Flexible experimental designs for valid single-cell RNA-sequencing experiments allowing batch effects correction

Fangda Song¹, Ga Ming Angus Chan¹ & Yingying Wei¹✉

Despite their widespread applications, single-cell RNA-sequencing (scRNA-seq) experiments are still plagued by batch effects and dropout events. Although the completely randomized experimental design has frequently been advocated to control for batch effects, it is rarely implemented in real applications due to time and budget constraints. Here, we mathematically prove that under two more flexible and realistic experimental designs—the reference panel and the chain-type designs—true biological variability can also be separated from batch effects. We develop Batch effects correction with Unknown Subtypes for scRNA-seq data (BUSseq), which is an interpretable Bayesian hierarchical model that closely follows the data-generating mechanism of scRNA-seq experiments. BUSseq can simultaneously correct batch effects, cluster cell types, impute missing data caused by dropout events, and detect differentially expressed genes without requiring a preliminary normalization step. We demonstrate that BUSseq outperforms existing methods with simulated and real data.
Single-cell RNA-sequencing (scRNA-seq) technologies enable the measurement of the transcriptome of individual cells, which provides unprecedented opportunities to discover cell types and understand cellular heterogeneity. However, like the other high-throughput technologies, scRNA-seq experiments can suffer from severe batch effects. Moreover, compared with bulk RNA-seq data, scRNA-seq data can have an excessive number of zeros that result from either biological zeros—that is, a gene is not expressed in a given cell—or dropout events—that is, the expression of some genes are not detected even though they are actually expressed in the cell due to amplification failure prior to sequencing. Consequently, despite the widespread adoption of scRNA-seq experiments, the design of a valid scRNA-seq experiment that allows the batch effects to be removed, the biological cell types to be discovered, and the missing data to be imputed remains an open problem.

One of the major tasks of scRNA-seq experiments is to identify cell types for a population of cells. The cell type of each individual cell is unknown and is often the target of inference. Classic batch effects correction methods, such as ComBat and SVA, are designed for bulk experiments and require knowledge of the subtype information of each sample a priori. For scRNA-seq data, this subtype information corresponds to the cell type of each individual cell. Clearly, these methods are thus infeasible for scRNA-seq data. Alternatively, if one has knowledge of a set of control genes whose expression levels are constant across cell types, then it is possible to apply RUV. However, selecting control genes is still challenging for scRNA-seq experiments, and recently there has been active research on identifying stably expressed genes that are reproducible and conserved across species for single cells.

To identify unknown subtypes, MetaSparseKmeans jointly clusters samples across batches. Unfortunately, MetaSparseKmeans requires all subtypes to be present in each batch. Suppose that we conduct scRNA-seq experiments for blood samples from a healthy individual and a leukemia patient, one person per batch. Although we can anticipate that the two batches will share T cells and B cells, we do not expect that the healthy individual will have cancer cells as the leukemia patient. Therefore, MetaSparseKmeans is too restrictive for many scRNA-seq experiments.

The mutual-nearest-neighbor (MNN) based approaches, including MNN and Scanorama, allow each batch to contain some but not all cell types. However, these methods require batch effects to be almost orthogonal to the biological subspaces and much smaller than the biological variations between different cell types. These are strong assumptions and cannot be validated at much smaller than the biological variations between different cell designs of scRNA-seq experiments under which their methods incorporate dropout events into the factor model, an open problem.

We develop a hierarchical model BUSseq that closely mimics the data generating procedure of scRNA-seq experiments (Fig. 1a, Supplementary Fig. 1 and Supplementary Note 1). Given that we have measured B batches of cells each with a sample size of nbi, let us denote the underlying gene expression level of gene g in cell i of batch Xbig, Xbig follows a negative binomial distribution with mean expression level µbig and a gene-specific and batch-specific overdispersion parameter δbig. The mean expression level is determined by the cell type Wbi with the cell type effect βg, the log-scale baseline expression level αg, the location batch effect ζg, and the cell-specific size factor δg. The cell-specific size factor δg characterizes the impact of cell size, library size and sequencing depth. It is of note that the cell type Wbi of each individual cell is unknown and is our target of inference. Therefore, we assume that a cell on batch b comes from cell type k with probability \( \text{Pr}(W_{bi} = k) = \pi_{bk} \) and the proportions of cell types \( \pi_{bk} \) vary among batches.

Unfortunately, it is not always possible to observe the expression level Xbig. Without dropout (Zbig = 0), we can directly observe Ybig = Xbig. However, if a dropout event occurs (Zbig = 1), then we observe Ybig = 0 instead of Xbig. In other words, when we observe a zero read count Ybig = 0, there are two possibilities: a non-expressed gene—biological zeros—or a dropout event. When gene g is not expressed in cell i of batch b (Xbig = 0), we always have Ybig = 0; when gene g is actually expressed in cell i of batch b (Xbig > 0) but a dropout event occurs, we can only observe Ybig = 0, and hence Zbig = 1. It has been noted that highly expressed genes are less-likely to suffer from dropout events. We thus model the dependence of the dropout rate \( \text{Pr}(Z_{big} = 1|X_{big}) \) on the expression level using a logistic regression with batch-specific intercept intercept \( y_{0b} \) and log-odds ratio \( \gamma_{bi} \).

Noteworthy, BUSseq includes the negative binomial distribution without zero inflation as a special case. When all cells are from a single cell type and the cell-specific size factor δg is estimated a priori according to spike-in genes, BUSseq can reduce to a form similar to BASICS.

We only observe Ybig for all cells in the B batches and the total G genes. We conduct statistical inference under the Bayesian framework and adopt the Metropolis-within-Gibbs algorithm for the Markov chain Monte Carlo (MCMC) sampling.
expressed gene, and (III) the ratios of mean expression levels are different for each cell-type pair \((k, \hat{k})\) (see Theorem 1 in “Methods”). Condition (I) requires that the highly expressed genes are less likely to have dropout events, which is routinely observed for scRNA-seq data. Condition (II) always holds in reality. Because scRNA-seq experiments measure the whole transcriptome of a cell, condition (III) is also always met in real data. For example, if there exists one gene \(g\) such that for any two distinct cell-type pairs \((k_1, k_2)\) and \((k_3, k_4)\) their mean expression levels’ ratios \(\frac{\exp(\beta_{gk_1})}{\exp(\beta_{gk_2})}\) and \(\frac{\exp(\beta_{gk_3})}{\exp(\beta_{gk_4})}\) are not the same, then condition (III) is already satisfied.

The commonly advocated completely randomized experimental design is a special case of the complete setting design. In a completely randomized design, cells are assigned to different batches completely at random. As a result, all of the batches have similar compositions. In contrast, under the complete setting design, cells from different cell types can be distributed to different batches very unevenly. The requirement that each batch has similar cellular compositions is crucial for traditional batch effects correction methods developed for bulk experiments such as ComBat to work well for scRNA-seq data. In contrast, BUSseq is not limited to this balanced design constraint and is applicable to not only the completely randomized design but also the general complete setting design.

Ideally, we would wish to adopt completely randomized experimental designs. However, in reality, it is always very challenging to implement complete randomization due to time and budget constraints. For example, when we recruit patients sequentially, we often have to conduct scRNA-seq experiments patient-by-patient rather than randomize the cells from all of the patients to each batch, and the patients may not have the same set of cell types. Fortunately, we can prove that BUSseq also applies to two sets of flexible experimental designs, which allow cell types to be measured in only some but not all of the batches.

Assuming that conditions (I)-(III) are satisfied, if there exists one batch that contains cells from all cell types and the other batches have at least two cell types (Fig. 1d), then BUSseq can...
tease out the batch effects and identify the true biological variability (see Theorem 2 in “Methods”). We call this setting the reference panel design.

Sometimes, it can still be difficult to obtain a reference batch that collects all cell types. In this case, we can turn to the chain-type design, which requires every two consecutive batches to share two cell types (Fig. 1e). Under the chain-type design, given that conditions (I)–(III) hold, BUSseq is also identifiable and can estimate the parameters well (see Theorem 3 in “Methods”).

A special case of the chain-type design is when two common cell types are shared by all of the batches, which is frequently encountered in real applications. For instance, when blood samples are assayed, even if we perform scRNA-seq experiment patient-by-patient with one patient per batch, we know a priori that each batch will contain at least both T cells and B cells, thus satisfying the requirement of the chain-type design.

The key insight is that despite batch effects, differences between cell types remain constant across batches. The differences between a pair of cell types allow us to distinguish batch effects from biological variability for those batches that measure both cell types. Therefore, BUSseq can separate batch effects from cell type effects under more general designs beyond the easily understood and commonly encountered reference panel design and chain-type design. If we regard each batch as a node in a graph and connect two nodes with an edge if the two batches share at least two cell types, then BUSseq is identifiable as long as the resulting graph is connected (Supplementary Fig. 2 and Theorem 4 in “Methods”).

For scRNA-seq data, dropout rate depends on the underlying expression levels. Such missing data mechanism is called missing not at random (MNAR) in statistics. It is very challenging to establish identifiability for MNAR. Miao et al. showed that for many cases even when both the outcome distribution and the missing data mechanism has parametric forms, the model can be nonidentifiable. However, fortunately, despite the dropout events and the cell-specific size factors, we are able to prove Theorems 1–4 (Supplementary Note 3). The reference panel design, the chain-type design, and the connected design liberalize researchers from the ideal but often unrealistic requirement of the completely randomized design.

BUSseq accurately learns the parameters and the missing data. We first evaluated the performance of BUSseq via a simulation study. We simulated a dataset with four batches and a total of five cell types under the chain-type design (Fig. 2a–d and Theorem 3). Every two consecutive batches share at least two cell types, but none of the batches contains all of the cell types. The sample sizes for each batch are \((n_1, n_2, n_3, n_4) = (300, 300, 200, 200)\) (Supplementary Table 1), and there are a total of 3000 genes, out of which 500 genes are differentially expressed between cell types. The remaining 2500 genes have no biological differences between different cell types, so they are pure noises with only batch effects. In real datasets, batch effects are often much larger than the cell type effects (Fig. 3a) and not orthogonal to the cell type effects (Supplementary Fig. 3). In the simulation study, we choose the magnitude of the batch effects, cell type effects, the dropout rates, and the cell-specific size factors to mimic real data scenarios (Fig. 3a). The simulated observed data suffer from severe batch effects and dropout effects (Figs. 2d, 3c). The dropout rates for the four batches are 26.79%, 24.53%, 28.36%, and 31.29%, with the corresponding total zero proportions given by 44.13%, 48.85%, 53.07%, and 61.38%.

BUSseq correctly identifies the presence of five cell types among the cells (Fig. 2e). Moreover, despite the dropout events, BUSseq accurately estimates the cell type effects \(\beta_{k,s}\) (Fig. 2a, f), the batch effects \(\nu_{k,b}\) (Fig. 2b, g), and the cell-specific size factors \(\delta_{k,s}\) (Fig. 2i). In particular, BUSseq outperforms existing normalization methods, including DESeq normalization34, trimmed mean of M-values (TMM) normalization35, library size normalization, and the deconvolution normalization method36, in estimating the cell-specific size factors \(\delta_{k,s}\) (Supplementary Fig. 4 and Supplementary Note 4). When controlling the Bayesian False Discovery Rate (FDR) at 5%37,38, we identify all intrinsic genes that differentiate cell types with the true FDR being 2% (“Methods”).

Figure 2h demonstrates that BUSseq can learn the underlying expression levels \(X_{kig,s}\) well based on the observed data \(Y_{kig,s}\), which are subject to dropout events. This success arises because BUSseq uses an integrative model to borrow strengths both across genes and across cells from all batches. In comparison, we also benchmarked BUSseq with three state-of-the-art imputation methods for scRNA-seq data—SAVER39, DrImpute40, and scImpute41. Once again, BUSseq performs the best in identifying the true biological zeros and recovering the underlying expression levels \(X_{kig,s}\) for the dropout events (Supplementary Table 2 and Supplementary Note 5).

ComBat offers a version of data that have been adjusted for batch effects. Here, we also provide batch-effects-corrected count data based on quantile matching (“Methods”). The adjusted count data no longer suffer from batch effects and dropout events, and they even do not need further cell-specific normalization (Fig. 2i). Therefore, they can be treated as if measured in a single batch for downstream analysis.

To evaluate the robustness of BUSseq, we conducted extensive sensitivity analyses, and they show that BUSseq is robust to the choice of hyperparameters, high zero rates, model misspecification and gene filtering (Supplementary Figs. 5–7, Supplementary Tables 3 and 4, and Supplementary Note 6).

BUSseq outperforms existing methods in simulation study. We benchmarked BUSseq with the state-of-the-art methods for batch effects correction for scRNA-seq data—LIGER18, MNN14, Scanorama15, scVI25, Seurat17, and ZINB-WaVE41. The adjusted Rand index (ARI) measures the consistency between two clustering results and is between zero and one, a higher value indicating better consistency42 (Supplementary Note 7). The ARI between the inferred cell types \(\hat{W}_{k,b}\) by BUSseq and the true underlying cell types \(W_{k,b}\) is one. Thus, BUSseq can perfectly recover the true cell type of each cell. In comparison, we applied each of the compared methods to the dataset and then performed their own clustering approaches (Supplementary Note 8). The ARI is able to compare the consistency of two clustering results even if the numbers of clusters differ, therefore, we chose the number of cell types by the default approach of each method rather than set it to a common number. The resulting ARI s are 0.837 for LIGER, 0.654 for MNN, 0.521 for Scanorama, 0.480 for scVI, 0.632 for Seurat, and 0.571 for ZINB-WaVE. Moreover, the t-SNE plots (Fig. 3c, d) show that only BUSseq can perfectly cluster the cells by cell types rather than batches. We also calculated the silhouette score for each cell for each compared method (Supplementary Note 7). A high silhouette score indicates that the cell is well matched to its own cluster and separated from neighboring clusters. Figure 3b shows that BUSseq gives the best segregated clusters.

BUSseq outperforms existing methods on hematopoietic data. We re-analyzed the two hematopoietic datasets previously studied by Haghverdi et al. (Fig. 4a and Supplementary Fig. 8a, b). The two datasets shared at least three cell types, including the common myeloid progenitors (CMP), megakaryocyte-erythrocyte
progenitors (MEP) and granulocyte-monocyte progenitors (GMP), thus follow the chain-type design.

BUSseq fits the zero rates (Table 1 and Supplementary Note 9) and the mean-variance trends (Fig. 5a, Supplementary Fig. 9 and Supplementary Note 7) of the real data very well. In order to compare BUSseq with existing methods, we compute the ARIs between the clustering of each method and the FACS labels. The resulting ARIs are 0.582 for BUSseq, 0.307 for LIGER, 0.575 for MNN, 0.518 for Scanorama, 0.197 for scVI, 0.266 for Seurat, and 0.200, respectively. The simulated observed data Y. The overall dropout rate is 27.3%, whereas the overall zero rate is 50.8%. e The BIC plot. The BIC attains the minimum at K = 5, identifying the true cell type number. f The estimated log-scale mean expression levels for each cell type \( \alpha + \beta \). g Estimated batch effects. h Imputed expression levels \( X \). i Corrected count data \( X \) grouped by batches. j Scatter plot of the estimated versus the true cell-specific size factors.

Specifically, BUSseq learns six cell types from the dataset. According to the FACS labels (Methods), Cluster 2, Cluster 5, and Cluster 6 correspond to CMP, MEP, and GMP, respectively (Figs. 4c, 6a–c). Cluster 1 is composed of long-term hematopoietic stem cells and multi-potent progenitors (MPP). These are cells from the early stage of differentiation. Cluster 4 consists of a mixture of MEP and CMP, while Cluster 3 is dominated by cells labeled as other. Comparison between the subpanel for BUSseq in Figs. 4c and 6b indicates that Cluster 4 are cells from an intermediate cell type between CMP and MEP. In particular, according to Fig. 6e, the marker genes Apoe and Gata2 are highly expressed in Cluster 4 but not in CMP (Cluster 2) and MEP (Cluster 6), and the marker gene Ctsε is expressed in MEP (Cluster 6) but not in Cluster 4 and CMP (Cluster 2). Therefore, cells in Cluster 4 do form a unique group with distinct expression patterns. This intermediate cell stage between CMP and MEP is missed by all of the other methods considered.

Moreover, we find that well known B-cell lineage genes \( Ebf1, Vpreb1, Vpreb3, \) and \( Igl1 \), are highly expressed in Cluster 3, but not in the other clusters (Fig. 6c, e). To identify Cluster 3, which is dominated by cells labeled as other by Nestorowa et al. 43, we map the mean expression profile of each cluster learned by BUSseq to the Haemopedia RNA-seq dataset46. It turns out that Cluster 3 aligns well to common lymphoid progenitors (CLP) that give rise to T-lineage cells, B-lineage cells and natural killer cells (Fig. 6d). Therefore, Cluster 3 represents cells that differentiate from lymphoid-primed multipotent progenitors (LMPP)44. Once again, all the other methods fail to identify these cells as a separate group.

Thus, although BUSseq does not assume any temporal ordering between cell types, it is able to preserve the differentiation trajectories (Fig. 6a, b); although BUSseq assumes that each cell belongs to one cell type rather than conducts semisoft clustering47, it is capable of capturing the subtle changes across cell types and within a cell type due to continuous processes such as development and differentiation (Supplementary Fig. 11 and Supplementary Note 10).

We further inspect the functions of the intrinsic genes that distinguish different cell types. BUSseq detects 1,419 intrinsic genes at the Bayesian false discovery rate (FDR) cutoff of 0.05 (“Methods”). The gene set enrichment analysis48 shows that 51
BUSseq outperforms existing method on pancreas data. We further studied the four scRNA-seq datasets of human pancreas cells, analyzed in Haghverdi et al.14. These cells were isolated from deceased organ donors with and without type 2 diabetes. As each patient has at least two pancreas cell types—alpha cells and beta cells, the four datasets follow the chain-type design. We obtained 7095 cells after quality control (Methods) and treated each dataset as a batch following Haghverdi et al.14.

BUSseq recapitulates the properties of real scRNA-seq data very well in terms of the zero rates (Table 1 and Supplementary Note 9) and the mean-variance trend (Fig. 5b and Supplementary Fig. 12). In particular, the posterior predictive check shows that BUSseq fits the zero rates much better than a model that ignores dropout events, especially when scRNA-seq data are assayed by protocols that do not incorporate UMI counts, such as SMART-seq2.

We can compare the clustering results from each batch effects correction method with the cell-type labels provided by Segerstolpe et al.52 and Lawlor et al.51 (Fig. 7a, b and Supplementary Fig. 8c, d). The pancreas is highly heterogeneous and consists of two major categories of cells: islet cells and non-islet cells. Islet cells include alpha, beta, gamma, and delta cells, while non-islet cells include acinar and ductal cells. BUSseq identifies a total of eight cell types: five for islet cells, two for non-islet cells and one for the labeled other cells. Specifically, the five islet cell types identified by BUSseq correspond to three groups of alpha cells, a group of beta cells, and a group of delta and gamma cells. The two non-islet cell types identified by BUSseq correspond exactly to the acinar and ductal cells. Compared with all of the other methods, BUSseq gives the best separation between islet and non-islet cells, as well as the best segregation within islet cells. In particular, the median silhouette coefficient by BUSseq is higher than that of any other method (Fig. 7c and Supplementary Fig. 10c).

The ARIs of all methods are 0.608 for BUSseq, 0.542 for LIGER, 0.279 for MNN, 0.527 for Scanorama, 0.282 for scVI, 0.287 for Seurat, and 0.380 for ZINB-WaVE (“Methods” and Supplementary Table 5). Thus, BUSseq outperforms all of the other methods in being consistent with the cell-type labels
Fig. 4 t-SNE and Principal Component Analysis (PCA) plots for the hematopoietic data. a t-SNE plots colored by batch. b t-SNE plots colored by FACS cell type labels. c PCA plots colored by FACS cell type labels. BUSseq adjusts batch effects and correctly clusters cells according to their FACS cell type labels in the hematopoietic study.

Table 1 Zero-count rates and dropout rates of the hematopoietic and pancreas studies.

| Study             | Protocol   | UMI | ρ₀ | ρₚ | ρₚₛ | ρ筚ₚₛ | ρₚₛₑ | ρₚₛₑₑ |
|-------------------|------------|-----|----|----|-----|-------|-------|-------|
| Hematopoietic     | MARS-seq   | Yes | 0.892 | <0.001 | 0.887 | 0.005 | 0.874 | 0.018 |
| Hematopoietic     | SMART-seq2 | No  | 0.421 | <0.001 | 0.424 | 0.003 | 0.445 | 0.024 |
| Pancreas          | CEL-seq2   | Yes | 0.689 | <0.001 | 0.625 | 0.064 | 0.682 | 0.007 |
| Pancreas          | CEL-seq2   | Yes | 0.517 | 0.017  | 0.558 | 0.041 | 0.617 | 0.100 |
| Pancreas          | SMART-seq2 | No  | 0.609 | 0.167  | 0.531 | 0.078 | 0.430 | 0.179 |
| Pancreas          | SMART-seq2 | No  | 0.480 | 0.551  | 0.485 | 0.005 | 0.329 | 0.161 |

ρ₀ denotes the observed zero rate in each batch; ρₚ represents the inferred dropout rate by BUSseq; ρ筚ₚₛ denotes the posterior mean of zero rate inferred by BUSseq; and ρ筚ₚₛₑₑ represents the posterior mean of zero rate inferred by a reduced model of BUSseq that ignores dropout events and hence uses negative binomial distribution without zero inflation, abbreviated as BUSseq-nzf. BUSseq detects the existence of dropout events automatically and performs better than BUSseq-nzf in terms of the posterior predictive check of zero rates.

according to marker genes. In Fig. 7d, the locally high expression levels of marker genes for each cell type show that BUSseq correctly clusters cells according to their biological cell types.

BUSseq identifies 426 intrinsic genes at the Bayesian FDR cutoff of 5% (Methods). We conducted the gene set enrichment analysis18 with the KEGG pathways49 (Supplementary Note 11). There are 14 enriched pathways (p values < 0.05). Among them, three are diabetes pathways; two are pancreatic and insulin secretion pathways; and another two pathways are related to metabolism (Supplementary Table 7). Recall that the four datasets assayed pancreas cells from type 2 diabetes and healthy individuals, therefore, the pathway analysis once again confirms that BUSseq provides biologically and clinically valid cell typing.

BUSseq is applicable to droplet-based scRNA-seq data. We further analyzed a dataset that contains samples assayed by droplet-based scRNA-seq protocols. Comparing the performance of different methods on real scRNA-seq data is challenging due to the lack of true cell type labels in real application. Fortunately, Tian et al.33 created scRNA-seq datasets with known cell type labels by profiling cells from cancer cell lines. In one experiment, they assayed three lung adenocarcinoma (LUAD) cell lines—HCC827, H1975, and H2228 on three platforms with CELseq2, 10x Chromium and Drop-seq protocols, respectively. As a result, 1401 cells were totally measured on three batches. Each batch consists of all of the three cell types, and data from different batches have different levels of sparsity. Consequently, this study satisfies the complete setting, which is a special case of both the reference-panel design and the chain-type design.

We selected the top 6000 highly variable genes (HVGs) within each batch and obtained 2267 common HVGs across three batches (“Methods”). The t-SNE and PCA plots of the raw count data show that significant effects occur across the three
protocols (Fig. 8a, b and Supplementary Fig. 13a, b). We applied BUSseq and varied the number of cell type K from 2 to 6. Although the BIC selects four cell types instead of three cell lines (Supplementary Fig. 14), two of the four identified clusters correspond to two subpopulations of the H1975 cell lines (Supplementary Table 8). We further visualized the log-scale mean expression levels of intrinsic genes of the four learned cell types (Fig. 8c). The first two cell types have similar expression patterns, but some differentially expressed genes are observed between them. Moreover, the t-SNE (Fig. 8c, d) and PCA (Supplementary Fig. 13c, d) plots demonstrate the high level of similarity of the first two estimated cell types and confirm that the corrected count data \( \mathbf{x}_{\text{seq}} \) obtained by BUSseq cluster cells by cell type instead of by batch (Fig. 8f).

We also applied the benchmarked methods to compare their clustering accuracy. The ARIs of all methods are 0.841 for BUSseq, 0.825 for LIGER, 0.650 for MNN, 0.637 for Scanorama, 0.429 for scVI, 0.324 for Seurat, and 0.398 for ZINB-WaVE. Thus, BUSseq outperforms all of the other methods in clustering accuracy. We further compared BUSseq with a recently proposed
Discussion

For the completely randomized experimental design, it seems that everyone is talking but no one is listening. Due to time and budget constraints, it is always difficult to implement a completely randomized design in practice. Consequently, researchers often pretend to be blind to the issue when carrying out their scRNA-seq experiments. In this paper, we mathematically prove and empirically show that under the more realistic reference panel and chain-type designs, batch effects can also be adjusted for scRNA-seq experiments. We hope that our results will alarm researchers of confounded experimental designs and encourage them to implement valid designs for scRNA-seq experiments in real applications.

BUSseq provides one-stop services. In contrast, most existing methods are multi-stage approaches—clustering can only be performed after the batch effects have been corrected and the differential expressed genes can only be called after the cells have been clustered. The major issue with multi-stage methods is that uncertainties in the previous stages are often ignored. For instance, when cells have been first clustered into different cell types and then differential gene expression identification is conducted, the clustering results are taken as if they were the underlying truth. As the clustering results may be prone to errors in practice, this can lead to false positives and false negatives. In contrast, BUSseq simultaneously corrects batch effects, clusters cell types, imputes missing data, and identifies intrinsic genes that differentiate cell types. BUSseq thus accounts for all uncertainties and fully exploits the information embedded in the data. As a result, BUSseq is able to capture subtler changes between cell types, such as the cluster corresponding to LIMPP lineage that is missed by all of the state-of-the-art methods.

BUSseq employs MCMC algorithm for statistical inference. Although MCMC algorithms are well-known for heavy computation load, fortunately, the computational complexity of BUSseq is \(O(\sum_{B} n_B G_K)\), which is both linear in the number of genes \(G\) and in the total number of cells \(N = \sum_{B} n_B\). Moreover, most steps of the MCMC algorithm for BUSseq are parallelizable (Supplementary Note 12). We implement a parallel multi-core-CPU version and a parallel GPU version of the algorithm, respectively. Running the GPU version of the algorithm with a single core of an Intel Xeon Gold 6132 Processor and one NVidia Tesla P100 GPU took 0.35, 1.15, 1.5 h for the simulation, the hematopoietic, and the human pancreas data, respectively (Supplementary Table 10). Experiments show that the running time and random-access memory (RAM) usage are indeed linear in the number of genes \(G\) and the number of cells \(N\) for both the CPU and the GPU parallel version of BUSseq (Fig. 9 and Supplementary Note 13). Moreover, by writing the posterior samples to the hard disk every a few iterations, we can further reduce the
RAM usage so that BUSseq is affordable by a commonly available cluster node rather than a high-end one (Supplementary Table 11 and Supplementary Fig. 15). Compared with the time for preparing samples and conducting the scRNA-seq experiments, the computation time of BUSseq is affordable and worthwhile for the accuracy.

Practical and valid experimental designs are urgently required for scRNA-seq experiments. We envision that the flexible
reference panel and the chain-type designs will be widely adopted in scRNA-seq experiments and BUSseq will greatly facilitate the analysis of scRNA-seq data.

**Methods**

**BUSseq model.** The hierarchical model of BUSseq can be summarized as:

$$
\Pr(W_{bi} = k) = \pi_{bk}, \quad \sum_{k=0}^{K} \pi_{bk} = 1.
$$

$$
X_{bi} \sim W_{bi} = k \sim \text{NB}(\mu_{bk}, \phi_{bk}), \quad \log(\mu_{bk}) = \alpha_k + \beta_{bk} + v_{bi} + \delta_i.
$$

$$
Z_{bi} | X_{bi} \sim \text{Bernoulli}(\pi_{bk}), \quad \log(\frac{\pi_{bk}}{1 - \pi_{bk}}) = \gamma_0 + \gamma_1 X_{bi}.
$$

$$
Y_{bi} = X_{bi} | Z_{bi} = 0, Y_{bi} = 0 | Z_{bi} = 1.
$$

Collectively, $Y = \{Y_{bi}\}_{b=1,...,B; i=1,...,n_b}$ are the observed data; the underlying expression levels $X = \{X_{bi}\}_{b=1,...,B; i=1,...,n_b}$, the dropout indicators $Z = \{Z_{bi}\}_{b=1,...,B; i=1,...,n_b}$, and the cell type indicators $W = \{W_{bi}\}_{b=1,...,B; i=1,...,n_b}$ are all missing data; the log-scale baseline gene expression levels $\alpha = \{\alpha_k\}_{k=1,...,K}$, the cell type effects $\beta = \{\beta_{bk}\}_{k=1,...,K}$, the location batch effects $\nu = \{\nu_{bi}\}_{b=1,...,B}$, the overdispersion parameters $\phi = \{\phi_{bk}\}_{b=1,...,B}$, the cell-specific size factors $\Delta = \{\Delta_{bk}\}_{b=1,...,B}$ and the dropout parameters $\Gamma = \{\gamma_0, \gamma_1\}$, and the cell compositions $\pi = \{\pi_{bk}\}_{b=1,...,B; k=1,...,K}$ are the parameters. Without loss of generality, for model identifiability, we assume that the first batch is the reference batch measured without batch effects with $\nu_2 = 0$ for every gene and the first cell type is the baseline cell type with $\beta_{b2} = 0$ for every gene. Similarly, we take the cell-specific size factor $\delta_0 = 0$ for the first cell of each batch. We gather all the parameters as

$$\Theta = (\alpha, \beta, \nu, \phi, \Delta, \Gamma, \pi).$$

Consequently, the observed data likelihood function becomes

$$
L_s(\Theta|y) = \prod_{b=1}^{B} \prod_{i=1}^{n_b} \sum_{k=1}^{K} \pi_{bk} \Pr(Y_{bi} = y_{bi}|W_{bi}, \Theta),
$$

where

$$
\Pr(Y_{bi} = y_{bi}|W_{bi}, \Theta) = \begin{cases} 
\sum_{k=1}^{K} \frac{\exp(\alpha_k + \beta_{bk} + v_{bi} + \delta_i)}{\text{NB}(\mu_{bk}, \phi_{bk})} & \text{if } y_{bi} = 0, \\
\frac{\mu_{bk}^{y_{bi}} \exp(-\mu_{bk})}{\text{NB}(\mu_{bk}, \phi_{bk})} & \text{if } y_{bi} > 0.
\end{cases}
$$

and

$$f_{\text{NB}}(x; \mu, \phi) = C_{\mu}^{\phi+1} \frac{x^\phi}{\Gamma(\phi + 1)} \exp(-\mu x)$$

denotes the probability mass function of the negative binomial distribution $\text{NB}(\mu, \phi)$. For $y_{bi} = 0$, $f_{\text{NB}}(0; \exp(\alpha_k + \beta_{bk} + v_{bi} + \delta_i), \phi_{bk})$ corresponds to a biological zero, whereas $\sum_{k=1}^{K} \frac{\exp(\alpha_k + \beta_{bk} + v_{bi} + \delta_i)}{\text{NB}(\mu_{bk}, \phi_{bk})}$ corresponds to a dropout event.

**Experimental designs.** By creating a set of functions similar to the probability generating function, we prove that BUSseq is identifiable, in other words, if two sets of parameters are different, then their probability distribution functions for the observed data are different, for not only the complete setting but also the reference panel and the chain-type designs (see the proofs in Supplementary Note 3).

Theorem 1 (The Complete Setting) If $\pi_{bk} > 0$ for every batch $b$ and cell type $k$, given that (I) $\gamma_{bi} < 0$ for every $b$, (II) for any two cell types $k_1$ and $k_2$, there exist at least two differentially expressed genes $g_1$ and $g_2$, $g_j - \beta_{bk} \neq g_2 - \beta_{bk}$, and (III) for any two distinct cell-type pairs $(k_1, k_2) \neq (k_2, k_1)$, their differences in cell-type effects are not the same $\beta_{bk} - \beta_{bk_2} \neq \beta_{bk_1} - \beta_{bk_2}$, then BUSseq is identifiable (up to label switching) in the sense that $L_s(\Theta|y) = L_s(\Theta|y)$ for any $y$ implies that $\pi_{bk} = \pi_{bk_2}$ for all $y$. In other words, BUSseq is identifiable (up to label switching).

Theorem 2 (The Reference Panel Design) If there is a total of $K$ cell types $\{C_1, C_2, ..., C_K\}$, $K \geq 2$ for every batch $b$, and there exists a batch $b$ such that it contains all of the cell types $\{1, 2, ..., K\}$, then given that conditions (I)–(III) hold, BUSseq is identifiable (up to label switching).

Theorem 3 (The Chain-Type Design) If there are a total of $K$ cell types $\{C_1, C_2, ..., C_K\}$ and every two consecutive batches share at least two cell types $\{C_1 \cap C_2, C_2 \cap C_3, ..., C_{K-1} \cap C_K\}$, then given that conditions (I)–(III) hold, BUSseq is identifiable (up to label switching).

Therefore, even for the reference panel and chain-type designs that do not assay all cell types in each batch, batch effects can be removed; cell types can be clustered...
and missing data due to dropout events can be imputed. Both the reference panel design and the chain-type design belong to the more general connected design.

**Theorem 4** (The Connected Design) We define a batch graph $G = (V, E)$. Each node $b \in V$ represents a batch. There is an edge $e \in E$ between two nodes $b_1$ and $b_2$ if and only if batches $b_1$ and $b_2$ share at least two cell types. If the batch graph is connected and conditions (I)–(III) hold, then BUSSeq is identifiable (up to label switching).

### Statistical inference

We conduct the statistical inference under the Bayesian framework. We assign independent priors to each component of $\Theta$ as follows (Supplementary Table 3):

$$\pi_g = (\pi_{g1}, \ldots, \pi_{gK}) \sim \text{Dirichlet}(\xi, \ldots, \xi), \quad 1 \leq b \leq B;$$

$$\gamma_{1i} \sim \text{Gamma}(\alpha_{g}, \beta_{g}), \quad 1 \leq b \leq B;$$

$$\alpha_{g} \sim \text{N}(\mu_{g}, \sigma_{g}^2), \quad 1 \leq g \leq G;$$

$$\delta_{bg} \sim \text{N}(\mu_{bg}, \sigma_{bg}^2), \quad 1 \leq b \leq B, 2 \leq g \leq n_{bg};$$

$$\theta_{gk} \sim \text{Gamma}(\kappa, r), \quad 1 \leq b \leq B, 1 \leq g \leq G.$$ 

We are interested in detecting genes that differentiate cell types. Therefore, we impose a spike-and-slab prior\(^{39}\) using a normal mixture to the cell-type effect $\beta_{bg}$. The spike component concentrates on zero with a small variance $\tau_{bg}$, whereas the slab component tends to deviate from zero, thus having a larger variance $\tau_{bg}$. We introduce another latent variable $L_{bgk}$ to indicate which component $\beta_{bg}$ comes from.

$L_{bgk} = 0$ if gene $g$ is not differentially expressed between cell type $k$ and cell type one, and $L_{bgk} = 1$, otherwise. We further define $D_{bg} = \sum_{k=1}^{K} L_{bgk}$. If $D_{bg} > 0$, then the expression level of gene $g$ does not stay the same across cell types. Following Huo et al.\(^{13}\), we call such genes intrinsic genes, which differentiate cell types. To control for multiple hypothesis testing, we let $L_{bgk} = \text{Bernoulli}(p)$ and assign a conjugate prior $\text{Beta}(\alpha_{bg}, \beta_{bg})$ to $p$. We set $\tau_{bg}$ to a large number and let $\tau_{bg}$ follow an inverse-gamma prior $\text{InvGamma}((\alpha_{bg}, \beta_{bg})$ with a small prior mean.

We develop an MCMC algorithm to sample from the posterior distribution (Supplementary Note 2). After the burn-in period, we take the mean of the posterior samples to estimate $\pi_{gK}$, $\alpha_{g}$, $\beta_{bg}$, $\gamma_{bg}$, and $\theta_{gk}$ and use the mode of posterior samples of $\theta_{gk}$ to infer the cell type for each cell.

We have actually also implemented an Expectation-Maximization (EM) algorithm (Supplementary Tables 13 and 14), we consider the log-scale of the estimated $L_{bgk}$ is not differentially expressed between cell type $k$ and cell type one, and $L_{bgk} = 1$, otherwise. We further define $D_{bg} = \sum_{k=1}^{K} L_{bgk}$. If $D_{bg} > 0$, then the expression level of gene $g$ does not stay the same across cell types. Following Huo et al.\(^{13}\), we call such genes intrinsic genes, which differentiate cell types.

**Selection of cell type numbers.** BUSSeq allows the user to input the total number of cell types $K$ according to prior knowledge. When $K$ is unknown, BUSSeq selects the number of cell types $K$ such that it achieves the minimum BIC.\(^{39}\) BIC adds a penalty term to the observed log-likelihood $L_{bgk}(\Theta)$ as Eq. (1).

$$\text{BIC}(K) = -2 \log(L_{bgk}(\Theta)) + K[B + G] + 2B (2B - 1) G + \frac{1}{2} \sum_{k=1}^{K} (n_{bg} - 1) \log(n_{bg}),$$

where $\Theta = (\alpha, \beta, \gamma, \phi, \delta, \pi)$ denotes the prior mean of parameters. As a result, the penalty in BIC helps the model selection to balance between goodness-of-fit and the model complexity (Supplementary Figs. 17–19, Supplementary Tables 15 and 16, and Supplementary Note 15).

### Inference of dropout events

In the BUSSeq model, a dropout event occurs for gene $g$ in cell $i$ of batch $b$ if the observed value $y_{bg} = 0$ but the imputed count data $\tilde{y}_{bg} > 0$. The identification allows us to calculate the frequency of dropout events in each batch. We calculate the zero rate of each batch as following:

$$\rho_d = \frac{1}{G} \sum_{g=1}^{G} \sum_{i=1}^{n_g} I(y_{bg} = 0)$$

and compute the dropout rate as the proportion of dropout events among the observations with zero counts.

**Posterior predictive check.** We evaluate how well BUSSeq fits the data via posterior predictive check\(^{49}\). In particular, we focus on the zero rates. In the posterior predictive check, we take MCMC samples of all the parameters after the burn-in iterations to simulate replicated datasets $y_{ji}^{rep}, j = 1, 2, \ldots, J$ for gene $g$ and $N = \sum_{b=1}^{B} n_{bg}$ cells, where $J$ denotes the total number of collected iterations after burn-ins. In our real data analyses, we ran 8000 iterations with the first 4000 iterations as burn-ins, so we generated $J = 8000 - 4000 = 4000$ replicated datasets for both the hematopoietic and Pancreas studies. For each generated replicate dataset, we calculated the zero rates of each batch according to Eq. (2). Finally, we average the zero rates over all the $J$ iterations to calculate the posterior mean $\bar{\rho}_d$ of the zero rate of each batch and compare it with the corresponding observed zero rate. Moreover, we also compare BUSSeq with a reduced model of BUSSeq which ignores dropout events and hence uses negative binomial distribution without zero inflation, abbreviated as BUSSeq-nz (Supplementary Note 9), via the posterior predictive check (Supplementary Note 16).

**Batch-effects-corrected values.** To facilitate further downstream analysis, we also provide a version of count data $\tilde{X} = [\tilde{X}_{bg1}, \tilde{X}_{bg2}, \ldots, \tilde{X}_{bgK}]$ for which the batch effects are removed and the biological variability is retained similar to that of ComBat\(^{7}\). Ideally, if $X_{bgk}$ is the arthentic percentile of NB(exp(\tilde{\alpha} + \tilde{\beta} + D_{bgk} + \delta_{gb})), we aim to take the arthentic percentile of NB(exp(\tilde{\alpha} + \tilde{\beta} + \delta_{gb})), as the corrected value $\tilde{X}_{bgk}$. However, the negative binomial distribution is a discrete distribution. As a result, several aliases may lie between the Pr(\tilde{X}_{bgk} = 1) percentile and Pr(\tilde{X}_{bgk} = 0) percentile of the distribution of $X_{bgk}$. For example, if $X_{bgk} \sim \text{NB}(\exp(2), 3)$, $X_{bgk} \sim \text{NB}(\exp(3), 5)$, and our observed value $X_{bgk} = 8$, then $\text{Pr}(\tilde{X}_{bgk} = 7)$ and $\text{Pr}(\tilde{X}_{bgk} = 8)$ correspond to the 58.676% and 65.766% percentile of NB(exp(2), 3). However, three numbers—21, 22, and 23—lie between 58.676% and 65.766% percentile of NB(exp(3), 5). Thus, to avoid bias, we draw one number uniformly from 21, 22, and 23 rather than take the maximum or the minimum to calculate $\tilde{X}_{bgk}$. 

---

**Article** Nature Communications | https://doi.org/10.1038/s41467-020-16905-2 | www.nature.com/naturecommunications
Thus, we develop a quantile matching approach based on inverse sampling. Specifically, given the fitted model and the inferred underlying expression level $\lambda_{ij}$, we first sample $u_{ij}$ from $Unif(\lambda_{ij} (\alpha + \beta + \gamma + \delta_1 + \delta_2))$. The corrected data $\tilde{X}$ are no longer protected from batch effects but also impute the missing data due to dropout events. Moreover, further cell-specific normalization is not needed. Meanwhile, the biological variability is retained thanks to the quantile transformation and sampling step. Therefore, we can directly perform downstream analysis on $\tilde{X}$.

### Preprocessing of the real datasets

**Gene filtering.** A common practice of scRNA-seq data analysis is to focus on the set of HVGs across all of the cells may be low, and hence such a gene will be selected, $\Phi_{ij} = \eta_{ij} u_{ij}$, as the corrected value $\tilde{X}_{ij}$. We corrected the data $X$ are not only protected from batch effects but also impute the missing data due to dropout events. Moreover, further cell-specific normalization is not needed. Meanwhile, the biological variability is retained thanks to the quantile transformation and sampling step. Therefore, we can directly perform downstream analysis on $\tilde{X}$.

**Hemopoietic study.** For the two hemopoietic datasets, we downloaded the read count matrix of the 1920 cells profiled by Paul et al. and the 2729 cells labeled as myeloid progenitor cells by Nestorova et al. from the NCBI Gene Expression Omnibus (GEO) with accession numbers GSE72857 and GSE81682. Following Brennecke et al., we first labeled cells using FACS labels and then performed the size factor normalization within each batch. Next, we filtered out the common HVGs identified by Nestorova et al. between two datasets. These HVGs were denoted by Ensembl ID. The genes in the GSE81682 dataset were named by Ensembl ID, and the genes in the GSE72857 dataset were named by Gene Symbol. The R package biomart was used to query the corresponding Gene Symbol by Ensembl ID. Finally, we obtained 3,470 common HVGs shared by the two datasets.

**Pancreas study.** Two of the pancreatic datasets profiled by the CEL-seq2 platform were downloaded from GEO with accession numbers GSE80176 and GSE85241. The two datasets assayed by the SMART-seq2 platform were obtained from GSE64735 and from ArrayExpress accession number E-MATB-5061. Following Haghverdi et al., we excluded cells with low library sizes (<100,000 reads), low numbers of expressed genes (>40% total counts from ribosomal RNA genes), or high ERCC content (>20% of total counts from spike-in transcripts) resulting in 7095 cells. We selected the 2480 HVGs shared by the four datasets according to mean expression level across all of the cells may be low, and hence such a gene will be selected, $\Phi_{ij} = \eta_{ij} u_{ij}$, as the corrected value $\tilde{X}_{ij}$. We corrected the data $X$ are not only protected from batch effects but also impute the missing data due to dropout events. Moreover, further cell-specific normalization is not needed. Meanwhile, the biological variability is retained thanks to the quantile transformation and sampling step. Therefore, we can directly perform downstream analysis on $\tilde{X}$.

**Software availability**

The C++ source code of the parallel multi-core-CPU version of BUSseq is available on GitHub [https://github.com/songfd2018/BUSseq] and the CUDA C source code of the GPU version of BUSseq is available on GitHub [https://github.com/AngelMaus/BUSseq_gpu]. All code is released under the GNU General Public License v3. In the two datasets assayed by the SMART-seq2 platform, we transformed the cluster mean learned by BUSseq as $m_{ij} = \log(1 + \exp(\alpha + \beta x_{ij}))$ for gene $g$ in the cluster $k$ and scaled $m_{ij}$ across all cell types as well. Finally, we calculated the correlation between the cluster means inferred by BUSseq and the reference expression profiles in Haemopedia for 37 hemopoietic cell types. The 37 marker genes were retrieved from Paul et al. (31 marker genes for HSPC) and Herman et al. (6 marker genes for LMPP).

**Data availability**

The published datasets used in this manuscript are available through the following accession numbers: SMART-seq2 platform hemopoietic data with GEO GSE81682 by Nestorova et al.; MARS-seq platform hemopoietic data with GEO GSE72857 by Paul et al.; CEL-seq platform pancreas data with GEO GSE81076 by Grün et al.; CEL-seq platform pancreas data with GEO GSE85241 by Muraro et al.; SMART-seq2 platform pancreas data with GEO GSE86473 by Lawlor et al.; and SMART-seq2 platform pancreas data with ArrayExpress E-MATB-5061 by Segerstolpe et al. All datasets are available in the published manuscript and figures in this manuscript are also available on GitHub [https://github.com/songfd2018/BUSseq-1.1.implementation]. Furthermore, we wrap C++ source code as an R package, BUSseq [https://github.com/songfd2018/BUSseq-Rpackage].

**Reporting summary**

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

References

1. Bacher, R. & Kendziorski, C. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biol.*, 17, 63 (2016).
2. Irizarry, R. A. et al. Multiple-laboratory comparison of microarray platforms. *Nat. Methods*, 2, 345–350 (2005).
3. Leek, J. T. et al. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.*, 11, 733–739 (2010).
4. Taub, M. A., Corrada-Bravo, H. & Irizarry, R. A. Overcoming bias and systematic errors in next-generation sequencing data. *Genome Med.*, 2, 87 (2010).
5. Hicks, S. C., Townes, F. W., Teng, M. & Irizarry, R. A. Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*, 19, 562–578 (2018).
6. Scherchenko, P. V., Silverstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nat. Methods*, 11, 740 (2014).
7. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8, 118–127 (2007).
9. Leek, J. T. Svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res.* 42, e161–e164 (2014).
10. Risso, D., Ngai, J., Speed, T. P. & Dudoit, S. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nat. Biotechnol.* 32, 896 (2014).
11. Jaganathan, L., Gagnon-Bartsch, J. A. & Speed, T. P. Correcting gene expression data when neither the unwanted variation nor the factor of interest are observed. *Biostatistics* 17, 16–28 (2015).
12. Lin, Y. et al. Evaluating stably expressed genes in single cells. *GigaScience* 8, giz106 (2019).
13. Huo, Z., Ding, Y., Liu, S., Oesterreich, S. & Tseng, G. Meta-analytic framework for sparse k-means to identify disease subtypes in multiple transcriptomic studies. *J. Am. Stat. Assoc.* 111, 27–42 (2016).
14. Haghighi, L., Lun, A. T., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* 36, 421 (2018).
15. Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat. Biotechnol.* 37, 685 (2019).
16. 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 (2018).
17. Stuarta, T. et al. Comprehensive integration of single-cell data. *Cell* 177, 1883–1890 (2019).
18. Welch, J. D. et al. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* 177, 1873–e1877 (2019).
19. Lin, Y. et al. scmerge leverages factor analysis, stable expression, and single-cell transcriptome pro-

Acknowledgements

This work was supported by the Hong Kong Ph.D. Fellowship F015-17417 and the General Research Funds 14306417 and 14305319 from the Hong Kong Research Grants Council of the Hong Kong Special Administrative Region of the People’s Republic of China and Direct Grants from the Research Committee of the Chinese University of Hong Kong. We acknowledge Dr. Xiangyu Luo for helpful comments on an early version of our paper.

Author contributions

F.S. developed the method and the proof, implemented the algorithm, prepared the software package, analyzed the data, and wrote the paper. G.M.A.C. implemented the algorithm and analyzed the data. Y.W. conceived and supervised the study, developed the method and the proof, and wrote the paper.

Competing interests

The authors declare no competing interests.
