Zero-preserving imputation of single-cell RNA-seq data

George C. Linderman\textsuperscript{1,8}, Jun Zhao\textsuperscript{2,8}, Manolis Roulis\textsuperscript{3}, Piotr Bielecki\textsuperscript{3,7}, Richard A. Flavell\textsuperscript{3,4}, Boaz Nadler\textsuperscript{5} & Yuval Kluger\textsuperscript{1,2,6} \textsuperscript{✉}

A key challenge in analyzing single cell RNA-sequencing data is the large number of false zeros, where genes actually expressed in a given cell are incorrectly measured as unexpressed. We present a method based on low-rank matrix approximation which imputes these values while preserving biologically non-expressed genes (true biological zeros) at zero expression levels. We provide theoretical justification for this denoising approach and demonstrate its advantages relative to other methods on simulated and biological datasets.
ingle-cell RNA-sequencing (scRNA-seq) techniques measure gene expression at the individual cell level. This requires amplification of truly minute quantities of mRNA, resulting in a phenomenon called “dropout” in which an expressed transcript is not detected and hence assigned a zero expression value. As a result, we define two types of zeros in the observed matrix: biological zeros (genes that were not expressed at the time of sequencing) and technical zeros (genes that were expressed at the time of sequencing but not measured). We note that our definition of “technical zeros” includes “sampling zeros,” which are the result of undersampling the transcripts in each cell. Roughly 15–40% of genes in bulk RNA-seq of different tissues are not expressed, suggesting that at least that percentage of genes are not expressed in individual cells and are thus biological zeros. However, due to technical zeros and cell-to-cell heterogeneity, the fraction of zeros in a typical scRNA-seq expression matrix is much higher, even exceeding 90% with recently developed droplet-based techniques.

Due to a large number of technical zeros, directly processing the raw data may be detrimental to downstream analysis, such as clustering and visualization. Imputation of the missing values has the potential to improve these analyses. An imputation method for scRNA-seq data should have (at least) the following two properties: (i) it should accurately impute the data, in particular preserving at zero true biological zeros while completing the technical ones. We illustrate this point below using a dataset from4, whereby detecting which matrix entries are true biological zeros, has crucial importance for a biological understanding of the particular system of interest. (ii) it should be able to process, in a reasonable time, large-scale scRNA-seq datasets that contain hundreds of thousands to millions of cells.

Over the past few years, several imputation methods have been developed, including DCA5, MAGIC6, SAVER7, and scImpute8. Some methods such as DCA and MAGIC treat all zeros as “missing data” and output a matrix in which every gene is expressed in every cell (i.e., there are no biological zeros). While other methods such as SAVER and scImpute do preserve biological zeros, as we show below, they either detect relatively few of them or otherwise impute a few of the technical ones. Finally, with their current implementations, SAVER and scImpute may be extremely slow on scRNA-seq datasets with over 100,000 cells.

To address these challenges, we present Adaptively thresholded Low-Rank Approximation (ALRA), a method for imputation of scRNA-seq data. ALRA takes advantage of the non-negativity and low-rank structure of an expression matrix to selectively impute technical zeros (Fig. 1). Due to widespread correlations between genes across different cells, we assume that the true expression matrix is non-negative, low-rank, and contains many zeros (biological zeros). The matrix measured in a scRNA-seq experiment is a corrupted version of this expression matrix where many technical zeros are introduced by measurement error or insufficient total read counts. The first step of ALRA is to compute a low-rank approximation of the observed matrix using the singular vector decomposition (SVD). The rank $k$ of the approximation is automatically determined by a simple procedure described in the Online Methods. The matrix resulting from this low-rank approximation, in general, has no zeros. Hence, the last step of ALRA is to restore the biological zeros in the matrix by thresholding its entries. As we prove mathematically in the supplement, under certain assumptions, the entries of this matrix that correspond to the true biological zeros are symmetrically distributed around zero. Due to this symmetry, the negative values provide an estimate for the error distribution of the elements corresponding to true biological zeros. As such, we set to zero all the entries of each row (or gene) that are smaller than the magnitude of the most negative value (see Methods for details). This ensures that with high probability all of the entries which correspond to biological zeros are set to zero. Our analysis may be of independent interest to other low-rank matrix completion or denoising problems.

We emphasize the simplicity—and hence computational efficiency—of our approach: it is SVD followed by a thresholding scheme that takes advantage of the non-negativity of the true matrix. Remarkably, as we demonstrate below with experiments on 11 real scRNA-seq datasets, this simple approach outperforms other more complex methods for the recovery of scRNA-seq expression data. In addition to these empirical results, we also give a rigorous theoretical justification for ALRA’s algorithm (see supplement).

Results and discussion

ALRA preserves biological zeros. Using a variety of datasets where ground truth is known, we empirically illustrate ALRA’s ability to preserve biological zeros. We first considered scRNA-seq of purified populations of peripheral blood mononuclear cells (PBMCs) generated by Zheng et al. (2017)9. We merged the purified populations into a single matrix and ran ALRA, DCA, MAGIC, SAVER, and scImpute on the merged matrix. We then focused our attention on B cells, natural killer cells, cytotoxic T cells, and helper T cells. These four cell types are well characterized and there are known marker genes specific to each. For example, PAX5 (BSAP), NCAM1 (CD56), CD8A, and CD4 are marker genes for peripheral B cells, NK cells, cytotoxic T cells, and helper T cells, respectively10–12. The expression values of a marker gene in a population that is known to not express that marker gene (e.g., NCAM1 in B cells) are biological zeros. Note that there may be further biological zeros in the marker genes due to cell-to-cell heterogeneity (e.g., some CD8+ T cells may not...
actually express CD8 at the time of sequencing), and for this reason, we focus on the known biological zeros: genes for which there is solid biological knowledge that these are not expressed in a cell population.

After low-rank approximation, all entries in the matrix are non-zero. However, as shown in Fig. 2A and in accordance with our theoretical analysis, the elements of the matrix correspond to known biological zeros (e.g., the gene PAX5 in NK cells, cytotoxic T cells, or helper T cells) are symmetrically distributed around zero. Thus in the second step of ALRA we threshold at the red dotted lines, which are the symmetrical mirror image of the 0.001 quantiles of each gene (blue dotted lines). As shown in Fig. 2B, this step recovers the known biological zeros of the marker genes for these four purified peripheral blood mononuclear cell populations. We note that some cytotoxic T cells with NK cell markers (i.e., CD8+ CD56+ cells) are known to be present in the peripheral blood, which justifies the completion of NCAM1 in a small fraction of cytotoxic T cells13.

In contrast, DCA, MAGIC, and SAVER output non-zero values for many entries in the matrix that are expected to be biological zeros. For example, both DCA and MAGIC result in a matrix where CD8+ T cells express CD4 at nontrivial levels. While there may be some crossover between these marker genes, particularly during development, it would be in a small fraction of the mature cells. Furthermore, the shapes of the distributions obtained by these three imputation methods have no obvious

---

**Fig. 2** Preserving biological zeros in purified PBMC populations. **A** After low-rank approximation, ALRA recovers biological zeros of each gene by thresholding (red line) the expression values of that gene in all cells at the magnitude of the $p = 0.001$ quantile of that gene (blue line). **B** Preserving biological zeros in genes known to be specific to each cell population. **C** Biological zeros were defined as genes which should not be expressed in a cell type (based on bulk RNA-seq). The ratio of these biological zeros preserved (ZP) is shown alongside the ratio of total zeros completed (TC). **D** Percentage of biological zeros preserved (left) and technical zeros completed (right) in a simulated scRNA-seq dataset as sequencing depth increases. DCA not shown for (C) and (D), as it imputes every zero to a positive number.
The importance of preserving biological zeros. A cell type or cellular process is often characterized by both positive marker genes (genes expressed by the cells) and negative marker genes (genes not expressed by the cells). Imputation methods that do not preserve biological zeros result in a matrix where every gene is expressed by every cell, requiring the practitioner to manually threshold the expression values to determine which cells express or do not express the markers. This thresholding can be arbitrary and lead to erroneous results. In contrast, ALRA does not require this thresholding step, as cells not expressing the gene will not be falsely imputed to non-zero values. We demonstrate the importance of preserving biological zeros using three real-life examples.

Gupta et al. recently found that Wnt signalling is necessary for hair follicle dermal condensate (DC) differentiation, an essential step for hair follicle development. This was established in mouse embryonic skin by conditionally knocking out beta-catenin, the key transducer of the Wnt pathway, and comparing the expression of DC marker genes with control using scRNA-seq. We imputed these expression values using ALRA and compared the ratio of cells expressing Sox2 (a marker for differentiated DC cells) and the ratio of cells expressing Axin2/Lef1 (markers for Wnt signalling) between the knockout and control to show that only dermal cells that retained Wnt activity were able to acquire DC cell fate. ALRA’s ability to preserve the biological zeros allows us to quantitatively compare the number of cells expressing the marker genes (Supplementary Table 1). The ratio of the proportions of cells expressing Sox2 and Axin2/Lef1 between the knockout and control is similar to the ratio computed from the pre-imputed data. In particular, ALRA preserves the conclusion that Wnt signalling was conditionally knocked out in a proportion of cells within the mutants, and that there was a corresponding impairment in DC differentiation caused by the absence of Wnt signalling. In contrast, MAGIC and DCA both output a matrix with no zeros, and hence, this proportion cannot be evaluated (Supplementary Figs. 2 and 3). Furthermore, based on their density plots and t-SNEs, it is not clear how to threshold the imputed values to restore the zeros (Supplementary Figure 3). SAVER’s results falsely suggest that the Wnt activity in the knockout and control is identical, whereas scImpute falsely suggests that DC differentiation is similar between knockout and control (Supplementary Table 1). In settings where lack of expression of a marker gene is used to characterize a population or has other biological importance, existing methods that do not preserve biological zeros are unsuitable.

Next, we demonstrate that ALRA allows practitioners to draw upon a rich literature of positive and negative markers for known cell types. In Supplementary Fig. 4, we demonstrate this application by identifying known subtypes of PBMCs from Zheng et al. using well-known markers (closely based on the Human Blood Cell Atlas, see methods). This is not possible using the raw data, as very few (if any) cells would express all the positive markers. But existing imputation methods that do not preserve biological zeros would require an arbitrary thresholding step to determine if a cell is “positive” or “negative” for the given gene. Effectively, ALRA allows for automatic gating of cells based on well-known marker genes, which can be especially useful for distinguishing cell subtypes within a cluster (e.g., Naïve B cells and Memory B cells).

In a third example, we apply ALRA to human bronchial epithelial cells from Lukassen et al. These authors sought to identify cells that expressed three SARS-CoV-2 entry proteins, ACE2, TMPRSS2, and FURIN. Their analysis was hampered by dropout, as only three cells were triple-positive. Using ALRA, we can identify 1309 of these triple-positive cells (Supplementary Fig. 5). When imputing with methods that do not preserve biological zeros, cells that are not actually triple-positive are imputed to express all three genes, hence requiring a thresholding step to set these incorrect values to zero. This thresholding is far from trivial, as different thresholds can lead to dramatically different results.
ALRA improves the separation of cell types. We next evaluated ALRA’s ability to improve the separation of cell types after synthetically introducing zeros. As in Huang et al., we downsampling the reads of Zeisel et al., Baron et al., Chen et al., and La Manno et al. We showed that after imputation, cell types are more clearly separated in the t-SNE (Fig. 3A and Supplementary Figs. 6 to 8). To quantitatively confirm that ALRA increases separation, we trained supervised random forests to classify the cells into cell types based on expression values, and the out-of-bag classification error is shown in each plot. B The effect of imputation on separation of previously annotated cell types in mouse visual cortical cells from Hrvatin et al. and in mouse neocortical cells from Tasic et al. Unlike in (A), the datasets were not downsampled prior to imputation as they already had high dropout rates. Additionally, we evaluate ALRA using t-SNE on two datasets with cells that the authors classified into distinct cell types. We applied ALRA, DCA, and MAGIC to the 65,539 mouse visual cortex cells from Hrvatin et al. and 21,874 neocortical cells from Tasic et al. Figure 3 compares the t-SNE before and after imputation. After imputation by ALRA, several cell types in the Hrvatin et al. dataset are more clearly separated. While on the Tasic et al. dataset ALRA does not seem to significantly improve cell-type separation, at least it does not decrease separation, which DCA appears to do. We computed the out-of-bag classification error as before, and found that ALRA yielded an improved error from 6.3 to 2.3% in Hrvatin et al., and down from 10.5 to 7.7% in Tasic et al., confirming that ALRA improves the separation of the cell types. With their current implementations, scImpute and SAVER do not scale well to large
datasets and hence were not run on these matrices (see later for scalability benchmarks).

Further validation of ALRA’s imputation. Having shown that ALRA successfully preserves biological zeros and improves the separation of cell types, we further validated ALRA performance with several additional analyses. First, we used the CITE-seq cord blood mononuclear cell (CBMC) dataset of Stoeckius et al. to show that after imputation, the RNA measurements of each gene are more consistent with the corresponding protein measurements (measured as antibody-derived tag (ADT)). We computed the $k$ nearest neighbors of each cell (for $k = 50, 100, 1000$) in the ADT space and also in the RNA space before and after imputation. ALRA nearly doubled the number of nearest neighbors that are consistent between the RNA and protein measurements (Supplementary Table 3). We also computed the pairwise distances between cells on the gene expression and on the protein expression data. We plotted the normalized distances between cells in gene expression data before and after imputation against the distances between cells in the protein expression data to show that ALRA improved their Spearman’s correlation (Supplementary Fig. 9) from 0.36 to 0.64.

As in Huang et al., we further evaluated our approach by comparing ALRA’s completion of Drop-seq data to RNA fluorescence in situ hybridization (FISH) of the same population of cells. In Torre et al. (2018), 8498 cells from a melanoma cell line were sequenced by Drop-seq. DNA FISH was also used to measure the expression of 26 markers in cells of the same cell line. Although the gene expression profiles of individual cells cannot be compared across these two datasets, the distribution of gene expression should be consistent, because the cells were sampled from the same cell line. Huang et al. (2018) applied SAVER to the Drop-seq data and demonstrated that, as compared to the raw data, the estimated distribution of expression values is more consistent with the FISH distributions. Their analysis computes the Gini coefficient of each gene, a measure of gene expression variability, which should be consistent across the Drop-seq and FISH technologies. Specifically, they showed that the Gini coefficients of a subset of marker genes in the FISH dataset were more consistent (correlation 0.6) with the SAVER estimates than either the raw Drop-seq data (correlation 0.3) or other completion methods (Supplementary Fig. 12). Our approach outperforms all other methods on this metric, obtaining a correlation of 0.8.

Next, we further evaluated ALRA using the downsampled datasets from before (Baron et al., Chen et al., La Manno et al., and Zeisel et al.). We computed the correlation of each gene/cell in the original dataset with the same gene/cell in the downsampled dataset before and after imputation by ALRA. We found that despite improving the clustering (see above), ALRA does not increase this correlation (Supplementary Fig. 15B). We attribute this to the presence of strong noise in the raw data which is not present after denoising using ALRA. To check if ALRA is preserving the signal, we computed the top $k$ PCs (where $k$ was chosen using the jackstraw method, see methods) of the original dataset with the top $k$ PCs of the downsampled dataset before and after imputation. We found that ALRA decreases the subspace angle between these signal PCs, from 0.34 to 0.25 for Baron et al., 0.30 to 0.17 for Chen et al, 0.36 to 0.24 for La Manno et al, and 0.21 to 0.15 for Zeisel et al. Thus ALRA effectively increases the correlation between the signals of the subsampled data and the original data. Additionally, the correlation of the imputed data with the original raw data (Supplementary Fig. 15B) is not a useful metric as it has a strong dependence on the noise of the original data.

We next focused on ALRA’s ability to preserve subtle differences in expression values. A recent study uncovered the existence of two major lineages of fibroblasts in the intestine characterized by differential Pdgfra gene expression levels: $Pdgfra_{\text{high}}$ and $Pdgfra_{\text{low}}$ fibroblasts. We tested whether ALRA would preserve this expression pattern by performing an independent scRNA-seq experiment and sequencing fewer cells, thus generating a smaller dataset to be completed by ALRA and other methods. We isolated mesenchymal/lamina propria cells from the colon of $Pdgfra_{\text{Egfp/+}}$ knock-in mice and used flow cytometry to confirm the presence of two major populations with non-zero $Pdgfra$-EGFP expression: $Pdgfra_{\text{low}}$ cells and $Pdgfra_{\text{high}}$ cells (Supplementary Fig. 16). After imputation by ALRA, a Pdgfra expression pattern very similar to that observed by flow cytometry can be clearly seen. In contrast, other methods do not show two distinct levels of Pdgfra expression (Supplementary Fig. 17).

Finally, using the same simulated dataset as before, we computed the correlation between each cell’s expression profile and the bulk RNA-seq profiles for the 9 immune cell populations they were generated from. If the correlation between a cell and its corresponding bulk RNA-seq sample is larger than the others, we say it is correctly classified. We would expect that after imputation, the correlation between a cell and its corresponding bulk RNA-seq sample would be clearly larger than its correlation to the other samples, and hence the classification rate would improve. As shown in Supplementary Fig. 10A, most cells are classified correctly in the original data. ALRA successfully increases the correlation of each cell with its corresponding bulk RNA-seq sample more than with the other cell-type bulk profiles. DCA, MAGIC, scImpute, and SAVER, in contrast, do not improve the correlations; in fact, the misclassification error is higher after matrix completion using these methods.

We also noticed that ALRA appears to perform best on datasets that include a diversity of cell types. In the presence of diverse cells, the leading singular values, corresponding to biological variation, are large relative to the technical noise. Consequently, the corresponding singular vectors can be more accurately estimated, resulting in better imputation results. When we applied ALRA to the individual purified PBMC populations from above (Fig. 2), it did not preserve the biological zeros as effectively as when we applied ALRA to the merged matrix. In the mouse intestine cells, we similarly found that when ALRA is applied to a biological replicate that had the epithelial cells removed, the two populations of fibroblasts did not have as clearly distinguishable Pdgfra expression (Supplementary Fig. 18). For this reason, we recommend using ALRA on the full dataset, as opposed to purified subsets.

ALRA is highly scalable. ALRA only requires a single rank-$k$ SVD of the observed matrix, which is very fast to compute. Furthermore, it can scale to huge matrices with modern algorithms and software. In Fig. 4A we compare the runtime of ALRA, MAGIC, SAVER, DCA, and scImpute on subsets of mouse visual cortex cells from Hrvatin et al. The resulting matrices ranged from 1000 to 50,000 cells, each with 19,155 genes, and all methods were restricted to using only a single core. ALRA and MAGIC are significantly faster than the other methods, taking ~40 min on 50,000 cells. DCA requires about 4 h to run on the same number of cells, whereas scImpute and SAVER both take over 10 h for 10,000 cells and were not attempted on larger matrices. We emphasize that these are single-core times, several of the above methods can multithread, substantially improving performance on machines with multiple cores. ALRA also supports multithreading via the Intel Math Kernel Library.
Methods

ALRA procedure: Given a measured expression matrix, as outlined in Algorithm 1, ALRA consists of the following steps. First, as commonly done with scRNA-seq data, the first step is standard pre-processing of the expression matrix by library normalization and log-transformation. Specifically, we normalize each column to \( a = 10,000 \) transcripts per cell, add 1 to every entry (i.e., a “pseudocount”), and take the logarithm of each entry to obtain the normalized expression matrix. It has been noted that different values of \( \alpha \) can affect downstream results\(^{30}\), but we show that ALRA’s preservation of biological zeros is not dependent on this choice (Supplementary Tables 6 to 9, Supplementary Fig. 19). As discussed in the Appendix, a variance stabilizing transform is necessary for good downstream results, and we chose the log-transform due to its widespread use. Other kinds of normalization can also be used.

After normalization, the next three steps are as follows: (i) Estimate the rank \( k \) and (ii) compute the rank-\( k \) SVD using randomized SVD and (iii) threshold each gene of the rank-\( k \) SVD by the absolute value of that gene’s \( p = 0.001 \) quantile. Finally, we rescale the resulting values such that the mean and standard deviation of the non-zero values of each gene in the resulting matrix match those of the normalized matrix (prior to low-rank approximation). A small number of non-zero values in the normalized matrix may be smaller than the threshold and may thus get set to zero by the thresholding operation in step (iii) of ALRA. For improved consistency with the observed matrix, we restore these entries to their original normalized values. We note that this process not only imputes technical zeros, but also denoises the originally non-zero values. An intuitive measure of how much the non-zero values were adjusted is the Pearson correlation between these values before and after ALRA (e.g., for the purified PBMCs and Hrvatin et al. datasets, the correlations were 0.56 and 0.23, respectively).

Implementing ALRA using standard SVD procedures in common software may be rather slow and not scale to large matrices. To allow ALRA to scale to matrices with hundreds of thousands to millions of cells, we apply recently developed methods for randomized SVD\(^{31}\). Randomized SVD can be used to compute the leading singular values and singular vectors of large matrices with high accuracy yet typically manifests as a long tail of singular values whose singular vectors point to statistical noise. The goal is to choose an appropriate threshold \( k \) computed in a data-driven manner.

Motivated by this work, here we also propose a similar approach. Noise typically manifests as a long tail of singular values whose singular vectors point to random directions in space with no scientific significance. The goal is to choose \( k \) such that the singular values \( \sigma_1, \ldots, \sigma_k \) correspond to biological variability of interest or signal, whereas the remaining values are noise. We assume an upper bound \( k < 100 \), and thus first use rsvd to compute the top 100 singular values. For this computation to be fast, we apply rsvd with the default setting of only \( q = 2 \)

(MKL) by passing the ‘use.mkl’ flag to the ALRA function, but we do not compare multithreaded performance here. ALRA’s scalability is especially crucial for the imputation of modern scRNA-seq datasets, which can exceed 1 million cells (e.g., ref. 29).

In this paper, we present ALRA, an efficient imputation method for scRNA-seq data. ALRA’s ability to preserve the vast majority of biological zeros rest on a solid theoretical foundation and its overall performance has been validated across simulated and real scRNA-seq datasets. ALRA improves the separation of cell types in both t-SNE and the original high dimensional space, imputes values consistent with external measurements, and computationally scales better than other methods. We also note that ALRA has only one parameter, the approximate rank \( k \) of the matrix, which is selected automatically based on statistics of the spacings between consecutive singular values.

Rank estimation. As described above, the only parameter in ALRA is the rank \( k \) of low-rank approximation. Estimating the rank of a matrix, given a noisy and/or corrupted version of it, is a classical and well-studied problem, see for example refs. 33–36. Most works on rank estimation, however, assume that all entries of the matrix are observed, often with each entry corrupted by additive noise. In ref. 37, in an econometric context, an estimator was based on the spacing of eigenvalues. Specifically, the estimated rank is the largest value of \( k \) for such that \( \lambda_k - \lambda_{k-1} > \delta \) for an appropriate threshold \( \delta \) computed in a data-driven manner.

Motivated by this work, here we also propose a similar approach. Noise typically manifests as a long tail of singular values whose singular vectors point to random directions in space with no scientific significance. The goal is to choose \( k \) such that the singular values \( \sigma_1, \ldots, \sigma_k \) correspond to biological variability of interest or signal, whereas the remaining values are noise. We assume an upper bound \( k < 100 \), and thus first use rsvd to compute the top 100 singular values. For this computation to be fast, we apply rsvd with the default setting of only \( q = 2 \)

Algorithm 1. Adaptively thresholded Low-Rank Approximation

Input: Raw expression matrix \( X \), number of rows (genes) \( m \), number of columns (cells) \( n \)
Output: Imputed expression matrix \( Y \)

Step 0. Pre-processing

Normalise each column to \( 10,000 \) transcripts: \( X_{ij} \leftarrow 10,000 \times \frac{X_{ij}}{\sum_{j=1}^n X_{ij}} \)

Add a pseudocount and take the logarithm: \( X_{ij} \leftarrow \log(X_{ij} + 1) \)

Step 1. Estimate \( k \).

Compute the rank-100 SVD \( \sum_{i=1}^{100} \sigma_i u_i^T v_i \) of \( X \).

Define the spacings \( \sigma_i = \sigma_i - \sigma_{i+1} \).

Compute the mean \( \mu \) and standard deviation \( \sigma \) of \( \sigma_1, \ldots, \sigma_q \).

Let \( k = \max \{ j \mid \sigma_j > \mu + 6 \sigma \} \).

Step 2. Imputation by low-rank approximation.

Compute the rank-\( k \) randomized SVD \( Y = \sum_{i=1}^{k} u_i^T \sigma_i v_i \) of \( X \).

Step 3. Restoration of biological zeros.

For \( i = 1 \) to \( m \) do

Let \( \sigma_i \) denote the \( p = 0.001 \) quantile of \( Y_i \).

For \( j = 1 \) to \( n \) do

If \( Y_{ij} < \sigma_i \), then \( Y_{ij} = 0 \).

end

end

Step 4. Shifting and rescaling.

For \( i = 1 \) to \( m \) do

Let \( \mu_i, \sigma_i \) denote the mean and standard deviation of the non-zero entries of \( X_i \).

Let \( \eta_i \) denote the mean and standard deviation of the non-zero entries of \( Y_i \).

Shift and rescale the imputed values

\[ Y_i = \left( Y_i - \eta_i \right) \frac{\mu_i}{\sigma_i} + \mu_i \]

end

Fig. 4 Runtime of ALRA versus other imputation methods using expression matrices of \( N \) cells and \(-19,000\) genes. All experiments performed on a single core.
additional power iterations. Next, we compute the spacings between consecutive singular values, denoted as $s_j = \sigma_j - \sigma_{j-1}$ for $j = 2, 3, \ldots, 100$. As can be seen in the plots of the two datasets (Fig. 3), the majority of the spacings for the observed matrix of the symmetrically distributed around zero, asymmetry in this distribution is due to threshold and hence kept as zeros. This is why we consider ALRA to be a conventional phase transition, whereby weak signals cannot be reliably detected by SVD or PCA, and implicate that ALRA is relatively insensitive to the specific choice of a rank $k$; the number of biological zeros preserved does not significantly change with $k$ (Supplementary Table 4). Similarly, ALRA's improved separation of cell types is also robust to the exact value of the rank (Supplementary Fig. 13).

### Excess zeros

After low-rank approximation, ALRA restores biological zeros by thresholding each gene based on the distribution of negative values. This allows the user to be confident that any imputed values do not correspond to biological zeros. However, values corresponding to technical zeros will inevitably be below the threshold and hence kept as zeros. This is why we consider ALRA to be a conventional phase transition, whereby weak signals cannot be reliably detected by SVD or PCA, and implicate that ALRA is relatively insensitive to the specific choice of a rank $k$; the number of biological zeros preserved does not significantly change with $k$ (Supplementary Table 4). Similarly, ALRA's improved separation of cell types is also robust to the exact value of the rank (Supplementary Fig. 13).

#### Assumptions, theory, and relation to low-rank matrix completion

The key assumption underlying ALRA is that the true scRNA-seq data matrix is low-rank, with the expression values lying in a linear subspace of dimension significantly lower than the number of cells or genes. It has been appreciated that many biological effects do not act independently, but rather in concert, forming groups of highly correlated genes often referred to as gene modules. Therefore, the true expression matrix is often modelled as a low-rank matrix. Due to this assumed low-rank structure, indeed nearly every scRNA-seq pipeline first reduces dimensionality to the top few principal components (e.g., ref. 1). At first sight, imputing an scRNA-seq data matrix containing many zeros and assumed to be of low rank, may look like a particular instance of a low-rank matrix completion (LRMC) problem. This may seem a promising venue as over the past 10 years, a rich theory and several methods have been developed to complete missing values in low-rank matrices (e.g., refs. 43–45). However, as we now explain, imputing scRNA-seq data are a different problem. Specifically, in the low-rank matrix completion literature, the assumption is that the set of indices of the observed entries are perfectly known, and have been sampled either uniformly at random, or as in ref. 46, with probability proportional to the leverage scores of the matrix, under the LRMC model, and some ad hoc incoherence assumptions, LRMC methods can exactly recover the true low-rank matrix with high probability. In scRNA-seq, in contrast, the precise indices where the values of the matrix were sampled are not perfectly known, since some of the observed zeros are true zeros of the underlying data matrix. Namely, the missing entries are some unknown subset of the observed zeros. Direct application of an LRMC method to scRNA-seq data (e.g., as in refs. 47,48) while treating all the zeros as missing implies a highly biased sampling, which consequently may lead to incorrect imputation of the matrix. Moreover, the theoretical guarantees found in the LRMC literature do not apply to this biased sampling scheme.

To account for zero preservation for scRNA-seq data, our method thus relies on a different yet simple observation: In datasets where the biological zeros were known, the non-zero values recovered by a low-rank SVD reconstruction of the observed matrix were symmetrically distributed around zero. In the supplement, we present the rigorous theoretical foundation for this empirical observation, relying on a perturbation analysis together with results from random matrix theory. Specifically, we prove that under certain assumptions, the distribution of these entries is indeed Gaussian with zero mean. Therefore, for each gene, any element larger in magnitude than the threshold is unlikely to be a biological zero. Conversely, values smaller in magnitude than this threshold may be biological zeros, since if not all technical zeros are set as zeros. This is why we consider ALRA to be a conventional phase transition, whereby weak signals cannot be reliably detected by SVD or PCA, and implicate that ALRA is relatively insensitive to the specific choice of a rank $k$; the number of biological zeros preserved does not significantly change with $k$ (Supplementary Table 4). Similarly, ALRA's improved separation of cell types is also robust to the exact value of the rank (Supplementary Fig. 13).

#### Evaluation zero preservation with bulk RNA-seq data

To comprehensively evaluate the ability of various imputation methods to preserve biological zeros, we used bulk RNA-seq data to define biological zeros. In ref. 1, several different types (B cells, monocytes, T cells, and NK cells) of peripheral blood mononuclear cells (PBMCs) were analyzed with bulk RNA-seq, with two biological replicates for each.
For each cell type, genes with zero counts in both replicates were defined as "biological zero genes" for this specific cell type. In the scRNA-seq data of the corresponding cell type, we calculate the zero preservation rate as a number of zero entries in all biological zero genes after imputation, divided by a number of zero entries in all biological zero genes before imputation.

**CITE-seq data analysis.** Cellular Indexing of Transcriptomes and Epitopes by sequencing (CITE-seq)\(^1\) is a method for simultaneous epitope and transcriptome measurement in single cells. We obtained the PBMC dataset from GEO (GSE100866 [https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100866]) and proceeded to identify cell types based on their surface protein markers. After filtering cells for which more than 90% of the UMIs came from mouse genes and removing genes that had expression in less than 10 cells, there were 7667 cells and 13,458 genes. Surface protein markers are measured by counting antibody-derived tags (ADT) on each cell, and gene expression is measured in the same cells using scRNA-seq. We used the ADT counts to determine cell type by "gating" the cells by their marker epitopes. We used markers CD3, CD19, and CD14 to define T cells, B cells, and monocytes, respectively. Specifically, we applied \(k\)-means with \(k = 2\) to transformed ADT counts for each marker individually in order to classify cells as expressing a marker epitope or not. Then, we classify cells into types as follows: CD3+CD19−CD14− T cells, CD3−CD19+CD14− B cells, and CD3−CD19−CD14+ monocytes. We ran ALRA and the other imputation methods on the scRNA-seq data; the rank for ALRA was estimated to be \(k = 22\) using the procedure described above.

We obtained the cord blood mononuclear cell (CBMC) dataset from the same paper and imputed the data using ALRA and the other imputation methods. After filtering cells for which more than 90% of the UMIs came from mouse genes, there were 8005 cells and 20,400 genes. For ALRA, the rank was estimated to be \(k = 25\). We used CLR-transformed epitope ADT counts and calculated the correlation between each protein marker and its corresponding gene, before and after imputation. We also calculated the correlation between each protein marker and 1000 random genes and compared the correlation of each protein marker and its expression in the same cells using scRNA-seq. We used the ADT counts to determine cell type by "gating" the cells by their marker epitopes. We used markers CD3, CD19, and CD14 to define T cells, B cells, and monocytes, respectively. Specifically, we applied \(k\)-means with \(k = 2\) to transformed ADT counts for each marker individually in order to classify cells as expressing a marker epitope or not. Then, we classify cells into types as follows: CD3+CD19−CD14− T cells, CD3−CD19+CD14− B cells, and CD3−CD19−CD14+ monocytes. We ran ALRA and the other imputation methods on the scRNA-seq data; the rank for ALRA was estimated to be \(k = 22\) using the procedure described above.

**Processing of mouse dermal cells.** Raw count scRNA-seq data of E14.5 Wild Type (WT: two replicates) and Loss of Function (LOF: one replicate) was obtained from\(^4\). For each sample, we used Seurat to perform data normalization, scaling, variable gene selection, PCA, clustering, and t-SNE embedding calculation. Cells with expression in less than 1000 genes were filtered out. The ColIα1 gene was used to select dermal clusters, and only cells in clusters with high expression (>85% expressing cells) of ColIα1 were retained for further analysis. The resulting matrix had 11,121 dermal cells and 28,000 genes. After the selection of dermal cells, we first merged the two biological replicates for the WT samples, then used canonical correlation analysis (CCA) to merge data from WT and LOF and removed batch effect. The top 15 CCA components were used for t-SNE calculation. The expression matrices were normalized as in the Purified PBMC analysis detailed above. The rank for ALRA was estimated to be \(k = 49\) using the procedure described above.

**Identification of PBMCs using known markers.** The 68k peripheral blood mononuclear cell dataset was downloaded from 10X Genomics website\(^6\). Cells with expression in less than 400 genes were filtered out, and genes with expression in less than 6 cells were filtered out. The resulting dataset had 58,674 cells and 16,137 genes. The rank for ALRA was estimated to be \(k = 19\) using the procedure described above, and ALRA was run on the dataset. Positive and negative markers were determined based on ref.\(^{19}\) and supplemented with expert knowledge (Supplementary Table 5). If a cell expressed all positive markers and no negative markers after imputation, then it was identified as belonging to that corresponding class.

**FISH experiments.** We obtained the Drop-seq and RNA FISH datasets from\(^{27}\). The Drop-seq data consists of 8640 cells and 32,287 genes. The RNA FISH consists of 26 genes across 7000–88,000 cells (depending on the gene). To directly compare with analysis published by the authors of SAVER, we followed their same filtering procedure, removing cells with library size greater than 20,000 or less than 500, and also removing genes with mean expression less than 0.1. The resulting dataset contains 8498 cells and 12,241 genes, with 16 genes shared between Drop-seq and FISH. As in Huang et al., cells in the bottom and top tenth percentiles of the housekeeping gene GAPDH were filtered out, and all genes were then normalized.
by GAPDH expression. We then computed the Gini coefficient of each of the other 15 genes (excluding GAPDH) in RNA FISH, original Drop-seq, and Drop-seq after imputation, using this filtered subset (Supplementary Fig. 12). The rank selected by ALRA was $k = 14$.

**Runtime comparison.** We sampled random subsets of sizes 5000, 10,000, 20,000, 30,000, 40,000, and 50,000 cells from Hrvatin et al. and compared the runtime of ALRA was P5_TSO_Hybrid657 and Nextera XT primers, N701 and N702 (Illumina). Both beads (Agencourt). The cDNA from an estimated 1500 cells was prepared and amplified as described by Macosko et al. to generate a digital expression matrix with 

**Intestinal fibroblast experiments.** PdgfraEGFP+ mice were obtained from the Jackson Laboratory and bred in the facilities of the Yale Animal Resources Center and maintained on a C57BL/6J genetic background. Mice were housed in standard cages, on a 12 h day/night cycle and were fed a standard rodent chow. All animal experimentation was performed in compliance with Yale Institutional Animal Care and Use Committee protocols.

**Isolation of mouse intestinal mesenchymal cells.** The intestine was dissected, flushed, opened longitudinally, and then cut into 1 cm pieces. The tissues were incubated in HBSS containing 1 mM EDTA, 1 mM DTT, 0.2% FBS, 4°C, 10 min each, at 37°C, 200 rpm. Epithelial cells were released by vigorous shaking. After epithelial cell removal, the remaining stromal part of the intestine was incubated in DMEM 10% FBS containing Collagenase XI (300 units/ml, Sigma, C7657), Dispase II (0.1 mg/ml, Sigma, D4693), and DNase II Type V (50 units/ml, Sigma, D8764) for 1 h, at 37°C, 200 rpm. Cells released after vigorous shaking were passed through a 70 μm strainer and washed with 2% sorbitol. Such cell preparations were directly processed by Drop-seq or by flow cytometry.

**Flow cytometry in intestinal mesenchymal cells.** Flow cytometry analysis was performed at the Yale Flow Cytometry Facility. Freshly isolated intestinal mesenchymal cells from PdgfraEGFP+ mice were analyzed for direct EGFP fluorescence with a BD LSRII cytometer equipped with FACSDiva software. Cells isolated from the intestine of WT mice were used as a non-fermenting control. Data analysis was performed with the FlowJo software. Single cells were selected first by a FSC-H vs FSC-A and then by a SSC-H vs SSC-A gate and then a FSC-A vs SSC-A gate was applied.

**Drop-seq in intestinal mesenchymal cells.** Mesenchymal cells were isolated from the intestine of WT C57BL/6J mice ($n = 2$). The cells were diluted to a concentration of 100 cells/ml and 1 ml aliquots were used as input to the Drop-seq protocol which was performed as described previously42,57 with minor modifications. The beads were purchased from ChemGenes Corporation, Wilmington MA (catalog number Macosko201110) and the PDMS co-flow microfluidic droplet generation device was generated by NanoShift, Emeryville CA. For both conditions, the 1 ml DropSeq collection was performed. Samples were processed for cDNA amplification within ~15 min of collection. Populations of 5000 beads (~150 cells) were separately amplified for 15 cycles of PCR (conditions identical to those previously described) and pairs of PCR products were co-purified by the addition of 0.6x AMPure XP beads (Beckman Coulter). The cDNA from an estimated 1500 cells was prepared and tailed by Nextera XT using 1000 pg of cdna input, and the custom primers P5_TSO_Hybrid57 and Nextera XT primers, N701 and N702 (ILLUMINA). Both libraries were sequenced on the Illumina NextSeq 500 using 2.0 PM in the volume of 1.3 ml HT1, and 2 ml of 0.3 μM Read1 CustomSeqB676 for priming of read 1. Read 1 was 20 bp; read 2 (paired-end) was 60 bp. Single-cell RNA-seq data were processed as described by Macosko et al. to generate a digital expression matrix with transcript count data. This matrix was filtered retaining cells with more than 1000 transcripts and less than 10% mitochondria transcripts. The expression matrices were normalized as in the Purified PBMC analysis described above, except that SAVER was applied to all genes, and hence it was also library normalized (unlike in the purified PBMC analysis).

**Code availability**

An R implementation of ALRA is freely available at https://github.com/KlugerLab/ALRA. The codes for analyses this paper are available at https://github.com/KlugerLab/ALRA-paper.

**References**

1. Svensson, V. Droplet scRNA-seq is not zero-inflated. *Nat. Biotechnol.* 38, 147–150 (2020).
2. Silverman, J. D., Roche, K., Mukherjee, S., David, L. A. Naught all zeros in sequence count data are the same. *Comput. Struct. Biotechnol. J.* 18, 2789–2798 (2020).
3. Ramsköld, D., Wang, E. T., Burge, C. B. & Sandberg, R. An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data. *PLoS Comput. Biol.* 5, e1000598 (2009).
4. Gupta, K. et al. Single-cell analysis reveals a hair follicle dermal niche molecular differentiation trajectory that begins prior to morphogenesis. *Dev. Cell* 48, 17–31 (2019).
5. 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).
6. van Dijk, D. et al. Recovering gene interactions from single-cell data using data distillation. *Cell* 174, 716–729.e27 (2018).
7. Huang, M. et al. Saver: gene expression recovery for single-cell rna sequencing. *Nat. Methods* 15, 539 (2018).
8. Li, W. V. & Li, J. J. An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nat. Commun.* 9, 997 (2018).
9. X.Y. Zheng, G. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* 8, 14049 (2017).
10. Torlakovic, E., Torlakovic, G., Nguyen, P. L., Bruning, R. D. & Delabie, J. The value of anti-pax-5 immunostaining in routinely fixed and paraffin-embedded sections: a novel pan pre-b and b-cell marker. *Am. J. Surg. Pathol.* 26, 1343–1350 (2002).
11. Van Acker, H. H., Caspomidis, A., Smits, E. L. & Van Tendeloo, V. F. Cd56 in the immune system: more than a marker for cytotoxicity? *Front. Immunol.* 8, 892 (2017).
12. Germain, R. N. T-cell development and the cd4–cd8 lineage decision. *Nat. Rev. Immunol.* 2, 309–322 (2002).
13. Ohkawa, T. et al. Systematic characterization of human cd8+ t cells with natural killer cell markers in comparison with natural killer cells and normal cd8+ t cells. *Immunology* 103, 281–290 (2001).
14. Hoek, K. L. et al. A cell-based systems biology assessment of human blood to monitor immune responses after influenza vaccination. *PloS ONE* 10, e0118528 (2015).
15. Stoeckius, M. et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* 14, 865 (2017).
16. Eng, T. S. P. et al. The immunological genome project: networks of gene expression in immune cells. *Nat. Immunol.* 9, 1091 (2008).
17. Sun, Z. et al. Dimm-sc: a dirichlet mixture model for clustering droplet-based single cell transcriptomic data. *Bioinformatics* 34, 139–146 (2017).
18. Heimberg, G., Bhatnagar, R., El-Samad, H. & Thomson, M. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell Syst.* 2, 239–250 (2016).
19. Uhlenet, M. et al. A genome-wide transcriptomic analysis of protein-coding genes in human blood cells. *Science* 366, 6472 (2019).
20. Lükassen, S. et al. Sars-cov-2 receptor ace 2 and imprs 2 are primarily expressed in bronchial transient secretory cells. *EMBO J.* 39, e115114 (2020).
21. Zeisel, A. et al. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science* 347, 1138–1142 (2015).
22. Baron, M. et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst.* 3, 346–360 (2016).
23. Chen, R., Wu, X., Jiang, L. & Zhang, Y. Single-cell rna-seq reveals hypothalamic cell diversity. *Cell Rep.* 18, 3227–3241 (2017).
24. Manno, G. L. et al. Molecular diversity of midbrain development in mouse, human, and stem cells. *Cell 167*, 566–580 (2016).
25. Hrvatin, S. et al. Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex. *Nat. Neurosci.* 21, 120 (2018).
26. Tasic, B. et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature* 563, 72 (2018).
27. Torre, E. Rare cell detection by single-cell rna sequencing as guided by single-molecule rna fish. *Cell Syst.* 6, 171–179 (2018).
28. Roulis, M. et al. Paracrine orchestration of intestinal tumorigenesis by a mesenchymal niche. Nature 580, 524–529 (2020).
29. 10X Genomics. Transcriptional profiling of 1.3 million brain cells with the chromium single cell 3' solution. Application Note. https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3M_neurons. (2016).
30. Lun, A. Overcoming systematic errors caused by log-transformation of normalized single-cell rna sequencing data. BioRxiv https://www.biorxiv.org/content/10.1101/404962v1 (2018).
31. Halko, N., Martinsson, P.-G. & Tropp, J. A. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. 53, 217–288 (2011).
32. Ericson, N. B., Voronin, S., Brunton, S. L. & Kutz, J. N. Randomized matrix decompositions using R. J. Stat. Softw. 89, 1–48 (2019).
33. Cotrell, B. B. The screen test for the number of factors. Multivar. Behav. Res. 1, 245–276 (1966).
34. Kritchman, S. & Nadler, B. Determining the number of components in a genome-wide expression data processing and modeling. Proc. Natl. Acad. Sci. USA 97, 10101–10106 (2000).
35. Cleary, B. et al. Compressed sensing for highly efficient imaging transmics. Nat. Biotechnol. 39, 936–942 (2021).
36. Kapur, A., Marwah, K. & Alterovitz, G. Gene expression prediction using low-rank matrix completion. BMC Bioinformatics 17, 243 (2016).
37. Onatski, A. Determining the number of factors from empirical distribution of eigenvalues. Rev. Econ. Stat. 92, 1094–1106 (2010).
38. Alter, O., Brown, P. O. & Botstein, D. Singular value decomposition for data analysis. Ann. Stat. 30, 794–815 (2002).
39. Borchers, H. W. & Ziegler, A. ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. J. Stat. Soft. 77, 1–17 (2017).
40. Borchers, H. W. pracma: Practical numerical multi functions. r package version 2.0. 7 https://cran.r-project.org/web/packages/pracma/pracma.pdf (2017).
41. Heath, T., Klingshofer, R. A., Corrin, P. D. & Soriano, P. Evolutionary divergence of platelet-derived growth factor alpha receptor signaling mechanisms. Mol. Cell. Biol. 23, 4013–4025 (2003).
42. Shekhar, K. et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. Cell 166, 1308–1323 (2016).
43. Candès, E. J. & Recht, B. Exact matrix completion via convex optimization. Found. Comput. Math. 9, 717 (2009).
44. Candès, E. J. & Tao, T. The power of convex relaxation: near-optimal matrix completion. IEEE Trans. Inf. Theory 56, 2053–2080 (2010).
45. Keshavan, R. H. & Montanari, A. Matrix completion from a few entries. In Information Theory, 2009. ISIT 2009. IEEE International Symposium on, pages 324–328. (IEEE, 2009).
46. Chen, Y., Bhojanapalli, S., Sanghavi, S. & Ward, R. Completing any low-rank matrix, provably. J. Mach. Learn. Res. 16, 2999–3034 (2015).
47. Chen, C., He, B. & Yuan, X. Matrix completion via an alternating direction method. IMA J. Numer. Anal. 32, 227–245 (2012).
48. Mitra, S. & Zou, H. ROAST: Robust Orthogonal Array-Based Spline Transform. BCB 2010 636, 1217 (2010).
49. Wright, M. N. & Ziegler, A. ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. J. Stat. Soft. 77, 1–17 (2017).
50. Borchers, H. W. pracma: Practical numerical multi functions. r package version 2.0. 7 https://cran.r-project.org/web/packages/pracma/pracma.pdf (2017).
51. Hamilton, T. G., Klingshofer, R. A., Corrin, P. D. & Soriano, P. Evolutionary divergence of platelet-derived growth factor alpha receptor signaling mechanisms. Mol. Cell. Biol. 23, 4013–4025 (2003).
52. Muloscoz, K. et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 161, 1202–1214 (2015).
53. Linderman, G. C. et al. ALRA source code. Github https://doi.org/10.5281/zenodo.5634257 (2021).
54. Linderman, G. C. et al. ALRA paper source code. Github https://doi.org/10.5281/zenodo.5634339 (2021).

Acknowledgements
We would like to thank Stefan Steinerberger, Jeremy Hoskins, Jonathan Levinsohn, Ohr Lindenbaum, Dmitry, Kobak, Peggy Myung, and Holly Steach for many useful discussions. G.C.L. acknowledges support by the NIH NHGRI Grant #R01HG008383. Y.K. acknowledges support by NIH grant # RO1GM31642, R01GM135928, U1IDA051410, P50CA121974, U01DA053628 and R01HG008383.

Competing interests
R.A.F. is a consultant for GlaxoSmithKline and Zai Lab Ltd. P.B. is an equity holder in Celsius Therapeutics. The remaining authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-021-27729-z.

Correspondence and requests for materials should be addressed to Yuval Kluger.

Peer review information Nature Communications thanks the anonymous reviewers for their contribution to the peer review of this work. Peer reviewer reports are available.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2022