Published online 14 April 2022

Nucleic Acids Research, 2022, Vol. 50, No. 12 ***e71***
https://doi.org/10.1093/nar/gkac216

MarcoPolo: a method to discover differentially expressed genes in single-cell RNA-seq data without depending on prior clustering

Chanwoo Kim [1,2,†], Hanbin Lee [3,†], Juhee Jeong[4], Keehoon Jung[4,5,6,*] and Buhm Han [4,7,*]

[1]Department of Electrical and Computer Engineering, Seoul National University, Seoul, Republic of Korea, [2]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA, [3]Department of Medicine, Seoul National University College of Medicine, Seoul, Republic of Korea, [4]Department of Biomedical Sciences, BK21 Plus Biomedical Science Project, Seoul National University College of Medicine, Seoul, Republic of Korea, [5]Department of Anatomy and Cell Biology, Seoul National University College of Medicine, Seoul, Republic of Korea, [6]Institute of Allergy and Clinical Immunology, Seoul National University Medical Research Center, Seoul, Republic of Korea and [7]Interdisciplinary Program in Bioengineering, Seoul National University, Seoul, Republic of Korea

Received March 02, 2021; Revised March 16, 2022; Editorial Decision March 19, 2022; Accepted March 22, 2022

ABSTRACT

The standard analysis pipeline for single-cell RNA-seq data consists of sequential steps initiated by clustering the cells. An innate limitation of this pipeline is that an imperfect clustering result can irreversibly affect the succeeding steps. For example, there can be cell types not well distinguished by clustering because they largely share the global structure, such as the anterior primitive streak and mid primitive streak cells. If one searches differentially expressed genes (DEGs) solely based on clustering, marker genes for distinguishing these types will be missed. Moreover, clustering depends on many parameters and can often be subjective to manual decisions. To overcome these limitations, we propose MarcoPolo, a method that identifies informative DEGs independently of prior clustering. MarcoPolo sorts out genes by evaluating if the distributions are bimodal, if similar expression patterns are observed in other genes, and if the expressing cells are proximal in a low-dimensional space. Using real datasets with FACS-purified cell labels, we demonstrate that MarcoPolo recovers marker genes better than competing methods. Notably, MarcoPolo finds key genes that can distinguish cell types that are not distinguishable by the standard clustering. MarcoPolo is built in a convenient software package that provides analysis results in an HTML file.

INTRODUCTION

Single-cell RNA (scRNA) sequencing technology has offered opportunities to study gene expressions of individual cells of a biological system. The two important goals of scRNA analysis are to define clusters of cell types existing in the data and to characterize the expression signature of each cell type. To achieve these goals, people carry out the standard pipeline consists of two steps. First, unsupervised clustering is applied to define clusters in the data. Second, informative genes that can characterize clusters are identified by examining differentially expressed genes (DEGs). These two steps, namely the clustering step and the characterization step, are performed sequentially because DEGs between the clusters cannot be found without clustering by definition.

However, this standard practice with two sequential steps has an innate limitation. The limitation is that the whole procedure is dependent on the first step, the clustering step. The clustering result may not be perfect for two reasons. First, the clustering is performed under the assumption that overall gene expressions differ by cell types. Typically, one reduces the dimension of gene expression data by calculating the principal components (PCs) from highly variable genes (HVGs). Applying a clustering algorithm to PCs is equivalent to assuming that many genes will differ by cell types. However, it is possible that some cell types may not show large differences in the global structure of genes, while a few informative DEGs can easily discriminate those cell types. For example, in our analyses of real datasets with curated cell type labels, the standard clustering pipeline failed to distinguish between the anterior primitive streak (APS)

*To whom correspondence should be addressed. Email: buhm.han@snu.ac.kr
Correspondence may also be addressed to Keehoon Jung. Email: keehoon.jung@snu.ac.kr
†These authors contributed equally.

cells and mid primitive streak (MPS) cells in the human embryogenic stem cell (hESC) data (1) and between NK cells and gamma delta (γδ) T cells in the human liver data (2), while one informative DEG could almost perfectly segregate these types. Thus, if the analysis depends entirely on the clustering result, one can neither find these informative marker genes nor distinguish these types. Second, there are many parameters and settings for the clustering step, which in turn underscores that no single clustering result is undoubtedly perfect for interpretation. The variable settings include the method for HVGs selection (3–5), the number of HVGs used (3,4), the method for dimension reduction (6,7), and the parameter for the clustering resolution (8,9). It is common that the clustering procedure is performed repeatedly to get the seemingly best clustering result to human eyes (4). In such a case, the final clustering result may be susceptible to subjective decision and data-specific overfitting.

In this respect, there have been demands for methods that can extract informative DEGs from data itself in a manner independent of the clustering result (10,11). One possible approach would be to utilize existing HVG selection methods. However, HVG methods were only designed to select genes as input for the dimension reduction but were not specifically designed to sort out DEGs. A more advanced approach is singleCellHaystack (12), a recently developed method that extracts a list of candidate DEGs by examining whether the expression of a gene agrees with the placement of cells in a low-dimensional space. The limitation of this method is that it, by default, uses the median read count of each gene to define whether a gene is expressed or not in each cell. This simple binarization can result in a loss of quantitative information because subsets of a cell type may express a gene with different intensities (13–15).

We here propose MarcoPolo, a novel clustering-independent approach to identifying DEGs in scRNA-seq data. MarcoPolo identifies informative DEGs without depending on prior clustering and therefore is robust to uncertainties from clustering or cell type assignment. The term 'DEGs' is usually used for situations when we first fix the group labels of cells and then observe the differences in expression levels between those groups. The DEGs in our MarcoPolo lack such fixed group information, so we note that the genes MarcoPolo presents are DEG candidates in the strict sense. Since DEGs are identified independent of clustering, one can utilize them as additional DEGs that can complement the DEGs found by the standard clustering, can utilize them to detect subtypes of a cell population that are not detected by the standard clustering, or can utilize them to augment HVG methods to improve clustering. An advantage of our method is that it automatically learns whether a cell expresses a gene or not by fitting a bimodal distribution, which can be helpful for interpretation. Additionally, our framework provides analysis results in the form of an HTML file so that researchers can conveniently review the visualization of the results.

MarcoPolo finds informative DEGs by using three strategies. First, it takes advantage of the fact that the expression patterns of an informative gene can be bimodal. (13–15) To this end, it fits a bimodal mixture model and calculates a score for how well the model fits the observed expression pattern. This procedure divides cells into two groups and provides the grouping labels of the cells, which can be useful for downstream interpretation. Second, it uses a *voting system* that compares a gene's expression pattern with other genes. The underlying reasoning is that a group of cells in a similar biological status will co-express a subset of genes (16). Hence, if an expression pattern of a gene is shared by many other genes, we consider the gene informative. Third, MarcoPolo examines whether the cells expressing a gene are placed proximal to each other in a low-dimensional space. MarcoPolo combines these three scores (bimodality score, voting system, and proximity score) into one to winnow genes of which expressions are noteworthy.

Using extensive simulations and real data analyses, we demonstrate that DEGs identified by MarcoPolo are informative. Using real datasets with curated cell labels, we defined marker genes based on the cell labels and let methods find these markers. MarcoPolo achieved the best accuracy compared to competing methods including the HVG methods, singleCellHaystack, and the standard DEG pipeline dependent on clustering. In the challenging tasks of distinguishing between the APS and MPS cells in the hESC data and between NK cells and γδ T cells in the Liver data, MarcoPolo assigned high ranks to the marker genes that could distinguish these types. Finally, when we used MarcoPolo-identified genes to augment HVGs for dimension reduction, we found that the clustering results became more robust against changes in the resolution and other parameters.

MATERIALS AND METHODS

MarcoPolo method

Linear Poisson mixture model. To identify the expression modality in scRNA-seq data, we fit the following Poisson mixture model to each gene's count data.

$$\log \mu_{nt} = \beta_0 + \sum_p \beta_p x_{np} + \delta_t + \log s_n$$

Here, $t \in \{0, 1\}$ indicates the two cell groups. We call the group of cells with a larger mean on-cells and the group of cells with a smaller mean off-cells. Conditional on that cell n belongs to group t, μ_{nt} is the mean of the Poisson distribution that y_{gn}, the observed read count of cell n, follows. β_0 is an intercept, and β_p is coefficients corresponding to covariate x_{np}. δ_t is on-cells-specific overexpression. s_n is the size factor (17) of cell n. A similar modeling of read counts has been previously used by Zhang *et al*. (18).

Based on this model, the loss function Q of each gene is defined using the log-likelihood.

$$Q = -\log\left[\prod_n \left(\sum_t \text{Poisson}(y_{gn}|\mu_{nt})\right)\right]$$

We optimize this loss function using the Adamax optimizer implemented in the PyTorch computing library. We used the default learning rate of 2e-3 for the optimizer. As PyTorch is a tensor computation software with a strong GPU acceleration support, users can easily utilize GPU for MarcoPolo.

After the loss function optimization, for each gene, we learn how the cells are divided into two groups according to the expression modality. Without loss of generality, we assume that the mean expression of group $t = 1$ is larger than the mean expression of group $t = 0$.

Multiple-criteria ranking system. MarcoPolo uses the following three criteria to sort out informative DEGs. For description purposes, we define an indicator variable $I_{gn} \in \{0, 1\}$ to denote the cluster assignment of cell n according to gene g.

Voting system. The voting system prioritizes genes that exhibit a common expression pattern with other genes. If a gene is truly related to a biological status, it is likely that there are more genes that are co-expressed in a way similar to the gene. We examine how many times the segregation pattern of a gene is repeated by calculating the voting score of gene g,

$$v_g = \sum_{g'} v_{gg'}$$

where $v_{gg'} = 1$ if $\frac{\sum_n (I_{gn} \cdot I_{g'n})}{\min(\sum_n I_{gn}, \sum_n I_{g'n})} > t$ or else $v_{gg'} = 0$ (The default value for the threshold t is 0.7). Thus, the more times a gene is supported by other genes, the higher the gene's voting score becomes. Note that our formula calculates what proportion of the cells that express a gene with a smaller on-cell count also expresses a gene with a larger on-cell count. We use this formulation because sometimes, one gene can be a marker gene of a group, and another gene can be a marker gene of a subtype of that group. In such a case, we wanted to consider them as supportive of each other in our voting system.

Proximity score system. A hierarchical structure is pervasive in scRNA-seq data due to the cell lineage. That is, heterogeneity from higher-level grouping can determine the global structure of the scRNA data. We assume that the expression pattern of a gene corresponding to a meaningful biological status tends to align well with this global structure. This idea is similar to the one on which singleCellHaystack is based (12). For each gene g, we calculate the proximity of the on-cells ($I_{gn} = 1$) in a low-dimensional space. The intuition is that if a gene can explain the underlying structure, the cells expressing that gene will be proximal to each other. To this end, we perform principal component analysis (PCA) following the same standard procedure used in Seurat (8) as follows. We first divide read counts of the genes in a cell by the total count within the cell and multiply by 10,000. This is then natural-log transformed using $\log(1 + x)$. Next, we center the data to 0 and divide it by the standard deviation. Finally, the values are truncated to 10. We obtain principal components of the data and then calculate the proximity score as follows,

$$p_g = \sum_i^{N_{PC}} \sqrt{\frac{\sum_n I_{gn} \cdot (U_{in} - (\sum_n I_{gn} \cdot U_{in}) / \sum_n I_{gn})^2}{\sum_n I_{gn}}}$$

where U_{in} is the projection of the nth cell's data onto the ith principal component. The default value for the number of PCs (N_{PC}) is 2. The smaller this score is, the more informative we interpret the gene as. Note that although this score tends to capture genes whose on-cells cluster together in the standard clustering approach, our method is not strictly dependent on a specific clustering algorithm. Thus, our method can avoid uncertainties induced by fixing the clusters and can be interpreted as utilizing the clustering information in the PC space in a soft way.

Bimodality score system. The bimodality score system prioritizes genes of which expression is bimodal. We measure the discrepancy between the high and low expression components of a given gene using two statistics. First, we compare the log-likelihood of the data (namely Q score) under the null hypothesis with a single Poisson distribution versus the alternative hypothesis with $K = 2$ Poisson distributions with different means. We have developed a statistic defined as the ratio of the two log-likelihoods, namely QQ ratios,

$$QQ_{\text{ratio},g} = \frac{Q_{\text{null},g}}{Q_{\text{alt},g}}$$

This modeling may look unconventional because the subtraction, rather than division, of the two log-likelihoods is more common in other statistical areas. We have found that the ratio statistic fits this problem well because the ratio is robust against the observed differences in the absolute read counts between different genes. Since read counts can differ drastically from gene to gene, a simple subtraction of Q scores can be affected largely by the read counts rather than by how well the data fit a bimodal distribution.

In addition to using the QQ ratio statistic, we also use a statistic that compares the on-cells' mean expression value with all cells' mean expression value. That is, we compare how much the mean of the alternative hypothesis shifts from the mean of the null hypothesis as follows.

$$MS_g = \frac{\sum_n (I_{gn} \cdot y_{gn})}{\sum_n I_{gn}} - \frac{\sum_n (y_{gn})}{n}$$

The bimodality score is obtained by merging the two measures in a nonparametric way,

$$b_g = \min\left(\text{rank}\left(QQ_{\text{ratio},g}\right),\ \text{rank}\left(MS_g\right)\right)$$

Final step of obtaining MarcoPolo score. Finally, we aggregate the aforementioned statistics to select genes of interest. We generate the MarcoPolo score, a nonparametric rank-based score of each gene, by combining v_g, p_g, and b_g.

$$\text{MarcoPolo}_g = \min\left(\text{rank}\left(v_g\right),\ \text{rank}\left(-p_g\right),\ b_g\right)$$

Note that the proximity score was negated in rank function because genes with smaller scores are more informative, and b_g was put without rank function because it is already based on ranks. In case two genes have the same rank, the gene with a larger fold change is prioritized. After calculating this statistic, we remove outlier genes that satisfy any of the following conditions: (1) log fold change between on-cells and off-cells is < 0.6, (2) the number of on-cells ($\sum_n I_{gn}$) is < 10, or (3) the number of on-cells ($\sum_n I_{gn}$) is > 70 percent of the number of all cells. We rank all remaining genes

based on their MarcoPolo scores and present the results in the form of an HTML file.

DATASETS

Simulation dataset

We generated multiple scRNA-seq simulation datasets using Symsim (19), a simulator of single-cell RNA-seq experiment. Each dataset contained 1,000, 2,000, 5,000 or 10,000 cells with 5,000 genes sequenced. In the simulation scheme of Symsim, each gene in each cell has its own parameters for modeling its transcriptional kinetics: promoter on-rate (k_{on}), off-rate (k_{off}), and RNA synthesis rate (s). The parameters are determined by the product of gene-specific coefficients (i.e., gene effects) and cell-specific coefficients, called extrinsic variability factors (EVFs). EVFs are a low-dimensional manifold, on which the cells lie, representing factors of cell-to-cell variability. For generating EVFs of each cell, we used the tree structure of numerous subpopulations shown in Supplementary Figure S1. The expected value of each EVF was determined according to the cell's position in the tree. For generating gene effect values, we modulated a parameter called η in the simulator (1e-2, 5e-3, 1e-3, and 5e-4), which controls the probability that each gene effect is not zero (i.e., the relative probability that a gene has a non-zero type-specific overexpression). The smaller the parameter becomes, the fewer the number of cell-type markers available in the dataset becomes. For example, when the number of cells was 10,000, the observed number of genes with non-zero overexpression became (196, 107, 23, and 10) with parameters (1e-2, 5e-3, 1e-3, and 5e-4) respectively. We ran simulations ten times for each combination of parameters with different random seeds. In total, 160 datasets were generated (4 dataset sizes $\times$ 4 η parameters $\times$ 10 trials). For other parameters, we used the following fixed values. The number of extrinsic variability factors (EVFs) was 20. The number of different EVFs between subpopulations (Diff-EVFs) was 5. The mean of a normal distribution from which gene effects were sampled was 1. The parameter bimod, which modifies the amount of bimodality in the transcript count distribution, was 1.

Real dataset

Embryonic stem cell scRNA data. The Koh *et al.* (1) dataset consists of 531 fluorescence-activated cell sorting (FACS)-purified human embryonic stem cells (hESCs) at various stages of differentiation. We extracted the data from the R package DuoClustering2018 (20), which can be installed using Bioconductor package manager. We used the following cell types with both bulk RNA-seq data and scRNA-seq data: hESC (day 0), anterior primitive streak (day 1), mid primitive streak (day 1), DLL1 + paraxial mesoderm (day 2), lateral mesoderm (day 2), early somite (day 3), sclerotome (day 6), and central dermomyotome (day 5).

Human liver scRNA data. The MacParland *et al.* (2) liver dataset consists of 8,444 cells of 11 cell types collected from 5 patients. We extracted the data from the R package HumanLiver, which can be downloaded from https://github.com/BaderLab/HumanLiver. MacParland *et al.* determined the identity of each cell type using manual curation based on known gene expression profiles after clustering cells. This dataset included 11 unique cell types: hepatocytes, $\alpha\beta$ T cells, macrophages, plasma cells, NK cells, $\gamma\delta$ T cells, LSECs, mature B cells, cholangiocytes, erythroid cells, and hepatic stellate cells.

PBMC 4k scRNA data. We obtained the PBMC (peripheral blood mononuclear cell) 4k dataset (21), namely Zhengmix8eq, from the R package DuoClustering2018. This dataset is a mixture of 3,994 FACS-purified PBMC cells of 8 cell types, which are B cells, monocytes, naive cytotoxic cells, regulatory T cells, memory T cells, helper T cells, naive T cells, and natural killer cells.

Tabula Muris consortium data. We downloaded the Tabula Muris consortium data (22) from the consortium website (https://tabula-muris.ds.czbiohub.org/). It consists of 20 datasets of different organs and tissues: aorta, bladder, brain (myeloid), brain (non-myeloid), diaphragm, fat, heart, kidney, large intestine, limb muscle, liver, lung, mammary gland, marrow, pancreas, skin, spleen, thymus, tongue, and trachea. The consortium provided datasets obtained by using two different approaches, microfluidic droplet-based 3'-end counting and FACS-based full-length transcript analysis. We used the FACS-based datasets for our analysis.

Data preprocessing

For each dataset, we only used the genes of which mean expression (log-normalized count) value across all cells was in the top 30th percentile. We used the t-SNE coordinates included in the downloaded metadata for drawing low-dimensional plots. The size factors of cells were calculated using calculateSumFactors (17) implemented in scran package (23). We note that all of the methods we compared used the same preprocessed data.

Marker gene identification

For simulation datasets, we used each gene's meta information to obtain gold standard marker sets. Symsim outputs the number of different external variability factors (EVF) for each gene, an indicator of how much it had a cell-type-specific effect. We regarded genes of which the number of different EVFs is not zero as markers. For real datasets, we used two criteria for defining markers. The first criterion was using the fold change values between cell types. Because the real datasets included cell type labels obtained either by FACS purification or by manual curation, we were able to accurately define markers based on cell types. To this end, we used the approach recently suggested by Zhang *et al.* (16), which can effectively detect markers that are shared by multiple cell types. Briefly, for each gene, we sorted the cell types in ascending order based on the mean expression level. We then calculated fold change between two consecutive types in this order. We then examined the maximum

value among the *N-1* fold change values, given *N* types. After calculating this maximum value for all genes, we used the genes with the large maximum values as marker genes. We selected a different number of top genes (100, 200, 300, 400, and 500 genes). We filtered out genes with the maximum log fold change not larger than 2. The second criterion was using genes reported in marker databases such as CellMarker database (24) and Panglao database (25). To this end, first, we downloaded the list of markers from the databases. Then we extracted the markers of the corresponding tissue and assigned them to our datasets. To obtain markers of high confidence, we excluded markers that were reported only once. We excluded the datasets which had less than 10 markers after the marker filtering procedure from our benchmarking.

Highly variable genes

We used the FindVariableFeatures function of Seurat (8) to obtain highly variable genes (HVG). The two selection methods we used to sort out HVGs were VST and DISP, which were selected by the selection.method parameter. This function output an ordered list of genes.

Standard DEG pipeline based on clustering

We followed the default pipeline of Seurat (v3.2.1) for identifying DEGs based on clustering. We selected 2,000 HVGs using the FindVariableFeatures function with the VST option. We normalized and scaled the read count data using the NormalizeData and ScaleData functions. Then, we calculated 50 principal components (PCs) using the RunPCA function. We obtained clustering results using the default number of PCs, the top 10 PCs, by running the FindNeighbors function and the FindClusters function with default parameters sequentially. For the FindClusters function, we used multiple resolutions of 0.5, 1.0, and 1.5 to obtain clustering results with various numbers of clusters. Finally, for each clustering result, we ran the FindAllMarkers function to identify differentially expressed genes across clusters. The FindAllMarkers function used a Wilcoxon Rank Sum test by default. In addition, we ran the function with and without a log fold change threshold of 0.25. In total, we ran Seurat pipeline with 6 combinations of parameters for Seurat. We ranked the genes by the ascending order of *P*-values. In case two genes had the same *P*-value, we prioritized genes with larger fold change values.

singleCellHaystack

Two input parameters required by singleCellHaystack were (1) the coordinates of cells in a low-dimensional space and (2) binarized detection data, a table showing which genes are detected in which cells. To obtain low-dimensional coordinates, we calculated 10 PCs using the functions implemented in Seurat, as we did in the standard DEG pipeline. To obtain the detection data, we used the median of each gene as the threshold for determining whether cells are detected or not, which was the default usage. We then ran the haystack function with those two input parameters. Finally, we ranked the genes by the ascending order of singleCellHaystack *P*-values.

Local web server generation

To help researchers interpret and make use of the result for various purposes, we provide the analysis result as a local database in the form of an HTML file. For each gene, the output file provides the fold change based on the tentative group, a two-dimensional plot of cells, a histogram of expression with annotated group information, and the statistics that were used to prioritize genes. In addition, it contains a biological description of each gene that was adopted from the NCBI Gene database (26), which was downloaded from the NCBI FTP server (https://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia/Homo_sapiens.gene_info.gz).

Benchmarking hardware

All benchmarking took place on an Intel® Xeon® Gold 6136 CPU (3GHz). We used a single CPU while limiting the number of threads to four using the taskset command in CentOS 7 operating system. A single Nvidia® RTX® 2080 was used for the GPU mode of MarcoPolo.

RESULTS

Overview of MarcoPolo Method

The standard analysis pipeline for scRNA-seq data consists of two steps, the clustering step and the characterization step, as shown in Figure 1A. A drawback of this pipeline is that any errors that occur in the first clustering step will sequentially affect the remaining analysis. The second drawback is that DEGs cannot suggest new subtypes unidentified by the clustering, because DEGs are entirely dependent on the clustering result. To solve these challenges, we propose MarcoPolo, a clustering-independent DEG identification method. Since MarcoPolo does not depend on clustering results, it is free from errors that can occur during clustering. MarcoPolo-identified genes can serve as additional DEGs, suggest novel subtypes of a cell type unidentified by the standard clustering, and augment HVGs to improve clustering (Figure 1B). To select DEGs, MarcoPolo uses a multiple-criteria ranking approach to rank genes after fitting a two-component Poisson mixture model (Methods). For a given gene, we name the cells that are more likely to be in the high expression component as on-cells and the cells that are more likely to be in the low expression component as off-cells (Figure 1C). The first component of MarcoPolo is the *voting system* that prioritizes genes exhibiting a shared expression pattern with other genes (Figure 1D). The intuition is that a true biological entity will express multiple markers; therefore, a gene that reflects this true entity will share its expression pattern with other genes. For example, gene 1, gene 3, gene 4, and gene 5 in Figure 1D show similar on-cell patterns, and therefore the voting system assigns high ranks to these genes. By contrast, gene 2 and gene 6 have no other genes sharing on-cell patterns and thus are assigned low ranks. In addition to the voting system, MarcoPolo implements two more criteria, the *proximity score* and the *bimodality score*. We can expect that biologically similar cells will be close in the distance in a low-dimensional space. Based on this idea, the prox-

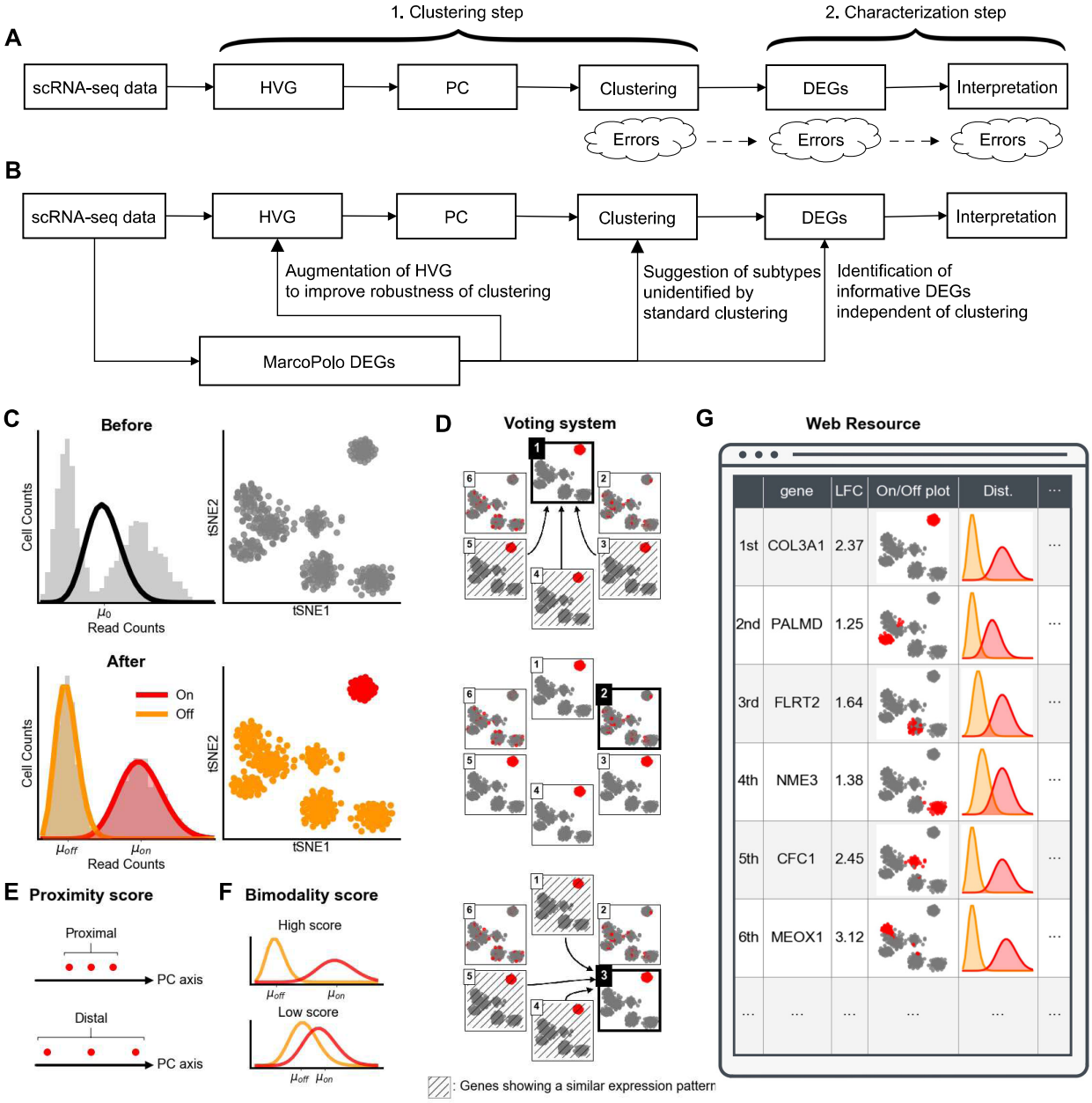

Figure 1. Overview of MarcoPolo. **(A)** the standard analysis pipeline for scRNA-seq data, consisting of two consecutive steps. In this pipeline, any errors in the clustering step irreversibly affect the succeeding step. **(B)** MarcoPolo analysis pipeline. MarcoPolo identifies informative DEGs independent of clustering, and these genes can be utilized for various purposes to complement the standard pipeline. **(C)** MarcoPolo fits a single Poisson distribution and a two-component Poisson mixture model separately. In the tSNE plots, the cells were colored by the groups they belonged to. In the bottom plots, on-cells (high expression component) are colored red, and off-cells (low expression component) are colored orange. **(D)** MarcoPolo's voting system prioritizes genes exhibiting a shared expression pattern with other genes. The on/off patterns of 6 different genes are plotted. Arrows indicate that a gene supports (votes for) another gene because they share expression patterns. Gene 1 and gene 3 got three votes, while gene 2 got zero vote. **(E)** MarcoPolo's proximity score system calculates the variance of the principal component (PC) values of on-cells for each gene. Higher ranks are assigned to genes with low variance. **(F)** MarcoPolo's bimodality score system gives higher scores to genes whose expressions follow a bimodal distribution. **(G)** MarcoPolo offers the analysis result in a local database (HTML file). For each gene, the web server provides the log fold change values, expression on/off plots, histograms, scores, and the biological description of the gene.

imity score computes the variance of the principal component (PC) values of on-cells and assigns high ranks to genes with low variance (Figure 1E). The bimodality score system measures how well the bimodal two-component model fits a given gene (Figure 1F). This system compares the log-likelihood (Q score) of the two-component model versus the one-component model. Accordingly, genes with a bigger reduction in the Q score are ranked higher. In addition, the system assigns higher ranks to genes of which on-cells' mean expression value is much higher than all cells' mean expression value. MarcoPolo determines the ranking of all genes by combining these three scoring systems. Our framework also provides the analysis result in the form of an HTML file so that researchers can conveniently interpret and visualize results (Figure 1G). The output file provides a fold change value based on the tentative group, a two-dimensional plot of cells, a histogram of expression with annotated group information, and the statistics that were used to rank genes.

Standard DEG analysis depends on the clustering performance

Here we show that if a clustering algorithm fails to cluster a group of cells, informative genes can be missed in the downstream DEG analysis. We demonstrate this using realistic simulation datasets generated by Symsim (19), a simulator of single-cell RNA-seq experiments. We generated 160 different simulation datasets while varying the parameter η (1e-2, 5e-3, 1e-3, and 5e-4), which controls the probability that a gene has a non-zero type-specific overexpression, and the dataset size (i.e., the number of cells) (1,000, 2,000, 5,000, and 10,000) (Methods). The simulated number of cell-type markers in the dataset was linearly proportional to the parameter η (Supplementary Table S1). The smaller the parameter η became, the fewer the number of cell-type markers became. For each combination of the two parameters, we generated 10 independent datasets to obtain 160 datasets in total. As expected, the smaller the parameter η became, the less clear the boundaries between the cell populations became (Supplementary Figure S2), and the lower the quality of clustering results became (Supplementary Figure S3). In addition, interestingly, we found that the quality of clustering results was the highest when the number of cells was 5,000. Since we know which genes are true DEGs in this simulation, we evaluated the performance of the methods for finding DEGs. The methods we compared were MarcoPolo, singleCellHaystack, two widely used HVG methods implemented in Seurat package (8) (VST and DISP), and the standard DEG workflow based on clustering (Seurat clustering with default parameters followed by Seurat FindAllMarkers function both with and without its default filter, a filter for excluding genes of log fold difference smaller than 0.25). We examined each method's area under the receiver operating characteristic curve (AUC) to see how accurately each method sorts out true DEGs (Figure 2). We found that the performance of the standard workflow based on clustering was good when the number of DEGs was high. This was expected because if there are many DEGs of a cell type, the clustering based on global structures will be successful, and the DEGs can be easily identified from clusters. Specifically, we observed that the standard workflow performed well when the number of cells was large (especially when it was 5,000). This was also expected because the quality of clustering can be low when the number of cells is low. For example, when the parameter η was set as high as 1e-2 and the number of cells was 5,000, the standard workflow's median AUCs were between 0.888 and 0.915 (with filter) and between 0.828 and 0.848 (without filter) depending on the resolution parameters (Supplementary Table S2). However, when the parameter η was lowered to 5e-4 and the number of cells was lowered to 1,000, the AUC decreased; the medians were between 0.769 and 0.906 (with filter) and between 0.597 and 0.634 (without filter), and the inter-quantile ranges (IQRs) were large (0.048∼0.167 with filter and 0.207∼0.294 without filter). This demonstrated that the standard pipeline can have difficulties in catching DEGs if the clustering becomes unclear due to a small number of DEGs for a specific population or due to an insufficient number of cells. In contrast, MarcoPolo consistently showed high performance. When the parameter η was large (1e-2) and the number of cells was 5,000, the median was 0.900, and the IQR was 0.051. When the parameter η was small (5e-4) and the number of cells was 1,000, the median was 0.833, and the IQR was 0.114. The HVG methods also showed good performances. The performance of DISP was particularly notable for this simulation benchmark; its AUC was lower than MarcoPolo when $\eta =$ 1e-2 but was higher than MarcoPolo when $\eta \leq$5e-3. Interestingly, the performance of singleCellHaystack showed a similar trend to the standard workflow. When $\eta =$ 1e-2, the median AUC was relatively high (0.895), but when $\eta =$ 5e-4, the median AUC went down to 0.601, and the IQR became large (0.364). Although this simulation does not perfectly represent real situations, the result suggests that when it is hard to cluster cells appropriately, the standard DEG analysis based on clustering could easily miss the target populations' DEGs while MarcoPolo and other methods showed a capability to retrieve them.

MarcoPolo identifies marker genes

We benchmarked if MarcoPolo could identify marker genes in real datasets for which we know the true markers. We used 23 real datasets: the human embryogenic stem cell (hESC) dataset (1), the human liver cell dataset (2), the human peripheral blood mononuclear cell (PBMC) dataset (21), and the Tabula Muris consortium datasets of 20 different organs and tissues (22) (see Methods). In contrast to simulation datasets for which we know true DEG sets, the true markers are not perfectly known for real datasets. We here used two separate criteria to define marker answer sets and repeated the same benchmarking procedure twice. The first criterion was using the fold change values between cell types. In these datasets, the cell types of the cells were already labeled with high confidence either by fluorescence-activated cell sorting (FACS) or by manual curation based on the known markers. Assuming that the cell type labels provided by the original studies were correct, we defined marker genes that were cell-type-specifically expressed (Methods). Briefly, for each gene, we sorted cell types in ascending order based on the mean expression

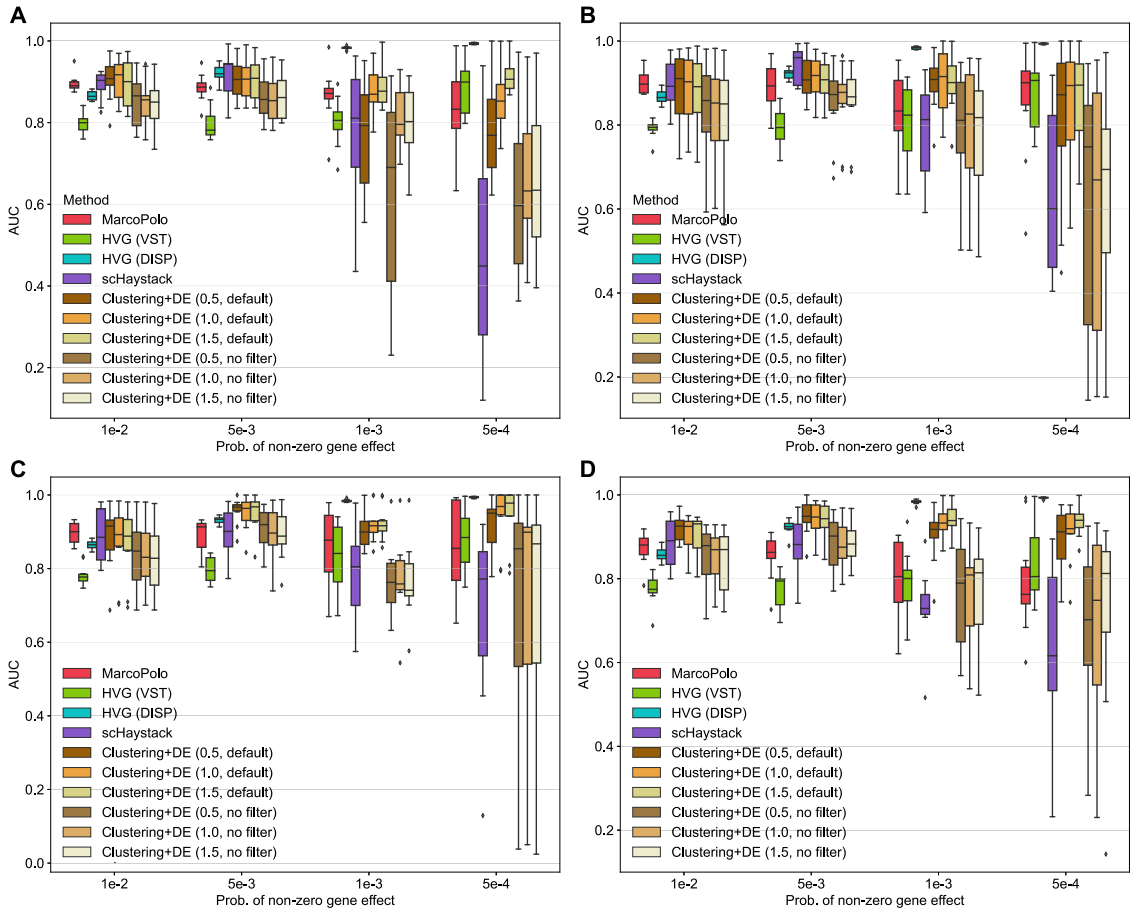

Figure 2. Simulation data analysis. Using Symsim, we simulated scRNA-seq data while varying the parameter η, which controls the probability that a gene has a non-zero type-specific overexpression. We plotted the performance of different methods for finding true markers in simulation data in terms of the area under the curve. For the standard DEG pipeline, the number in parentheses denotes the resolution parameter of the clustering algorithm. **(A)** The number of cells is 1,000. **(B)** The number of cells is 2,000. **(C)** The number of cells is 5,000. **(D)** The number of cells is 10,000.

level and calculated the maximum fold change between any two consecutive cell types. We prioritized genes with large maximum fold changes and selected the top 100 genes. To verify this method's validity, we applied it to simulation datasets and found that these fold-change-based markers belonged to EVF-based gold standard marker sets to a large extent (Supplementary Figure S4). We then examined how well different methods could identify these marker genes if the true labels of cell types were not given. As we did in the previous analysis, we compared MarcoPolo, the two HVG methods, singleCellHaystack, and the standard DEG pipeline based on clustering. We measured the AUC of the methods for each dataset and plotted the distribution of AUC over 23 datasets. Figure 3A shows that MarcoPolo achieved the best AUC among all methods (median AUC = 0.847). The median AUCs of other methods were 0.723 for DISP (HVG), 0.798 for VST (HVG), 0.784 for singleCellHaystack, and around 0.794∼0.829 for standard DEG pipelines. The inter-quartile range of MarcoPolo was relatively small (0.128). Figure 4 and Supplementary Table S3 show the actual receiver operating characteristic (ROC) curves and their AUCs of the methods for each dataset. When we examined which method performed the best for each dataset, MarcoPolo performed the best or the second-best among all methods in 16 out of 23 datasets (70% of datasets; the best in 13 datasets and the second-best in 3 datasets). For singleCellHaystack, the number was 4 (17% of datasets; 2 best and 2 second-best). For the standard pipeline based on clustering, the number was 11 (48% of datasets; 4 best and 7 second-best). This result shows that MarcoPolo's multiple-criteria ranking approach worked well in sorting out informative marker genes for a variety of datasets. As MarcoPolo determines the final ranking of the genes by combining the three scoring systems, we wanted to see how each scoring system contributed to the final performance. To this end, we examined the performance of using each single scoring system alone. Supplementary Table S4 and Supplementary Figure S5 show that the scoring system with the best performance differed by datasets. Thus, MarcoPolo needed all three systems to achieve consistent performance over different datasets. In many cases, the combined score performed comparably to the best single system, implying that it effectively captured information from the three systems. We repeated the same analysis for various numbers of top genes (200, 300, 400, and 500 genes) and observed similar trends

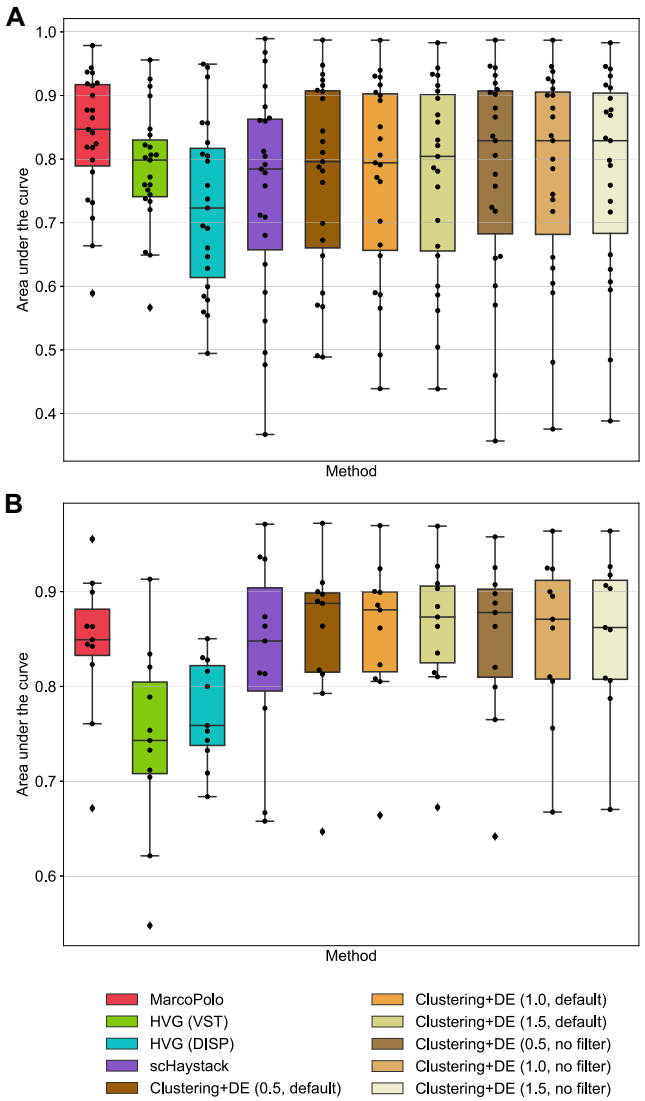

Figure 3. Performance of different methods for finding true markers in real datasets. For the standard DEG pipeline, the number in parentheses denotes the resolution parameter of the clustering algorithm. **(A)** In each of 23 real datasets, we defined true markers based on the cell type labels provided by the original study. We used the top 100 genes with large maximum log fold change values as a true marker set. **(B)** We used genes reported in CellMarker database and Panglao database as a true marker set.

(Supplementary Tables S5–S8 and Supplementary Figures S6–S9); the median AUC of MarcoPolo was always the highest.

As this result can depend on the specific DEG extraction method we used, we tried a second criterion to define marker genes; we used genes reported in marker databases such as CellMarker database (24) and Panglao database (25). We defined the true marker answers for each tissue using the list of markers in the databases (Methods and Supplementary Table S9) and repeated the same analysis above (Figure 3B, Supplementary Figure S10, and Supplementary Table S10). In this setting, the standard DEG pipelines showed the best performance in terms of the median AUCs. This was expected because the records in marker databases are based on standard DEG pipelines in most cases. The performances of MarcoPolo and singleCellHaystack were comparable to those of the standard DEG pipelines. We here note that as we cannot guarantee that the sample of the records in the databases has similar properties to our dataset's property, marker database-based benchmarking is also not a perfect measure. For example, it is not guaranteed that our dataset has the same cell-type composition as the one in the database, as we only used tissue names to match records.

Bimodal mixture model helps interpretation

A notable advantage of MarcoPolo compared with other previous methods such as singleCellHaystack is that MarcoPolo provides the grouping of the cells after fitting the bimodal mixture model, which can help interpretation. For example, singleCellHaystack requires a predefined threshold to determine whether a gene is either expressed or not in each cell (median is used by default). However, MarcoPolo adaptably learns the model parameter from data and classifies the cells into two groups: on-cells that are in the high expression component of the mixture model and off-cells that are in the low expression component. The on-cell and off-cell information that MarcoPolo provides can be utilized to complement the standard pipeline for determining clusters. Figure 5A-C show some real data examples where the on/off information from MarcoPolo DEGs were concordant with the true cell types. These genes were ranked high ($<30^{th}$) by MarcoPolo, and their distributions were bimodal. MarcoPolo distinguished on-cells (red in the histogram) from off-cells after model fitting, and those on-cells corresponded well to a specific cell type in the t-SNE plot. We note that the median read count (blue vertical line in the histogram if plotted and zero otherwise) was not fully effective for distinguishing the two bimodal groups in these examples.

We then assumed an extreme situation that we were completely blind to the cluster information from the standard clustering analysis. We assumed that we only had the on/off information from MarcoPolo DEGs. Since the on/off information can correspond to the membership of a cell type, we wanted to accumulate the on/off information of top rank DEGs to retrieve the cell types. To this end, we evaluated how many top-rank MarcoPolo DEGs are required to retrieve the cell types. We regarded a gene to be distinguishing a cell type if the on-cell group contains more than 70% of the cell type and less than 20% of the other remaining cell types. We gradually increased the number of top-ranked genes and counted how many cell types were distinguishable by those genes. In total, among a total of 147 cell types included in the 23 real datasets, MarcoPolo's grouping information was able to segregate 97 cell types when the top 300 genes in MarcoPolo were considered (Figure 5D and Supplementary Table S11). For comparison, we considered simple grouping by using the medians of read count as a threshold to define on/off cells, following the default use of singleCellHaystack. Figure 5D shows that grouping based on the medians of read counts gave inferior results in retrieving the cell types compared to MarcoPolo.

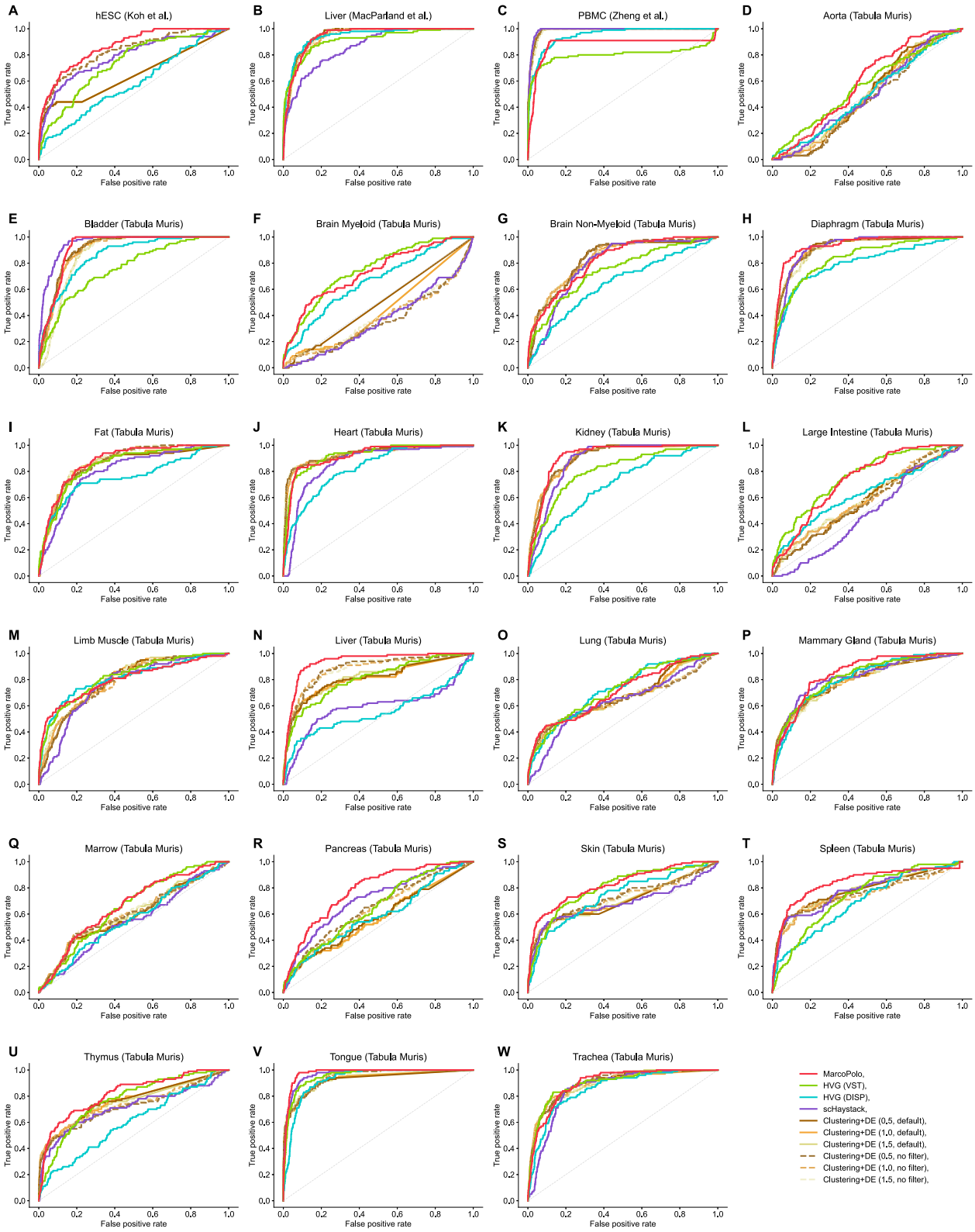

Figure 4. ROC curves and their AUCs for identifying true markers using different methods for each real dataset. As a true marker set, we used the top 100 genes with large maximum log fold change values. The candidate gene lists were obtained by using MarcoPolo, HVG methods, singleCellHaystack, or standard DEG pipeline with clustering, and were compared to the true markers. For the standard DEG pipeline, the number inside parentheses denotes the resolution parameter of the clustering algorithm.

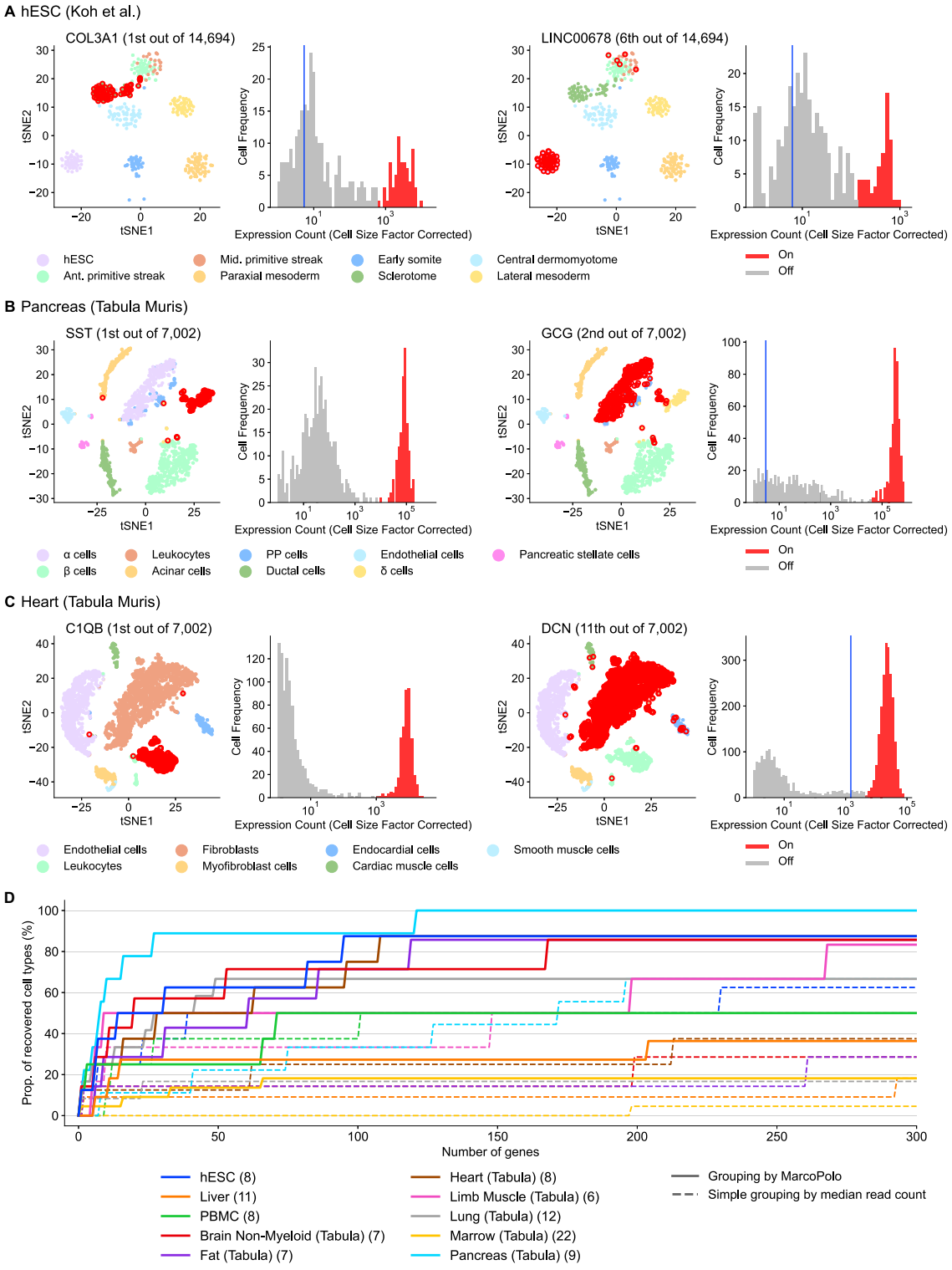

Figure 5. MarcoPolo's bimodal mixture model fitting. **(A)-(C)** Three examples showing that the on/off information from MarcoPolo's bimodal fitting is concordant with the true cell types provided by the original study. The red colors (both in dot plot and histogram) denote on-cells (the cells in the high expression component of the bimodal mixture). Next to the gene name is the gene's rank in the MarcoPolo result. Cells with zero expression count were excluded from histograms for simplicity, although the zero count cells were accounted for in the model fitting. We plotted blue vertical lines to indicate the median read count of each gene in case the median is non-zero. The x-axis of histogram denotes the bins of expression counts divided by cell-specific size factors. For each bin of expression count, the height indicates the number of cells in it (cell frequency). **(D)** The proportion of recovered cell types when we only used the on/off information from the MarcoPolo result. We used the top-ranked genes and examined how many cell types were distinguishable using these genes. For comparison, we also used simple grouping by using the medians of read counts (equivalent to the default use of SingleCellHaystack). Only datasets containing more than five cell types are shown. The number inside the parentheses denotes the number of cell types in each dataset.

MarcoPolo DEGs can identify cell types not distinguishable by the standard pipeline

Sometimes, the standard clustering analysis may fail to distinguish some cell types. This is because cell types with similar global structures can be placed too closely in the reduced-dimensional space. Even in such situations, a few key genes may be able to segregate these types. Therefore, if informative genes are given, researchers can have opportunities to manually examine each gene and find those types. MarcoPolo DEGs can provide useful candidates for the distinction of the cell types.

The human embryogenic stem cell (hESC) dataset of Koh *et al*. (1), the liver dataset of MacParland *et al*. (2), and the lung dataset of Tabula Muris consortium (22) provide examples where the standard clustering approach fails to distinguish certain cell types. In the 2D t-SNE coordinates provided by the original studies, the APS and MPS cells in the hESC dataset, the gamma delta (γδ) T cells and NK cells in the liver dataset, and the NK cells and T cells in the lung dataset show unclear boundaries (Supplementary Figure S11). The standard clustering can have difficulties in distinguishing these types, because t-SNE coordinates reflect PCs based on which the clustering is performed. To confirm this, we re-analyzed these datasets from scratch using the standard Seurat pipeline. We used the default HVG selection pipeline (VST method to select 2,000 genes) and calculated PCs. For the clustering step, we used the widely used method (FindNeighbors and FindClusters function in Seurat package). For the FindClusters function's resolution parameter, we used the value of 2.0 because the default value of 1.0 gave worse results. In our analysis, as was expected, the clustering algorithm failed to cluster the abovementioned cell types properly (second column of Figure 6). This implies that if one simply uses the standard pipeline, one would neither distinguish these cell types nor find DEGs between them.

We then applied MarcoPolo to these datasets. We found that a highly ranked gene (NODAL, ranked 11th out of 14,694 genes) by MarcoPolo was able to distinguish the APS and MPS cells in the hESC data. The on-cells inferred from the bimodal distribution were highly concordant to the MPS cells, and the off-cells were concordant to the APS cells (Figure 6A). MarcoPolo's grouping information attributed 92% of APS and MPS cells into on-cells or off-cells correctly. This shows that examining the top-scoring DEGs predicted by MarcoPolo along with their grouping information can help identify the cell types. We also found that a highly ranked gene (GNLY, ranked 12th out of 6,002 genes) by MarcoPolo was able to distinguish γδ T cells and NK cells in the liver data. The on-cells inferred from the bimodal distribution were concordant with the γδ T cells, and the off-cells were concordant with the NK cells. MarcoPolo's grouping information attributed 83% of the γδ T cells and NK cells into on-cells or off-cells correctly. This is an interesting example because GNLY is known as a marker gene for both γδ T cells and NK cells. GNLY gene encodes the antimicrobial peptide granulysin, which kills microbial pathogens such as Mycobacteria, Listeria, and Plasmodium vivax by assembling itself into a pore which disrupts the pathogens' membrane (27–29). Although GNLY gene is commonly used as a marker for NK cells, it is also known to be expressed by γδ T cells (27). Thus, this gene is expressed in both types. Indeed, our data showed that this gene's distribution over the cells shows continuously connected bimodal distribution (the last column in Figure 6B). Since there were two groups of cells with different expression intensities, MarcoPolo was able to learn the two peaks from data. In addition, we found that a highly ranked gene (GZMA, ranked 42nd out of 7,002 genes) by MarcoPolo was able to distinguish the NK cells and T cells in the lung data. The on-cells inferred from the bimodal distribution were highly concordant to the T cells, and the off-cells were concordant to the NK cells (Figure 6C). MarcoPolo's grouping information attributed 92% of the NK cells and T cells into on-cells or off-cells correctly. We want to emphasize that the 11th, 12th, and 42nd ranks can be considered notably high, because MarcoPolo examined a total of 14,694 genes in the hESC data, 6,002 genes in the liver data, and 7,002 genes in the lung data.

Augmenting MarcoPolo DEGs to HVGs improves the robustness of clustering

Since MarcoPolo analysis is independent of clustering (Figure 1B), MarcoPolo DEGs can complement the standard pipeline in various ways. So far, we have shown that MarcoPolo DEGs can suggest marker DEGs and potential subtypes of a cluster after the clustering is done. Here, we show that MarcoPolo can also help before the clustering is done by updating HVGs. In the standard scRNA-seq analysis pipeline, a subset of genes selected by HVG methods is used to construct a low-dimensional representation before the clustering. For a clustering result that well reflects the structure of the underlying biological data structure, it is important to use informative HVGs as an input. As MarcoPolo and singleCellHaystack are designed to pick genes with informative differential expression, we can use them as a feature selection method in the standard pipeline as well.

In this experiment, we augmented MarcoPolo (or singleCellHaystack) genes to HVGs and used them as input for the dimension reduction step. Briefly, we compared three categories of feature selection methods: only using genes selected by the standard HVG method, using a mixture of HVG genes and MarcoPolo genes (namely HVG with MarcoPolo), and lastly using a mixture of HVG genes and singleCellHaystack genes (namely HVG with Haystack). In case we mixed two different criteria such as mixing HVG and MarcoPolo, we extracted the same number of genes from the top-ranked genes in each criterion. We used the datasets from the previous analysis: we aimed to distinguish the APS and MPS cells in the hESC data, the γδ T cells and NK cells in the liver data, and the NK cells and T cells in the lung data. We note that although we evaluated methods by checking whether those two cell types were clustered correctly, we used all of the cells contained in the datasets for the feature selection and clustering steps. In order to test the robustness of each feature selection method against parameter changes, we ran the same method multiple times using different parameters and settings and measured how many times a method gave a successful clustering. We varied the parameters and settings as follows. First, we tried

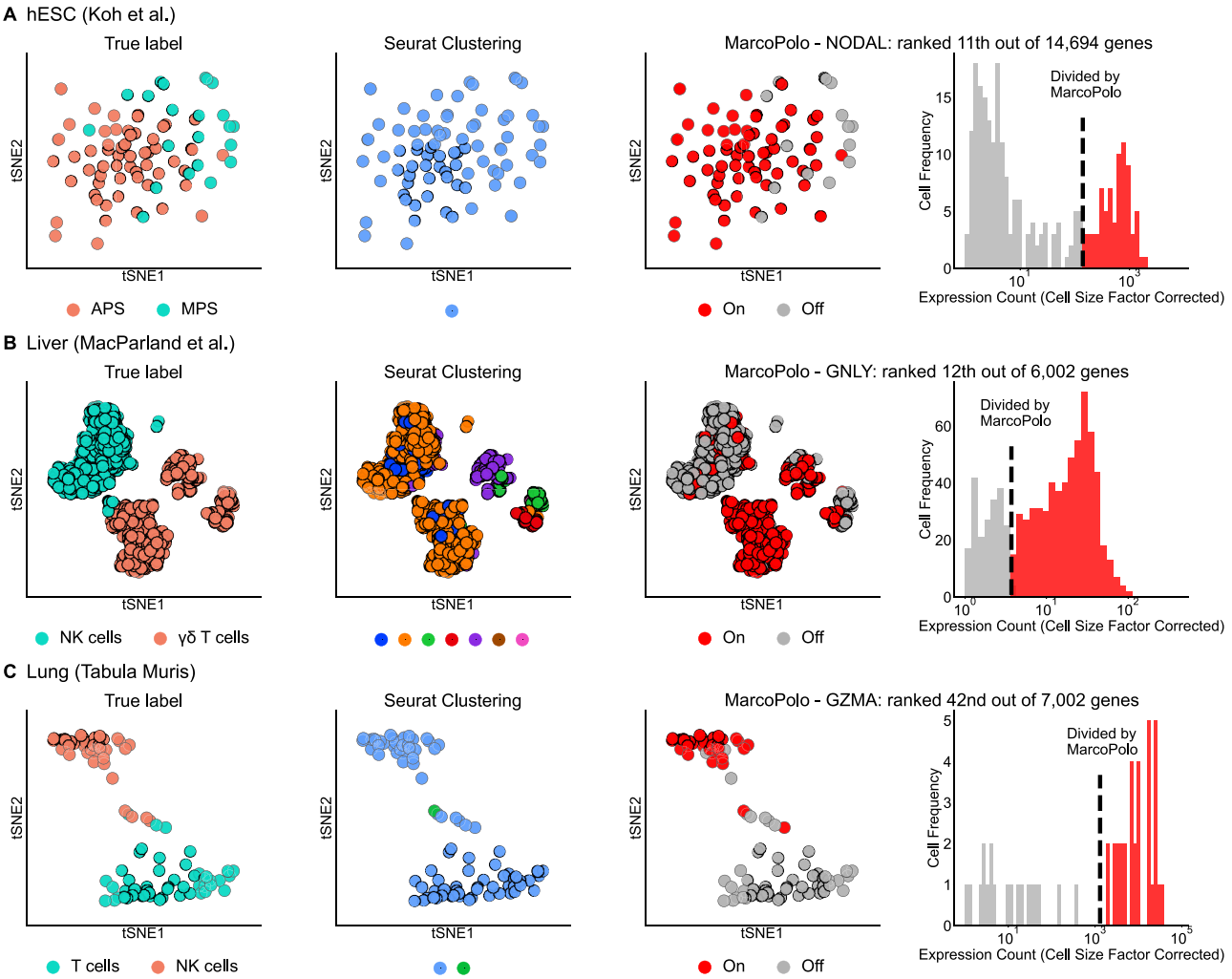

Figure 6. Cell types not distinguished by the standard pipeline but distinguished by top ranked MarcoPolo genes. **(A)** the APS and MPS cells in the hESC dataset of Koh *et al*., **(B)** the NK cells and the γδ T cells in the liver dataset of MacParland *et al*., and **(C)** the T cells and NK cells in the lung dataset of Tabula Muris consortium. The first column shows the true cell type labels of the original study, obtained by either FACS **(A, C)** or manual curation **(B)**. We used t-SNE coordinates included in the original datasets. Only cell types of our interest were shown for better visualization. The second column shows the standard clustering result by Seurat. The distinction of the cell types was unclear even with an increased resolution parameter of 3.0 (default: 1.0). The third column shows the MarcoPolo results, where we colored cells based on a high-rank gene found by MarcoPolo. The cells in the on-cell group (high expression group) were colored red. The fourth column shows the histograms, where cells with zero expression count were excluded. The x-axis denotes the bins of expression counts divided by cell-specific size factors. For each bin of expression count, the height indicates the number of cells in it (cell frequency).

two widely used HVG methods (VST and DISP). Second, we varied the number of HVGs from 200 to 1,000 with an interval of 100 (9 numbers). For example, when we used a mixture of standard HVG method and MarcoPolo with 200 genes, we used the top 100 genes from MarcoPolo and the top 100 genes from standard HVGs, respectively. Third, we varied the number of top PCs used by the clustering algorithm from 10 to 30 with an interval of 5 (5 numbers). Fourth, we varied the resolution parameter in the Louvain clustering algorithm from 0.6 to 1.4 with an interval of 0.2 (5 parameters). To sum up, for each feature selection method, we repeated the analysis 450 times with different settings (2 HVG methods × 9 HVG numbers × 5 top PC numbers × 5 resolution parameters). We then calculated how many of these 450 trials succeeded in isolating the target populations. We concluded that the two cell types were distinguished in the clustering result if the proportion of correctly mapped cells of two cell types was larger than 0.8.

Overall, employing MarcoPolo as a feature selection method improved the robustness of clustering (Figure 7A). When MarcoPolo genes were employed for the hESC dataset, the frequency of separating the target populations dramatically improved from 36.4% to 96.9% compared to using HVGs alone. For the liver dataset, augmenting MarcoPolo genes gave a comparable result with a slight drop (from 78.7% to 73.8%). In the case of the lung dataset, MarcoPolo increased the success rate from 41.8% to 52.9%. singleCellHaystack genes also improved the robustness when augmented to HVGs in the hESC dataset (from 36.4% to 60.9%) and the lung dataset (from 41.8% to 54.2%), but the improvement was smaller than MarcoPolo. When applied to the liver data, singleCellHaystack showed a considerable

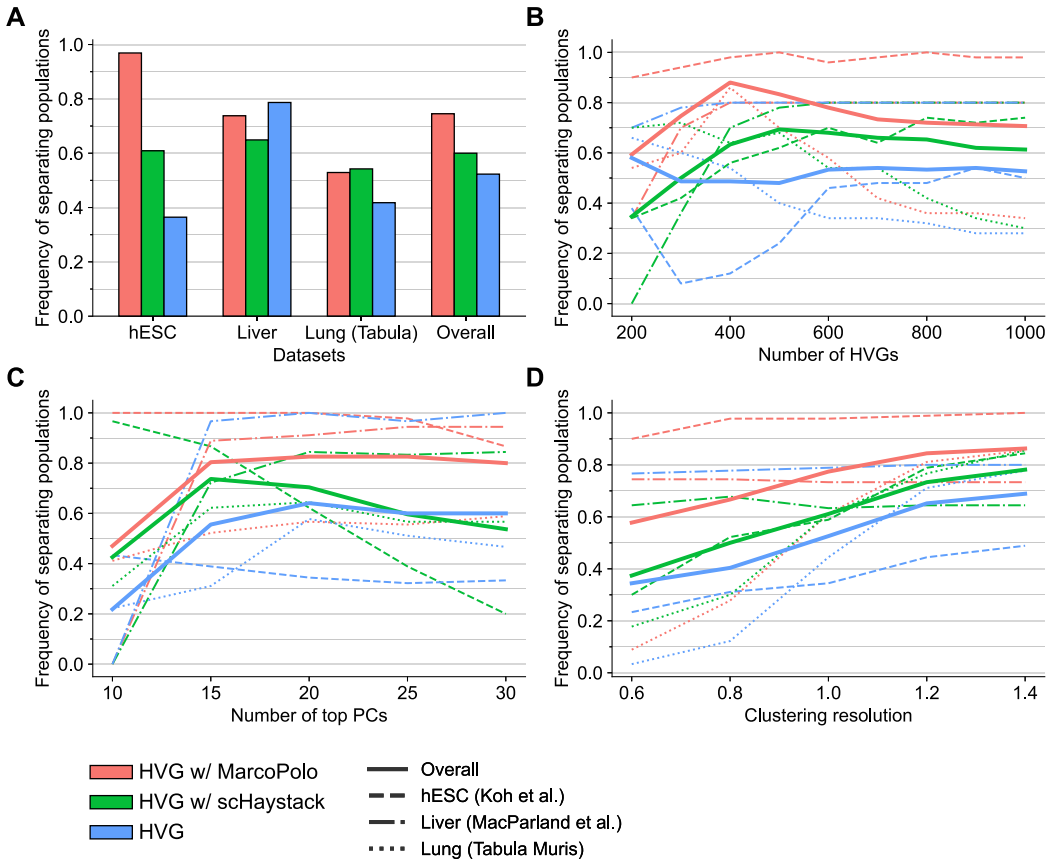

Figure 7. Robustness of clustering against parameter changes. We focused on the challenging tasks of distinguishing the APS and MPS cells in the hESC data, the NK and γδ T cells in the liver data, and the T and NK cells in the lung data. We compared three different HVG selection methods: the standard HVG method and the augmented versions using MarcoPolo or SingleCellHaystack. We tried 450 different parameters and settings for the clustering analysis and measured the frequency of successfully separating the populations of interest over multiple trials. **(A)** Overall performances of the three methods. **(B)-(D)** Comparison of the three methods with respect to the changes of a single parameter.

performance drop (78.7% to 64.9%). When it comes to the overall performance of the three datasets, HVG with MarcoPolo was the best; HVG with MarcoPolo, HVG with singleCellHaystack, and HVG alone showed success rates of 74.5%, 60.0%, and 52.3%, respectively. We also visualized the methods' performances with respect to each variable parameter (Figure 7B-D), which shows that HVG with MarcoPolo performed the best in many different settings.

Analysis of MarcoPolo's dependency on hyperparameters

We investigated MarcoPolo's dependency on hyperparameters by running it with different hyperparameter settings. We used the Tabula Muris bone marrow and lung datasets for this analysis. First, the voting system of MarcoPolo involves a threshold determining whether two genes' expression patterns support each other. We ran MarcoPolo with default hyperparameter settings except for the threshold of the voting system being varied as 0.5, 0.6, 0.7, 0.8, and 0.9 (Supplementary Figure S12A). We observed that the Spearman correlations between any two runs of MarcoPolo were all over 0.86, where the Spearman correlation was calculated based on MarcoPolo's gene ranking of all genes. We also measured how many genes coexisted in the top 100 gene lists of the two runs. At worst, the number of genes coexisting in the two top 100 gene lists was 70. Second, we varied the maximum value in the normalization step of the proximity score system: 5, 10, 15, and 20, and found that the MarcoPolo result was not affected by this hyperparameter (Spearman correlation of 1.0; Supplementary Figure S12B). Third, we varied the number of PCs used in the proximity system: 1, 2, 3, 4, and 5 (Supplementary Figure S12C). We observed that Spearman correlations were all over 0.95. At worst, the number of genes coexisting in the two top 100 gene lists was 88. The final step of MarcoPolo involves the process of removing outlier genes that satisfy the following conditions (1) log fold change between on-cells and off-cells is < 0.6, (2) the number of on-cells ($\sum_n I_{gn}$) is < 10, or (3) the number of on-cells ($\sum_n I_{gn}$) is > 70 percent of the number of all cells. We varied the log fold change threshold: 0.4, 0.5, 0.6, 0.7, and 0.8 (Supplementary Figure S12D), the minimum number of on-cells: 5, 10, 15, 20, 25, and 30 (Supplementary Figure S12E), and the maximum percentage of on-cells: 50, 60, 70, 80, and 90 (Supplementary Figure S12F). In our experiments varying each of the three hyperparameters, the Spearman correlations were all over 0.99, 0.99, and 0.97, respectively, and the number of genes coexisting

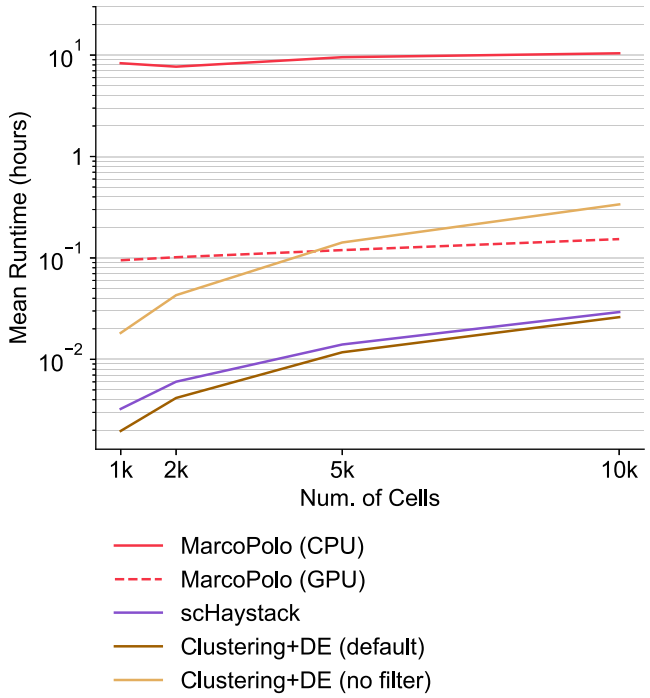

Figure 8. Mean runtime of different methods for simulated datasets of various sizes.

in the two top 100 gene lists was at worst 92, 96, and 83, respectively.

Comparison of runtime with different methods

We measured the runtime of different methods using simulation datasets (Figure 8 and Supplementary Table S12). As HVG methods finished in seconds, we excluded them from comparison. MarcoPolo involves the process of fitting the linear Poisson mixture model for each gene and thus was the slowest among all methods tried. For all sizes of datasets, MarcoPolo (CPU) took about 8∼10 hours, which we consider a feasible amount of time for typical research use. Because MarcoPolo was implemented in PyTorch, which enables tensor computation with strong GPU acceleration, users can opt whether to run MarcoPolo with GPU or only with CPU (Methods). We found that MarcoPolo (GPU) is 60∼90 times faster than MarcoPolo (CPU). Except for MarcoPolo (CPU), for datasets of 1,000 cells, MarcoPolo (GPU) was the slowest method, and for datasets of 10,000 cells, the standard clustering-based DEG pipeline with high resolution was the slowest method. Interestingly, unlike other methods, MarcoPolo's runtime increased sublinearly with dataset size, suggesting its potential utility for very large data in the future.

DISCUSSION

We developed MarcoPolo, a clustering-independent method to identify DEGs in scRNA-seq data. The standard DEG pipeline has the limitation that the results can be affected by any errors or uncertainties in the clustering step. We showed that MarcoPolo can overcome this limitation and has three practical usages. First, it can identify biologically informative genes accurately in a manner independent of clustering. Second, it can suggest groups of cells that are not identified in the standard clustering process. Third, it can be used as a feature selection method to improve the clustering robustness.

MarcoPolo suggests a philosophically different perspective on analyzing scRNA-seq data. In the standard workflow, groups of cells are first defined by clustering, and then DEGs among them are identified. In contrast, MarcoPolo identifies DEGs first, and then the DEGs can provide tentative grouping information of the cells. The two directions of approaches can complement each other. If a cell population has a unique global structure of expression profile, the DEG detection followed by clustering will work well. If there is a certain marker gene of which bimodal expression can suggest a population, MarcoPolo can work well. In our analysis, MarcoPolo's grouping information was shown to recover the cell types in real datasets and was often able to distinguish cell types that were not distinguished by the standard pipeline.

MarcoPolo has differences from another DEG identification method, singleCellHaystack. Two methods are similar in that DEGs are identified in a manner independent of clustering. A notable difference is that singleCellHaystack requires a predefined threshold to determine whether a gene is expressed or not in the cell. We showed that this binarization can decrease the performance of DEGs for distinguishing putative clusters (Figure 5D). MarcoPolo flexibly learns the on-cells and off-cells by fitting a bimodal distribution and does not require a predefined threshold.

One strength of MarcoPolo that differentiates it from other methods is the automatic generation of the summary results and the figures in the form of an HTML file (Supplementary Figure S13). This can be considered as a convenient local web database where the user can explore the top-ranked markers and how the cells are visually segregated as on/off-cells by those markers. In addition, MarcoPolo prepares the on/off information of the cells in an R object file so that the information can be used in the downstream analysis. We expect that this detailed and friendly software implementation of our method can help maximize the accessibility to general users as well as experts.

MarcoPolo has the following limitations. First, like many other methods, MarcoPolo is dependent on hyperparameters. Among the five hyperparameters that we tested for sensitivity, the voting system threshold and the number of PCs were the two that MarcoPolo was the most sensitively dependent on. Second, the support of real data analysis for the cell type identification was suggestive rather than definitive. The situation that the standard pipeline failed to distinguish cell types was only observed in some datasets but not in others. However, we expect that the information provided by MarcoPolo could be useful for the identification of cell types in future studies. Third, MarcoPolo assumes there are two peaks in the mixture distribution, but there can be genes with more peaks. When we extended our model to include three peaks, the overall performance did not improve, though. Our own manual examination of the datasets also suggested that there might not be many genes with more

than two peaks, which was the reason we decided to go on with our two-component Poisson mixture model.

The real datasets used in our analyses presented a large range of proportions of zeros. In scRNA-seq analysis, there have been strong beliefs that zeros are over-presented by technical issues, and therefore the missing values should be imputed (30,31). In contrast, recently, some studies suggested that there was no evidence of zero inflation when the read counts were fitted with the Poisson model (32,33). To see how much the amount of zeros in the data impacts MarcoPolo's performance, we did the following analysis. Beginning with the dataset used for Figure 5A, we subsampled the read counts. We applied a binomial distribution with $P = 0.5$ to each read count, so the probability of detecting expression was reduced to 50% compared to that of the original dataset. The Spearman correlation between MarcoPolo scores from the original dataset and the subsampled dataset was very high as 0.96, and the number of genes that coexisted in the top 100 gene lists from the runs for the two datasets was 88 genes (Supplementary Figure S14).

One of the interesting observations in our study was that the HVG DISP method accurately identified marker genes in simulation datasets especially when the parameter η, which controls the probability that a gene has a non-zero type-specific overexpression, was low. DISP is an HVG identification method that prioritizes genes with a large variance-to-mean ratio. We checked whether the variance-to-mean ratios of marker genes were large when the parameter η was low in simulation datasets (Supplementary Table S13). The lower the parameter η became, the larger the variance-to-mean ratios of marker genes became. For this reason, DISP worked well in simulation datasets with low non-zero type-specific expression probabilities. We then performed the same analysis for real datasets (Supplementary Table S14). In contrast to simulation datasets, in real datasets, we observed that the variance-to-mean ratios of marker genes were not dramatically larger than those of non-marker genes. Except for the liver dataset of MacParland *et al.*, the mean variance-to-mean ratio of marker genes was at most 15 times higher than that of non-marker genes, whereas it was 50 times higher in the simulation datasets.

CONCLUSION

We presented MarcoPolo, a clustering-independent approach to the exploration of differentially expressed genes in single-cell RNA-seq data. Our method exploits the bimodality of expression to find informative genes and learns the on/off group information of the cells from the data. Using simulations and real data analyses, we showed that our method puts informative genes at the top ranks, helps identify cell types not separated well in the standard clustering result, and selects features to improve the robustness of the clustering. We believe that our approach can complement the standard pipeline by suggesting a different perspective of scRNA-seq analysis for finding informative genes.

DATA AVAILABILITY

MarcoPolo is available at the GitHub repository (https://github.com/chanwkimlab/MarcoPolo). It is coded in Python 3.7 using PyTorch v1.4. The required packages are NumPy, SciPy, scikit-learn, and Pandas.

We made the MarcoPolo analysis results for the real datasets used in the paper available online. The local database (HTML) reports generated by MarcoPolo are accessible at the following addresses.

hESC (Koh *et al.*) (https://chanwkimlab.github.io/MarcoPolo/hESC/index.html)

Liver (MacParland *et al.*) (https://chanwkimlab.github.io/MarcoPolo/HumanLiver/index.html)

PBMC (Zheng *et al.*) (https://chanwkimlab.github.io/MarcoPolo/Zhengmix8eq/index.html)

The followings are for Tabula Muris consortium datasets.

Aorta (https://chanwkimlab.github.io/MarcoPolo/TabulaAorta/index.html)

Bladder (https://chanwkimlab.github.io/MarcoPolo/TabulaBladder/index.html)

Brain Myeloid (https://chanwkimlab.github.io/MarcoPolo/TabulaBrainMyeloid/index.html)

Brain Non-Myeloid (https://chanwkimlab.github.io/MarcoPolo/TabulaBrainNonMyeloid/index.html)

Diaphragm (https://chanwkimlab.github.io/MarcoPolo/TabulaDiaphragm/index.html)

Fat (https://chanwkimlab.github.io/MarcoPolo/TabulaFat/index.html)

Heart (https://chanwkimlab.github.io/MarcoPolo/TabulaHeart/index.html)

Kidney (https://chanwkimlab.github.io/MarcoPolo/TabulaKidney/index.html)

Large intestine (https://chanwkimlab.github.io/MarcoPolo/TabulaLargeIntestine/index.html)

Limb muscle (https://chanwkimlab.github.io/MarcoPolo/TabulaLimbMuscle/index.html)

Liver (https://chanwkimlab.github.io/MarcoPolo/TabulaLiver/index.html)

Lung (https://chanwkimlab.github.io/MarcoPolo/TabulaLung/index.html)

Mammary Gland (https://chanwkimlab.github.io/MarcoPolo/TabulaMammaryGland/index.html)

Marrow (https://chanwkimlab.github.io/MarcoPolo/TabulaMarrow/index.html)

Pancreas (https://chanwkimlab.github.io/MarcoPolo/TabulaPancreas/index.html)

Skin (https://chanwkimlab.github.io/MarcoPolo/TabulaSkin/index.html)

Spleen (https://chanwkimlab.github.io/MarcoPolo/TabulaSpleen/index.html)

Thymus (https://chanwkimlab.github.io/MarcoPolo/TabulaThymus/index.html)

Tongue (https://chanwkimlab.github.io/MarcoPolo/TabulaTongue/index.html)

Trachea (https://chanwkimlab.github.io/MarcoPolo/TabulaTrachea/index.html)

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

This work was supported by the National Research Foundation of Korea (NRF) (Grant number

2022R1A2B5B02001897 to BH and Grant number 2020R1C1C1015062 to KJ) funded by the Korean government, Ministry of Science, and ICT. BH and KJ were supported by the Creative-Pioneering Researchers Program funded by Seoul National University (SNU).

Conflict of interest statement. Buhm Han is the CTO of Genealogy Inc. No other authors have any competing interests.

REFERENCES

1. Koh,P.W., Sinha,R., Barkal,A.A., Morganti,R.M., Chen,A., Weissman,I.L., Ang,L.T., Kundaje,A. and Loh,K.M. (2016) An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development. *Sci. Data*, **3**, 160109.
2. MacParland,S.A., Liu,J.C., Ma,X.-Z., Innes,B.T., Bartczak,A.M., Gage,B.K., Manuel,J., Khuu,N., Echeverri,J., Linares,I. *et al.* (2018) Single cell RNA sequencing of human liver reveals distinct intrahepatic macrophage populations. *Nat. Commun.*, **9**, 4383.
3. Yip,S.H., Sham,P.C. and Wang,J. (2018) Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Brief Bioinform.*, **20**, 1583–1589.
4. Luecken,M.D. and Theis,F.J. (2019) Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol. Syst. Biol.*, **15**, e8746.
5. Hwang,B., Lee,J.H. and Bang,D. (2018) Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Medicine*, **50**, 96.
6. Tsuyuzaki,K., Sato,H., Sato,K. and Nikaido,I. (2020) Benchmarking principal component analysis for large-scale single-cell RNA-sequencing. *Genome Biol.*, **21**, 9.
7. Townes,F.W., Hicks,S.C., Aryee,M.J. and Irizarry,R.A. (2019) Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biol.*, **20**, 295.
8. Stuart,T., Butler,A., Hoffman,P., Hafemeister,C., Papalexi,E., Mauck,W.M., Hao,Y., Stoeckius,M., Smibert,P. and Satija,R. (2019) Comprehensive integration of single-cell data. *Cell*, **177**, 1888–1902.
9. Butler,A., Hoffman,P., Smibert,P., Papalexi,E. and Satija,R. (2018) Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.*, **36**, 411–420.
10. Lähnemann,D., Köster,J., Szczurek,E., McCarthy,D.J., Hicks,S.C., Robinson,M.D., Vallejos,C.A., Campbell,K.R., Beerenwinkel,N., Mahfouz,A. *et al.* (2020) Eleven grand challenges in single-cell data science. *Genome Biol.*, **21**, 31.
11. Zhang,J.M., Kamath,G.M. and Tse,D.N. (2019) Valid Post-clustering differential analysis for single-cell RNA-Seq. *Cell Syst.*, **9**, 383–392.
12. Vandenbon,A. and Diez,D. (2020) A clustering-independent method for finding differentially expressed genes in single-cell transcriptome data. *Nat. Commun.*, **11**, 4318.
13. Dobrzyński,M., Nguyen,L.K., Birtwistle,M.R., Kriegsheim,A.von, Fernández,A.B., Cheong,A., Kolch,W. and Kholodenko,B.N. (2014) Nonlinear signalling networks and cell-to-cell variability transform external signals into broadly distributed or bimodal responses. *J. Roy. Soc. Interface*, **11**, 20140383.
14. Birtwistle,M.R., Rauch,J., Kiyatkin,A., Aksamitiene,E., Dobrzyński,M., Hoek,J.B., Kolch,W., Ogunnaike,B.A. and Kholodenko,B.N. (2012) Emergence of bimodal cell population responses from the interplay between analog single-cell signaling and protein expression noise. *Bmc Syst. Biol.*, **6**, 109.
15. Korthauer,K.D., Chu,L.-F., Newton,M.A., Li,Y., Thomson,J., Stewart,R. and Kendziorski,C. (2016) A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biol.*, **17**, 222.
16. Finak,G., McDavid,A., Yajima,M., Deng,J., Gersuk,V., Shalek,A.K., Slichter,C.K., Miller,H.W., McElrath,M.J., Prlic,M. *et al.* (2015) MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.*, **16**, 278.
17. Lun,A.T.L., Bach,K. and Marioni,J.C. (2016) Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol.*, **17**, 75.
18. Zhang,A.W., O'Flanagan,C., Chavez,E.A., Lim,J.L.P., Ceglia,N., McPherson,A., Wiens,M., Walters,P., Chan,T., Hewitson,B. *et al.* (2019) Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. *Nat. Methods*, **16**, 1007–1015.
19. Zhang,X., Xu,C. and Yosef,N. (2019) Simulating multiple faceted variability in single cell RNA sequencing. *Nat. Commun.*, **10**, 2611.
20. Duò,A., Robinson,M.D. and Soneson,C. (2018) A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000research*, **7**, 1141.
21. Zheng,G.X.Y., Terry,J.M., Belgrader,P., Ryvkin,P., Bent,Z.W., Wilson,R., Ziraldo,S.B., Wheeler,T.D., McDermott,G.P., Zhu,J. *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.*, **8**, 14049.
22. Schaum,N., Karkanias,J., Neff,N.F., May,A.P., Quake,S.R., Wyss-Coray,T., Darmanis,S., Batson,J., Botvinnik,O., Chen,M.B. *et al.* (2018) Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature*, **562**, 367–372.
23. Lun,A.T.L., McCarthy,D.J. and Marioni,J.C. (2016) A step-by-step workflow for low-level analysis of single-cell RNA-seq data with bioconductor. *F1000research*, **5**, 2122.
24. Zhang,X., Lan,Y., Xu,J., Quan,F., Zhao,E., Deng,C., Luo,T., Xu,L., Liao,G., Yan,M. *et al.* (2018) CellMarker: a manually curated resource of cell markers in human and mouse. *Nucleic Acids Res.*, **47**, gky900.
25. Franzén,O., Gan,L.-M. and Björkegren,J.L.M. (2019) PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database*, **2019**, baz046.
26. Coordinators,N.R., Agarwala,R., Barrett,T., Beck,J., Benson,D.A., Bollin,C., Bolton,E., Bourexis,D., Brister,J.R., Bryant,S.H. *et al.* (2017) Database resources of the national center for biotechnology information. *Nucleic Acids Res.*, **46**, D8–D13.
27. Dotiwala,F. and Lieberman,J. (2019) Granulysin: killer lymphocyte safeguard against microbes. *Curr. Opin. Immunol.*, **60**, 19–29.
28. Dotiwala,F., Mulik,S., Polidoro,R.B., Ansara,J.A., Burleigh,B.A., Walch,M., Gazzinelli,R.T. and Lieberman,J. (2016) Killer lymphocytes use granulysin, perforin and granzymes to kill intracellular parasites. *Nat. Med.*, **22**, 210–216.
29. Fragoso,R.C., Su,M.W.-C. and Burakoff,S.J. (2002) Encyclopedia of cancer (Second Edition). *Immunol Article Titles T*, https://doi.org/10.1016/b0-12-227555-1/00235-5.
30. Gong,W., Kwak,I.-Y., Pota,P., Koyano-Nakagawa,N. and Garry,D.J. (2018) DrImpute: imputing dropout events in single cell RNA sequencing data. *Bmc Bioinformatics*, **19**, 220.
31. Huang,M., Wang,J., Torre,E., Dueck,H., Shaffer,S., Bonasio,R., Murray,J.I., Raj,A., Li,M. and Zhang,N.R. (2018) SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods*, **15**, 539–542.
32. Svensson,V. (2020) Droplet scRNA-seq is not zero-inflated. *Nat. Biotechnol.*, **38**, 147–150.
33. Kim,T.H., Zhou,X. and Chen,M. (2020) Demystifying "drop-outs" in single-cell UMI data. *Genome Biol.*, **21**, 196.