Optimal marker genes for *c*-separated cell types with SepSolve

Bartol Borozan, Tomislav Prusina, Luka Borozan, et al.

Genome Res. published online October 9, 2025
Access the most recent version at doi:10.1101/gr.280637.125

Optimal marker genes for *c*-separated cell types with SepSolve

Bartol Borozan[1], Tomislav Prusina[1], Luka Borozan[1], Domagoj Ševerdija[1], Francisca Rojas Ringeling[2], Domagoj Matijević[*1], and Stefan Canzar[†2]

[1]School of Applied Mathematics and Informatics, University of Osijek, 31000 Osijek, Croatia

[2]Faculty of Informatics and Data Science, University of Regensburg, 93053 Regensburg, Germany

Abstract

The identification of cell types in single-cell RNA-seq studies relies on the distinct expression signature of marker genes. A small set of target genes is also needed to design probes for targeted spatial transcriptomics experiments and to target proteins in single-cell spatial proteomics or for cell sorting. While traditional approaches have relied on testing one gene at a time for differential expression between a given cell type and the rest, more recent methods have highlighted the benefits of a joint selection of markers that together distinguish all pairs of cell types simultaneously. However, existing methods either consider all pairs of individual cells which becomes intractable even for medium-sized datasets, or ignore intra-cell type expression variation entirely by collapsing all cells of a given type to a single representative. Here we address these limitations and propose to find a small set of genes such that cell types are *c*-separated in the selected dimensions, a notion introduced previously in learning a mixture of Gaussians. To this end, we formulate a linear program that naturally takes into account expression variation within cell types without including each pair of individual cells in the model, leading to a highly stable set of marker genes that allow to accurately discriminate between cell types and that can be computed to optimality efficiently.

[*]domagoj@mathos.hr
[†]stefan.canzar@ur.de

Introduction

A common first step in the analysis of single-cell data, such as those generated in atlases of the brain and other human organs and tissues (He et al. 2020b; Chen et al. 2022), is the characterization of the different cell types and their states based on the distinct expression of so-called marker genes. Typically, cells are first grouped according to their overall transcriptional similarity by a computational clustering method such as Seurat (Hao et al. 2021) and SCANPY (Wolf et al. 2018). Then genes are statistically tested for differential expression between one cluster and the rest (one-vs-rest) (Hao et al. 2021; Wolf et al. 2018), or between all pairs of clusters (package cran) (Lun et al. 2016). The detected marker genes are then used to label and biologically interpret cell clusters in an experimental context. Other general purpose statistical or machine learning based methods include the ANOVA F-test and decision trees (Quinlan 1986). Minimum-redundancy-maximum-relevance (mRMR) (Peng et al. 2005), on the other hand, does not only consider the relevance of a feature for an outcome but also takes into account redundancy among features. Similarly, ReliefF (Kononenko 1994) uses nearest neighbors to indirectly account for feature dependencies. Other ranking based methods include SMaSH (Nelson et al. 2022) and RankCorr (Vargo and Gilbert 2020) which have recently been benchmarked (Pullin and McCarthy 2024) against several other methods. All of these feature-selection methods belong to the class of filter methods that do not interact with a classifier.

While a one-vs-rest comparison of gene expression allows to differentiate between major cell types, it might fail to detect genes whose expression differs only between similar subtypes, different tissues, or cell types in different disease states (Hasanaj et al. 2022; Dumitrascu et al. 2021). In addition, such a cluster-by-cluster, gene-by-gene approach will introduce redundancy into the set of selected genes, which makes it unsuitable for applications that rely on a small set of informative genes. For example, prominent fluorescence in situ hybridization (FISH)-based spatial transcriptomics technologies can assay only a limited number of genes that need to be pre-selected. Similarly, fluorescence-activated cell sorting (FACS) relies on a small number of distinct surface markers to sort cells by their types.

Rather than finding genes whose expression is distinct in a given cluster, recent methods have therefore studied a variant of the marker gene selection problem, which seeks to identify a single set of a small number of genes that jointly distinguish all clusters simultaneously. Existing methods

model this task as a combinatorial optimization problem and differ mostly in how they measure how much a given gene contributes to the discrimination between two clusters or cell types. scGeneFit (Dumitrascu et al. 2021) and G-PC (Hasanaj et al. 2022) select marker genes such that distances between all pairs of cell types are above a certain threshold. The former method uses a linear program (LP) to minimize, for a fixed number of genes, the violation of the minimum required separation. The latter phrases the problem of minimizing the number of genes necessary to satisfy all distance requirements as a variant of the set cover problem which it solves using a greedy approach. Similarly, the method used in Langlieb et al. 2023 uses set cover to minimize the number of genes to combinatorially distinguish a given cell type from all others in a single-nucleus RNA-seq data set of the mouse brain. It solves this set cover problem for each cell type using integer linear programming (ILP) and then uses a second ILP to integrate the resulting gene sets while reducing redundancy. To measure the separation between cell types, scGeneFit and G-PC use squared Euclidean or Manhattan distance, respectively, in the dimensions of the selected marker genes. Both methods compute the distance between single representatives (e.g. centroid) of each cell type. Alternatively, scGeneFit allows to explicitly model the separation of each pair of individual cells from two different types. The combinatorial method used in Langlieb et al. 2023 does not take into account any form of distance between cell types but considers 'distinguishable' as a binary property, where each gene either distinguishes between two cell types or not, depending on the difference in the fraction of cells expressing it.

While one-vs-rest methods based on a statistical test of differential expression naturally take into account expression variation of a given gene across cells of the same type, combinatorial methods that aim to cover or separate all pairs of cell types with a single set of marker genes ignore within-cell type variation entirely by collapsing all cells of a given type to a single representative. On the other hand, including all pairs of individual cells in an LP, as scGeneFit does, becomes intractable even for small- and medium-sized datasets. It therefore selects a small set of constraints (5000 by default) among typically many millions of possible cell pairs at random, thus ignoring large parts of the generated data. Here we address these limitations and propose a combinatorial method, SepSolve, that takes into account gene expression variation. Our method's objective is to find a small set of genes such that cell types are c-separated in the selected dimensions. In previous work (Dasgupta 2000), two Gaussians have been defined to be c-separated if their centers

are c radii apart. Random projection was then used to reduce dimensionality while approximately retaining the separation of a mixture of Gaussians. Instead, here we formulate an ILP based on a linear approximation, which we heuristically solve to search for such a subspace without making the assumption of normality. We provide experimental evidence that this strategy yields a compact, biologically meaningful set of genes that can accurately discriminate between distinct and closely related cell types.

Results

Overview of SepSolve

SepSolve is designed to identify a small subset of genes such that the spheres enclosing most of the cells of a given type overlap minimally in the space of selected genes (Fig. 1A). To formalize this objective, we build on the concept of c-separation, as introduced in Dasgupta 2000 for Gaussian mixtures. Specifically, SepSolve achieves c-separation of cell types by selecting genes whose mean expressions are separated by at least the scaled radius of the spheres, with the scaling determined by a separation parameter c. Larger values of c imply stricter separation with less overlap. A 2-separated mixture will be almost entirely separated, while 1- or $\frac{1}{2}$-separated clusters will have a small overlap (Dasgupta 2000). The radii of the spheres are calculated using the trace of the covariance matrices, which we demonstrate captures the majority of intra-cell-type expression variation (Methods and Supplemental Fig. 1).

We cast the search for a gene subspace that (almost) induces c-separation as a constrained optimization problem where for a given number m of marker genes, the violation of c-separation between pairs of cell types is minimized. After linearizing constraints using Taylor expansions we obtain an integer linear programming formulation of the problem. Rather than solving it to optimality (e.g. via branch-and-bound), we solve the linear relaxation and select genes corresponding to variables with the m largest values. A detailed description of our method can be found in Methods.

Fig. 1B demonstrates that 50 marker genes selected by SepSolve among 10,000 highly variable genes (hvg) in a human lung scRNA-seq dataset (Madissoon et al. 2019) are sufficient to separate cell types in a uniform manifold approximation and projection (UMAP) embedding.

To illustrate the benefits of taking variance into account in the combinatorial model, we have

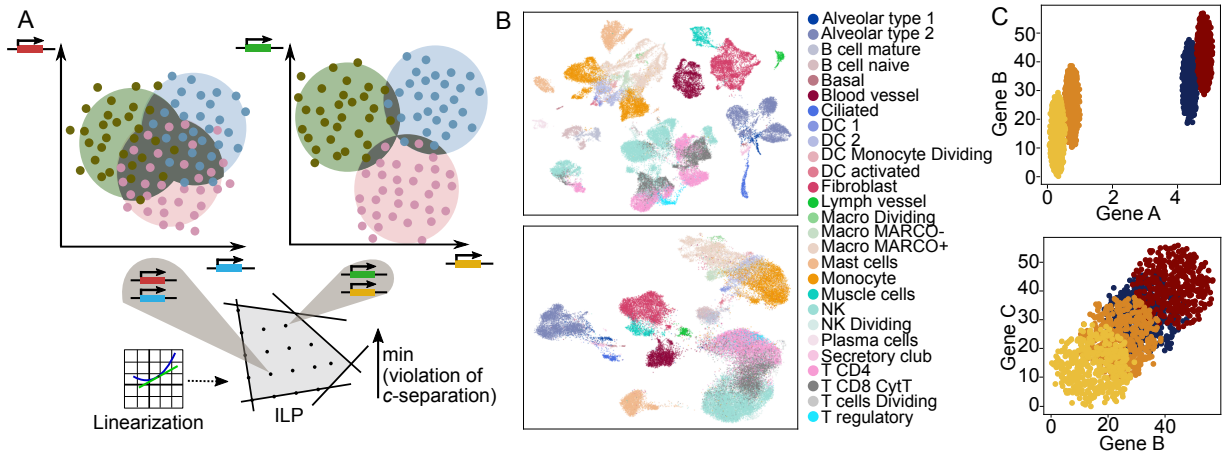


Figure 1: SepSolve overview and proof of principle. *(A)* Overview of the SepSolve algorithm. Using linearizations, SepSolve models the search for marker genes as an integer linear program. Each grid point within the feasible region defined by hyperplanes corresponds to a candidate set of marker genes (two in this example). The optimal solution (here, the highest grid point) minimizes the overlap (dark area) between balls enclosing most cells of a given type, or more formally, minimizes total violation of c-separation between cell types. *(B)* UMAP embedding of cells in a human lung dataset (Madissoon et al. 2019), using 10,000 hvg (top) or 50 markers selected by Sepsolve (bottom). *(C)* Cells from four different types in the projected space of marker genes computed by SepSolve (top) and scGeneFit and G-PC (bottom). The plots show simulated expression levels of two genes (specified on the axes).

created a synthetic three-dimensional spherical dataset consisting of four different labels (i.e. cell types) and three genes A, B and C. The mean expression levels of gene A do not differ much between the first two cell types and between the last two, with a difference of only 0.45 units. However, across all cell types, the variance of expression levels of gene A is low, namely 0.50. The expression means of genes B and C are equidistantly distributed, with a relatively large distance of 8 between neighboring cell types. However, the expression of genes B and C exhibit a high variance of 12 across all cell types. As illustrated in Fig. 1C, the marker gene selection methods scGeneFit and G-PC, which ignore gene expression variation, choose genes B and C. SepSolve, on the other hand, leverages very small variation in the expression of gene A and selects it as a marker, yielding a much sharper separation of cell types.

SepSolve selects genes that classify cells more accurately

Similar to the benchmarks performed in Dumitrascu et al. 2021 and Hasanaj et al. 2022, we evaluated the discriminatory properties of the identified marker sets by the ability of a classifier to

predict cell types based on them.

We compared the performance of SepSolve to most recent combinatorial methods scGeneFit (Dumitrascu et al. 2021) and G-PC (Hasanaj et al. 2022). On smaller datasets, scGeneFit was run in both pairwise and centers mode, including constraints for all pairs of cells or only between single representatives (centers) for each cell type, respectively. The authors' experiments have shown that the centers mode is the most efficient and stable one. Indeed, attempts to run scGeneFit in pairwise mode on the larger datasets failed because it exceeded available memory or a time limit of 12 hours. No implementation of the core combinatorial optimization algorithm described in Langlieb et al. 2023 was available. However, we note that the ILP formulation in Langlieb et al. 2023 tackles a variant of the set cover problem, in the same spirit as G-PC but based on a different notion of distinguishability. In addition, we compared SepSolve with the selection of differentially expressed genes (DE), and filter methods SMaSH (Nelson et al. 2022), RankCorr (Vargo and Gilbert 2020), minimum-redundancy-maximum-relevance (mRMR) (Peng et al. 2005), and mutual information (Kraskov et al. 2004). SMaSH failed to complete within the 12-hour time limit on all but the smallest dataset.

We benchmarked methods on seven different scRNA-seq datasets (Table 1). We included all three datasets used in Hasanaj et al. 2022 from the human lung (IPF), mouse cortex, and a human cell atlas (HCA). Consistent with Hasanaj et al. 2022, we restricted the IPF dataset to the healthy samples. While IPF and MC samples were obtained from a single tissue, in HCA identical cell types occurred in multiple tissues. We defined cell type labels by either merging cell types across tissues or by distinguishing identical cell types originating from different tissues. We used an additional dataset (Zheng8eq) of peripheral blood mononuclear cells (PBMCs) in our benchmark which included both distinct and similar cell types Pan et al. 2023a. We further included two human lung datasets (MaL and MeL) that were used in Kuemmerle et al. 2024 to benchmark probe set selection method Spapros. We used them in the next Section to assess SepSolve's cross-dataset classification performance. True cell-type labels were taken from the original publications. In the Zheng8eq dataset the authors randomly mixed roughly equal proportions of presorted B cells, CD14 monocytes, naive cytotoxic T cells, regulatory T cells, CD56 NK cells, memory T cells, CD4 T helper cells, and naive T cells. In this case, true labels were independent of scRNA-seq measurements.

Dataset	#Cells	#Genes	#Cell Types	Reference
Idiopathic pulmonary fibrosis (IPF)	96,303	45,947	38	Adams et al. 2020
Human Cell Atlas (HCA)	84,363	22,653	24/196	He et al. 2020a
Mouse cortex (MC)	3,005	20,006	7	Zeisel et al. 2015
PBMC (Zheng8eq)	3,971	1,571	8	Duò et al. 2018
Meyer lung (MeL)	193,108	33,538	80	Madissoon et al. 2023
Madissoon lung (MaL)	57,020	25,204	28	Madissoon et al. 2019
Fetal liver (FL)	113,063	27,080	27	Popescu et al. 2019

Table 1: Datasets used in this study. For HCA, the number of cell types is shown when merging labels across tissues (24) and when distinguishing the tissue of origin (196).

Consistent with the analysis in Dumitrascu et al. 2021 and Hasanaj et al. 2022, we used the marker genes found by each method in k-nearest neighbors (k-NN) and logistic regression classifiers. For the logistic regression classifier we split the data into 70% for training and 30% for testing. As in Hasanaj et al. 2022, we measured the performance of the classifiers using the macro-average F1 score which corrects for an unbalanced cell type composition and gives higher weight to rare cell types. It is computed as the mean F1 score across all cell types.

Fig. 2A and Supplemental Fig. 2 highlight the improved performance of the logistic regression classifier in predicting cell type labels based on marker genes identified by SepSolve. The increase in F1 score is particularly noticeable when using a small number of marker genes in the FL dataset (Fig. 2A) and in datasets HCA (with merged cell type labels) and Zheng8eq (Supplemental Fig. 2), where F1 scores of best-performing methods converge to similar F1 values as the number of markers increases. In contrast, on the HCA (with labels split between tissues), MeL and IPF datasets (Fig. 2A), SepSolve's improvement over competing methods becomes more pronounced with an increasing number of marker genes. In the case of the MeL dataset, cell classification with k-NN outperforms logistic regression, particularly with a smaller set of markers (Supplemental Fig. 3). For both classifiers, the performance of methods which perform competitively with fewer markers, namely scGeneGit and mutual information, declines as the number of markers increases. In contrast, DE shows the opposite trend, with performance improving as the number of markers increases but showing worse performance for a small number of markers. Notably, on the Zheng8eq dataset (Supplemental Fig. 2), only the performance of DE approaches that of SepSolve, leaving a significant gap between SepSolve and all other methods, including the combinatorial algorithms G-PC and scGeneGit. The smallest improvement can be observed on the IPF dataset, but the advantage

of our method becomes more evident when using the slightly more accurate k-NN classifier. Overall, using k-NN instead of logistic regression yields consistent results (Supplemental Fig. 3). The mouse cortex (MC) is the only instance where SepSolve's gene sets of fewer than 30 genes yielded lower F1 scores than classical differential expression testing (Supplemental Fig. 2). Notably, even RankCorr and mRMR–two of the poorest-performing methods across all other datasets–achieved competitve results on this dataset. Notably, SMaSH completed successfully within the 12-hour time limit on this dataset (it failed on all others), but performed the worst across all tested numbers of marker genes. This finding is consistent with the result in Kuemmerle et al. 2024, where SmASH took 15 hours to complete on the MeL dataset and performed poorly in classification.

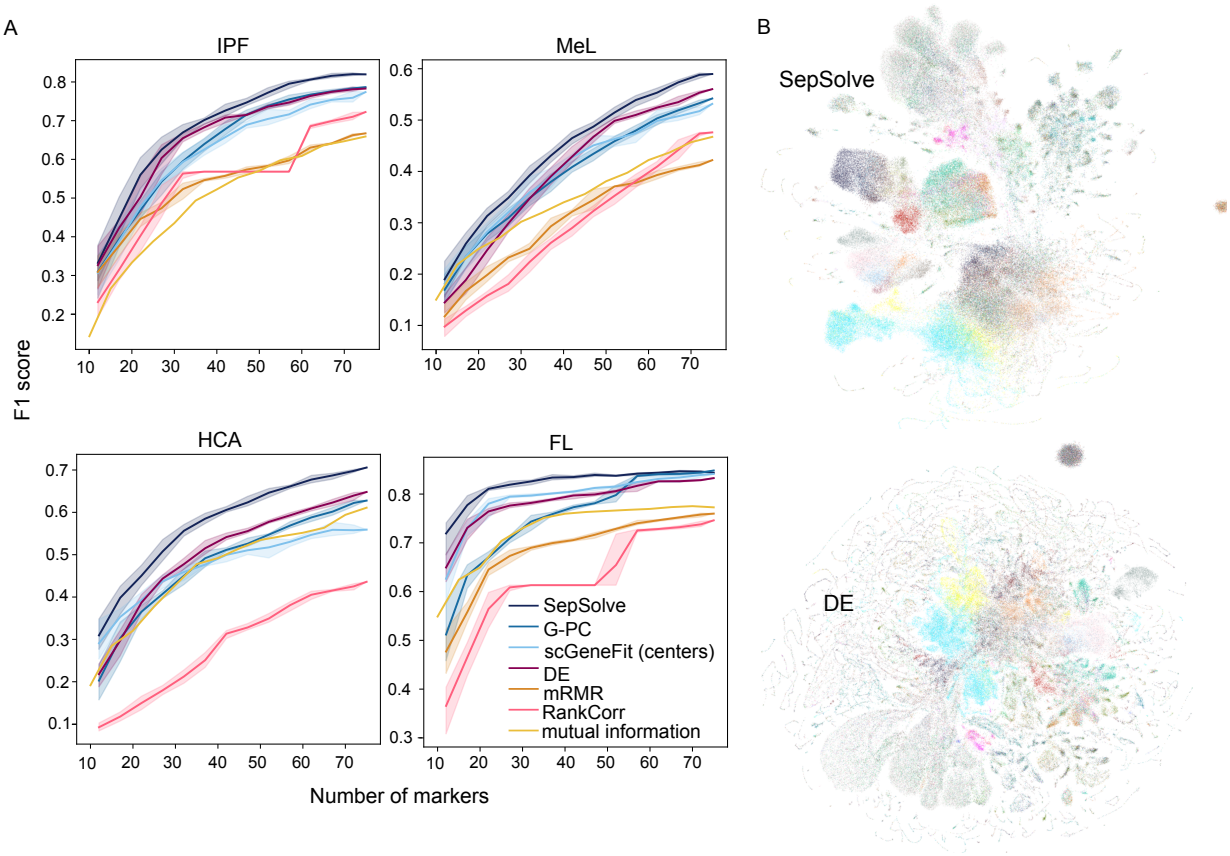


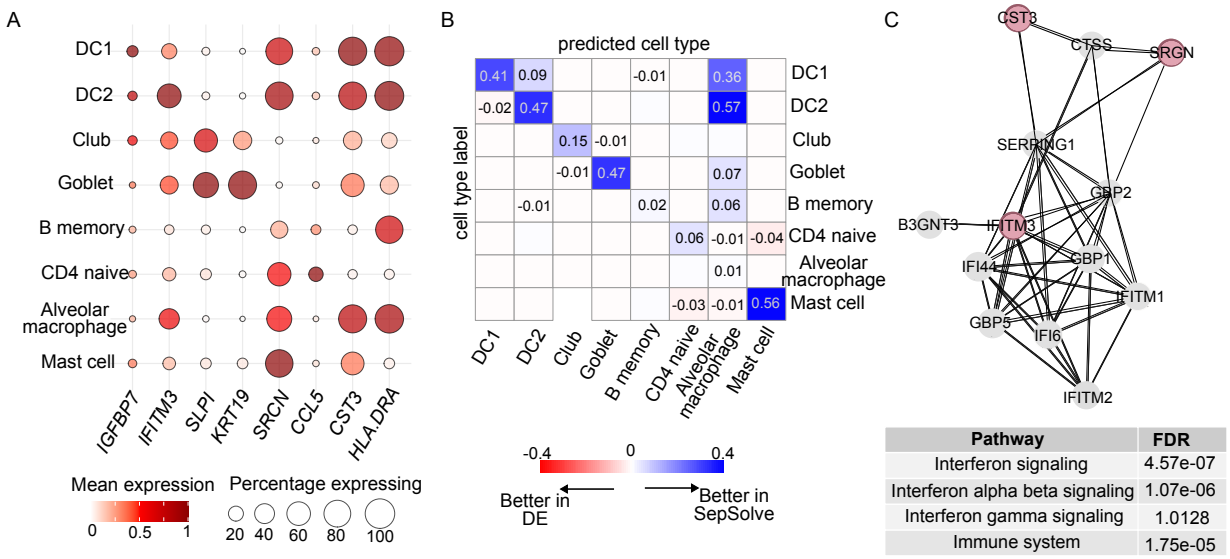
Figure 2: Improvements in cell type classification and visualization. *(A)* F1 scores of a logistic regression classifier when provided varying numbers of marker genes computed by the different methods on the four datasets. On HCA, cell type labels distinguished the tissue of origin. scGeneFit ran successfully in pairwise mode only on the two smallest datasets, SMaSH only on Zheng8eq (see text for details). Shaded regions depict standard deviation. *(B)* UMAP embeddings generated from the 20 marker genes selected by SepSolve and DE on the human lung dataset MeL. UMAP embeddings of the original space and of marker genes selected by the remaining methods are in Supplemental Fig. 5, along with the complete cell type legend.

Even when compared to recent method Spapros (Kuemmerle et al. 2024) that does not belong to the class of filter methods but uses the results of a classifier to select marker genes, SepSolve's performance is competitive (Supplemental Fig. 4) and again shows advantages for small numbers of marker genes on the three largest datasets (MeL, HCA, and FL). The greatest advantage of SepSolve is evident when distinguishing identical cell types across different tissues in the human cell atlas. On the two smallest datasets, Zheng8eq and mouse cortex, classifier-based marker selection by Spapros performed slightly better than SepSolve for certain marker set sizes. However, in principle, access to a classifier could also enhance SepSolve's classification performance by allowing optimization of its separation parameter c (Methods). As shown in Supplemental Fig. 4, SepSolve's performance improves slightly on most datasets, and substantially on some (HCA, and MC for < 30 marker genes) when c is tuned to the classification task. In the same figure, we also compare this strategy to scGeneFit, which supports hyperparameter tuning of its target cell separation parameter via a dual annealing approach. scGeneFit also benefits from classifier-based parameter tuning, especially on datasets Zheng8eq and HCA, but does not reach the performance of tuned SepSolve. Among combinatorial methods, scGeneFit is unique in offering built-in hyperparameter optimization.

SepSolve's combinatorial gene marker approach enhances the accuracy of cell type classification in a single-cell lung dataset (MeL) even when choosing a limited set of 20 markers (Fig. 2A), underscoring the efficiency of this approach. Overall, this improvement is further supported by a visual comparison of UMAP embeddings generated from the 20 marker genes selected by the top-performing methods (Fig. 2B and Supplemental Fig. 5). For instance, the simultaneous assessment of *IFITM3*, *SRGN*, and *CST3* expression enables a more accurate distinction between type 1 and type 2 dendritic cells compared to traditional differential expression methods (Fig. 3A and Fig. 3B). Constructing a protein-protein interaction network centered on these three genes reveals an enrichment in interferon signaling pathways (Fig. 3C), which are integral to dendritic cell differentiation (Simmons et al. 2012). This observation underlines the biological relevance of selecting these specific markers.

SepSolve performs robustly across datasets, parameters, and perturbations

In the context of feature selection, stability refers to the degree of consistency with which a feature is identified as relevant across different conditions. To be practical–such as when designing probes



Pathway	FDR
Interferon signaling	4.57e-07
Interferon alpha beta signaling	1.07e-06
Interferon gamma signaling	1.0128
Immune system	1.75e-05

Figure 3: Markers selected by Sepsolve on the MeL lung single-cell dataset. *(A)* Each dot's color represents the scaled mean expression of a marker within a specific cell type, normalized across all cell types. The size of each dot corresponds to the percentage of cells within that cell type expressing the marker. *(B)* Confusion matrix heatmap comparing cell classification accuracy between markers identified by Sepsolve and those selected via differential expression analysis. Positive values (blue) indicate superior classification accuracy using Sepsolve markers, whereas negative values (red) denote better performance with differential expression markers. *(C)* Protein-protein interaction network centered on IFITM3, SRGN, and CST3, constructed using the STRING database. Nodes represent proteins, and edges denote predicted functional associations based on various evidence types, including curated databases and experimental data. The accompanying table lists pathways enriched within this network, along with their associated false discovery rate (FDR) values. DC1, type 1 dendritic cell; DC2, type 2 dendritic cell; DE, Differential expression

for a targeted spatial transcriptomics experiment based on a reference scRNA-seq sample–a set of marker genes must not only accurately discriminate between cell types but also remain stable across varying experimental conditions, including the number of profiled cells and different noise levels. Fig. 4A shows that markers selected by SepSolve based on one human lung dataset can be used to accurately classify cells in another scRNA-seq dataset generated from the same tissue. To more comprehensively assess marker stability, we followed the approach in Hasanaj et al. 2022 and calculate the stability of a collection of marker gene sets $\{S_1, S_2, \ldots, S_n\}$ as the average Jaccard similarity of all pairs of sets:

$$s = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j>i}^{n} \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$$

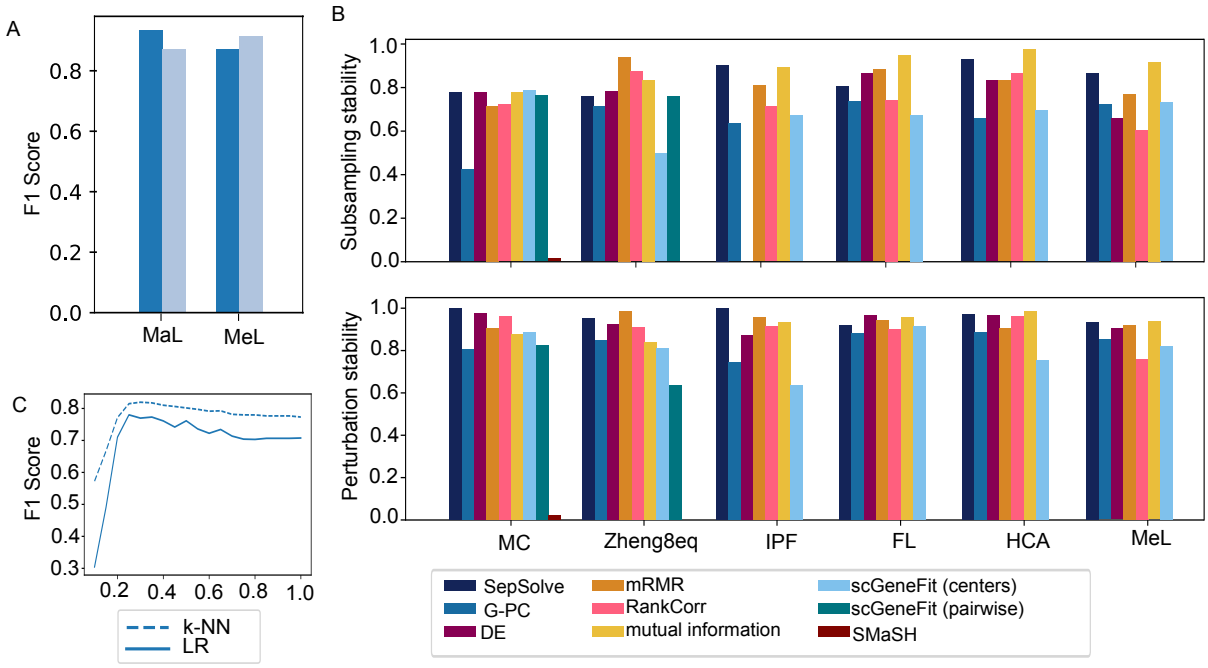


Figure 4: Stability of selected markers. *(A)* Cross dataset F1 scores of a logistic regression classifier. SepSolve selected 20 marker genes in the MaL (left) or MeL (right) human lung dataset. Classification performance was evaluated either on the same dataset (dark blue) or the other (light blue). *(B)* Stability of 50 marker genes computed by the different methods on random subsamples of cells (top) or perturbed counts (bottom). DE crashed on subsampled IPF data since an insufficient number of cells per cell type remained. *(C)* F1 scores of a logistic regression and a k-NN classifier on dataset IPF when using 50 marker genes selected by SepSolve for varying separation constant c.

We evaluated the stability of markers computed by the different methods in two experiments. To avoid a potential confounding influence of the robustness of highly variable gene (hvg) detection on overall stability, we omitted the selection of hvgs in these experiments. First, consistent with Hasanaj et al. 2022, we randomly sampled 50% of input cells in 5 random trials. We applied each method to every subsample and then calculated the stability of the marker gene sets identified by each method. Fig. 4B (top) shows that SepSolve returns a highly stable set of marker genes with between around 76% and 93% overlap across all six datasets. The slightly higher stability observed on datasets IPF, HCA, and MeL might be due to the large number of profiled cells. In contrast, the markers selected by the two combinatorial methods scGeneFit and G-PC showed lower stability across almost all dataset. In the worst case, only half or less than half of marker genes returned by scGeneFit and G-PC overlapped across runs. Notably, taking into account intra-cell-type variability

by including constraints between pairs of cells in scGeneFit (pairwise mode) instead of collapsing cell types to a single representative (centers mode) improved scGeneFit's stability (Fig. 4B (top) and Supplemental Fig. 6 (top)). The most stable method is mutual information; however, as shown in the previous section, it did not yield a discriminative set of genes.

In addition, we assessed stability with respect to a small data perturbation. To this end, for each entry e in the raw count matrix, we sampled from a Poisson distribution with expectation $0.05 \cdot e$ and either added or subtracted the obtained value from the matrix entry with equal probability. Again, this perturbation was applied 5 times, and the stability of detected markers was evaluated. Overall, the marker gene sets were less impacted by this perturbation compared to the subsampling experiment (Fig. 4B (bottom)). Notably, marker genes found by SepSolve exhibited over 90% overlap across datasets. Again, combinatorial methods, scGeneFit and G-PC, demonstrated lower stabiliy than SepSolve across all datasets. With 64% and 75%, they achieved the lowest stability values on the IPF dataset. Subsampling and permutation stability showed consistent patterns when using 25 instead of 50 markers (Supplemental Fig. 6).

Beyond SepSolve's robustness to data perturbations, we also evaluated its sensitivity to the main algorithmic parameter, the target separation c. As shown in Fig. 4C and Supplemental Fig. 7, both classifiers maintain consistently high F1 scores when the separation exceeds a minimum threshold of approximately 0.3, with only a gradual decline observed as c increases further.

SepSolve scales to large datasets

Fig. 5 displays the average running times of the different methods across all numbers of target marker genes evaluated in Fig. 2 for the six datasets. All experiments were run on an AMD Ryzen Threadripper 3990X 64-Core Processor @ 4.3GHz, 256 GB of RAM, and Python version 3.10.12. Except for scGeneFit run in pairwise mode, running times exhibit minimal variability with respect to the number of target genes (Supplemental Fig. 8). As expected, G-PC's greedy heuristic is the fastest, completing in a maximum of 2.46 seconds on the largest dataset (MeL). SepSolve follows as the second-fastest method. Solving a linear program and subsequent rounding in SepSolve took only 49 seconds on the largest dataset (MeL), while scGeneFit, run in the more efficient centers mode, took 285 seconds to complete. In pairwise mode, the running time of scGeneFit increased to 160 seconds on average on the smaller mouse cortex data, with a single instance requiring up to

9,206 seconds on the same dataset. In this mode, scGeneFit failed to run on all but the two smallest datasets due to memory and time limits (12h). Similary, SMaSH successfully ran only on the MC dataset. In Kuemmerle et al. 2024, SMaSH took 15h to find markers in the MeL dataset. Spapros, the only non-filter based method included as a reference in this benchmark, was the slowest method. On the largest lung dataset (MeL) it took 13,105 seconds ($\approx$ 3.5 hours) to complete, compared to under a minute for SepSolve. Even on the smallest MC dataset, it required 159 seconds, whereas SepSolve completed the task in under a second. With few exceptions, the remaining methdos mRMR, RankCorr, DE, and mutual information were at least an order of magnitude slower than SepSolve.

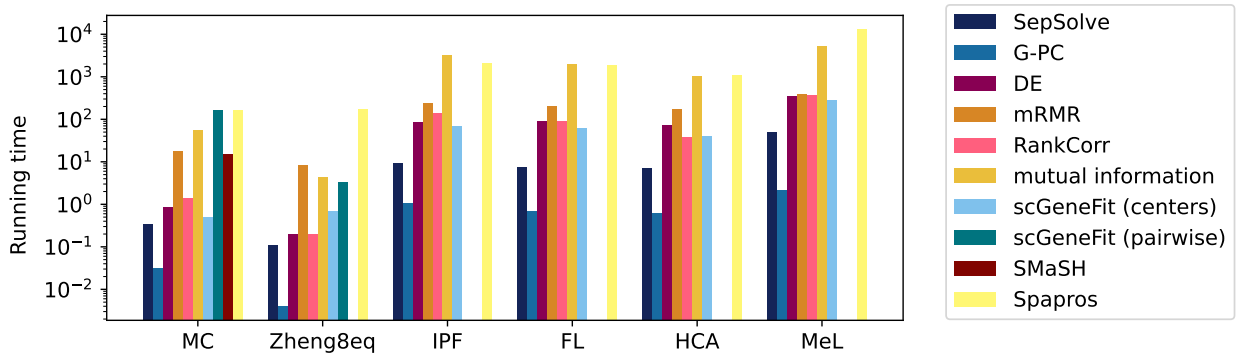


Figure 5: Average running times (log scale, in seconds) over all evaluated target numbers of marker genes.

Discussion

We have proposed method SepSolve, a combinatorial approach that tries to discriminate between all cell types simultaneously in the space of marker genes. In contrast to existing methods in this class, it takes into account gene expression variability. Transcriptional heterogeneity within cell types can be due to different, potentially unknown, cell states and transitions between them, but can also have technical reasons such as noisy experimental measurements.

SepSolve annotated cell types more accurately, especially for a small number of marker genes which is the most common application scenario, for example when designing probes for a targeted spatial transcriptomics assay. We attribute the higher classification accuracy with fewer markers to SepSolve's combinatorial selection of genes. It allows to discriminate between similar subtypes,

which may have overlapping sets of differentially expressed genes. This was demonstrated by the accurate distinction between closely related type 1 and type 2 dendritic cells in a complex lung dataset and by the significant improvement over competing methods when distinguishing identical cell types across different tissues in the human cell atlas. In addition, SepSolve effectively leverages the additional information contained in the within-cell-type expression variation. The value of this is indicated by the substantial improvements in the F1 score by scGeneFit on the Zheng8eq dataset, when constraints for all cell pairs (rather than single representatives) were included in an otherwise identical optimization model.

Our method falls under the category of filter methods, which do not optimize a specific classifier during feature selection, in contrast to wrapper and embedded methods (Hasanaj et al. 2022). As a result, the low-dimensional subspace in which we represent cell cluster structures can be beneficial for other tasks, such as the deconvolution of bulk expression data, as demonstrated in Hasanaj et al. 2022, or for visualization purposes. Despite this, SepSolve performed favorably in cell type classification compared to Spapros, a recently proposed embedded method that utilizes the result of a classifier (decision trees) to identify marker genes.

An additional benefit of our method is that it provides a more stable set of markers compared to other combinatorial methods when perturbing or re-sampling the data. We showed that this also allows for an accurate cross-dataset classification of cell types. This is a particularly useful property, for example, when designing spatial probes based on dissociated reference data. Again, we attribute this mainly to taking into account transcriptional heterogeneity within cell types in our model without including every pair of individual cells as scGeneFit does in the pairwise mode. In addition, we optimally solve an LP formulation of the problem, compared to the greedy scheme applied by G-PC that might be more sensitive to small data perturbations (Hasanaj et al. 2022). Nevertheless, like all selection methods, SepSolve will be susceptible to technical variations, such as batch effects. Whenever possible, selected genes should be compared across batches.

Furthermore, even though SepSolve takes into account within-cell-type variability, it is fast and can be applied to very large datasets. This is in contrast to scGeneFit, especially when trying to account for expression variability in pairwise mode.

Regarding the selection of the main algorithmic parameter in our method, the target separation c, our experiments showed that a single value (0.4)) effectively produced discriminative sets of

marker genes across a diverse range of samples. These included different tissues and species, with varying numbers of profiled cells (ranging from approximately 3,000 to 193,000) and cell types (ranging from 7 to 38). Furthermore, we showed that, beyond a certain minimum threshold, a broad range of values for c yielded similar classification accuracy, highlighting the robustness of the method. However, as shown in Supplemental Fig. 4, there exist scenarios where a tailored choice of c for a specific downstream task (e.g., classification) can significantly enhance SepSolve's performance. While SepSolve was not originally designed for classifier-specific marker gene selection, the software package provides a grid-search option to fine-tuning c when adapting the method for this purpose.

Furthermore, by leveraging linear programming as the underlying framework for identifying marker genes, SepSolve can seamlessly incorporate prior biological knowledge or technical constraints into the model (Kuemmerle et al. 2024). For instance, disease-relevant genes can be prioritized by setting the corresponding variable to 1. This approach ensures that these pre-selected genes are included, while the algorithm identifies additional genes that best complement them, enhancing the relevance and specificity of the selected marker genes.

The number of marker genes required to distinguish cell types depends on several factors, including the intended application and dataset characteristics. First, experiments typically impose constraints on the number of marker genes. When designing a targeted spatial experiment, for example, this number is defined by the size of the probe panel. Spapros, for example, was used in Kuemmerle et al. 2024 to find 64 genes that were targeted in a SCRINSHOT (Sountoulidis et al. 2020) experiment on human lung samples.

In addition, the complexity and transcriptional heterogeneity of cell types and the states they are in dictate the necessary number of markers. For example, when aiming to separate only the seven major cell types in our experiments on the mouse cortex sample, a straightforward DE approach was sufficient to classify cells with more than 90% accuracy using just 10 markers. In the original publication (Zeisel et al. 2015), however, 47 subclasses were identified that potentially require not only a larger number but also a combinatorial selection of genes to highlight subtle transcriptional differences between similar subtypes. In such scenarios involving fine-grained cell type distinctions, our experiments suggest that SepSolve's combinatorial selection of genes reduced the number of genes necessary to distinguish subtle transcriptional differences compared to one-vs-all approaches.

Moreover, the discriminative power of gene sets typically increases with size. But rather than

simply increasing the number of genes to achieve a better global separation of cell types, our optimization model can be geared towards the separation of specific cell types that might be known, for example, to be relevant in a given disease context. By introducing a penalty term for the corresponding variable in our objective function, the separation of the two cell types can be forced towards a given target separation. The success of this strategy can be validated by the user by examining the achieved separation for each pair of cell types reported by SepSolve. Future work will focus on tailoring our model to such use cases in complex biological settings.

A current limitation of our method is that total expression variance, i.e. the trace of the sample covariance, is used as target separation between cell types, inspired by previous work (Dasgupta 2000) on random projections of high-dimensional Gaussians. Even though we empirically showed that this target separation is justified for single-cell data, more tailored measures might exist. The flexibility of an LP formulation will allow us to investigate alternative measures of separation in future work. A second limitation is the simple ranking scheme we applied to convert fractional solutions of the LP to an integral one. Exact ILP solving with Gurobi already became impractical for moderately sized instances. We will explore more accurate approaches such as branch and bound with heuristic speed-ups and (randomized) rounding schemes with potential guarantees in the future.

Methods

SepSolve optimization

Our approach exploits the notion introduced in Dasgupta 2000 that formalizes that a mixture of Gaussians is c-separated if its component Gaussians are pairwise c-separated.

Definition 1. Let $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$ represent two Gaussians in $\mathbb{R}^d$, where μ_1 and μ_2 are the means, and Σ_1, Σ_2 are the covariance matrices, and let tr() denote the trace of a matrix. We say that two Gaussians are c-separated if

$$||\mu_1 - \mu_2|| \geq c\sqrt{\max\{\mathrm{tr}(\Sigma_1), \mathrm{tr}(\Sigma_2)\}}.$$

The motivation in Dasgupta 2000 for choosing $\sqrt{\text{tr}(\Sigma)}$ as the target separation of the means in the above definition is that $\text{tr}(\Sigma)$ is the expected squared Euclidean distance from the mean:

$$E||X - \mu||^2 = \text{tr}(\Sigma). \tag{1}$$

Furthermore, for a Gaussian with bounded eccentricity, where eccentricity was used in Dasgupta 2000 to measure how non-spherical a Gaussian is, the distribution will be concentrated around this *radius* of $\sqrt{\text{tr}(\Sigma)}$. The degree of separation is controlled by parameter c. Single-cell count data restricted to a single cell type typically do not follow a Gaussian distribution Pan et al. 2023b. However, as noted in Dasgupta 2000, random projections of many high-dimensional distributions look more Gaussian due to the central limit theorem. In fact, in the marker gene subspace we observe (Supplemental Fig. 1) that most of the probability mass lies close to distance $\sqrt{\text{tr}(\Sigma)}$ from the mean, rendering Definition 1 suitable as a target separation between cell types.

c-Separated point sets in α-subspace

Let S denote a sample of points in $\mathbb{R}^d$. With a slight abuse of notation, below we do not distinguish between population parameters and (estimated) sample statistics. Analogous to Definition 1 we can compute the total sample variance $\text{tr}(\Sigma)$ of S in the following way:

$$\text{tr}(\Sigma) = \sum_{k=1}^{d} \sum_{x \in S} \frac{(x_k - \mu_k)^2}{|S| - 1}$$

where μ denotes the sample mean of set S, k stands for the k-th component of a vector in $\mathbb{R}^d$ and $|S|$ denotes the cardinality of S. Let S^α denote the set of points induced by a vector $\alpha \in \{0, 1\}^d$, such that

$$S^\alpha = \{x^\alpha \in \mathbb{R}^d : x_k^\alpha = \alpha_k x_k, \forall x \in S\}.$$

Namely, $\alpha_k = 1$ denotes that the k-th coordinate is included, while $\alpha_k = 0$ denotes that the k-th coordinate is ignored. We consider all coordinates for which $\alpha_k = 1$ to define a point in a

α-subspace. Then, the trace of the sample covariance in the α-subspace can be computed as follows:

$$\mathrm{tr}_\alpha(\Sigma) = \sum_{k=1}^{d} \sum_{x \in S} \frac{(\alpha_k x_k - \alpha_k \mu_k)^2}{|S| - 1} = \sum_{k=1}^{d} \alpha_k^2 \sum_{x \in S} \frac{(x_k - \mu_k)^2}{|S| - 1} \tag{2}$$

Now, for two samples of points S_1 and S_2, we are ready to rewrite the inequality from Definition 1 in some arbitrary α-subspace:

$$\sum_{k=1}^{d} \alpha_k^2 (\mu_k^{(1)} - \mu_k^{(2)})^2 \geq c^2 \max\{\mathrm{tr}_\alpha(\Sigma_1), \mathrm{tr}_\alpha(\Sigma_2)\}, \tag{3}$$

where $\mu^{(1)}, \mu^{(2)}$ denote the sample means of S_1 and S_2, respectively, and $\mathrm{tr}_\alpha(\Sigma_1)$ and $\mathrm{tr}_\alpha(\Sigma_2)$ are computed as in (2), for sample covariance matrices Σ_1, Σ_2 of S_1 and S_2, respectively. We call any two point sets S_1 and S_2 in $\mathbb{R}^d$ c-separated in α-subspace if (3) holds.

c-Separation as optimization problem

To identify an α-subspace of fixed dimension m such that $m < d$, where cells of different types achieve c-separation, we formulate a corresponding constrained optimization problem. Specifically, the goal is to minimize a linear objective function that imposes a penalty when the separation of cell types within the α-subspace is not fully realized, subject to a set of nonlinear constraints ensuring distinct separation properties for each cell type. To enable tractable computation, we further linearize these constraints locally around a chosen point, approximating the original problem as an instance of an Integer Linear Program (ILP).

Given input point set $S \subset \mathbb{R}^d$, let $S_i \subseteq S$, $i = 1, \ldots, l$ denote all cells of label i. For every pair of sets S_i and S_j, we aim to separate them according to (3). For an arbitrary value of $c \in \mathbb{R}$, finding an α-subspace such that all constraints (3) are satisfied will in general not be feasible. Therefore, we rephrase (3) by introducing slack variables β as follows:

$$\sum_{k=1}^{d} \alpha_k^2 (\mu_k^{(i)} - \mu_k^{(j)})^2 \geq (c^2 - \beta_{ij}) \max\{\mathrm{tr}_\alpha(\Sigma_i), \mathrm{tr}_\alpha(\Sigma_j)\}.$$

By increasing values of the slack variables β_{ij}, we allow for a relaxation of the original requirement that the two sets S_i and S_j must be c-separated. Let $L = \{\{i, j\} : i = 1, \ldots, l, j = 1, \ldots, l, i \neq j\}$

be the set of all unordered pairs of cell labels. We can now formulate the search for an appropriate α-subspace as the following optimization problem:

$$\begin{aligned}
\text{minimise} \quad & \sum_{(i,j)\in L} \beta_{ij} \\
\text{s.t.} \quad & \sum_{k=1}^{d} \alpha_k^2 (\mu_k^{(i)} - \mu_k^{(j)})^2 \geq (c^2 - \beta_{ij}) \max\{\mathrm{tr}_\alpha(\Sigma_i), \mathrm{tr}_\alpha(\Sigma_j)\} \quad \forall \{i,j\} \in L \\
& \beta_{ij} \geq 0 \quad \forall \{i,j\} \in L \\
& \alpha_k \in \{0,1\} \quad \forall k \in \{1,\ldots,d\} \\
& \sum_{k=1}^{d} \alpha_k = m
\end{aligned} \tag{4}$$

Note that m is the dimension of the subspace induced by the vector α. Furthermore, since α_k can only take the values zero or one, we can substitute α_k^2 with α_k in (4). Constraints (4) are not linear since variables β_{ij} are multiplied with the trace of the covariance matrix of either S_i or S_j that in turn depend on variables α. Problem (4) is NP-hard (see proof in Supplemental Material) and cannot be solved efficiently. In fact, the Gurobi QP solver failed to solve even the smallest instances, Zheng8eq and MC, within a full day of computation. To that end, we will linearize the problem in the next Section.

Linear program (LP) formulation

In this subsection, we aim to relax our model to enable solving it in polynomial time. Initially, we attempt to convert it into a Quadratic Program (QP), but we will later show that even the QP is not guaranteed to be polynomial-time solvable. As a result, we further relax the model into a Linear Program (LP), which can be solved efficiently. Let $\sigma_k^{(i)}$ denote the k-th diagonal element of the covariance matrix Σ_i, which is given by:

$$\sigma_k^{(i)} = \sum_{x \in S_i} \frac{(x_k - \mu_k^{(i)})^2}{|S_i| - 1},$$

where $\mu_k^{(i)}$ is the mean of gene k across the cells in set S_i, and $|S_i|$ represents the size of S_i. For a fixed $\alpha \in \{0,1\}^d$, the trace of the covariance matrix Σ_i is computed as:

$$\text{trace}_\alpha \Sigma_i = \sum_{k=1}^{d} \alpha_k \sigma_k^{(i)}.$$

Relaxation. To simplify the problem, we first relax the binary constraints:

$$\alpha_k \in \{0,1\}$$

to the following box constraints:

$$\alpha_k \in [0,1].$$

This relaxation aims to transform the problem into a QP that can be solved in polynomial time. Next, we attempt to replace the max function present in the constraints.

Max substitution. To replace the constraints involving the max function, we aim to reformulate them in a more computationally tractable form. We begin by observing that the max function can be expressed as:

$$\max_{l \in \{i,j\}} \{\text{trace}_\alpha(\Sigma_l)\} = \max\left\{\sum_{k=1}^{d} \alpha_k \sigma_k^{(i)}, \sum_{k=1}^{d} \alpha_k \sigma_k^{(j)}\right\}.$$

We can distribute the multiplication inside the maximum expression:

$$(c^2 - \beta_{ij}) \max\left\{\sum_{k=1}^{d} \alpha_k \sigma_k^{(i)}, \sum_{k=1}^{d} \alpha_k \sigma_k^{(j)}\right\} = \max\left\{(c^2 - \beta_{ij}) \sum_{k=1}^{d} \alpha_k \sigma_k^{(i)}, (c^2 - \beta_{ij}) \sum_{k=1}^{d} \alpha_k \sigma_k^{(j)}\right\}.$$

To resolve the maximum, we impose that the left-hand side must be greater than or equal to both elements in the maximum expression. This results in the following two constraints:

$$\sum_{k=1}^{d} \alpha_k (\mu_k^{(i)} - \mu_k^{(j)})^2 \geq (c^2 - \beta_{ij}) \sum_{k=1}^{d} \alpha_k \sigma_k^{(i)},$$
$$\sum_{k=1}^{d} \alpha_k (\mu_k^{(i)} - \mu_k^{(j)})^2 \geq (c^2 - \beta_{ij}) \sum_{k=1}^{d} \alpha_k \sigma_k^{(j)}.$$

At this stage, we have a QP. However, these constraints are not convex, meaning the problem is not guaranteed to be polynomial-time solvable. We demonstrate this non-convexity with a simple example.

Example of non-convexity. Consider the following example with three genes and two Gaussian distributions:

$$\mu_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

$$\sigma_1 = \sigma_2 = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

and

$$c = 1, \quad k = 2.$$

Substituting these values into the equations results in the following QP:

$$\begin{aligned} \text{minimize} \quad & \beta_{12}, \\ \text{s.t.} \quad & \alpha_1 + \alpha_2 + \alpha_1 \geq (1 - \beta_{12})\,(3\alpha_1 + \alpha_2 + 2\alpha_3)\,, \\ & \alpha_1 + \alpha_2 + \alpha_1 \geq (1 - \beta_{12})\,(3\alpha_1 + \alpha_2 + 2\alpha_3)\,, \\ & \beta_{12} \geq 0, \quad \alpha_k \in [0, 1], \\ & \alpha_1 + \alpha_2 + \alpha_3 = 2. \end{aligned}$$

In this case, the points

$$\alpha_1 = 1,\ \alpha_2 = 0,\ \alpha_3 = 1,\ \beta_{12} = 0.5,$$

and

$$\alpha_1 = 1,\ \alpha_2 = 1,\ \alpha_3 = 0,\ \beta_{12} = 0.5$$

are feasible. However, their convex combination

$$\alpha_1 = 1,\ \alpha_2 = 0.5,\ \alpha_3 = 0.5,\ \beta_{12} = 0.5$$

is not feasible, as it violates the first constraint ($2 \not\geq 2.5$). This shows that the problem is non-convex, and therefore is not guaranteed to be solvable in polynomial time.

Linearization. To make the problem solvable, we linearize the term involving $\beta_{ij}\alpha_k$ using Taylor expansion. Recall the approximation:

$$\beta_{ij}\alpha_k\sigma_k^{(l)} \approx \frac{1}{2}\beta_{ij}\sigma_k^{(l)} + \frac{c^2}{2}\alpha_k\sigma_k^{(l)} - \frac{c^2}{4}\sigma_k^{(l)}.$$

Using this linearization, we can rewrite the original inequality constraints as follows:

$$\sum_{k=1}^{d}\alpha_k(\mu_k^{(i)} - \mu_k^{(j)})^2 \geq \sum_{k=1}^{d}\frac{c^2}{2}\alpha_k\sigma_k^{(i)} - \frac{1}{2}\beta_{ij}\sigma_k^{(i)} + \frac{c^2}{4}\sigma_k^{(i)},$$
$$\sum_{k=1}^{d}\alpha_k(\mu_k^{(i)} - \mu_k^{(j)})^2 \geq \sum_{k=1}^{d}\frac{c^2}{2}\alpha_k\sigma_k^{(j)} - \frac{1}{2}\beta_{ij}\sigma_k^{(j)} + \frac{c^2}{4}\sigma_k^{(j)}.$$

LP formulation. Finally, we can formulate the Linear Program (LP) as follows. The objective is to minimize the sum of slack variables β_{ij}, subject to the following constraints:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in L}\beta_{ij}, \\
\text{s.t.} \quad & \sum_{k=1}^{d}\alpha_k(\mu_k^{(i)} - \mu_k^{(j)})^2 \geq \sum_{k=1}^{d}\frac{c^2}{2}\alpha_k\sigma_k^{(i)} - \frac{1}{2}\beta_{ij}\sigma_k^{(l)} + \frac{c^2}{4}\sigma_k^{(i)}, \quad \forall(i,j)\in L, \\
& \sum_{k=1}^{d}\alpha_k(\mu_k^{(i)} - \mu_k^{(j)})^2 \geq \sum_{k=1}^{d}\frac{c^2}{2}\alpha_k\sigma_k^{(j)} - \frac{1}{2}\beta_{ij}\sigma_k^{(l)} + \frac{c^2}{4}\sigma_k^{(j)}, \quad \forall(i,j)\in L, \\
& \beta_{ij} \geq 0, \quad \forall(i,j)\in L, \\
& \alpha_k \in [0,1], \quad \forall k \in \{1,\ldots,d\}, \\
& \sum_{k=1}^{d}\alpha_k = m.
\end{aligned}$$

Benchmarks

Methods

We implemented the ILP described in the previous Section in method SepSolve. We used Gurobi (Gurobi Optimization, LLC 2024) to solve its linear relaxation and convert fractional α to integral

values through a simple ranking method which rounds the m largest values to 1 and sets the remaining α-variables to zero. The same rounding scheme was applied in Dumitrascu et al. 2021. When comparing with embedded method Spapros and with scGeneFit's hyperparameter optimization in Supplemental Fig. 4, we tuned the parameter c by interacting with a classifier. Specifically, we split the data into 40% training, 30% validation, and 30% test sets. For each candidate c between 0.1 and 1, marker genes were selected and a logistic regression model was trained on the training set and evaluated by F1 score on the validation set. The best c was then re-applied for marker gene selection and model training on train+validation, and performance was reported on the held-out test set.

Most methods, including SepSolve, were run using default values for all parameters. In SepSolve we set the default value for the separation constant c to 0.4. The authors of G-PC did not provide a default value for the coverage factor (K) in their implementation. Consistent with Hasanaj et al. 2022, we therefore varied K in the range of $\{5, \ldots, 20\}$ and chose the value that yielded the highest average accuracy on the IPF dataset. The same value was then applied to all datasets.

When running mRMR, we performed a binary search over the parameter lambda to find the value that yields the desired number of marker genes. If no such value exists, we selected the value that produces the largest set of marker genes smaller than the target number of genes. Spapros was run in the cell type recovery mode (SpaprosCTo). DE genes were obtained using the rank_genes_groups function from the SCANPY package with the default t-test method by sequentially selecting the highest-ranked gene for each cell type until the desired number of marker genes was obtained. Methods were downloaded from the repositories listed in Table 2.

Method	Source	GitHub/GitLab
scGeneFit	Dumitrascu et al. 2021	`https://github.com/solevillar/scGeneFit-python`
G-PC	Hasanaj et al. 2022	`https://github.com/euxhenh/phenotype-cover`
mRMR	Peng et al. 2005	`https://github.com/smazzanti/mrmr`
RankCorr	Vargo and Gilbert 2020	`https://github.com/ahsv/RankCorr`
Spapros	Kuemmerle et al. 2024	`https://github.com/theislab/spapros`
SMaSH	Nelson et al. 2022	`https://gitlab.com/cvejic-group/smash`
Mutual Inf.	Pedregosa et al. 2011	`https://github.com/scikit-learn/scikit-learn`
DE genes	Wolf et al. 2018	`https://github.com/scverse/scanpy`

Table 2: Methods used in this study.

Data processing

Following the same preprocessing steps as applied in Hasanaj et al. 2022 for assessing classification performance, we removed cell types with less than 50 cells and genes expressed in less than 10 cells. We omitted the step of cell type removal when assessing the stability of marker selection. Counts were normalized to 10,000 using the SCANPY Python package and log transformed after adding a pseudocount of 1. Finally, we identified highly variable genes using the highly_variable_genes function from SCANPY, applying the same parameters as those used in Hasanaj et al. 2022 for all datasets. Following the authors' recommendation, we have additionally standardized the data to zero mean and unit variance when running G-PC. When testing SepSolve's marker genes across the two lung datasets (Fig. 4A), we considered only the cell types and genes common to both datasets.

Software Availability

The SepSolve software is available at GitHub (`https://github.com/bborozan/SepSolve`) and as Supplemental Code.

Competing interest statement

The authors declare no competing interests.

Acknowledgements

We are grateful to the members of the Canzar group for their valuable feedback and to the anonymous reviewers for their constructive comments on earlier versions of this manuscript.

Author contributions: All authors conceived the algorithm. B.B. performed the computational experiments. All authors wrote the manuscript. D.M. and S.C. supervised the study.

References

Adams TS et al. 2020. Single-cell RNA-seq reveals ectopic and aberrant lung-resident cell populations in idiopathic pulmonary fibrosis. *Science Advances*. **6**: eaba1983.

Chen S et al. 2022. hECA: The cell-centric assembly of a cell atlas. *iScience*. **25**: 104318.

Dasgupta S 2000. Experiments with Random Projection. In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. UAI '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 143–151.

Dumitrascu B, Villar S, Mixon DG, and Engelhardt BE. 2021. Optimal marker gene selection for cell type discrimination in single cell analyses. *Nature Communications*. **12**: 1186.

Duò A, Robinson MD, and Soneson C. 2018. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*. **7**: 1141.

Gurobi Optimization, LLC 2024. Gurobi Optimizer Reference Manual.

Hao Y et al. 2021. Integrated analysis of multimodal single-cell data. *Cell*. **184**: 3573–3587.e29.

Hasanaj E, Alavi A, Gupta A, Póczos B, and Bar-Joseph Z. 2022. Multiset multicover methods for discriminative marker selection. *Cell Reports Methods*. **2**: 100332.

He S, Wang LH, Liu Y, Li YQ, Chen HT, Xu JH, Peng W, Lin GW, Wei PP, Li B, et al. 2020a. Single-cell transcriptome profiling of an adult human cell atlas of 15 major organs. *Genome biology*. **21**: 1–34.

He S et al. 2020b. Single-cell transcriptome profiling of an adult human cell atlas of 15 major organs. *Genome Biology*. **21**: 294.

Kononenko I 1994. Estimating attributes: Analysis and extensions of RELIEF. In: *Machine Learning: ECML-94*. Ed. by F Bergadano and L De Raedt. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 171–182.

Kraskov A, Stögbauer H, and Grassberger P. 2004. Estimating mutual information. *Physical Review E*. **69**: 066138.

Kuemmerle LB et al. 2024. Probe set selection for targeted spatial transcriptomics. *Nature Methods*. **21**: 2260–2270.

Langlieb J et al. 2023. The molecular cytoarchitecture of the adult mouse brain. *Nature*. **624**: 333–342.

Lun A, McCarthy D, and Marioni J. 2016. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor [version 2; peer review: 3 approved, 2 approved with reservations]. *F1000Research*. **5**:

Madissoon E et al. 2019. scRNA-seq assessment of the human lung, spleen, and esophagus tissue stability after cold preservation. *Genome Biology*. **21**: 1.

Madissoon E et al. 2023. A spatially resolved atlas of the human lung characterizes a gland-associated immune niche. *Nature Genetics*. **55**: 66–77.

Nelson ME, Riva SG, and Cvejic A. 2022. SMaSH: a scalable, general marker gene identification framework for single-cell RNA-sequencing. *BMC Bioinformatics*. **23**: 328.

Pan Y, Justin TL, Moorad R, Wu D, Marron JS, and Dittmer DP. 2023a. The Poisson distribution model fits UMI-based single-cell RNA-sequencing data. *BMC Bioinform.* **24**: 256.

Pan Y, Landis JT, Moorad R, Wu D, Marron JS, and Dittmer DP. 2023b. The Poisson distribution model fits UMI-based single-cell RNA-sequencing data. *BMC Bioinformatics*. **24**: 256.

Pedregosa F et al. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. **12**: 2825–2830.

Peng H, Long F, and Ding C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **27**: 1226–1238.

Popescu DM et al. 2019. Decoding human fetal liver haematopoiesis. *Nature*. **574**: 365–371.

Pullin JM and McCarthy DJ. 2024. A comparison of marker gene selection methods for single-cell RNA sequencing data. *Genome Biology*. **25**: 56.

Quinlan JR. 1986. Induction of decision trees. *Machine Learning*. **1**: 81–106.

Simmons DP, Wearsch PA, Canaday DH, Meyerson HJ, Liu YC, Wang Y, Boom WH, and Harding CV. 2012. Type I IFN Drives a Distinctive Dendritic Cell Maturation Phenotype That Allows Continued Class II MHC Synthesis and Antigen Processing. *The Journal of Immunology*. **188**: 3116–3126.

Sountoulidis A, Liontos A, Nguyen HP, Firsova AB, Fysikopoulos A, Qian X, Seeger W, Sundström E, Nilsson M, and Samakovlis C. 2020. SCRINSHOT enables spatial mapping of cell states in tissue sections with single-cell resolution. *PLOS Biology*. **18**: 1–32.

Vargo AHS and Gilbert AC. 2020. A rank-based marker selection method for high throughput scRNA-seq data. *BMC Bioinformatics*. **21**: 477.

Wolf FA, Angerer P, and Theis FJ. 2018. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*. **19**: 15.

Zeisel A et al. 2015. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*. **347**: 1138–1142.