

MULTIVARIATE RARE CELL MINING AND BIOMARKER IDENTIFICATION FROM SINGLE-CELL DATA

Alexander Gerniers

*Thesis submitted in partial fulfillment of the requirements for
the Degree of Doctor in Engineering Sciences and Technology*

October 2024

ICTEAM/INGI

Louvain School of Engineering

Université catholique de Louvain

Louvain-la-Neuve

Belgium

Thesis Committee:

Prof. Pierre Dupont, *supervisor*

Prof. Siegfried Nijssen, *supervisor*

Prof. Charles Pecheur, *chairperson*

Prof. Pierre Coulie, *secretary*

Prof. Yvan Saeys

Prof. Mark Robinson

UCLouvain/ICTEAM, Belgium

UCLouvain/ICTEAM, Belgium

UCLouvain/ICTEAM, Belgium

UCLouvain/DDUV, Belgium

Universiteit Gent/VIB, Belgium

Universität Zürich/SIB, Switzerland

Multivariate rare cell mining and biomarker identification from single-cell data

by Alexander Gerniers

© Alexander Gerniers 2024

ICTEAM

UCLouvain

Place Sainte-Barbe, 2

1348 Louvain-la-Neuve

Belgium

Abstract

Identifying rare cell types is an important task to capture the heterogeneity of single-cell data. Indeed, many cell types are present in very small amounts, and are consequently easily missed by classical clustering approaches. As they may play a crucial role in some pathologies (such as cancer), their accurate identification is critical and has prompted the development of clustering methods specifically targeted to spot rare cells. These approaches are often restricted to be univariate due to computational restrictions and scalability concerns. In contrast, this thesis investigates the use of multivariate criteria, specifically, a global sum criterion which is robust with respect to the high technical and biological variability of scRNA-seq data. It jointly identifies rare cells and a corresponding set of genes characterizing them, which proves effective at identifying biologically relevant rare subpopulations of cells.

Different variations of this approach have also been developed. A hybrid procedure allows the use of this multivariate criterion in large-scale single-cell data in a computationally efficient way. The global sum criterion is also robust with respect to the batch effects that appear when combining single-cell data from multiple biological samples (such as multiple donors). Indeed, one generally needs to computationally integrate the different samples into one coherent dataset before applying clustering methods (which thus ignores the initial partition into multiple samples). As an alternative, this thesis shows how the sum criterion can be adapted to explicitly search for shared rare expression patterns across samples, without the need for prior data integration.

Acknowledgments

I would like to express my gratitude to a lot of people who have, either directly or indirectly, contributed to the realization of this thesis.

First of all, I would like to thank my supervisors, Pierre Dupont and Siegfried Nijssen. Pierre has guided me since day one, providing many insights and challenging my work (often encouraging me to take a step back; or in the words of the cycling enthusiast that Pierre is: “*lever le nez du guidon*”). His eagle-eyed attention to details has without any doubt raised the quality of my work. I’m also grateful to Siegfried, who accepted to join in the middle of this thesis. His expertise, complementary to that of Pierre, has provided me with different perspectives and enriched my work.

I also thank Pierre Coulie, Nicolas van Baren, and Orian Bricard from the de Duve institute, which have familiarized me with the biological context of this thesis. The frequent visits I paid them during the first year were very instructive and laid the foundations of this work. I further thank Vincent Branders and Victor Hamer for the many research conversations we had sharing an office. I’m also grateful for the insightful comments and suggestions given by the members of my thesis committee.

I have had the privilege to work in the best working environment imaginable. I thank my fellow researchers and the staff of the INGI department for all the wonderful moments we had, from the many belote games¹ during the lunch breaks all the way to going to concerts and festivals (some were even brave enough to become housemates!). I have also had the pleasure to interact with many students during the practical sessions or projects I gave for several courses. These teaching duties have always been an enriching experience for me (and I hope for the students as well!).

¹I recall in particular this memorable game, which is a true testament to never giving up no matter how bleak the situation appears!

Nous	2	/	18	/	/	28	44	/	56	60	77	93	118→
Eux	14	39	/	45	61	67	/	83	87	99	100	/	/

Note: the goal is to be the first team to reach 101 points.

My last (but far from least) words of thanks go to my family and friends. They have provided support during all these years, been there for the good moments and (knowingly or not) helped me through more difficult ones. The present manuscript would not exist if not for them.

Contents

Abstract	i
Acknowledgements	iii
Table of contents	v
1 Introduction	1
1.1 Background	1
1.2 Contributions and outline of the thesis	3
1.3 Bibliographic notes	7
2 Single-cell data and the search for rare cells	9
2.1 Overview of single-cell technologies	9
2.1.1 Single-cell imaging	9
2.1.2 Flow and mass cytometry	10
2.1.3 Single-cell omics	11
2.2 Single-cell transcriptomics	12
2.2.1 Sequencing workflow	12
2.2.2 Bias sources	14
2.2.3 The typical scRNA-seq pipeline	16
2.2.4 Batch effects	18
2.3 Rare cell identification	24
2.3.1 Motivation	24
2.3.2 Clustering based approaches	27
2.3.3 Rareness scores	33
2.3.4 Deep learning clustering	36
2.3.5 A new approach	41
3 A global sum criterion	43
3.1 The maximal sum submatrix problem	43
3.1.1 Definition	43

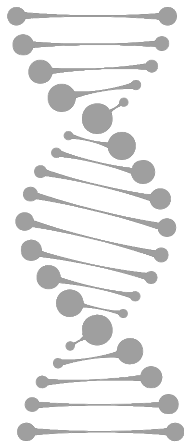
3.1.2	CP Global Constraint	45
3.1.3	Comparison to traditional biclustering	50
3.2	Application to scRNA-seq data	52
3.2.1	Data scaling	52
3.2.2	Robustness w.r.t. scRNA-seq variability	53
4	Mining rare and highly specific subpopulations	57
4.1	MicroCellClust definition	57
4.1.1	Objective function	58
4.1.2	Constraints	59
4.1.3	Identifying multiple biclusters	61
4.2	Comparison to alternative methods	63
4.2.1	Identification of rare GARP+ Tregs among CD8 T cells . .	64
4.2.2	Identification of rare Jurkat cells among 293T cells	70
4.2.3	Rareness versus distinctiveness	71
4.3	Case studies on scRNA-seq data	75
4.3.1	Identification of a T regulatory related subpopulation within CD4 T cells	75
4.3.2	Identification of a cell-cycle related subpopulation in tu- mor GARP+ Tregs	79
4.3.3	Identification of rare subpopulations in mouse embryonic stem cells	82
4.4	Discussion	86
5	Efficient solving of the optimization problem	91
5.1	Implicit search space	91
5.1.1	Constraint on negative values	92
5.1.2	Objective value computation	94
5.2	Solving small scale instances	95
5.2.1	Branch and bound approach	95
5.2.2	Beam search	97
5.2.3	Execution time and memory analysis	104
5.3	Discussion	107
6	Multivariate rare cell mining in large-scale single-cell data	109
6.1	MicroCellClust 2: a hybrid approach	110
6.2	Case studies on large-scale scRNA-seq data	116
6.2.1	Identification of rare megakaryocytes within 68k PBMCs .	116

6.2.2	Identification of rare ependymal cells within 160k mouse brain cells	118
6.2.3	Scalability	121
6.3	Additional features	122
6.3.1	Automated parameter tuning	122
6.3.2	Generalization of the heuristics	124
6.4	Discussion	127
7	Searching for rare cells across multiple samples	129
7.1	Motivation	129
7.2	scCross definition	133
7.3	Results on scRNA-seq data	135
7.3.1	Identification of cell-cycle genes across GARP+ Treg samples	136
7.3.2	Identification of rare Jurkat cells with artificial batch effects	138
7.3.3	Identification of a cilium subpopulation across different donors	141
7.3.4	Identification of subpopulations in pancreas data from two studies	148
7.3.5	Identification of rare subpopulations across different mouse organs	148
7.4	Multiple beam solver	161
7.5	Discussion	164
8	Perspectives for multi-modal analysis	169
8.1	General framework	169
8.2	Combining transcriptomics and chromatin accessibility	171
8.2.1	Single-cell epigenomics	171
8.2.2	Enhancer co-activity	173
8.2.3	Applying MicroCellClust on 10X Multiome	174
8.3	Discussion	177
9	Conclusions and perspectives	179
	Bibliography	185

Appendix	199
A Data description and availability	199
A.1 Breast cancer data	199
A.2 Jurkat and 293T mixture	200
A.3 Artificial data for rareness versus distinctiveness	200
A.4 Mouse embryonic stem cells	201
A.5 68k PBMCs	201
A.6 160k mouse brain cells	201
A.7 Non-small cell lung cancer dissociated tumor cells	201
A.8 8 pancreas samples	202
A.9 Tabula muris	202
A.10 10X Multiome PBMCs	202

Introduction

| 1



This thesis relates to the task of identifying biomarkers from medical data, i.e. finding factors of interests that play a potential role in the pathology under study. In particular, we here look for interesting genes expressed by a group of cells, so as to understand the function of these cells. Identifying rare cells, i.e. cells types that are only present in tiny amounts, is an important challenge in this context in order to accurately understand the pathology under study.

Section 1.1 lays the background of this thesis. Section 1.2 sketches the contributions of this thesis to the task of identifying rare cells in single-cell data, as well as the outline of the dissertation.

1.1 Background

During the last decades, cancer research has been one of the most important fields in medical sciences. This has lead to the development of many different kind of therapies, one of which is *immunotherapy*. It consists in activating the immune system of the patient to eliminate cancer cells ([Mellman et al., 2011](#)). This approach has, for instance, been particularly effective in the treatment of melanoma ([Franklin et al., 2017](#)), with several such treatments proving to be a valuable alternative to more aggressive treatments (e.g. chemotherapy). For other kind of cancers, such as breast cancer, such treatments are still under development. Such research relies on the identification of biomarkers, i.e. biological factors potentially linked to the pathology under study. For instance, monitoring which proteins are

synthesized within cells involved in the immune response against a tumor could lead the way to new treatments.

This can be performed by measuring *gene expression*, i.e. the process that uses the information contained in a gene to synthesize a protein. It starts with transcription, during which a messenger RNA molecule corresponding to a specific gene is synthesized. Measuring the amount of RNA transcripts present in a biological sample thus indicates which genes are expressed (and in what amounts). This is known as *transcriptomics*.

Originally, researchers relied on so called *bulk* methods, such as DNA microarrays and RNA-Seq (Wang et al., 2009), which analyze the transcriptome of a group of cells. These technologies therefore yield the average gene expression values of a cell population as a whole, thus plainly ignoring the cell type heterogeneity that can be present within such populations (Kolodziejczyk et al., 2015). The development of *single-cell* technologies has enabled the analysis of cells separately, providing a richer source of information. For instance, single-cell RNA sequencing (scRNA-seq) measures the RNA transcripts within each cell of a population. Rather than having one vector of average gene expression for each biological sample, one now has a matrix describing which genes are expressed in which cells, allowing the identification of different subpopulations.

From a computational point of view, single-cell data somewhat avoid the *curse of dimensionality*, whereby a few data points (typically bulk samples) are scattered inside a high dimensional space (corresponding to tens of thousands of genes) leading to empty spaces (Bellman, 1957; Helleputte, 2010). With single-cell data, aggregating a few biological samples can lead to huge matrices with hundreds of thousands of cells, exceeding the number of dimensions, which benefits the learning process. Yet, this gold mine of information comes at a cost. Since each cell is sequenced individually, the small amount of available material makes it difficult to accurately detect all RNA transcripts, which can lead to lots of technical noise. This requires dedicated methods to accurately cluster the data to identify the different cell types that are present (Todorov and Saeys, 2019).

An important area of focus in single-cell analysis is the identification of *rare cell types*. Indeed, the amount of cells from each type can vary significantly within a cell population, from abundant subpopulations to tiny ones represented by only a few cells. Typical clustering algorithms, which tend to group cells into large clusters, often miss these rare cell types. Yet, they have proven to be crucial to the understanding of certain pathologies. Notable examples include the identification of cancer stem cells, whose self-renewing abilities have been shown to

drive hematopoietic, brain, and breast cancers (Jordan et al., 2006), and circulating tumor cells, which circulate in the bloodstream, making their detection challenging (Yu et al., 2011).

Dedicated methods for rare cell¹ identification have been able to highlight rare cell types in scRNA-seq data based on their gene expression (Grün et al., 2015; Jiang et al., 2016; Jindal et al., 2018). This allows the identification of *differentially expressed* genes (i.e. genes whose expression within these cells significantly differs from the other cells in the population), which characterize the function of these rare cells and their potential role in the immune response to a pathology. Yet, these methods often rely on univariate analysis, considering every gene independently from the others. While such a computationally tractable approach has proven effective to uncover new rare cell types, it plainly ignores situations where genes interact with each other. Considering multivariate approaches could enable the identification of new patterns, albeit at a higher computational cost.

1.2 Contributions and outline of the thesis

This thesis presents a novel multivariate approach for rare cell identification. It jointly identifies a subset of cells and a subset of genes representative of a rare subpopulation, based on a global sum criterion which is well suited to the high variability of single-cell data. Moreover, the joint clustering of cells and genes, which can be considered a kind of *biclustering*, departs from traditional methods which generally perform cell clustering and gene marker identification in separate steps. The main contributions of this thesis are:

- *The proposal of an optimization problem based on a global sum criterion for joint rare cells and marker genes identification.*

We propose MicroCellClust, an optimization problem based on a global sum criterion, whereby expression values are not compared in a pairwise fashion. Rather, the quality of a cell/gene bicluster is evaluated as a whole, based on the sum of the corresponding expression values. This approach is especially sensible with respect to the high variability of scRNA-seq data, and proves particularly efficient at identifying rare subpopulations in situations where they are only represented by a handful of cells. This is typically the case for single-cell data that is of limited availability, which complexifies the detection of rare cells. The proposed approach is evaluated on

¹Throughout this thesis, the term “rare cells” will be used to refer to rare cell types or rare expression patterns.

real single-cell data, and the results are biologically interpreted to prove its ability to retrieve biologically coherent results.

- *The proposal of a hybrid procedure to perform multivariate rare cell mining in large-scale data.*

Considering such a multivariate approach comes at the cost of solving a NP-hard optimization problem. Dedicated solvers are proposed to find good quality solutions in a reasonable time, such that they can be used by medical domain experts without requiring days of computation on supercomputers. In particular, a hybrid procedure combining MicroCellClust and a rareness metric, which efficiently scales to such data, is proposed to identify biclusters in data containing several hundreds of thousands of cells. This enables a multivariate mining of rare cells in large data generated for general cell type characterization, which contrasts with alternative methods (Chen et al., 2019; Jindal et al., 2018) designed to scale to such large data.

- *The proposal of an optimization problem for the identification of rare subpopulations present across multiple single-cell samples.*

With the widespread availability of scRNA-seq datasets, it is now common to integrate multiple biological samples (known as “horizontal integration”), for instance corresponding to multiple donors in a cancer study. We propose scCross, which adapts the optimization problem to find rare subpopulations that are common to all (or most) donors, rather than donor-specific subpopulations. scCross proves robust with respect to the batch effects that are present in such aggregated data, and consequently doesn’t require computational models to integrate the samples prior to their analysis. It therefore efficiently identifies rare gene expression patterns shared by different donors. This explicit search across samples contrasts with the traditional two-step approach of first integrating datasets, i.e. transforming the data into one sample, and then using traditional clustering algorithms (thus ignoring the partition into multiple samples).

- *The application of the proposed approaches on multi-omics data.*

Vertical integration consists in measuring several modalities (typically transcriptomics and another omic modality) in the same cells. The proposed optimization problem can thus be used to find links between biomarkers from both modalities in a completely unsupervised and multivariate way. In particular, new technologies are able to sequence the transcriptome and

chromatin accessibility in the same single-cells, allowing to study the link between enhancers and gene expression. Most studies only analyze the link between one gene and one enhancer. Only one study (Ziyani et al., 2023) used this data to investigate the role of shadow enhancers, i.e. multiple enhancers being involved in the expression of one gene. The multivariate optimization problem of MicroCellClust (or an adaptation thereof) can be of interest to link groups of enhancers and of genes together.

This thesis is organized as follows.

CHAPTER 2, [SINGLE-CELL DATA AND THE SEARCH FOR RARE CELLS](#), sets the background of the field of single-cell data. It starts with an overview of the different types of single-cell data. Then, it focuses specifically on single-cell transcriptomics, which is the main source of data used in this thesis, and highlights the challenges when analyzing such data. Finally, we focus on the task of identifying rare cells in scRNA-seq data. Different existing approaches are presented, and their advantages and limitations are discussed, motivating a new multivariate approach.

CHAPTER 3, [A GLOBAL SUM CRITERION](#), discusses the global sum criterion, which constitutes the common thread between the approaches proposed in this thesis. We state the problem of identifying the submatrix of maximal sum, as presented by Branders (2021), which has proven more successful at identifying gene-specific subgroups in bulk transcriptomics data than typical alternative bi-clustering methods. The advantages of the global sum criterion to deal with the high variability of scRNA-seq are discussed.

CHAPTER 4, [MINING RARE AND HIGHLY SPECIFIC SUBPOPULATIONS](#), presents MicroCellClust, which is based on the global sum criterion. It refines the objective function to be optimized and adds useful constraints to specifically search for rare and highly specific patterns of expressions within small subpopulations of cells. The proposed approach is compared to typical alternative methods using a controlled setting where rare cell types are artificially planted among an abundant type. The results show MicroCellClust achieves a higher F_1 score, specifically in cases where the rare cell type is only represented by a few individuals. Finally, several case studies show its ability to retrieve biologically coherent rare subpopulations from different single-cell populations.

CHAPTER 5, [EFFICIENT SOLVING OF THE OPTIMIZATION PROBLEM](#), details the solvers used to efficiently find the optimal solution to the MicroCellClust problem, which is NP-hard. Namely, we propose a beam search solver as this approach takes advantage of the expected similarity between all cells composing the result.

This allows to efficiently prune the search space, and produce a high quality solution with a quadratic time complexity in terms of the number of cells, well suited to the small and middle-scale data studied in chapter 4.

CHAPTER 6, [MULTIVARIATE RARE CELL MINING IN LARGE-SCALE SINGLE-CELL DATA](#), presents MicroCellClust 2, a hybrid approach based on a precomputed rareness score to efficiently solve the optimization problem for large-scale data. The use of appropriate heuristics reduces the complexity to a linear one, enabling the analysis of data containing hundreds of thousands of cells. This allows such data, typically generated to perform a general characterization of cell types within a heterogeneous tissue, to be analyzed in a multivariate way to search for rare cell types.

CHAPTER 7, [SEARCHING FOR RARE CELLS ACROSS MULTIPLE SAMPLES](#), presents scCross, a method to jointly identify rare subpopulations of cells and corresponding marker genes across multiple scRNA-seq samples. The objective function is refined to search for rare cell subpopulations present in all or most samples, ignoring sample-specific expression patterns. We show scCross is robust with respect to batch effects between samples as it identifies subpopulations of interest within unintegrated data. A modified solver, that takes advantage of the sample origin of each cell to diversify the search, is presented.

CHAPTER 8, [PERSPECTIVES FOR MULTI-MODAL ANALYSIS](#), presents a general framework to use the optimization problem of MicroCellClust (or scCross) on single-cell data where multiple modalities have been sequenced in the same single-cells. This chapter further presents a concrete application on data combining transcriptomics (scRNA-seq) and epigenomics (scATAC-seq). This approach jointly identifies highly expressed genes and open regions of chromatin for a specific subpopulation, allowing to link gene expression and enhancer activity.

CHAPTER 9, [CONCLUSIONS AND PERSPECTIVES](#), summarizes the main results and conclusions of this thesis, and gives some perspectives for further work relating to the multivariate identification of rare cells in single-cell gene expression data.

1.3 Bibliographic notes

Journal articles

- Gerniers, A., Bricard, O., and Dupont, P. (2021). MicroCellClust: mining rare and highly specific subpopulations from single-cell expression data. *Bioinformatics*, 37(19):3220–3227.
<https://doi.org/10.1093/bioinformatics/btab239>
- Gerniers, A., Nijssen, S., and Dupont, P. (2024). scCross: efficient search for rare subpopulations across multiple single-cell samples. *Bioinformatics*, 40(6):btae371.
<https://doi.org/10.1093/bioinformatics/btae371>

Conference paper

- Gerniers, A., and Dupont, P. (2022). MicroCellClust 2: a hybrid approach for multivariate rare cell mining in large-scale single-cell data. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 148–153.
<https://doi.org/10.1109/BIBM55620.2022.9995176>

Miscellaneous contributions

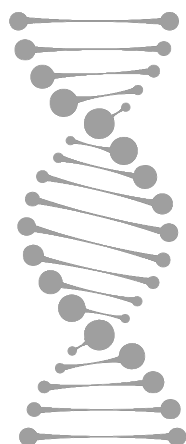
- Gerniers, A. MicroCellClust 1 & 2 implementations in R and Scala:
<https://github.com/agerniers/MicroCellClust/>
- Gerniers, A. scCross implementation in R and Scala:
<https://github.com/agerniers/scCross/>

Master thesis supervision

- Dubuisson, E. (2024). *Rareness quantification of groups of cells and application to large-scale single-cell (bi)clustering*. Master’s thesis, École polytechnique de Louvain, UCLouvain.
<https://hdl.handle.net/2078.1/thesis:45922>

Single-cell data and the search for rare cells

2



This chapter provides a broad overview of single-cell data, with a specific focus on single-cell transcriptomics and the identification of rare cell types.

Section 2.1 shows an overview of different types of single-cell data. Section 2.2 focuses specifically on scRNA-seq data and its challenges, namely the technical and biological variance that arises as well as the batch effects that appear between several biological samples. Section 2.3 focuses on the task of rare cell identification in scRNA-seq data, by showing a small case study and describing the different approaches that have been proposed in the literature to tackle this specific task.

2.1 Overview of single-cell technologies

The term single-cell encapsulates different technologies, which can be classified in three categories: single-cell imaging, flow or mass cytometry, and next-generation omics sequencing (Todorov and Saeys, 2019). This section gives a short overview of each category.

2.1.1 Single-cell imaging

The first approach for single-cell analysis consists in microscopy-based techniques. Initially performed manually, automated image acquisition and segmentation have now made it possible to conduct high-throughput screenings, allow-

ing the analysis of hundreds of thousands of cells in single-well plates ([Abraham et al., 2004](#)). Many other types of microscopy, including various forms of light and electron microscopy, also allow the extraction of information at the single-cell level, albeit with generally lower throughput. Unlike most other single-cell approaches where cells are dissociated in suspension, image-based single-cell profiling has the advantage of providing spatial information in two or three dimensions, which can be crucial for uncovering new biological insights ([Todorov and Saeys, 2019](#)).

2.1.2 Flow and mass cytometry

Flow cytometry measures multiple physical characteristics of single-cells that are suspended in a fluid streaming through a measuring device ([Adan et al., 2017](#)). Typically, cells are labeled with fluorescent markers, which bind to user defined targets (for instance surface markers, intracellular molecules, etc.). These cells then flow one at a time through a laser beam, and detectors capture the emitted fluorescence to measure which fluorescent marker (and thus corresponding biomarker) is present.

Mass cytometry broadly uses the same principle, but replaces the fluorescent markers by heavy metal isotopes. Cells are then analyzed by mass spectrometry to identify the different markers ([Bandura et al., 2009](#)). This technique avoids the spectral overlap which can occur with flow cytometry, increasing the number of markers that can be simultaneously detected. Imaging mass cytometry ([Giesen et al., 2014](#)) performs mass spectrometry on tissue slices, meaning each cell is analyzed on the spot rather than streaming through a detector, providing additional spatial information.

The main advantage of these methods is they have a very high throughput (flow cytometry easily processes millions of cells), but the counterpart is only a few tens of user-selected features are quantified. These technologies don't aim to identify new biomarkers, but rather identify well-known cell types. *Fluorescence-Activated Cell Sorting* (FACS) ([Bonner et al., 1972](#)) takes this a step further by combining flow cytometry with a cell sorter, which physically separates the cells based on the fluorescent markers. This can be useful to isolate a particular cell type, which can then be further analyzed using scRNA-seq, for instance.

2.1.3 Single-cell omics

Single-cell omics technologies have been revolutionized by advances in *next-generation sequencing* (NGS), which enables the rapid sequencing of DNA and RNA from individual cells with high accuracy and depth. These approaches offer high resolution as they sequence the entirety of a cell's content rather than only user-specified targets, as in cytometry. A single experiment can measure tens of thousands of parameters (e.g. RNA transcripts in the case of scRNA-seq) in up to several tens (or even hundreds) of thousands of cells. This allows in-depth study of the heterogeneity of cell populations, identification of rare cell types, and studying the dynamics of cellular systems (Todorov and Saeys, 2019).

One of the most widely used omics technology is single-cell *transcriptomics* (scRNA-seq), which involves sequencing the RNA (or transcriptome) of individual cells. It provides a snapshot of gene expression within a cell at a given moment, offering insights into the cell's functional state and identity. This enables to identify novel cell types, understand developmental pathways, and study disease processes at the resolution of individual cells. Other single-cell omics include:

- *Genomics*, which analyses the DNA at the single-cell level and provides detailed insights into the genetic makeup and variations within single cells (Gawad et al., 2016).
- *Proteomics*, which aim at directly measuring the protein composition of cells (Kelly, 2020). This is a challenging task given the lower abundance of proteins compared to RNA, but it provides a more precise measurement as it is less affected by transcriptional bursting (see section 2.2.2).
- *Epigenomics*, which explore the epigenetic landscape of individual cells, including DNA methylation patterns and chromatin accessibility (scATAC-seq). This helps in understanding how gene regulation and expression vary from cell to cell (Kelsey et al., 2017) (see section 8.2).

The field is further advancing by integrating single-cell RNA sequencing with other modalities, in so called *single-cell multi-omics* technologies. These emerging technologies, which combine various types of measurements from the same single cell, are likely to become a standard part of molecular biology research in the future.

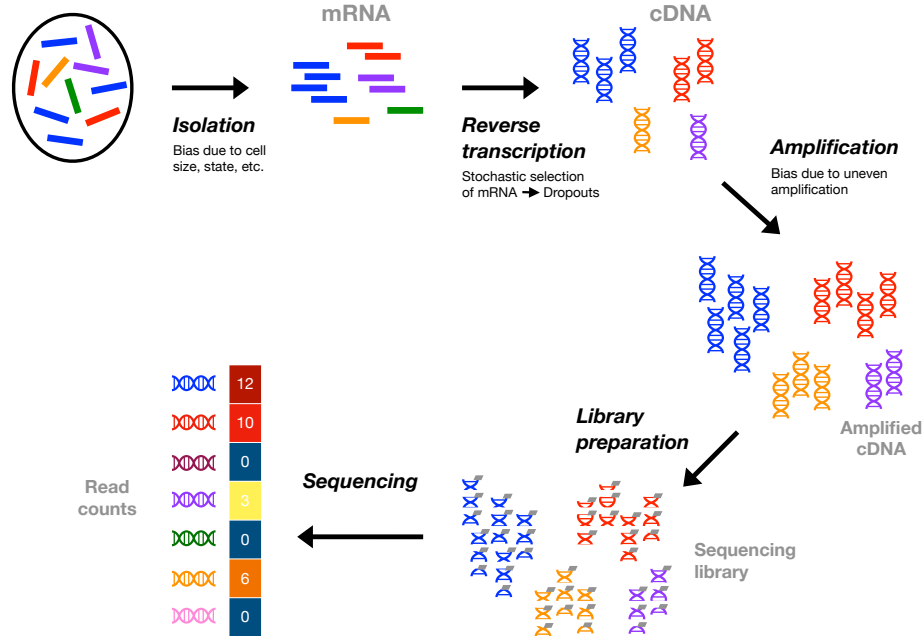


Figure 2.1: Sequencing workflow of scRNA-seq. The different steps occurring during a typical sequencing process are indicated, with the possible biases they can introduce in the final result.

2.2 Single-cell transcriptomics

This section focuses on single-cell transcriptomics, which is the main data type used throughout this thesis. It first outlines the typical workflow used by sequencing technologies to obtain read counts from a cellular tissue, as well as a description of the main sequencing technologies. The biases that can arise during this process are then discussed, as well as a typical analysis pipeline for biomarker identification from the obtained data.

2.2.1 Sequencing workflow

The typical workflow to sequence the transcriptome of single-cells (figure 2.1) starts by isolating the cells and extracting the mRNAs, transcribing them into cDNA which are then amplified and profiled (Stegle et al., 2015; Ziegenhain et al., 2017).

Single-cell isolation

The first step in scRNA-seq sequencing is to prepare a single-cell suspension from the tissue to be analyzed, i.e. dissociating it into individual cells floating in a liquid. This process can involve mechanical dissociation, enzymatic digestion, or a combination of both. The goal is to separate cells while minimizing stress and damage, as this can affect the quality of the result. Once a single-cell suspension is obtained, the individual cells must be isolated to ensure that the sequenced RNA comes from a single cell. Depending on the sequencing technology, this is either achieved using FACS or a microfluidics-based system. The cells are then lysed in order to extract the mRNA molecules.

Reverse transcription

Once the cells are isolated, the RNA is converted into complementary DNA (cDNA) as the latter is more stable and easier to amplify. This is achieved through *reverse transcription*, which uses a reverse transcriptase enzyme to transcribe the RNA into cDNA. During this process, the cDNA is tagged with a unique identifier, either a barcode or a unique molecular identifier (UMI) that marks which cell the cDNA originated from. This is particularly important in methods where cells are pooled during sequencing.

Amplification

As the amount of cDNA obtained from a single cell is extremely small, it is necessary to amplify it to generate sufficient material for sequencing. This is typically achieved using polymerase chain reaction (PCR), which can rapidly make millions of copies of a specific DNA segment. The challenge here is to amplify the cDNA evenly, i.e. avoiding preferential amplification of certain transcripts over others.

Library preparation and sequencing

After amplification, a sequencing library is prepared. The cDNA is fragmented, and adaptors are attached to the ends of the fragments, which will bind to the sequencing platform. This allows the detection of these short DNA sequences, or *reads*, during the sequencing. After a quality control to ensure that they are accurate and reliable, the reads are mapped to a reference genome or transcriptome to identify the genes they originate from. Finally, the number of reads is quantified for each gene in each cell, which provides a measure of gene expression levels.

The scRNA-seq sequencing technologies can be separated in two categories: *full-length* technologies, such as Smart-seq2 (Picelli et al., 2014), which sequence the entire transcripts and consequently detects more genes per cell, and *droplet-based* technologies, such as 10X Chromium (Zheng et al., 2017), which only sequence the 3' or 5' ends of transcripts but achieve a significantly higher throughput.

Choosing the appropriate sequencing technology should depend on the research question being investigated. With its higher sensitivity for gene detection but lower throughput, Smart-seq2 is especially appropriate to analyze complex samples that may be of limited availability, for instance tumor samples. Moreover, Smart-seq2 includes a FACS step that can separate cells into different major cell types before sequencing. One typically obtains datasets ranging from 100 to 1000 cells, often representing a population isolated as relatively homogeneous, with a high number of genes detected per cell. This allows a better detection of genes at low expression levels, as well as alternatively spliced transcripts (Wang et al., 2021; Ziegenhain et al., 2017).

On the other hand, the high throughput of 10X allow the simultaneous sequencing of tens to hundreds of thousands of cells, at a lower cost. This is especially appropriate to perform a detailed characterization of cell types in samples that are abundant and easily accessible (such as peripheral blood mononuclear cells, mouse organs, etc.), which are typically highly heterogeneous. While less genes are detected per cell, such large data generally allows the detection of more cell clusters, given the higher abundance of cells of each type (Wang et al., 2021).

2.2.2 Bias sources

Being able to sequence the transcriptome at the single-cell level gives a lot of information about the cell population under study. The drawback is that the measured expression values are more subject to noise, as technical variability can appear at different stages of the sequencing process. Additional biological variability, both at the sample level (e.g. differences between donors) or the cell level, further complexity the interpretability of the result. Computational methods should therefore mind these sources of bias in order to produce accurate results (Choudhary and Satija, 2022; Todorov and Saeys, 2019). This section details some typical sources of bias.

Dropouts

An important source of bias that appears in single-cell RNA-seq data are *dropouts*, whereby a gene that is actually expressed in a cell is not detected in the sequencing data, consequently leading to a “false zero”. Dropout events result in a sparse data matrix with many zeros, posing a challenge to differential expression analysis as they can lead to an underestimation of gene expression levels, and affect cell type identification (Vallejos et al., 2017; Wagner et al., 2016; Ziegenhain et al., 2017).

Genes with lower expression levels tend to have higher dropout ratios (Brennecke et al., 2013; Wang et al., 2021). Indeed, dropouts are a consequence of the small amount of RNA material that is present in a single cell. Since reverse transcription is stochastic by nature, a gene may not be detected if none of its corresponding RNA transcripts are converted into cDNA. Moreover, some transcripts may not be amplified sufficiently to be detected. Logically, a lowly expressed gene, i.e. with few RNA transcripts, has a higher probability to go undetected.

The sensitivity of scRNA-seq protocols to capture and convert a particular mRNA transcript into cDNA can vary, as high-throughput protocols typically show higher dropout rates (Ziegenhain et al., 2017). In particular, Smart-seq2 has been shown to have lower dropout ratios than 10X (Wang et al., 2021). However, high amounts of sequenced cells can help to infer dropout probabilities (Todorov and Saeys, 2019), using methods which identify zero counts that are most likely resulting from dropout events, such as ZIFA (Pierson and Yau, 2015) and ZINB-WaVE (Risso et al., 2018).

Amplification bias

Attention should be paid to the accuracy of sequencing protocols, i.e. how well the read quantification corresponds to the actual concentration of mRNAs, and in particular to the precision of the amplification process (Ziegenhain et al., 2017). *Uneven amplification* of different RNA transcripts can distort the quantitative assessment of gene expression levels, leading to some transcripts being over (or under) represented in the count data. Such discrepancies can make it challenging to compare expression levels across different cells or to identify low-abundance transcripts accurately.

Droplet-based protocols, such as 10X, have sacrificed full-length coverage in order to sequence part of the primer used for cDNA generation (Ziegenhain et al., 2017). This enables multiplexing, and hence a higher throughput, and allows the

incorporation of *unique molecular identifiers* (UMI), random nucleotide sequences that tag individual mRNA molecules. This allows for the distinction between original molecules and amplification duplicates that derive from the cDNA amplification (Kivioja et al., 2012). Regardless of how many times a particular RNA transcript has been amplified, it is only counted once based on its unique UMI. Counting UMIs rather than the total number of reads drastically improves the quantification of RNA transcripts and nearly eliminates amplification bias (Grün et al., 2014a; Islam et al., 2014).

Transcriptional bursting

Besides technical bias induced by sequencing protocols, biological variability may also affect the analysis of scRNA-seq data. Indeed, gene expression is characterized by *transcriptional bursting*: the production of mRNA transcripts occurs in a series of discontinuous bursts, rather than at a constant rate (Fukaya et al., 2016; Tunnacliffe and Chubb, 2020). This means that a gene can switch between states of high activity (bursting) and little or no activity (silence), consequently leading to varying levels of measured expression. Transcriptional bursting can lead to an unusually high count value for a gene in a particular cell, compared to others cells of the same type, further complexifying the direct comparison of expression levels between cells.

2.2.3 The typical scRNA-seq pipeline

A scRNA-seq sequencing machine outputs a matrix of read (or UMI) counts $\mathbf{R} \in \mathbb{N}^{|\mathcal{G}| \times |\mathcal{C}|}$, where the rows represent the set of genes $\mathcal{G}$ and the columns the set of cells $\mathcal{C}$. An entry r_{ij} of this matrix represents the measured expression of gene i in cell j .¹ More specifically, it contains the read count value, i.e. the number of times an RNA transcript of gene i has been detected in cell j . In order to perform a proper biological analysis, this raw expression data should undergo quality control and be normalized to remove unwanted variability (Luecken and Theis, 2019; Todorov and Saeys, 2019). One can then proceed to identifying the different cell types and corresponding biomarkers.

¹Throughout this thesis, uppercase bold letters are used to denote matrices (e.g. $\mathbf{A}$), lowercase bold for vectors (e.g. $\mathbf{v}$), and lowercase italic for scalars (e.g. x). Consequently, v_i denotes the i th element of vector $\mathbf{v}$, and a_{ij} the element on the i th row and j th column of matrix $\mathbf{A}$. When taking a vector from a matrix $\mathbf{A}$, the domain of the running index defines whether it is a row or a column: $i \in \mathcal{G}$ will always be used to identify rows (i.e. genes), and $j \in \mathcal{C}$ for columns (i.e. cells). Therefore, $\mathbf{a}_i$ represents the i th row of $\mathbf{A}$, whereas $\mathbf{a}_j$ represents the j th column of $\mathbf{A}$.

Quality control and normalization

One of the first steps is to perform on the raw read count data is *quality control* (Lun et al., 2016). Cells with an unusually low, or high, number of detected genes may indicate poor-quality cells, or doublets (i.e. two cells mistakenly sequenced as one). Two common measures of cell quality are the *library size*, defined as the total sum of counts across all features (i.e. genes and spike-in transcripts), and the number of expressed features in each library. Another measure of quality is the proportion of reads mapped to *mitochondrial genes*. A high proportion can indicate a cell is damaged or dying. All these cells are generally removed before further analysis.

After quality control, the data is *normalized* to try to remove unwanted biological and technical effects that can mask the signal of interest (Vallejos et al., 2017). Among others, normalization is performed to adjust differences in read counts between cells, such as the *sequencing depth* (i.e. the average number of times each base in the genome, i.e. target region, is read during sequencing). It aims at bringing cell-specific measures onto a common scale, which is for instance done by scaling to *reads per million* (RPM). This produces a normalized read count matrix $\mathbf{N} \in \mathbb{R}_+^{|\mathcal{G}| \times |\mathcal{C}|}$. scRNA-seq data is generally *log-normalized* to stabilize the variance across genes with different expression levels, since gene expression data is often skewed, with a few genes highly expressed and many genes expressed at low levels. A log-count matrix is for instance obtained using $\mathbf{L} = \log_2(\mathbf{N} + 1)$, where the 1 pseudocount is added to avoid the undefined $\log(0)$ case.

Feature selection

After normalizing the data, *feature selection* is often performed. The goal is to identify genes that significantly contribute to the cell heterogeneity and the biological processes under study, i.e. behave differently from a null model describing the assumed technical noise. These genes are generally referred to as *highly variable genes* (HVG). Cell type identification methods often include such a feature selection step before performing clustering or other analyses (see examples in sections 2.3.2 and 2.3.3). Given the high dimensionality of gene expression data (order of 10 000 dimensions), performing feature selection improves the computational efficiency and limits the curse of dimensionality, alleviating model overfitting in the downstream analysis (Yang et al., 2021).

Cell type identification

One of the main goals of analyzing scRNA-seq data is to identify the different cell types that constitute a cell population. To this end, several downstream analyses are typically performed on the normalized scRNA-seq data.

A semi-automatic way to identify cell types is to use *dimensionality reduction*. Contrary to feature selection, which doesn't alter the original features but selects a subset of them, dimensionality reduction methods combine features to produce new ones, defining a space of lower dimension of generally 2D or 3D which can then be visually inspected. Popular methods include PCA (Wold et al., 1987) and *t*-SNE (van der Maaten and Hinton, 2008). *Principal Component Analysis* (PCA) is a linear dimensionality reduction method that identifies the principal components, i.e. linear combinations of the genes that capture the maximum variance in the data. *t-distributed Stochastic Neighbor Embedding* (*t*-SNE) is a nonlinear dimensionality reduction technique that measures pairwise similarities between neighboring data points in the high-dimensional space, and optimizes a lower-dimensional representation to preserve these similarities.

Unsupervised clustering is a common task in the scRNA-seq pipeline. It consists in grouping cells in a fully automatic way using their expression values to highlight subpopulations in the cell tissue. Beyond classical generic clustering algorithms, several methods have been developed specifically for scRNA-seq data (Kiselev et al., 2019), such as SC3 (Kiselev et al., 2017) or Seurat (Satija et al., 2015), to be robust with respect to the specific sources of bias of this data.

As the ultimate goal when analyzing scRNA-seq data is to gain new insights into (the immune response to) a certain pathology, one typically looks for specific genes describing each cell type. Once different clusters have been identified, a *differential expression* analysis is therefore generally performed. It consists in comparing gene expression levels across clusters, to identifies marker genes that characterize each cell type (i.e. are expressed in a different way than in the remaining cells), so as to provide biological insights into their functions.

2.2.4 Batch effects

Nowadays, the availability of single-cell omics data is widespread. When designing an experiment, it is common to have access to single-cell data from multiple sources. For instance, one could aggregate single-cell datasets obtained by biopsies of several cancer patients in order to get a better understanding of the pathology. Such larger quantities of data can make it easier to extract information using

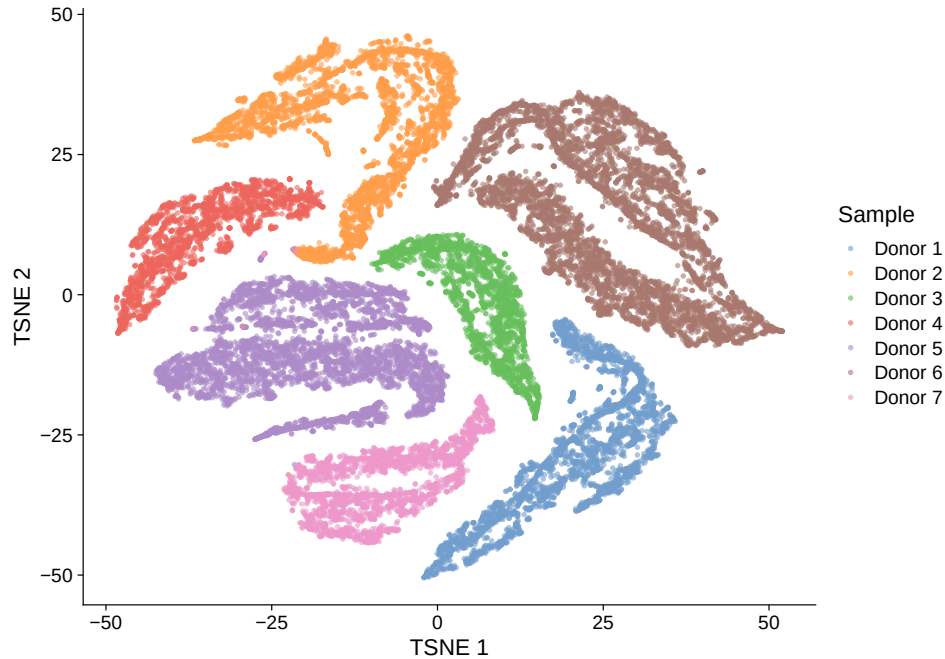


Figure 2.2: Dimensionality reduction of the NSCLC DTCs dataset using t -SNE. Several clusters are visible, but each one contains only cells coming from one single donor. t -SNE evidently captures variability between batches rather than grouping cells together by specific expression patterns.

data-mining techniques, allowing us to gain insights into biological phenomena that could not be retrieved from one single dataset. However, aggregating several single-cell samples comes with pitfalls: *batch effects* may appear due to the presence of technical variability (differences in sequencing technologies, experimental conditions, etc.) as well as biological variance between donors (Luecken et al., 2022). If not handled properly, unsupervised (bi)clustering algorithms could pick up this unwanted variation rather than biologically relevant information.

Figure 2.2 shows the result of t -SNE on a dataset containing 26 027 Non-Small Cell Lung Cancer (NSCLC) Dissociated Tumor Cells (DTCs) coming from 7 donors (see details about this dataset in appendix A.7). Several clusters seem to appear in the result. However, when assigning a color based on the origin of each cell, it turns out that the clusters simply identify the donors. The variance of expression captured in this projected and reduced space actually accounts for

the batch effects across samples rather than the different cell types present in the data.

To avoid such bias, several *data integration* techniques performing *batch correction* have been proposed (Argelaguet et al., 2021; Luecken et al., 2022). They aim at combining several single-cell samples into one consistent dataset, on which the standard (i.e. single-sample) scRNA-seq analysis pipeline can be subsequently applied. They generally rely on the definition of anchors that link different datasets, from which correction vectors can be computed.

Mutual Nearest Neighbors

The Mutual Nearest Neighbors (MNN) method (Haghverdi et al., 2018) identifies pairs of cells in different samples that are mutually nearest to each other, which define the most similar cells of the same type across samples. The difference in expression between MNN pairs, averaged across many such pairs, provides an estimation of the batch effect, from which a correction vector is derived that integrates the different samples.

More specifically, the MNN algorithm integrates 2 datasets by defining one as reference and correcting the expression values of the other (algorithm 2.1). It starts by performing a cosine normalization of the data to be robust to technical differences in sequencing depth. Next, for each cell j in batch 1, it identifies the k nearest neighbors within the cells of batch 2 (i.e. those with the smallest distance to j), and conversely. A pair of cells (j, j') from respectively batch 1 and batch 2 are *mutual nearest neighbors* whenever each one is a nearest neighbor of the other.

A pair of mutual nearest neighbors is interpreted as cells of the same type. Differences in expression levels between them are therefore assumed to be batch effects. Such an interpretation relies on the assumptions that:

- i. at least one cell type is present in both batches;
- ii. the batch effects are almost orthogonal to the biological subspace;
- iii. the resulting variations are much smaller than the biological difference between cell types.

The *biological subspace* refers to a set of vectors that each represent a particular biological process (e.g. the cell cycle, cell type-specific profiles, differentiation or activation states, etc.). The true expression profile of a cell could be represented as linear combination of these vectors. The actually measured expression profile

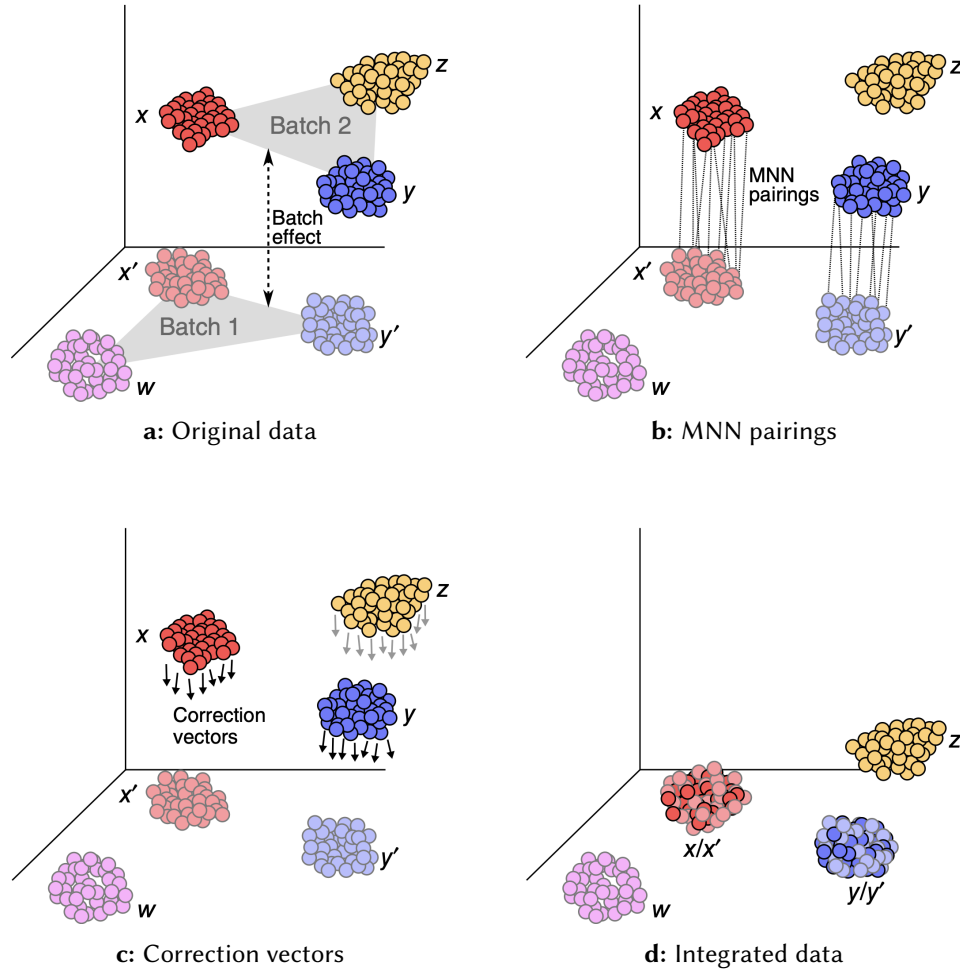


Figure 2.3: Batch correction using Mutual Nearest Neighbors (Haghverdi et al., 2018). **a** Two single-cell batches represented in the original space (we here assume for simplicity $|\mathcal{G}| = 3$), with an almost orthogonal batch effect difference between them. The colors represent the different cell types. **b** Mutual nearest neighbors are identified between the two batches. **c** Correction vectors are computed from these MNN pairs. **d** Batch 1 is regarded as the reference, and batch 2 is integrated into it by subtracting the correction vectors.

Algorithm 2.1: MNN

Input: $\mathbf{R}_1 \in \mathbb{N}^{|\mathcal{G}| \times |\mathcal{C}_1|}$, $\mathbf{R}_2 \in \mathbb{N}^{|\mathcal{G}| \times |\mathcal{C}_2|}$ two single-cell gene expression matrices

Input: $k \in \mathbb{N}$ the number of nearest neighbors to consider

Input: $\sigma \in \mathbb{R}$ the parameter of the Gaussian kernel

Output: $\mathbf{N} \in \mathbb{R}_+^{|\mathcal{G}| \times |\mathcal{C}_1 \cup \mathcal{C}_2|}$ an integrated single-cell matrix

```

// 1) Cosine normalization
for  $j \in \mathcal{C}_1$  do  $\mathbf{n}_{1j} \leftarrow \frac{\mathbf{r}_{1j}}{\|\mathbf{r}_{1j}\|}$ 

for  $j' \in \mathcal{C}_2$  do  $\mathbf{n}_{2j'} \leftarrow \frac{\mathbf{r}_{2j'}}{\|\mathbf{r}_{2j'}\|}$ 

// 2) Identification of MNNs
for  $j \in \mathcal{C}_1$  do  $NN_j \leftarrow k\_nearest\_neighbors(\mathbf{n}_{1j}, \mathbf{N}_2)$ 
for  $j' \in \mathcal{C}_2$  do  $NN_{j'} \leftarrow k\_nearest\_neighbors(\mathbf{n}_{2j'}, \mathbf{N}_1)$ 
 $MNN \leftarrow \emptyset$ 

for  $j \in \mathcal{C}_1; j' \in \mathcal{C}_2$  do
    if  $j' \in NN_j$  and  $j \in NN_{j'}$  then
         $MNN \leftarrow MNN \cup \{(j, j')\}$ 

// 3) Computation of correction vectors
for  $(j, j') \in MNN$  do // Pair-specific vectors
     $\mathbf{pv}_{jj'} \leftarrow \mathbf{n}_{1j} - \mathbf{n}_{2j'}$ 

for  $j'' \in \mathcal{C}_2$  do // Cell-specific vectors
    for  $(j, j') \in MNN$  do  $w_{jj'} \leftarrow \exp\left(-\frac{\|\mathbf{n}_{2j'} - \mathbf{n}_{2j''}\|^2}{2\sigma^2}\right)$ 
     $\mathbf{v}_{j''} \leftarrow \frac{\sum_{(j,j') \in MNN} w_{jj'} \mathbf{v}_{jj'}}{\sum_{(j,j') \in MNN_{j''}} w_{jj'}}$ 
     $\mathbf{n}'_{2j''} \leftarrow \mathbf{n}_{2j''} - \mathbf{v}_{j''}$ 

return (  $\mathbf{N}_1 \mathbf{N}'_2$  )

```

would be the result of adding a vector representing the batch effects. Under the stated assumptions, cells from the same type in different batches will form MNN pairs, as they will form parallel hyperplanes with respect to each other.

A pair-specific correction vector is defined for each MNN pair as the difference between their expression values. Cell-specific vectors are then computed for batch 2 as the weighted average of the pair-specific vectors, with weights computed using a Gaussian kernel. This stabilizes the correction for each cell and ensures smooth changes between adjacent cells, as each pair vector contributes to the batch correction for cells in the neighborhood of the corresponding pair. The correction vectors are then subtracted from the expression vectors of the cells of batch 2, effectively integrating them with batch 1. Whenever more than two batches need to be integrated, MNN follows an iterative process which considers the previously integrated batches as reference and computes the correction vectors for the next batch.

Other methods

Several other approaches have been build upon this principle, generally after reducing the dimensionality of the original data. Indeed, the accurate identification of MNNs can be complicated whenever the batch effects are on a similar scale to the biological differences between cell types. Seurat v3 (Stuart et al., 2019) therefore first jointly reduces the dimensionality of both datasets using diagonalized *canonical correlation analysis* (CCA), and applies L_2 -normalization to the canonical correlation vectors. It then identifies the MNNs in this lower-dimensional representation of the expression data.

Scanorama (Hie et al., 2019) was developed based on the observation that the integration results can be largely dependent on the order in which the batches are integrated. Rather than iteratively integrating one batch to a reference dataset, Scanorama integrates all batches at once by identifying MNNs among all pairs of batches. It first reduces the dimensions using *singular value decomposition* (SVD), and then searches for MNNs across all batches. Connected components are then iteratively merged into larger panoramas. The pairs of batches are ordered according to the percentages of cells in the batches that were involved in a MNN. Panoramas are built by successively merging a batch into a panorama based on this ordering.

As such, one loses the original interpretation in terms of genes when integrating datasets in lower dimensions. Such integrated data can thus be used to identify cell types, but it is not possible to perform a differential expression

analysis. Yet, adaptations of Seurat v3 and Scanorama allow to perform batch correction, i.e. upscaling the integration back to the original dimensions.

2.3 Rare cell identification

The high heterogeneity of a cell population is manifested by the presence of several cell types, which can be present in very different proportions. While a typical scRNA-seq pipeline is often effective at identifying abundant cell types, special attention must be given to the identification of *rare cell types*, which can play a major role in the biological process under study.

2.3.1 Motivation

In general, clustering tends to group cells in relatively large clusters. It is therefore very effective at identifying major cell types, but tends to miss rare subpopulations which only account for a small fraction of the cells. Let's consider a collection of T lymphocytes extracted from a human breast tumor, sorted by FACS into CD8 T cells and activated regulatory T cells (expressing the transforming growth factor beta activator LRRC32 protein, or GARP+) and processed following the Smart-seq2 protocol (Picelli et al., 2014) (see details in appendix A.1). Three datasets of 50 cells are sampled from the 179 available cells (96 GARP+ Tregs and 86 CD8 T cells after quality control), with varying GARP+/CD8 ratio. Figure 2.4 show the result when running SC3, a method performing a consensus clustering, on these datasets. The two types of lymphocytes have very distinct functions, which should be reflected in their gene expression. The SC3 method has no trouble to distinguish between both cell types when their relative proportion is, by design here, 50/50. Yet, when the GARP+ Tregs only represent a smaller fraction of the data (here 10%), SC3 clearly fails to identify them as forming a separate and specific cluster.

An alternative way to identify cell types is to rely on dimensionality reduction methods to represent these cells in a 2D-space, and visually inspect such representations to spot specific clusters. Figure 2.5 illustrates such an example by randomly sampling three datasets 5 GARP+ Tregs and 45 CD8 T cells and reducing the data in 2D-spaces using PCA or *t*-SNE. Cluster separation after PCA is relatively poor on this example, and it would be difficult to identify the GARP+ Treg subpopulation without any kind of supervision (the color code is only available for illustration purpose in such a controlled experiment). *t*-SNE produces more distinct clusters on this example, but also fails to separate the GARP+ Tregs

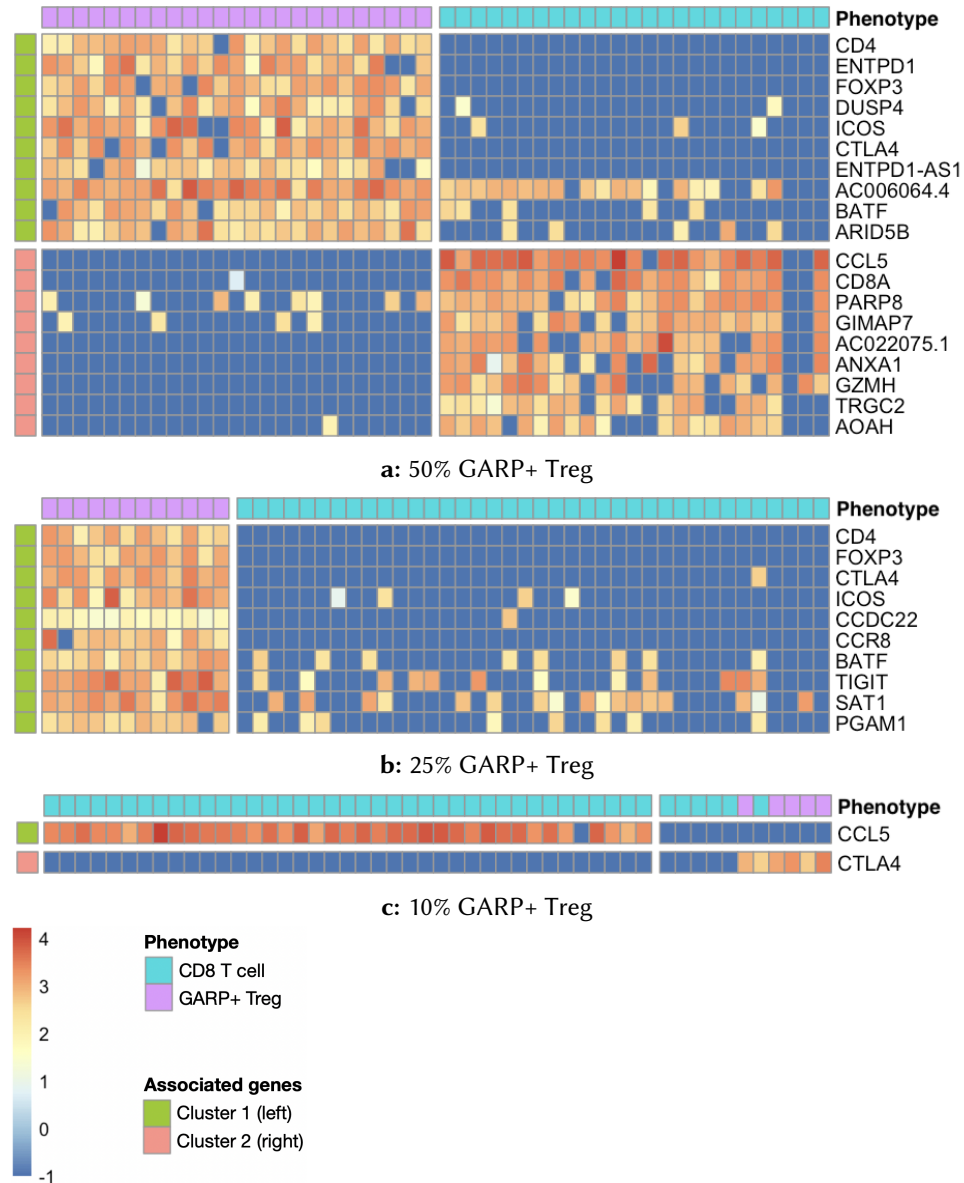


Figure 2.4: Result of SC3 on mixtures of GARP+ Tregs (purple) and CD8 T cells (turquoise). SC3 correctly clusters both cell types separately whenever their relative proportion is either 50/50% (a) or 25/75% (b). SC3 fails to identify the GARP+ Tregs as a separate and specific cluster when their relative proportion is only 10% of the cells (c). The reported genes are the marker genes identified by SC3 for each cluster. Expression values are log-normalized ($\log_{10}(x + 0.1)$).

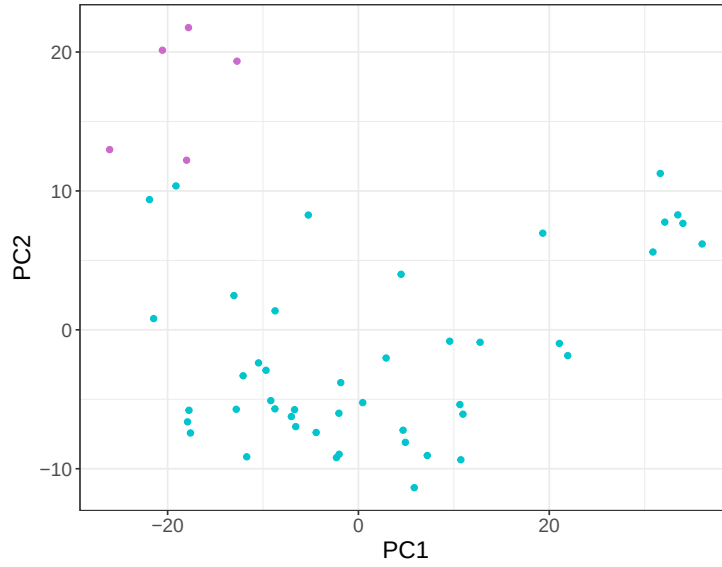
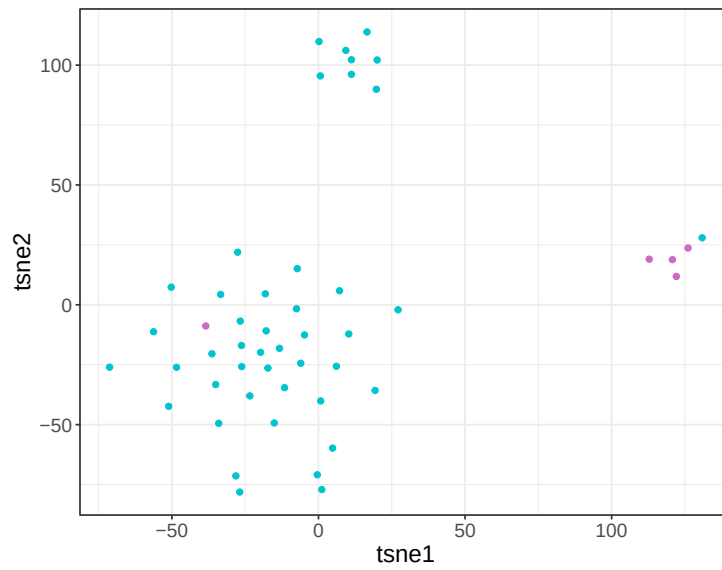
**a: PCA****b: *t*-SNE**

Figure 2.5: 2-D representation after dimensionality reduction of a GARP+ Tregs (purple, 10% of the data) and CD8 T cells (turquoise, 90%) mixture. Cluster separation is poor with PCA (a). A better separability is observed with *t*-SNE (b) but this method fails at identifying the GARP+ Tregs as forming a separate cluster.

from CD8 T cells. These examples illustrate the need for dedicated methods, beyond generic clustering methods, to identify rare subpopulations from scRNA-seq data.

2.3.2 Clustering based approaches

RaceID (Grün et al., 2015) and GiniClust (Jiang et al., 2016) are two popular methods to address this question. They perform a complete clustering of the data but include mechanisms to specifically focus on the discovery of small clusters.

RaceID

A first method for rare cell identification, RaceID, has been proposed by Grün et al. (2015). In a nutshell, it first runs a classical clustering approach, k -means, to divide the dataset into large clusters. Then, it identifies possible outliers inside each cluster, and performs a second partition of the data based on these outliers (algorithm 2.2).

More specifically, RaceID starts by constructing a similarity matrix using Pearson's correlation coefficients for all pairs of cells, and subtracts these coefficients from one to obtain a distance matrix. The k -means clustering (MacQueen, 1967) is then applied using the Euclidean distance. The determination of the optimal number of clusters relies on the gap statistic (Tibshirani et al., 2001), which quantifies the difference of average within-cluster dispersion between uniformly distributed data and the actual data. RaceID uses the first local maximum of the gap statistic to set the default cluster number. Cluster reproducibility is evaluated by bootstrapping and computing Jaccard's similarity. If more than one cluster has a similarity lower than 0.5, the clustering process is repeated with fewer clusters.

To identify rare clusters, RaceID looks for outlier cells within each cluster by assessing the transcript count variability for every gene across all cells in the cluster, and comparing it to a background model computed over all cells, that accounts for technical and biological noise. Specifically, a second-order polynomial is fitted to the transcript count variance as a function of the average transcript count in logarithmic space. This serves as an estimate of the expected variance-mean dependence under the assumption genes do not exhibit cluster specific expression.

Outlier cells are then identified by computing the transcript count probability in each cell for a given gene from a negative binomial distribution defined by the average transcript count of this gene across all cells in the cluster and the expected

Algorithm 2.2: RacelD

Input: $\mathbf{N} \in \mathbb{R}_+^{|\mathcal{G}| \times |\mathcal{C}|}$ a normalized single-cell gene expression matrix
Input: $k \in \mathbb{N}$ the number of clusters to use in the first partition
Output: $\mathbf{cl} \in \mathbb{N}^{|\mathcal{C}|}$ the cluster partition of the cell

```

// 1) k-means clustering
 $\mathbf{P} \in \mathbb{R}^{|\mathcal{C}| \times |\mathcal{C}|} \leftarrow \text{pearson\_correlation}(\mathbf{N})$ 
 $\mathbf{D} \leftarrow \mathbf{1} - \mathbf{P}$ 
 $\mathbf{cli} \leftarrow \text{kmeans}(\mathbf{D}, k)$ 

// 2) Outlier detection
 $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{G}|} \leftarrow \log_2(\text{rowmeans}(\mathbf{N}))$ 
 $\boldsymbol{\sigma}^2 \in \mathbb{R}^{|\mathcal{G}|} \leftarrow \log_2(\text{rowvars}(\mathbf{N}))$ 
 $f \leftarrow \text{poly\_regression}(x=\boldsymbol{\mu}, y=\boldsymbol{\sigma}^2, \text{degree}=2)$ 
 $C_{\text{outliers}} \leftarrow \text{find\_outliers}(\mathbf{N}, \mathbf{cli}, k, f)$  // Algorithm 2.3

// 3) Cluster outliers
 $P_{\text{others}} \leftarrow \{p_{ij} \mid i, j \in \mathcal{C} \setminus C_{\text{outliers}}, i < j\}$ 
for  $o \in C_{\text{outliers}}$  do
     $C_{\text{newcl}} \leftarrow \{o' \in C_{\text{outliers}} \mid p_{oo'} > q_3(P_{\text{others}})\}$ 
     $k \leftarrow k + 1$ ;  $\mathbf{cli}_{o''} \leftarrow k \quad \forall o'' \in C_{\text{newcl}}$ 
     $C_{\text{outliers}} \leftarrow C_{\text{outliers}} \setminus C_{\text{newcl}}$ 

// 4) Final clustering
for  $x \in \{1, \dots, k\}$  do
     $\mathbf{c}_x \leftarrow \text{rowmeans}(\mathbf{N}|_x)$  // Only cells from cluster  $x$ 

for  $j \in \mathcal{C}$  do
     $cl_j \leftarrow \text{argmax}_{x \in \{1, \dots, k\}} \{\text{pearson\_correlation}(\mathbf{n}_j, \mathbf{c}_x)\}$ 

return  $\mathbf{cl}$ 

```

Algorithm 2.3: RacelD – find_outliers

Input: $\mathbf{N} \in \mathbb{R}_+^{|\mathcal{G}| \times |\mathcal{C}|}$ a normalized single-cell gene expression matrix

Input: $\text{cli} \in \mathbb{N}^{|\mathcal{C}|}$ an initial clustering partition

Input: $k \in \mathbb{N}$ the number of clusters

Input: $f : \mathbb{R} \rightarrow \mathbb{R}$ a background model

Output: $C_{\text{outliers}} \subseteq \mathcal{C}$ the set of outliers

$C_{\text{outliers}} \leftarrow \emptyset$

for $x \in \{1, \dots, k\}$ **do**

$C_x \leftarrow \{c \in \mathcal{C} \mid \text{cli}_c = x\}$

for $j \in C_x$ **do**

$G_{\text{low}} \leftarrow \emptyset$

for $i \in \mathcal{G}$ **do**

$\mu_{ix} \leftarrow \frac{1}{|C_x|} \sum_{q \in C_x} n_{iq}$

$\sigma_i^2 \leftarrow f(i)$

$X \sim \text{NB}\left(r = \frac{\mu_{ix}^2}{\sigma_i^2 - \mu_{ix}}, p = \frac{\mu_{ix}}{\sigma_i^2}\right)$

$p \leftarrow \text{FDR_correction}(\text{F}_X(n_{ij}))$

if $p < 10^{-4}$ **then** $G_{\text{low}} \leftarrow G_{\text{low}} \cup \{i\}$

if $|G_{\text{low}}| \geq 2$ **then** $C_{\text{outliers}} \leftarrow C_{\text{outliers}} \cup \{j\}$

return C_{outliers}

variance computed by the second order polynomial. RaceID considers a cell as an outlier if the FDR-corrected transcript count probability for a specified number of genes (default 2) falls below a defined probability threshold (default 10^{-4}).

Finally, the outlier cells are grouped into small clusters based on expression correlation. New clusters are created by merging outlier cells whose transcriptome correlation exceeds the 75%-quantile (q_3) of the distribution of cell-to-cell correlation within the original clusters after outlier removal. New cluster centers are then computed for all clusters (original and outlier ones) by averaging the count values within each cluster. Each cell is reassigned to the most highly correlated cluster center, providing a full clustering of the data with abundant and rare cell types.

RaceID has been further improved by replacing the k -means algorithm by k -medoids clustering (Kaufman, 1990), and determining the cluster number using the saturation point of the within-cluster dispersion rather than the gap statistic, in order to obtain a more robust original clustering (RaceID2) (Grün et al., 2016). RaceID3 provides further improvements, including a univariate feature selection step before clustering, and a random-forest-based reclassification after outlier detection (Herman et al., 2018).

GiniClust

GiniClust has been proposed by Jiang et al. (2016) based on the observation that the effect of traditional differential expression metrics, such as the Fano factor (Grün et al., 2014b), diminishes as the cell subpopulation becomes rarer (figure 2.6a). The authors proposed to use the *Gini index* instead, which had originally been designed to analyze social inequality (Gini, 1912). The value of this index remains high for a gene that is differentially expressed even in a small proportion of the data (figure 2.6b).

The Gini index is calculated for each gene by sorting cells based on their expression levels and plotting the cumulative expression levels as cells are included in the ranked list (which produces a Lorenz curve). The Gini index is defined as two times the area between the Lorenz curve and the diagonal, which varies between 0 (most uniform) and 1 (most extreme). As the Gini index strongly correlates with the maximum gene expression levels, it is normalized by fitting a curve using LOESS regression, and then removing the fitted trend from the original Gini value (algorithm 2.4). GiniClust then selects the “high Gini genes”, and performs a clustering based on these genes using DBSCAN (Ester et al., 1996).

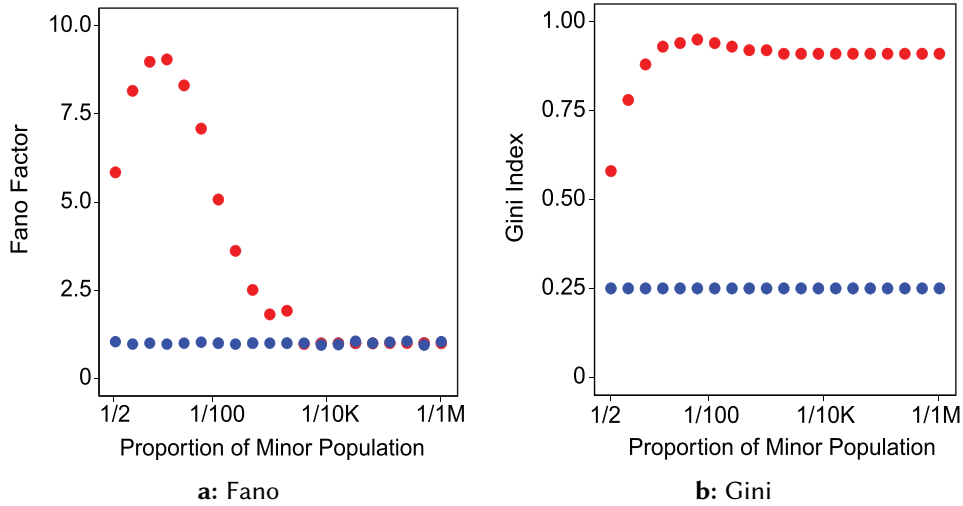


Figure 2.6: Comparison between Gini index and Fano factor in detecting differentially expressed genes (Jiang et al., 2016). Evolution of the metrics of two genes, in a mixed population of 1 000 000 cells containing two cell types, for varying proportions of the minor cell type (1/1 M stands for one in one million). Only the red gene presents a differential expression between the two cell types. **a** The Fano factor starts to decrease as soon as the minor cell type represents less than 10% of the cells, and altogether fails to detect any differential expression when this proportion falls below 0.1%. **b** The Gini index stays high for the differentially expressed gene even as the proportion of the minor cell type decreases.

Subsequent versions of GiniClust introduce a consensus clustering between the Gini and Fano factors (GiniClust2) (Tsoucas and Yuan, 2018) and scale up to large-scale datasets by replacing the clustering step by community detection algorithms for large networks (GiniClust3) (Dong and Yuan, 2020).

Discussion

Both RaceID(3) and GiniClust(3) are univariate methods in the sense that the gene expression analysis (either assessed through a count probability or a Gini index) is performed gene by gene. Such methods thus plainly ignore the possible dependence between different gene expression values when looking for rare subpopulations of cells but rather consider each gene as playing an independent role. These methods are therefore computationally efficient but are likely to miss interesting patterns when a set of genes forms a specific expression altogether.

Algorithm 2.4: GiniClust

Input: $\mathbf{N} \in \mathbb{R}_+^{|\mathcal{G}| \times |\mathcal{C}|}$ a normalized single-cell gene expression matrix**Output:** $\mathbf{cl} \in \mathbb{N}^{|\mathcal{C}|}$ the cluster partition of the cell

// 1) Compute normalized Gini index

for $i \in \mathcal{G}$ **do**

$$\left[\begin{array}{l} gini_i = \frac{\sum_{j \in \mathcal{C}} \sum_{j' \in \mathcal{C}} |n_{ij} - n_{ij'}|}{2|\mathcal{C}| \sum_{j \in \mathcal{C}} n_{ij}} \end{array} \right.$$

 $f \leftarrow \text{LOESS_regression}(\log_2(\text{rowmax}(\mathbf{N})))$ **for** $i \in \mathcal{G}$ **do** $gini_i \leftarrow gini_i - f(i)$

// 2) Select high Gini genes

$$\mathbf{p} \leftarrow \Phi\left(-\left|\frac{\mathbf{gini} - \mu_{\mathbf{gini}}}{\sigma_{\mathbf{gini}}}\right|\right) \quad // \text{ CDF of std. normal dist.}$$

$$G_{\text{gini}} \leftarrow \{i \in \mathcal{G} \mid p_i < 0.0001\}$$

// 3) Clustering using only high Gini genes

$$\mathbf{J} \in \mathbb{R}^{|\mathcal{C}| \times |\mathcal{C}|} \leftarrow \text{jaccard_distance}(\mathbf{N}_{G_{\text{gini}} \mathcal{C}})$$

$$\mathbf{cl} = \text{DBSCAN}(\mathbf{J})$$

return $\mathbf{cl}$

2.3.3 Rareness scores

The development of rare cell identification methods has been driven by scalability, i.e. the development of methods capable of handling large number of cells. This prompted the definition of rareness score methods. These approaches don't attempt to cluster data, but rather assign a score to each cell, indicating how unique this cell is with respect to the other ones in the data, which can be used for subsequent manual analysis.

FiRE

The FiRE method (Jindal et al., 2018) was designed as a way to estimate the rareness of a cell directly from the data, bypassing the clustering step. Indeed, the latter is often sensitive to various parameters and tends to perform poorly when the density of cells varies. Moreover, determining the appropriate level of granularity for clustering can be challenging, often necessitating the use of multi-level clustering to avoid overlooking minor clusters on the first pass (since other major cell types influence the expression variance in the data).

FiRE was designed based on the observation that rareness estimation of a particular cell is the opposite of measuring the density around it. FiRE employs the *sketching technique* (Lv et al., 2006; Wang et al., 2007), which encodes high-dimensional data points (i.e. the cells) into compact bit vectors (figure 2.7; algorithm 2.5). First, FiRE selects the 1000 most variable genes, based on their relative dispersion (i.e. the ratio between variance and mean). Sketches of size m are then generated for each cell by randomly selecting m of these genes and applying random thresholds to them, converting their expression into binary values. A weight vector is generated for each cell, and the dot product between this weight vector and the cell's sketch is used to map the cell to a predetermined hash code. The Hamming distance between sketches approximates the L_1 distance between the original high-dimensional data points.

The cells are then grouped by their hash code. These hash codes can be thought of as buckets that group cells together if they are close in the original high-dimensional space, i.e. have similar gene expression patterns. Indeed, two cells share the same hash code if they express the randomly selected genes in a similar way (according to the randomly chosen thresholds). A high score is given to cells in scarcely populated groups, and a low score to cells that are part of large groups. The density estimate for cell j (for a particular random run n) is

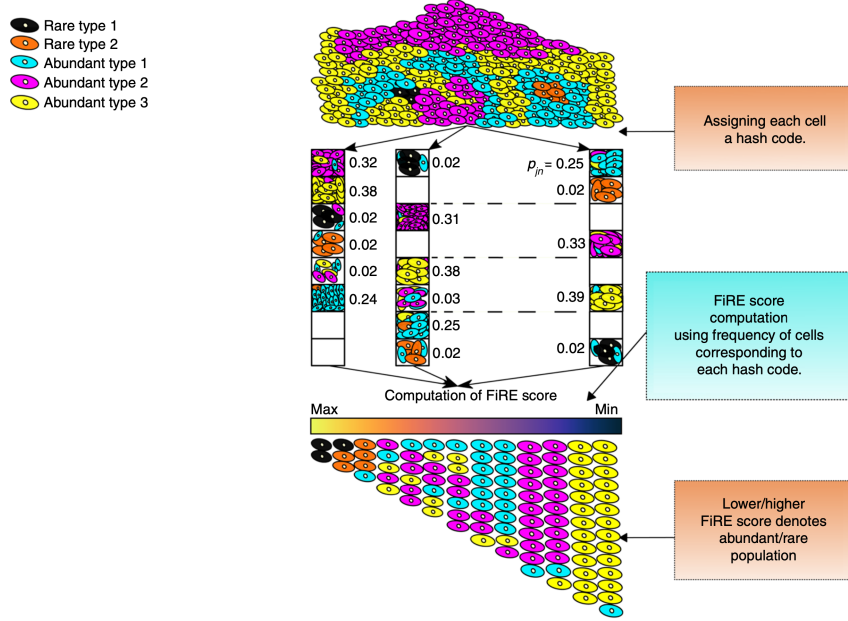


Figure 2.7: Overview of FiRE (Jindal et al., 2018). During several successive runs, each cell is assigned to a bucket based on its expression of a random subset of the genes. A probability value is assigned to each bucket based on the number of cells it contains. These probabilities are then combined to obtain a rareness score for each cell.

computed as:

$$p_{jn} = \frac{\text{Nb. of cell with same hash code as } j}{\text{Total number of cells}} \quad (2.1)$$

Several such runs (i.e. with new randomly chosen genes and thresholds) are aggregated to reduce the variance and produce a continuous rareness score, which should be high for rare cells:

$$r_j = -2 \sum_{n=1}^{n_r} \ln(p_{jn}) \quad (2.2)$$

This continuous score gives users the freedom to decide the degree of rareness of the cells to be further investigated. Yet, FiRE labels a cell j as rare if:

$$r_j \geq q_3(\mathbf{r}) + 1.5 \cdot IQR(\mathbf{r}) \quad (2.3)$$

where q_3 denotes the third quartile of the distribution and IQR the interquartile range.

Algorithm 2.5: FiRE

Input: $\mathbf{L} \in \mathbb{R}_+^{1000 \times |\mathcal{C}|}$ a log-normalized gene expression matrix containing the 1000 most variable genes

Input: $n_r \in \mathbb{N}$, $m \in \mathbb{N}$, $h \in \mathbb{P}$ the number of runs, the number of genes to randomly select in each run, and the hash table size (prime number)

Output: $\mathbf{r} \in \mathbb{R}^{|\mathcal{C}|}$ a vector containing the rareness score of each cell

for $n \in \{1, \dots, n_r\}$ **do**

- $G \leftarrow \text{random}(\text{range}=\{1, \dots, 1000\}, \text{size}=m)$
- $\mathbf{t} \leftarrow \text{random}(\text{range}=[\min \mathbf{L}, \max \mathbf{L}], \text{size}=m)$
- $\mathbf{w} \leftarrow \text{random}(\text{range}=\mathbb{R}, \text{size}=m)$
- for** $j \in \mathcal{C}$ **do**
 - for** $i \in G$ **do**
 - $\text{sketch}_i \leftarrow l_{ij} \geq t_i$
 - $\text{hash}_{jn} \leftarrow \text{sketch} \cdot \mathbf{w} \bmod h$

for $j \in \mathcal{C}$ **do**

- $r_j \leftarrow -2 \sum_{n=1}^{n_r} \ln \left(\frac{|\{j' \in \mathcal{C} \mid \text{hash}_{j'n} = \text{hash}_{jn}\}|}{|\mathcal{C}|} \right)$

return $\mathbf{r}$

DoRC

DoRC (Chen et al., 2019) provides an alternative using a similar approach, where the sketching is replaced by an *isolation forest* (Hariri and Kind, 2018), which proves effective to detect anomalies (here rare cells) in data (Susto et al., 2017). A tree is grown by splitting the data at each node according to randomly sampled genes (among the 1000 most variable ones). Cells located at a low depth are likely to be rare and are assigned a high score. As for FiRE, a continuous score is obtained by aggregation of several random runs.

Discussion

These rareness score methods have the advantage that they easily scale to large datasets, as they execute in linear time with respect to the number of cells. The continuous scores produced by FiRE and DoRC allow the identification of hierarchical structures, as one can vary the threshold to try to identify different groups. Yet, the cells passing the threshold (2.3) form a collection of rare cells, but not *one* rare subpopulation, as these cells might come from several different subpopulations. A subsequent manual analysis or clustering is therefore required to accurately identify the different rare cell types present in the data. This is also the case for biomarker identification, as the score gives no information of which genes might be differentially expressed in the rare cells. Indeed, the sketching technique and averaging of multiple runs make it difficult to tell on which genes the rareness estimation of the cell was based. One needs to rely on a univariate filter, such as the Wilcoxon's rank-sum test (Wilcoxon, 1945), applied *a posteriori* on the spotted cells to discover marker genes.

2.3.4 Deep learning clustering

The third category of rare cell identification methods use deep learning approaches to learn an embedding, which represents the data in lower dimensional spaces, in which clustering is performed. This allows to retain the full gene expression when reducing the matrix size, as opposed to the gene selection typically performed by clustering algorithms. Many such deep learning-based clustering methods have been proposed to cluster scRNA-seq data (Lei et al., 2023), but only few are designed to pay attention to the presence of rare cell types.

scAIDE

scAIDE (Xie et al., 2020) is an embedding method that has proven effective at identifying rare cells by paying attention to the expected imbalance of clusters in scRNA-seq data. It is composed of two components (algorithm 2.6), namely AIDE for dimensionality reduction and RPH-kmeans for clustering.

The *autoencoder-imputed distance-preserved embedding* (AIDE) (Xie et al., 2020) is designed to reduce the dimensionality of a scRNA-seq dataset (typically to 256 dimensions), overcoming the high level of noise present in such data. It consists of two main modules: an imputation module and a dimension reduction module. In the first, AIDE uses an *autoencoder* to correct noise in the data, such as dropout events. It captures the underlying data structure in the hidden layer, and generates imputed expression vectors. Dropout layers are added to prevent overfitting. Specifically, given a log-normalized scRNA-seq data matrix $\mathbf{L}$, the autoencoder learns a function:

$$\hat{\mathbf{l}}_j = f_{\text{AE}}(\mathbf{l}_j, \Theta_{\text{AE}}) \quad (2.4)$$

where Θ_{AE} represents the weights of the autoencoder. The *reconstruction loss* is defined as:

$$L_{\text{rec}} = \frac{1}{|\mathcal{C}|^2 - |\mathcal{C}|} \sum_{j \in \mathcal{C}} \sum_{\substack{j' \in \mathcal{C} \\ j < j'}} \left(\|\mathbf{l}_j - \hat{\mathbf{l}}_j\|_2^2 + \|\mathbf{l}_{j'} - \hat{\mathbf{l}}_{j'}\|_2^2 \right) \quad (2.5)$$

To make the imputed data suitable for Euclidean-based clustering methods, AIDE employs a *multidimensional scaling encoder* (Cox and Cox, 2000) in the dimension reduction module. This encoder transforms the imputed expression vectors into a low-dimensional representation while preserving the distance between them. Formally, the MDS encoder learns a function:

$$\mathbf{d}_j = f_{\text{MDS}}(\hat{\mathbf{l}}_j, \Theta_{\text{MDS}}) \quad (2.6)$$

The corresponding loss is defined as:

$$L_{\text{MDS}} = \frac{2}{|\mathcal{C}|^2 - |\mathcal{C}|} \sum_{j \in \mathcal{C}} \sum_{\substack{j' \in \mathcal{C} \\ j < j'}} \left| \|\mathbf{d}_j - \mathbf{d}_{j'}\|_2^2 - \|\hat{\mathbf{l}}_j - \hat{\mathbf{l}}_{j'}\|_2^2 \right| \quad (2.7)$$

The training of AIDE involves two stages: a pre-training and a joint tuning. In the first phase, the parameters of the autoencoder are optimized to minimize

Algorithm 2.6: scAIDE

Input: $\mathbf{L} \in \mathbb{R}_+^{|\mathcal{G}| \times |\mathcal{C}|}$ a log-normalized gene expression matrix
Input: $k \in \mathbb{N}$ the number of clusters
Input: $\alpha \in \mathbb{R}_+^*$ a balancing constant for the losses
Input: $n_r \in \mathbb{N}$ the number of runs of RPH-kmeans
Input: $q \in \mathbb{R}$, $n_h \in \mathbb{N}$, $n_s \in \mathbb{N}$, $n_i \in \mathbb{N}$ the parameters for the RPH center initialization
Output: $\mathbf{cl} \in \mathbb{N}^{|\mathcal{C}|}$ the cluster partition of the cell

```

// 1) AIDE
 $\Theta_{\text{AE}} \leftarrow \underset{\Theta_{\text{AE}}}{\operatorname{argmin}} L_{\text{rec}}(\mathbf{L}, \Theta_{\text{AE}})$ 
 $\Theta_{\text{AE}}, \Theta_{\text{MDS}} \leftarrow \underset{\Theta_{\text{AE}}, \Theta_{\text{MDS}}}{\operatorname{argmin}} \left( L_{\text{rec}}(\mathbf{L}, \Theta_{\text{AE}}) + \alpha L_{\text{MDS}}(\mathbf{L}, \Theta_{\text{AE}}, \Theta_{\text{MDS}}) \right)$ 
 $\mathbf{D} \in \mathbb{R}^{256 \times |\mathcal{C}|} \leftarrow f_{\text{MDS}}(f_{\text{AE}}(\mathbf{L}, \Theta_{\text{AE}}), \Theta_{\text{MDS}})$ 

// 2) RPH-kmeans
 $\mathbf{cl} \leftarrow \mathbf{0}$ ;  $\varphi \leftarrow 0$ 
for  $n \in \{1, \dots, n_r\}$  do
   $\{\mathbf{c}_x\}_{x=1}^k \leftarrow \text{RPH\_centers}(\mathbf{D}, k, q, n_h, n_s, n_i)$  // Algo. 2.7
   $\mathbf{cl}_n \leftarrow \text{kmeans}(\mathbf{D}, \{\mathbf{c}_x\}, k)$ 
   $\varphi_n \leftarrow \sum_{x=1}^k \sum_{\substack{j \in \mathcal{C} \\ \mathbf{cl}_{nj} = x}} \|\mathbf{d}_j - \mathbf{c}_x\|^2$  // Inertia
  if  $\varphi_n > \varphi$  then  $\mathbf{cl} \leftarrow \mathbf{cl}_n$ ;  $\varphi \leftarrow \varphi_n$ 
return  $\mathbf{cl}$ 
  
```

the reconstruction loss of the imputed data. In the second phase, both the autoencoder and multidimensional scaling encoder are trained together by minimizing a combined loss function that balances the reconstruction loss and the distance preservation loss.

The obtained data is then clustered using *random projection hashing-based k-means clustering* (RPH-kmeans) (Xie et al., 2020). It is designed to improve *k*-means for data with imbalanced cluster sizes, which is typically the case for scRNA-seq data. Indeed, traditional *k*-means is sensitive to initial cluster centers, which often results in cluster centers being biased towards larger cell populations, neglecting smaller ones. RPH-kmeans therefore initializes cluster centers using a *locality-sensitive hashing* (LSH) method (Gionis et al., 1999). It projects data points into a lower-dimensional space, using a *random projection-based hashing* (RPH), where similar points are more likely to be hashed to the same bucket.

The RPH function maps the data point $\mathbf{d}_j$ to an integer by generating a random vector $\mathbf{a} \in \mathbb{R}^{|\mathcal{G}|}$ from the $N(0, 1)$ distribution and a constant $b \in \mathbb{R}$ from $U(0, q)$ (where $q \in \mathbb{R}$ is the quantization step):

$$h(\mathbf{d}_j, q) = \left\lfloor \frac{\mathbf{a} \cdot \mathbf{d}_j + b}{q} \right\rfloor \quad (2.8)$$

A composite hash function is constructed by combining n_h hash functions:

$$g(\mathbf{d}_j, q) = (h_1(\mathbf{d}_j, q) \cdots h_{n_h}(\mathbf{d}_j, q)) \quad (2.9)$$

The data points are then grouped into buckets based on their hash vector, and all points within a bucket are merged into one composite data point. This process is repeated several times to produce a data skeleton with much fewer points.

Subsequently, weighted *k*-means with *k*-means++ initialization (Arthur and Vassilvitskii, 2007) is applied to the skeleton to generate initial cluster centers for RPH-kmeans. This significantly reduces the number of points in larger clusters, mitigating the bias caused by data imbalance. Finally, several runs of *k*-means clustering are performed using these cluster centers, and the partition with lowest inertia is returned as the final scAIDE clustering.

DeepScena

DeepScena (Lei et al., 2023) is a recently proposed deep learning clustering method designed for rare cell identification. Briefly, unlike scAIDE which learns

Algorithm 2.7: scAIDE – RPH_centers

Input: $\mathbf{D} \in \mathbb{R}^{256 \times |\mathcal{C}|}$ a low dimension representation of gene expression data

Input: $k \in \mathbb{N}$ the number of clusters

Input: $q \in \mathbb{R}$, $n_h \in \mathbb{N}$ the quantization step and the number of LSH functions to use

Input: $n_s \in \mathbb{N}$, $n_i \in \mathbb{N}$ the maximum number of of skeleton points, and of iterations

Output: $\{\mathbf{c}_x \in \mathbb{R}^{256}\}_{x=1}^k$ the cluster centers

$\mathbf{S} \in \mathbb{R}^{256 \times |C_s|} \leftarrow \mathbf{D}$; $t \leftarrow 0$

while $t < n_i$ **and** $C_s > n_s$ **do**

for $j \in C_s$ **do** $\text{hash}_j \leftarrow g(\mathbf{s}_j, q)$

for $j \in C_s$ **do**

$B \leftarrow \{j' \in C_s \mid \text{hash}_{j'} = \text{hash}_j\}$ // Bucket

$\mathbf{c} \leftarrow \frac{1}{|B|} \sum_{j' \in B} \mathbf{s}_{j'}$

$\mathbf{S} \leftarrow \begin{pmatrix} \mathbf{S}_{C_s \setminus B} & \mathbf{c} \end{pmatrix}$ // Replace cells by composite

$t \leftarrow t + 1$

$\{\mathbf{c}_x\} \leftarrow \text{kmeans}(\mathbf{S}, k)$

return $\{\mathbf{c}_x\}$

the lower-dimensional representation of cells separately from the clustering process, DeepScena integrates data imputation, dimensionality reduction, and hierarchical clustering directly into its deep-learning framework. It uses a negative binomial-based autoencoder to denoising the data, perform dimensionality reduction and a preliminary clustering.

Next, DeepScena performs a deep learning clustering with a self-supervision model (Sadeghi and Armanfard, 2023), i.e. a fully connected neural network which uses pairwise cell similarities in a reliable subspace to self-supervise the training procedure. Specifically, each cell is mapped within a k -dimensional space (k being the number of clusters), where the inner product is used for measuring cell-to-cell similarity. The self-supervision model strengthens similarities between two similar cells, and weakens it between two dissimilar ones. DeepScena performs a hierarchical clustering using a top-down loop that iteratively applies two modules to different levels of clusters.

Discussion

The use of autoencoders has the advantage of creating a lower dimensional space capturing the full gene expression spectrum, which reduces the curse of dimensionality (as the number of cells will often be larger than the dimensions used during the clustering) while minimizing loss of information. Yet, this comes at a high computational cost, as training such models may require hours of computations on supercomputers, even for relatively small datasets. Moreover, such deep learning models are harder to interpret as they are highly non-linear. In particular, learning new dimensions means one loses the interpretability in terms of genes and cells, requiring a separate differential expression analysis to identify marker genes for the different clusters.

2.3.5 A new approach

As illustrated in the previous sections, identifying rare cells with corresponding marker genes in scRNA-seq data is often done using methods that are univariate and/or perform a separate identification of cell types and marker genes. While they have been effective at identifying rare cell types in many situations, these methods could miss interesting patterns, specifically when multiple genes are co-expressed. This motivates the development of a multivariate approach to identify rare subpopulations of cells.

Biclustering is a multivariate approach that has been frequently used to analyze biological data (Madeira and Oliveira, 2004; Xie et al., 2018). It consists in a joint identification of rows (representing the genes) and columns (representing samples or cells) from a data matrix which together optimize a specific criterion. In particular, the *max-sum submatrix* method (Branders et al., 2019) has proven effective to identify gene-specific subgroups from bulk gene expression data, thanks to its global sum criterion which is robust to technical and biological variability (see chapter 3).

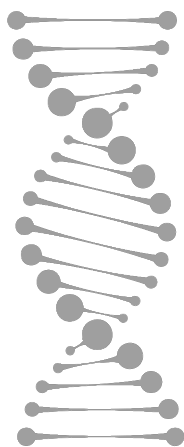
Using biclustering to identify rare cell types is not straightforward, as these methods tend to maximize the number of cells and genes included in the result, contrasting with the objective of identifying *small* subpopulations within the data. A biclustering approach specifically designed to search for *rare* and *highly specific* patterns of expressions within small subpopulations of cells is therefore presented in chapter 4.

The ability to easily detect rare cell types may depend on the availability of cells of the considered tissue. In large-scale datasets of tens or hundreds of thousands of cells, for instance generated by 10X for the purpose of general characterization of the different cell types within a generally highly heterogeneous tissue, identifying rare cells is relatively easy. Indeed, even very rare cell types of less than 1% of the data could be represented by hundreds of individuals, easing their detection (Wang et al., 2021). Yet, such data might not always be available, due for instance to the limited availability of material (e.g. intra-tumoral Tregs), or the choice to use full-length sequencing technologies which have a better gene detection rate (but are more expensive and have lower throughput). Rare cells are much more difficult to spot in such a setting, motivating the use of a multivariate biclustering approach.

The primary focus of this approach is thus the search for rare subpopulations in small to middle-scale data, as detailed in chapters 4 and 5. Yet, chapter 6 also proposes a hybrid approach for the identification of rare cells in large-scale data. Chapter 7 further motivates the relevance of a multivariate approach by showing its robustness to batch effects, allowing the detection of rare subpopulations that are present across multiple single-cell samples. Finally, chapter 8 shows an application of the proposed approaches to multi-modal data.

A global sum criterion

3



The end of the previous chapter shows that existing rare cell identification methods generally rely on univariate approaches, and/or separate identification of rare cells and corresponding marker genes. This chapter exposes the global sum criterion, which can be used to perform a joint and multivariate mining of cells and genes, and constitutes the common thread of the methods developed during this thesis.

Section 3.1 exposes the maximal sum submatrix problem, as it has been proposed in the context of identifying gene-specific subgroups from bulk gene expression data. Section 3.2 shows how this principle can be applied to single-cell RNA-seq data, and why it is well suited to such data.

3.1 The maximal sum submatrix problem

This section presents the max-sum submatrix problem, and as such summarizes part of the work of [Branders \(2021\)](#).

3.1.1 Definition

The problem of identifying a submatrix of maximal sum has been proposed by [Branders et al. \(2019\)](#) as an alternative to traditional biclustering methods to identify gene-specific subgroups within bulk gene expression data. Indeed, a frequent objective is to group genes according to their expression under certain conditions, or to group conditions based on the expression of a number of genes. This is commonly achieved through *biclustering*, i.e. the task of identifying spe-

cific subsets of rows (i.e. genes) and of columns (i.e. samples or conditions) which jointly form homogeneous entries (Madeira and Oliveira, 2004; Xie et al., 2018).

Biclustering is typically applied on a data matrix $\mathbf{M}$, where an entry m_{ij} represents the value of a specific row i (e.g. a gene) obtained for a specific column j (e.g. a sample). A *bicluster* is a submatrix of $\mathbf{M}$ defined by a subset of *selected rows* and a subset of *selected columns* (which need not be continuous in the original matrix). Traditional biclustering algorithms typically identify biclusters sharing similar expression levels, for instance by minimizing the variance across the selected genes and samples. Such an approach assumes a homogeneous distribution of the expression values across the genes/samples combinations to be identified. However, gene expression data is expected to contain some outlier values, which could make the identification of some relevant biclusters difficult.

As an alternative, the *max-sum submatrix* problem identifies subsets of genes and of samples with globally high values, i.e. genes that are highly expressed within a set of samples. This constitutes a generalization of the *maximal rank tiling* problem, where one looks for a maximal subset within a matrix where entries correspond to a discrete ranks (Le Van et al., 2016, 2014). While this problem is only suitable for data where each observation (e.g. each row) corresponds to a certain ranking, the max-sum submatrix problem can be applied on matrices without any constraints between values, which is the case for gene expression data.

Mathematically, this problem takes as input a matrix whose m_{ij} entries are assumed positive only if gene i is expressed in sample j , and negative otherwise. For gene expression data, this requires to choose a threshold on the original values to specify a minimal expression level above which a gene is considered expressed, and to scale the data accordingly. One looks for the subset of genes $I^* \subseteq \mathcal{G}$ and the subset of samples $J^* \subseteq \mathcal{C}$ such that the sum of the corresponding entries in $\mathbf{M}$ is higher than any other (I, J) bicluster:

$$(I^*, J^*) = \operatorname{argmax}_{\substack{I \subseteq \mathcal{G} \\ J \subseteq \mathcal{C}}} \sum_{i \in I} \sum_{j \in J} m_{ij} \quad (3.1)$$

As such, the result is less influenced by the presence of some outliers than traditional biclustering methods, as values are never compared in a pairwise fashion. Figure 3.1 shows the result of the max-sum submatrix on a toy gene expression dataset. The identified bicluster can include a few outliers in terms of high expression levels. For instance, the $m_{4,4} = -4.1$ entry is included in the optimal solution because this low value is compensated along its row and column by

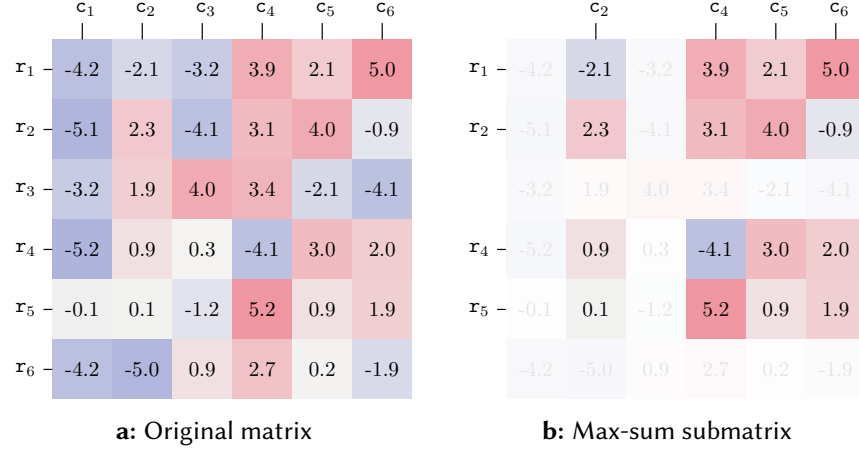


Figure 3.1: Toy example of an expression matrix (Branders, 2021). **a** The instance matrix M . **b** The submatrix of maximal sum within M , corresponding to bicluster $(\{r_1, r_2, r_4, r_5\}, \{c_2, c_4, c_5, c_6\})$. It has a global sum of 27.3, which is the highest value that can be obtained.

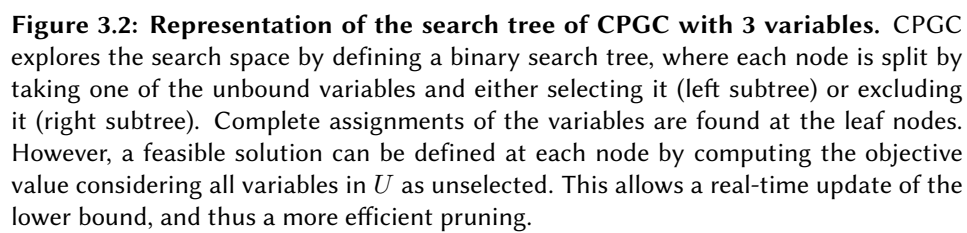
other positive values. Hence all selected rows and selected columns contribute positively to the objective function.

3.1.2 CP Global Constraint

Despite its NP-hard nature,¹ this problem has been efficiently solved using the CPGC algorithm (Branders, 2021; Branders et al., 2018). It uses a constraint programming approach which relies on a *global constraint* that captures the relations between the rows and columns and provides a bound on the sum of covered entries. It uses of a *branch-and-bound* approach (Land and Doig, 1960) with an efficient bounding procedure that allows to drastically filter the domains.

CPGC (see algorithm 3.1) operates on an implicit search space, where only the columns $\mathcal{C}$ are variables, as the optimal assignment of rows (i.e. genes) given

¹The max-sum submatrix can be solved in polynomial time in some specific cases, for instance when all the rows and columns in the bicluster are contiguous (Bentley, 1984). Yet, the max-sum submatrix problem has been shown NP-hard in the general case (Branders, 2021), by a reduction to the maximum edge weight biclique problem (Dawande et al., 1996).



Algorithm 3.1: CPGC

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ a gene expression matrix where positive values represent expression and negative values absence of (or negligible) expression

Output: $I^* \subseteq \mathcal{G}$, $J^* \subseteq \mathcal{C}$, $\Omega^* \in \mathbb{R}$ sets of genes and cells maximizing problem (3.1), and the corresponding objective value

$I^*, J^*, \Omega^* \leftarrow \emptyset, \emptyset, 0$

procedure $\text{dfs}(J^0 \subseteq \mathcal{C}, U \subseteq \mathcal{C}, P \subseteq \mathcal{G}, \mathbf{psums} \in \mathbb{R}^{|\mathcal{G}|})$:

if $U \neq \emptyset$ **then**

$j \leftarrow \text{random}(U)$

 // Evaluate partial solution

$J \leftarrow J^0 \cup \{j\}$

$U \leftarrow U \setminus \{j\}$

$I, \Omega, \mathbf{psums}_{\text{new}} \leftarrow \text{obj}(\mathbf{M}, j, P, \mathbf{psums})$ // Algo. 3.2

if $\Omega \geq \Omega^*$ **then** $I^*, J^*, \Omega^* \leftarrow I, J, \Omega$

 // Subtree with j included

$ub, \mathbf{ubg}, P \leftarrow \text{upper_bound}(\mathbf{M}, U, P, \mathbf{psums}_{\text{new}})$ // 3.3

if $ub > \Omega^*$ **then**

$J', U', \mathbf{psums}' \leftarrow \text{filter}(J, U, P, \mathbf{ubg}, \mathbf{psums}_{\text{new}}, \Omega^*)$

$\text{dfs}(J', U', P, \mathbf{psums}')$ // $\hookrightarrow$ 3.4

 // Subtree with j excluded

$ub, \mathbf{ubg}, P \leftarrow \text{upper_bound}(\mathbf{M}, U, P, \mathbf{psums})$ // 3.3

if $ub > \Omega^*$ **then**

$J', U', \mathbf{psums}' \leftarrow \text{filter}(J^0, U, P, \mathbf{ubg}, \mathbf{psums}, \Omega^*)$

$\text{dfs}(J', U', P, \mathbf{psums}')$ // $\hookrightarrow$ 3.4

$\text{dfs}(\emptyset, \mathcal{C}, \mathcal{G}, \mathbf{0})$

return I^*, J^*, Ω^*

an assignment of columns $J \subseteq \mathcal{C}$ can be found in linear time:

$$I_J^* = \left\{ i \in \mathcal{G} \mid \sum_{j \in J} m_{ij} \geq 0 \right\} \quad (3.2)$$

CPGC explores the search space by maintaining a set $U \subseteq \mathcal{C}$ of variables that are *unbound*, i.e. are neither included nor excluded from J at the current node in the search tree, and can consequently still be added to J in the subtree under the current node (figure 3.2). CPGC maintains a feasible solution at each node of the search tree by considering all unbound variables as excluded from the (I, J) -bicluster, which allows a real-time update of the lower bound (i.e. the best solution found so far). The tree is further explored by selecting a variable $u \in U$, and creating two children nodes: one where u is added to J , and the other where u is excluded from the solution.

The computation of the objective value is done efficiently by maintaining a partial sum $psums_i$ for each gene i , which contains the sum of its expression

Algorithm 3.2: CPGC – obj

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ a gene expression matrix

Input: $j \in \mathcal{C}$ the variable that is added to the current solution

Input: $P \subseteq \mathcal{G}$ the candidate genes

Input: $\mathbf{psums} \in \mathbb{R}^{|\mathcal{G}|}$ the partial sums of the genes

Output: $I \subseteq P$, $\Omega \in \mathbb{R}$, $\mathbf{psums} \in \mathbb{R}^{|\mathcal{G}|}$ the set of genes maximizing problem (3.1) given the current assignment of the variables, the corresponding objective value, and the updated partial sums

$I \leftarrow \emptyset$; $\Omega \leftarrow 0$

for $i \in P$ **do**

$psums_i \leftarrow psums_i + m_{ij}$

if $psums_i \geq 0$ **then**

$I \leftarrow I \cup \{p\}$

$\Omega \leftarrow \Omega + psums_i$

return I , Ω , $\mathbf{psums}$

values within the selected variables. Computing equation (3.2) when adding a variable j simply amounts to adding its expression value m_{ij} to the partial sum, rather than computing the whole sum from scratch (algorithm 3.2). These partial sums are implemented in practice using reversible variables, which maintain previous states on a stack, enabling them to return to a previous value when backtracking in the search.

CPGC computes an upper bound for the max-sum submatrix solution by summing the entries for the selected variables, and adding any positive expression of the unbound variables, as they can still potentially be added to the current solution. Let $P \subseteq \mathcal{G}$ be the set genes that can still possibly contribute positively to the objective. The upper bound is:

$$ub(J, U) = \sum_{i \in P} \max \left\{ 0, \sum_{j \in J} m_{ij} + \sum_{u \in U} \max\{0, m_{iu}\} \right\} \quad (3.3)$$

This allows to prune the search tree under a node whenever this upper bound is smaller than the best solution found so far. Computing this upper bound (al-

Algorithm 3.3: CPGC – upper_bound

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ a gene expression matrix

Input: $U \subseteq \mathcal{C}$, $P \subseteq \mathcal{G}$ the unbound variables and candidate genes

Input: $\mathbf{psums} \in \mathbb{R}^{|\mathcal{G}|}$ the partial sums of the genes

Output: $ub \in \mathbb{R}$, $\mathbf{ubg} \in \mathbb{R}^{|\mathcal{G}|}$, $P_{\text{new}} \subseteq P$ the upper bound given the current assignment of the variables, the upper bound values of each gene, and the updated set of candidate genes

$ub \leftarrow 0$; $P_{\text{new}} \leftarrow P$

for $i \in P$ **do**

$ubg_i \leftarrow psums_i$

for $u \in U$ **do** $ubg_i \leftarrow ubg_i + \max\{0, m_{iu}\}$

if $ubg_i > 0$ **then** $ub \leftarrow ub + ubg_i$

else $P_{\text{new}} \leftarrow P_{\text{new}} \setminus \{i\}$

return $ub, \mathbf{ubg}, P_{\text{new}}$

gorithm 3.3) also allows to identify genes that can no longer be included in the solution, as they would have a negative impact on the objective value even in the most optimistic case. They can consequently be removed from P .

Finally, CPGC uses an efficient bounding procedure that allows to drastically filter the variables (algorithm 3.4). It computes, for each unbound variable u individually, the upper bounds that are obtained when including or excluding u . Whenever the bound when including u is lower than the best solution found so far, one knows there is no point in selecting it and u can be removed from U . Similarly, if the bound when excluding u is lower than the best solution found so far, one must necessarily add u to J to have a chance of finding a better solution in the subtree. This strategy allows to drastically prune the search tree, and enables CPGC to efficiently find a solution to the max-sum submatrix problem.

3.1.3 Comparison to traditional biclustering

Branders et al. (2019) show the max-sum submatrix method outperforms typical biclustering algorithms to identify biologically relevant gene/sample biclusters from DNA microarray datasets. CPGC has been compared to the *Cheng and Church algorithm* (CCA), xMOTIFs, the *iterative signature algorithm* (ISA), QUBIC, Plaid and Spectral. These approaches generally rely in some way on pairwise comparisons of values, as opposed to the global max-sum approach.

CCA (Cheng and Church, 2000) is based on iteratively adding or removing rows and columns to the current assignment in order to minimize the variance within the bicluster. xMOTIFs (Murali and Kasif, 2002) finds biclusters in a discretized matrix by looking for simultaneously conserved genes in subsets of samples, i.e. for which the discretized values are identical. ISA (Bergmann et al., 2003) greedily adds or removes columns from a randomly selected bicluster until some minimal average value is reached across the selected columns and rows. QUBIC (Li et al., 2009) builds a graph from a discretized matrix, where each node corresponds to a gene and the weights of the edges represent the number of samples for which two genes have the same nonzero discretized value. It then searches for heavy subgraphs to identify biclusters. Plaid (Lazzeroni and Owen, 2002) fits a generative statistical model, from which all entries of the original matrix are assumed to have been generated, by minimizing the mean squared error between this modeled and the actual data. Spectral (Kluger et al., 2003) relies on singular value decomposition to cluster genes and samples simultaneously, looking for distinctive patterns such as contiguous rows and contiguous columns.

Algorithm 3.4: CPGC – filter

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ a gene expression matrix

Input: $J \subseteq \mathcal{C}$, $U \subseteq \mathcal{C}$, $P \subseteq \mathcal{G}$ the current assignments

Input: $\text{ubg} \in \mathbb{R}^{|\mathcal{G}|}$, $\text{psums} \in \mathbb{R}^{|\mathcal{G}|}$ the upper bounds of the genes, and the partial sums of the genes

Input: $\Omega^* \in \mathbb{R}$ the current best objective value

Output: $J \subseteq \mathcal{C}$, $U \subseteq \mathcal{C}$, $\text{psums} \in \mathbb{R}^{|\mathcal{G}|}$ the updated variable assignments and partial sums

for $u \in U$ **do**

$\text{ub}_u^\inleftarrow 0$ // upper bound if u was selected

$\text{ub}_u^\notinleftarrow 0$ // upper bound if u was excluded

for $i \in P$ **do**

if $m_{iu} > 0$ **then**

if $\text{ubg}_i - m_{iu} > 0$ **then** $\text{ub}_u^\notinleftarrow \text{ub}_u^\notin + \text{ubg}_i - m_{iu}$

$\text{ub}_u^\inleftarrow \text{ub}_u^\in + \text{ubg}_i$

else if $m_{iu} < 0$ **then**

if $\text{ubg}_i + m_{iu} > 0$ **then** $\text{ub}_u^\inleftarrow \text{ub}_u^\in + \text{ubg}_i + m_{iu}$

$\text{ub}_u^\notinleftarrow \text{ub}_u^\notin + \text{ubg}_i$

if $\text{ub}_u^\in \leq \Omega^*$ **then**

$U \leftarrow U \setminus \{u\}$

else if $\text{ub}_u^\notin \leq \Omega^*$ **then**

$U \leftarrow U \setminus \{u\}$

$J \leftarrow J \cup \{u\}$

$\text{psums} \leftarrow \text{psums} + \mathbf{m}_u$

return J, U

The max-sum submatrix method has been compared to these algorithms on several DNA microarray datasets (Branders et al., 2019). A Gene Ontology (GO) analysis (Ashburner et al., 2000; The Gene Ontology Consortium, 2019) performed on the different results shows that the max-sum submatrix method returns gene sets associated with more significantly enriched GO terms than its alternatives. This confirms the relevance of the global sum criterion to analyze gene expression data, as opposed to approaches relying on pairwise comparisons of values.

3.2 Application to scRNA-seq data

The max-sum submatrix method has been defined in the context of identifying gene-specific subgroups within bulk expression data. Yet, as the optimization problem is completely generic, it could in principle be applied to single-cell expression data in order to find gene/cell biclusters, i.e. a subgroup of genes that are highly expressed in a subset of the cells. Given its robustness with respect to outliers, this global sum criterion is well suited to the high technical and biological variability of scRNA-seq data. However, this requires to first scale this data (which contains only positive values) in an appropriate way, such that an m_{ij} entry is assumed positive whenever the gene i is expressed in the cell j , and negative otherwise.

3.2.1 Data scaling

Applying this global sum criterion on scRNA-seq data requires to transform the raw read count matrix $\mathbf{R} \in \mathbb{N}^{|\mathcal{G}| \times |\mathcal{C}|}$, or its normalized version $\mathbf{N} \in \mathbb{R}_+^{|\mathcal{G}| \times |\mathcal{C}|}$, into a matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ with positive values (for gene expression) and negative values (for negligible or absence of expression). Given the high sparsity of single-cell data, a natural threshold is to consider only zero counts to be negative values. A straightforward approach could be to binarize the (normalized) read counts:

$$m_{ij} = \begin{cases} 1 & \text{if } n_{ij} > 0 \\ -1 & \text{if } n_{ij} = 0 \end{cases} \quad (3.4)$$

Alternatively, one can take advantage of the log-scaling that is commonly applied to scRNA-seq (normalized) count values. Indeed, by adapting the pseudo-count that is added to avoid the undefined $\log(0)$ case, one can introduce negative values for genes that are not (or marginally) expressed, while conserving the dif-

ferences in expression levels:

$$m_{ij} = \log_a \left(n_{ij} + \frac{1}{a} \right) \quad (3.5)$$

For instance, considering $m_{ij} = \log_{10}(n_{ij} + 0.1)$ defines a threshold of 0.9 on the normalized read counts. Values greater than 0.9 will remain positive after transformation, while zero counts will become negative (-1). This typically scales the data in a $[-1, 5]$ range, and will be used throughout this thesis.

Optimizing the sum of selected entries in the data matrix naturally leads to selecting *highly expressed* genes in the selected cells, whenever the data matrix represent the original scRNA-seq count values (or a log-normalized version of them). Yet, appropriate data transformation enables the identification of other kind of patterns, which can in particular be useful for genes expressed in a lot of cells. Considering the *opposite* of the original values would lead to select cells with *low* expressed genes after solving the same problem. Other data normalizations can also lead to other interesting patterns (e.g. the genes departing the most of the median expression values for each cell). Nevertheless, we stick here for clarity to the original interpretation with selected entries corresponding to jointly *high* expression values. This choice is also consistent with the nature of scRNA-seq data because of the dropout phenomenon (Ziegenhain et al., 2017) which could lead to many false positives whenever one looks for low expressed values.

3.2.2 Robustness w.r.t. scRNA-seq variability

The global sum criterion used in equation (3.1) is less influenced by the presence of some outliers than typical biclustering criterions, and therefore seems well suited to the high technical and biological variability of scRNA-seq data described in section 2.2.2. It allows for variations within the expression values, as shown in figure 3.1. An unusually high expression of a gene in a particular cell, e.g. due to transcriptional bursting, would have little influence on the bicluster found since expression values are not compared in a pairwise fashion. Indeed, cell c_4 has a much higher expression level for gene r_5 compared to the other selected cells, which also express this gene but at a lower level. Using another biclustering criterion, e.g. based on the variance, would likely fail to group these cells together.

Moreover, the sum criterion allows for some genes within the selected bicluster to be lowly, or even negatively, expressed for some cells as long as their inclusion increases the objective value globally, which is consistent with the dropout phenomenon. Indeed, the negative values in $m_{1,2}$, $m_{5,4}$, and $m_{2,6}$ are largely

compensated by the other entries of the bicluster, which prove that the four selected cells coexpress all four selected genes. In other words, the sum allows for exceptions within the selected bicluster as only the global influence of the selected cells and selected genes matters.

As such, the max-sum submatrix problem is not designed for *rare* subpopulation identification as the maximization tends to identify large subgroups. Figure 3.3 shows the submatrix of maximal sum within a dataset containing 45 CD8 T cells and 5 GARP+ Tregs from a human breast cancer sample (similar to the data used in section 2.3). It corresponds to a large bicluster containing 42 out of the 50 cells, and 1 879 genes.

In principle, it is possible to identify *small* biclusters using the max-sum approach by increasing the threshold to consider a gene as expressed (i.e. $m_{ij} > 0$ if $n_{ij} > \theta$, for some $\theta > 0$). Many previously positive values in $\mathbf{M}$ would now become negative, logically leading to smaller biclusters. Yet, such a situation would not be ideal to analyze scRNA-seq data, as the result would be dominated by values representing very high expression, some of which might actually be the result of transcriptional bursting. Genes that are detected at lower transcript levels would then plainly be ignored.

Moreover, the max-sum submatrix objective function (3.1) ignores how the selected genes are expressed outside of the bicluster. In other words, it doesn't look for patterns of *differential expression*. The principal contribution of this thesis is to extend the max-sum approach by refining the objective function to be optimized and by adding useful constraints to specifically search for *rare* and *highly specific* patterns of expressions within small subpopulations of cells.

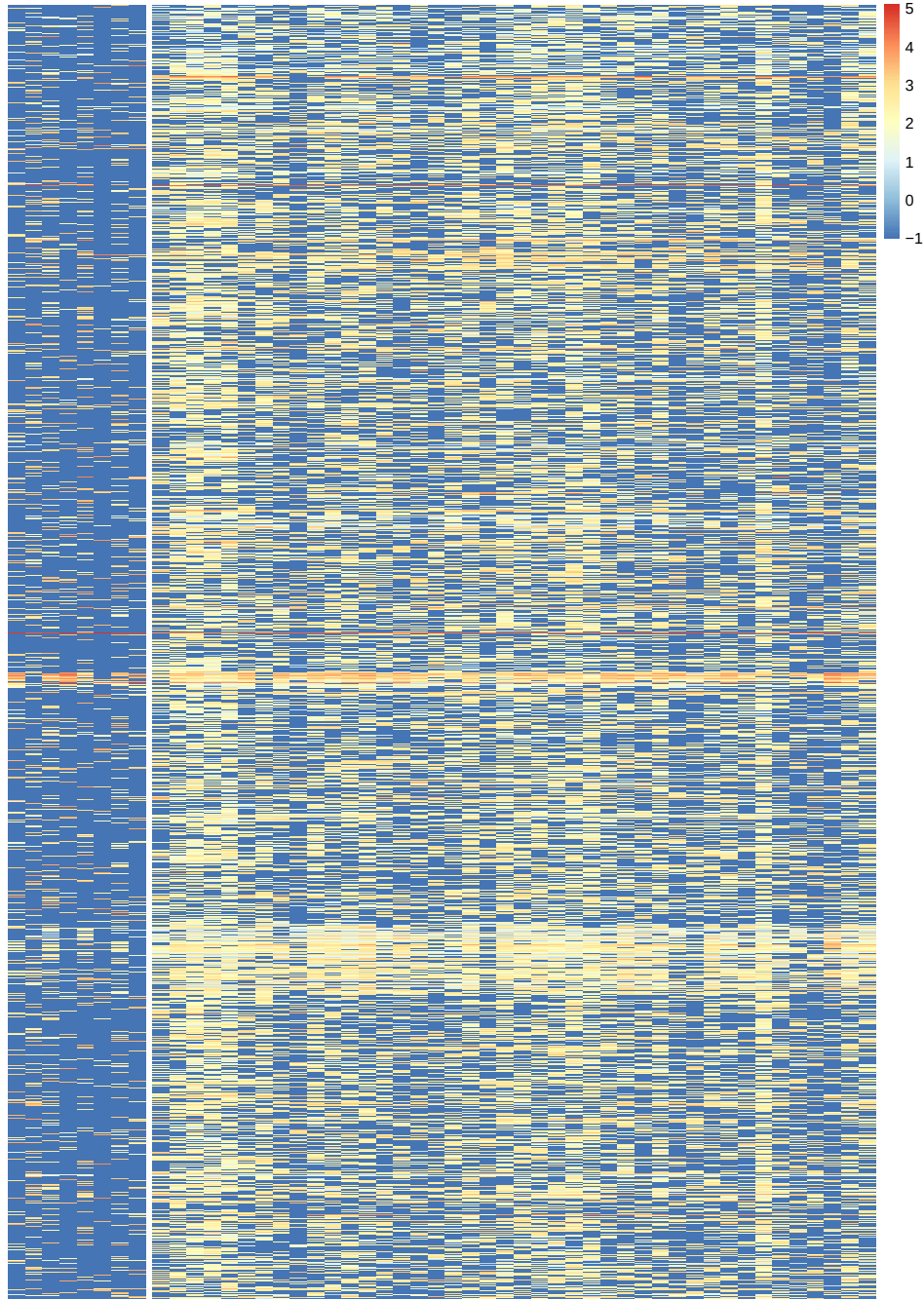
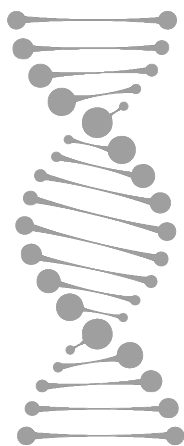


Figure 3.3: Submatrix of maximal sum on a dataset containing 45 CD8 T cells and 5 GARP+ Tregs. The bicluster is composed of 42 cells (on the right) and 1879 genes (the remaining genes are not displayed).

Mining rare and highly specific subpopulations

4



The previous chapter presents the max-sum submatrix problem, and shows the utility of this global sum criterion in light of the high variability of scRNA-seq data. Yet, it doesn't apply as such to the task of finding rare subpopulations of cells. This chapter extends this approach by refining the objective function to be optimized and by adding useful constraints to specifically search for rare and highly specific patterns of expressions within small subpopulations of cells.¹

Section 4.1 states the optimization problem of MicroCellClust. Section 4.2 compares the performance of MicroCellClust to alternative rare cell methods on two experiments where a rare cell subpopulation is artificially planted within an abundant cell type. Section 4.3 shows three case studies proving the ability of MicroCellClust to identify relevant rare subpopulations, which are biologically interpreted.

4.1 MicroCellClust definition

As such, the max-sum submatrix problem identifies a large gene/cell bicluster, whose value is only defined by the sum of the values inside the bicluster. Its objective function thus ignores what happens outside the bicluster, i.e. the expression of the selected genes in the cells that are not included in the bicluster (called *out-of-cluster* cells). If the latter also highly express a selected gene, it can't be seen as specific to the bicluster. Moreover, the unconstrained maximization of

¹*Related publication:* Gerniers, A., Bricard, O., and Dupont, P. (2021). MicroCellClust: mining rare and highly specific subpopulations from single-cell expression data. *Bioinformatics*, 37(19):3220–3227.

the max-sum submatrix method leads to the inclusions of many cells and genes, even though they might only be loosely related to the other cells and genes of the bicluster. We therefore introduce a new optimization problem to identify a bicluster (I^*, J^*) representative of a small subpopulation of cells with globally high values for a set of specific genes. Assuming again a matrix $\mathbf{M}$ whose values m_{ij} are positive whenever the gene i is expressed in the cell j , and negative otherwise (see section 3.2.1):

$$(I^*, J^*) = \operatorname{argmax}_{\substack{I \subseteq \mathcal{G} \\ J \subseteq \mathcal{C}}} \sum_{i \in I} \left(\sum_{j \in J} m_{ij} - \kappa \sum_{k \in \mathcal{C} \setminus J} \max\{0, m_{ik}\} \right) \quad (4.1)$$

$$\text{such that } \frac{|\{(i, j) \mid i \in I, j \in J, m_{ij} < 0\}|}{|I| \cdot |J|} \leq \mu \quad (4.2)$$

4.1.1 Objective function

The objective function in equation (4.1) is composed of two terms. Maximizing the first term, i.e. $\sum_{i \in I} \sum_{j \in J} m_{ij}$, corresponds to the max-sum submatrix objective, i.e. a bicluster for which the sum of all the corresponding expression values is maximal. The global sum criterion allows for variations within the expression values, which is well suited to the variability of scRNA-seq data (see section 3.2.2).

As such, the max-sum objective does not guarantee the solution to be highly specific. In other words, the genes selected within the bicluster could also be highly expressed in other cells. The second term of equation (4.1) prevents such situation by penalizing positive expression of these genes (hence the $\max\{0, \cdot\}$ operation) in out-of-cluster cells ($k \in \mathcal{C} \setminus J$), in order to find a group of genes which are differentially expressed in the selected cells.

The parameter $\kappa \in \mathbb{R}_+$ controls the relative influence of these two terms within the objective function, i.e. the strength of the out-of-cluster expression penalty. The higher κ , the less out-of-cluster expression is tolerated and only the most differential expressed genes are selected. This naturally leads to the inclusion of fewer genes in the solution. The κ parameter can thus be used as a way to monitor the size of the selected geneset. Selecting 10 to 30 genes is often relevant to perform a biological interpretation of the result. We suggest $\kappa = \frac{100}{|\mathcal{C}|}$ as default value. The results in section 4.3 include detailed analyses of the influence of this parameter.

Data preprocessing

The objective function (4.1) requires the expression matrix to be sparse. Indeed, many genes could be expressed in nearly all cells. Such a situation would lead to an optimal solution being a large bicluster that would no longer be specific to a small population of cells. In the limit, all cells could be part of the selected bicluster (i.e. $J = C$) and the second term of (4.1) would vanish. MicroCellClust therefore filters out initially any gene expressed in more than $x\%$ of the cells (typically 25%, which generally keeps around 80% of the genes).

This initial filtering is motivated by the search of gene markers of specific subpopulations rather than generic markers of high expression throughout the cell population. It turns out (see section 4.4) one can identify a large part of these genes by performing a preliminary execution of MicroCellClust on the entire data. These genes can then be removed from $\mathbf{M}$ to perform a new execution of MicroCellClust, which then identifies rare patterns. Yet, given the computational overhead and its very limited impact on the solution, we favor the simpler approach of filtering out genes based on a fixed threshold.

Instead of plainly ignoring these genes, one could rely on alternative data normalization (mentioned in section 3.2.1) to include these genes, but scaled to look for specific rare patterns. For instance, subtracting some threshold (e.g. the median or third quartile) from their original values could enable the identification of genes that are overexpressed in the selected cells, and lowly expressed in all the others. One could also check whether these genes present an absence of expression within the selected cells by replacing their values by the opposite. Nevertheless, we stick here for clarity to the original interpretation as such genes are marginal in number in the reported experiments experiments.

4.1.2 Constraints

Using a global sum criterion in the objective function (4.1) allows for some genes within the bicluster to be lowly (or even negatively) expressed in some cells, providing some tolerance for the dropout phenomenon. Single-cell RNA-seq data is however also often sparse in terms of expressed values. As shown in figure 3.3, unconstrained maximization on such data can lead to the inclusion of many negative values. Indeed, it allows for many cells to be included, even though many of them seem quite unrelated to one another as they only coexpress a minority of the selected genes.

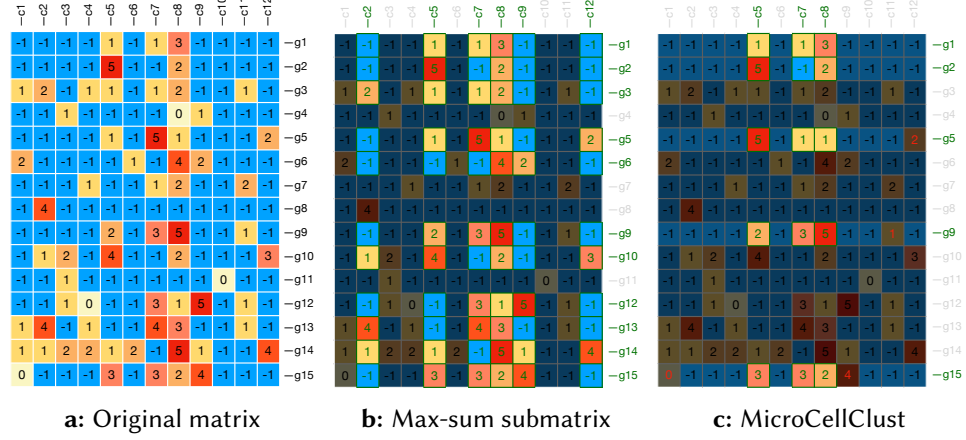


Figure 4.1: A toy example of scRNA-seq data. **a** Data after log-normalization and rounding to integer values for clarity of this example. **b** The optimal solution to the (unconstrained) max-sum submatrix problem. **c** The MicroCellClust solution with $\kappa = 1$ and $\mu = 10\%$.

The additional constraint (4.2) prevents this by putting a limit, defined by the parameter $\mu \in [0, 1]$, on the proportion of negative values allowed in the solution. It ensures the selected genes remain highly expressed across the bicluster as a whole, i.e. all cells coexpress, if not all, the vast majority of selected genes. One typically starts with a relatively conservative value of $\mu = 10\%$, which corresponds to what is observed in the SC3 result when clustering abundant cell types (figure 2.4). This values can be increased if needed after a visual inspection of the result (see section 4.3.1 for an example).

Figure 4.1a represents a toy example of scRNA-seq data made of 12 cells and 15 genes. The unconstrained submatrix solution (figure 4.1b) forms a relatively large bicluster with a low specificity. It includes, for instance, the cells c_2 and c_{12} as they positively contribute to the objective (the sum of the selected entries in their respective column is positive) despite many negative values and a low similarity with the other selected cells. Figure 4.1c represents the MicroCellClust solution, with $\kappa = 1$ in the objective function (4.1) under the constraint (4.2) with $\mu = 10\%$. The 3 cells inside the selected bicluster have a similar expression of their 5 selected genes. These 5 genes are also specifically expressed in this bicluster with only a few out-of-cluster expressions (red digits in one out-of cluster cell, respectively for g_5 , g_9 and g_{15} .). Other genes, such as g_3 , are no longer selected

since their out-of-cluster positive expression implies a negative contribution to the objective function.

Remarks

Equation (4.2) represents a generic constraint on the proportion of negative values allowed in the bicluster. In our implementation, we represent this generic constraint by a global constraint and an additional constraint on each gene:

$$-\sum_{i \in I} \sum_{j \in J} \min\{0, \text{sign}(m_{ij})\} \leq \mu |I| |J| \quad (4.3)$$

$$-\sum_{j \in J} \min\{0, \text{sign}(m_{ij})\} \leq \mu |J| \quad \forall i \in I \quad (4.4)$$

The global constraint (4.3) guarantees that at most a proportion μ of negative values are included inside the selected bicluster. Yet, such a global constraint has no influence on the distribution of these negative entries inside the bicluster. A gene could therefore have many negative values that are compensated by a few very high expression values. Such a gene would thus not be specific to the bicluster. The additional constraints (4.4) make sure that the maximal proportion of negative entries is also satisfied for each selected gene. Such constraints also make sense from a computational point of view, as we can know during the search for the optimal solution (see chapter 5) which genes can no longer satisfy the constraints, and discard them.

One could define different μ values for constraints (4.3) and (4.4), and typically be more permissive (i.e. a larger μ) for individual genes than for the selected bicluster overall. The current implementation uses a common μ value (by default, $\mu = 10\%$) which produces good results in the reported experiments. In practice, the $\mu |J|$ in equation 4.4 is rounded up to avoid that genes in biclusters with only a few cells wouldn't be allowed any negative entries. This *de facto* leads to a more permissive constraint.

4.1.3 Identifying multiple biclusters

Finding the solution to the optimization problem of MicroCellClust is achieved using combinatorial optimization techniques with specific heuristics to efficiently search among the $2^{|\mathcal{G}|+|\mathcal{C}|}$ possible solutions (i.e. corresponding to all possible subsets of genes and cells). This is described in depth in chapter 5.

Solving this optimization problem only identifies one bicluster, leaving the remaining cells unclustered (or *out-of-cluster*). Yet, one can easily identify other

biclusters by executing MicroCellClust again while excluding already clustered cells from the pool of candidate variables. These cells can no longer be selected in J , but their expression can still have an influence on the second term of the objective (4.1) which counts out of cluster expression. Such a strategy corresponds to searching for multiple local optima of the optimization problem (4.1)(4.2), each one representing a different subpopulation in the data.

In order to decide the number of executions to perform, we suggest to interpret biologically the results of each one (e.g. using a Gene Ontology analysis (Ashburner et al., 2000; The Gene Ontology Consortium, 2019)) and stop whenever additional ones seem to return uninteresting results (typically clusters of 2 cells with genes not linked to any GO terms, suggesting all rare subpopulations have already been found).

This approach follows a greedy procedure that is commonly used by several biclustering algorithms (Madeira and Oliveira, 2004; Pontes et al., 2015), such as K -CPGC (Branders et al., 2019) and CCA (Cheng and Church, 2000). Indeed, selecting the optimal assignment of K biclusters implies considering $2^{K(|\mathcal{G}|+|C|)}$ possible solutions, which is not feasible in practice. These approaches efficiently search for K biclusters by “cancelling” the values of previously found biclusters in the data matrix: K -CPGC sets these values to zero, which is neutral with respect to the max-sum criterion; CCA replaces these values by random values within the expression range, as it aims to minimize variance. Here, we choose not to modify the matrix entries but to exclude the previously identified cells from the candidate variables, such that these expression values can still be counted when summing out-of-cluster expression.

Performing K such executions of MicroCellClust approximates the following optimization problem:

$$(I^*, J^*) = \underset{\substack{I_1, \dots, I_K \subseteq \mathcal{G} \\ J_1, \dots, J_K \subseteq \mathcal{C}}}{\operatorname{argmax}} \sum_{n=1}^K \sum_{i \in I_n} \left(\sum_{j \in J_n} m_{ij} - \kappa \sum_{k \in \mathcal{C} \setminus J_n} \max\{0, m_{ik}\} \right) \quad (4.5)$$

$$\text{such that} \quad \frac{|\{(i, j) \mid i \in I, j \in J, m_{ij} < 0\}|}{|I| \cdot |J|} \leq \mu \quad (4.6)$$

$$\bigcap_{n=1}^K J_n = \emptyset \quad (4.7)$$

The greedy approach returns the optimal solution to this problem whenever cells of one bicluster never highly express the selected genes of any other bicluster. If this were to be the case, then there would be an overlap such that a group of cells could correspond to two biclusters. Depending on the order in which the biclusters are identified, the result can be a suboptimal solution to problem (4.5)–(4.7). Such a situation can typically arise when aggregating multiple-single-cell samples corresponding to different donors. We show in section 7.1 that cells can be clustered by donor-specific patterns, preventing them from being included in a cross-sample bicluster. Chapter 7 presents a refined optimization problem that specifically addresses multi-sample data.

From a biological viewpoint, it seems relevant to exclude previously identified cells and keep all genes, as a gene may participate in multiple pathways which may or may not be co-active under several conditions (Madeira and Oliveira, 2004). Alternatively, one could opt to exclude previously identified genes, and keep all cells, to identify new biclusters. Such a strategy might be useful to identify hierarchical structures as cells might be clustered multiple times. In this thesis, we stick to the first option to produce the results described in the subsequent sections.

4.2 Comparison to alternative methods

This section illustrates the benefits of using MicroCellClust to identify small and highly specific subpopulations of cells and their associated marker genes from scRNA-seq data, compared to alternative methods (described in section 2.3). Such a comparison is conducted in a controlled setting, where a small group of cells is planted among an abundant cell type. The first result compares MicroCellClust with the RaceID3 and GiniClust methods at the task of spotting few GARP+ Tregs cells among CD8 T cells, as in the motivating example described in section 2.3.1. Methods specifically designed to deal with large-scale data are not effective to solve this task as commented below. A second controlled experiment aims at spotting rare Jurkat cells among 293T cells from human cell lines. In this case, the fraction of rare cells artificially planted varies from .5% to 10%. The experiments are conducted on datasets with up to 1000 cells which allows further comparisons with large-scale methods such as FiRE, GiniClust3, and scAIDE. Finally, an experiment on synthetic data is conducted to evaluate whether generic (bi)clustering methods are able to identify rare subpopulations whenever they are highly differentiated, allowing to compare MicroCellClust with traditional (bi)clustering algorithms.

4.2.1 Identification of rare GARP+ Tregs among CD8 T cells

The original data is composed of T lymphocytes extracted from a human breast tumor, sorted by flow cytometry into CD8 T cells and activated regulatory T cells and processed following the Smart-seq2 protocol (Picelli et al., 2014) (see details in appendix A.1). Raw expression values are log-transformed ($\log_{10}(x+0.1)$), except for RaceID3 and GiniClust which start from raw expression data. We design a controlled dataset by sampling uniformly at random 5 GARP+ Tregs (from the 96 available ones) and 45 CD8 T cells (from the 83 available ones after quality control). Such a random sampling is repeated independently 100 times and, for each run, the scRNA-seq data matrix contains 50 cells and $\approx 18\,000$ genes on average (MicroCellClust filters out ≈ 2500 genes expressed in more than 25% of the cells).

We assess to which extent each method is able to identify as a separate and specific cluster the small population of GARP+ Tregs (10% of the cells) among the CD8 T cells. These two kinds of lymphocytes have very distinct functions. Yet, clustering them from their expression values into separate and specific subpopulations is not necessarily straightforward, as illustrated in figures 2.4 and 2.5.

RaceID3 and GiniClust perform a full clustering of the data, for which each cell is assigned to one cluster. In contrast, MicroCellClust spots a specific subpopulation of cells leaving the other cells not clustered. Yet, it is straightforward to execute MicroCellClust again on the remaining cells (as discussed in section 4.1.3). Repeating this strategy K times defines the top- K candidate subpopulations identified by MicroCellClust (it turns out that considering the top-2 biclusters is sufficient to find a Treg related cluster for every run, as illustrated below). For all tested methods, the cluster with the highest proportion of GARP+ Tregs is considered to be the solution found for a specific run.

Table 4.1 reports the precision (i.e. the proportion of cells included in the solution which are actual Tregs) and recall (i.e. the proportion of the 5 Tregs to be found actually included in the solution) for each method. Both metrics and their harmonic average, or F_1 score, are averaged over the 100 independent runs. The first column (Nb. subset) indicates the number of runs for which the solution is composed exclusively of Tregs (100% precision). The second column (Nb. exact) indicates the number of runs for which all 5 Tregs exactly form the solution (100% precision and recall). MicroCellClust clearly outperforms the competing methods with a F_1 score equal to 76%, and even equal to 92% when considering the top-2 strategy. MicroCellClust top-2 also identifies perfectly the 5 GARP+ Tregs in 54 out of the 100 runs. This is much less the case with RaceID3 and GiniClust which

Table 4.1: Identification results of rare GARP+ Tregs among CD8 T cells. Averaged values over 100 independent runs.

Method	Nb. subset	Nb. perfect	Precision	Recall	F_1 score
MicroCellClust top 1	71	45	0.79	0.74	0.76
MicroCellClust top 2	86	54	0.96	0.89	0.92
RaceID3	48	2	0.75	0.47	0.55
GiniClust	9	0	0.35	0.59	0.37
scAIDE	1	0	0.20	0.95	0.32

“Nb. subset” reports the number of runs with 100% precision.

“Nb. perfect” reports the number of runs with 100% precision and recall.

Bold results indicate which method outperforms the others for each metric.

illustrates that these methods are far less efficient at finding rare and specific subpopulations.

Figure 4.2 reports the bicluster found by MicroCellClust for a representative run. The 5 Tregs are perfectly identified and the selected subset of marker genes is informative. For instance, the marker genes include IL2RA and FOXP3, which are clear indicators of Treg cells, and CCR8, which is specific to Tregs present in the breast (Plitas et al., 2016).

Such a result is very rarely obtained with RaceID3. In most cases, the Tregs are split into different clusters and/or end up being clustered with CD8 T cells. The outlier detection procedure behaves poorly here as RaceID3 often identifies only a subset of Tregs as outliers, resulting in mixed clusters. Figure 4.3 illustrates a good performance run for RaceID3 that correctly identifies Treg related genes, such as VDR, IL2RA, CD4, CCR8, and FOXP3, also present in the MicroCellClust solution. However, 2 CD8 T cells are also present in the solution found and the last 3 marker genes are also highly expressed in many cells from the other clusters.

GiniClust performs poorly in this controlled experiment. GiniClust (with the p -value cutoff of the Gini index tuned to 0.005 since the default cutoff of 10^{-4} leads to select very few genes, none at all for some run) identifies ≈ 35 high Gini genes on average over 100 runs. Yet, it always fails at identifying exactly the 5 GARP+ Tregs. Figure 4.4 illustrates a best run, in relative terms, for GiniClust. For this run, 4 out of 5 Tregs are correctly identified in a separate cluster but the marker genes are not particularly expressed in this cluster. In other words, GiniClust fails at identifying highly specific genes which would be together in-

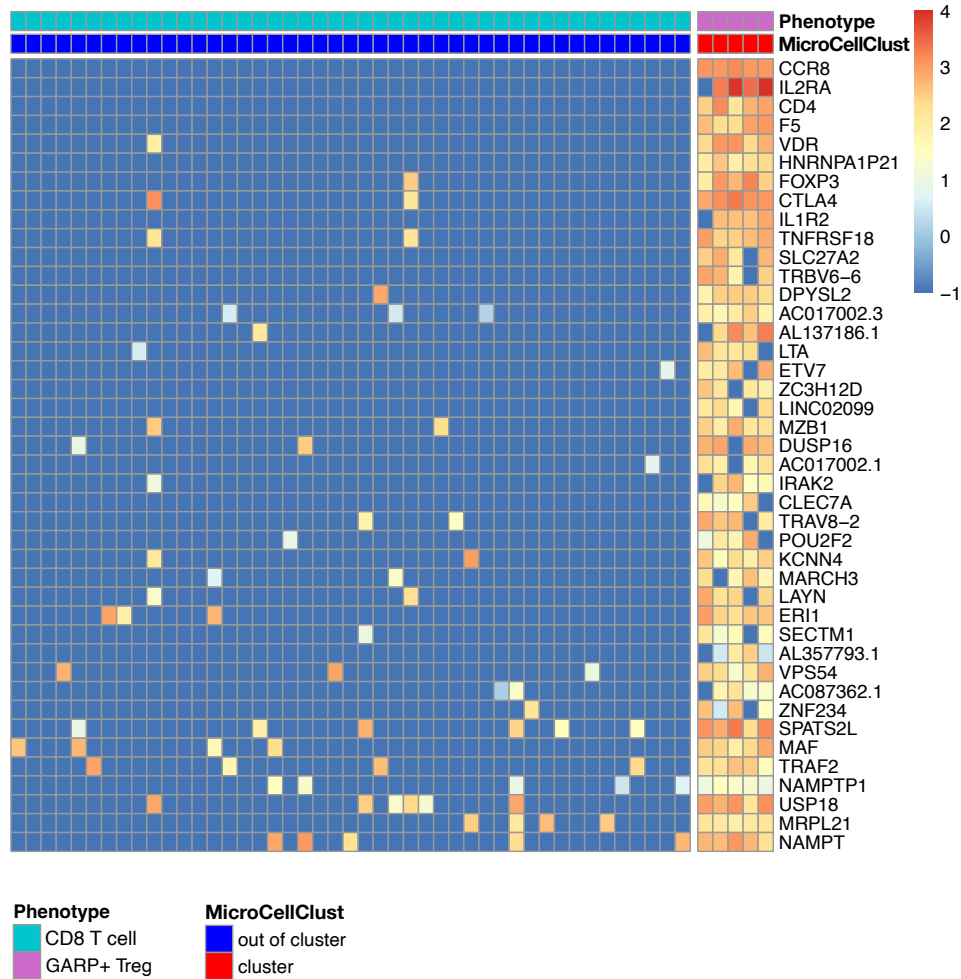


Figure 4.2: Representative results of GARP+ Tregs identification within CD8 T cells using MicroCellClust. Marker genes identified by MicroCellClust in a typical run where it identifies the 5 GARP+ Tregs perfectly. See figures 4.3 and 4.4 for representative results of alternative methods.

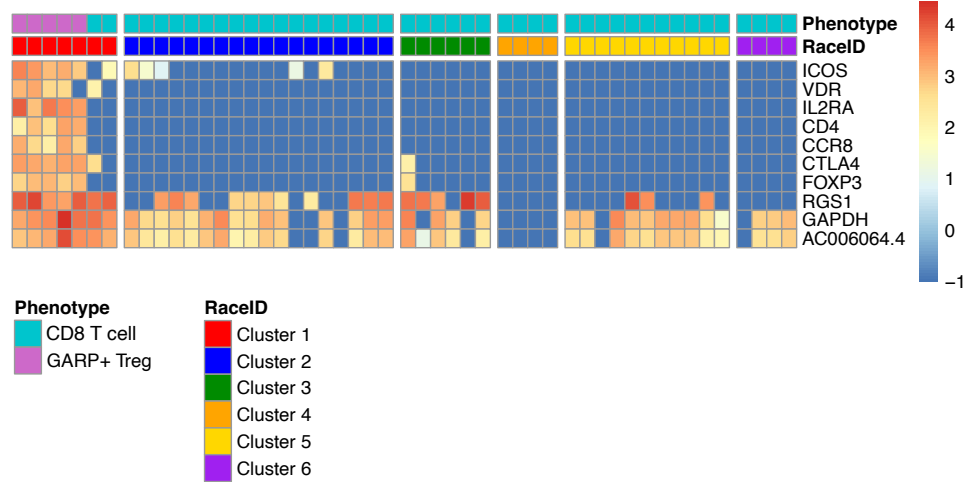


Figure 4.3: Representative results of GARP+ Tregs identification within CD8 T cells using RaceID3. Marker genes for cluster 1 found by RaceID3 in a typical run. RaceID3 correctly identified Treg-associated genes, but also includes CD8 T cells in the same cluster. The last 3 genes are also highly expressed in other cells.

formative of the identified cell clusters. This is most likely due to the univariate gene selection performed in this method which contrasts drastically with the MicroCellClust approach.

The methods for large-scale data are performing very poorly in this experiment. scAIDE has a very low precision with the Tregs assigned to large clusters of 25 cells on average. FiRE returns the same score to each cell and is therefore useless for this data. GiniClust3 finds only 1 candidate marker gene, no matter which Gini cut-off is considered, and is therefore useless as well.

Geneset reproducibility

We assess here whether, besides a high F_1 -score, MicroCellClust is able to consistently return similar genesets when performing multiple resamplings of the rare GARP+ Tregs versus CD8 T cells experiment. Figure 4.2 shows a representative result where the 5 Tregs are perfectly identified, together with Treg related genes such as FOXP3, IL2RA and CCR8. These characteristic genes are also reidentified in the vast majority of the 100 runs performed in this experiment. Table 4.2a shows that these genes, and several others, are present in nearly all of the 54 runs where MicroCellClust identifies the 5 Tregs perfectly. This illustrates the ability

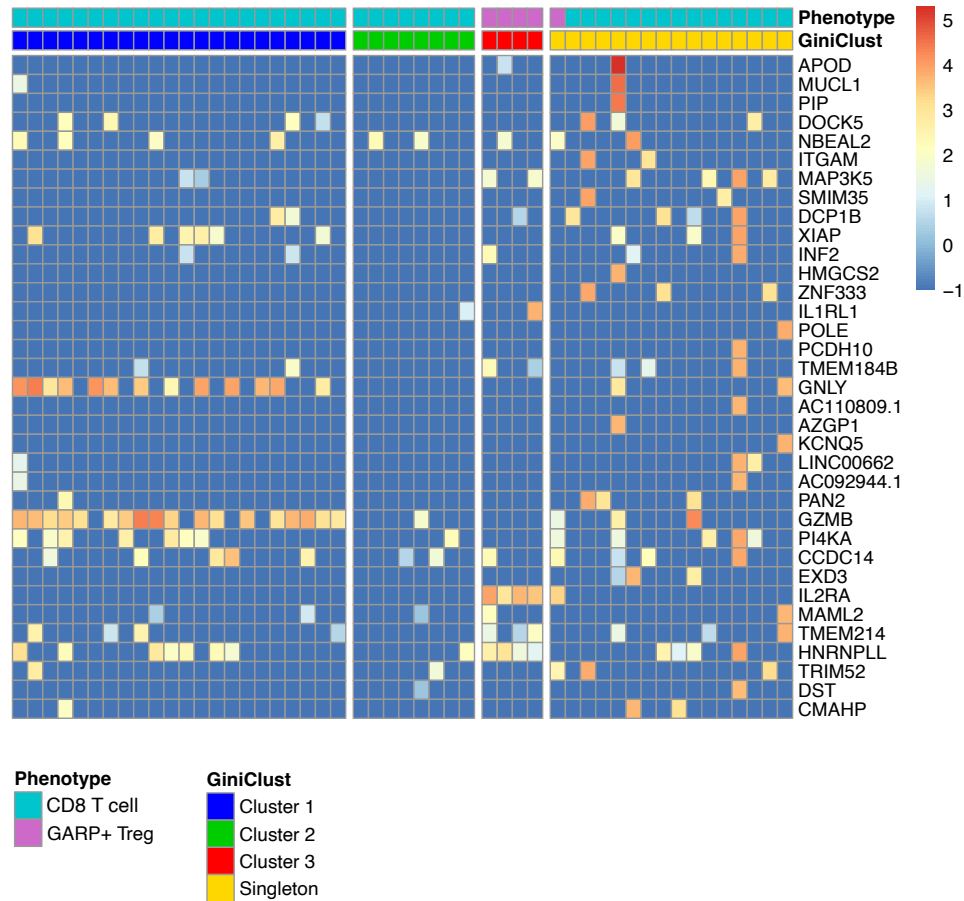


Figure 4.4: Representative results of GARP+ Tregs identification within CD8 T cells using GiniClust. High Gini genes identified by GiniClust for one of its best runs (i.e. 100% precision). 4 out of 5 Tregs are correctly identified in a separate cluster but the marker genes are not particularly expressed in this cluster.

Table 4.2: Occurrence of the 20 most recurring genes identified by MicroCell-Clust during the GARP+/CD8 experiment. a Occurrence for the 54 runs with perfect precision and recall only (i.e. the 5 Tregs form exactly the solution). **b** Occurrence for all of the 100 runs.

a: Perfect runs			b: All runs		
Rank	Gene	Occur. [%]	Rank	Gene	Occur. [%]
1	CD4	96	1	CTLA4	85
1	FOXP3	96	2	FOXP3	84
1	CTLA4	96	3	CCR8	83
4	CCR8	94	4	CD4	80
5	TNFRSF18	87	4	TNFRSF18	80
6	IL2RA	85	6	IL2RA	76
7	CCDC22	81	7	DUSP4	72
7	DUSP4	81	8	CCDC22	69
9	AL137186.1	78	9	AL137186.1	68
9	ICOS	78	10	IL1R2	64
11	IL1R2	74	10	ICOS	64
12	HNRNPA1P21	67	12	ENTPD1	61
13	ENTPD1	65	13	HNRNPA1P21	59
14	LINC02099	59	14	VDR	55
15	VDR	50	15	LINC02099	51
16	ENTPD1-AS1	48	16	ENTPD1-AS1	49
17	F5	46	17	F5	44
18	PMAIP1	44	18	TBC1D4	37
19	TBC1D4	39	19	TNFRSF4	34
20	DNPH1	37	20	DNPH1	33

Table 4.3: Identification results of rare Jurkat cells among 293T cells. Average F_1 score over 10 independent runs.

Tot. size	Rare cells		MCC	RaceID3	FiRE	GiniClust3	scAIDE
50	5	(10%)	0.99	0.71	0.00	0.00	0.74
100	10	(10%)	0.99	0.68	0.00	0.00	0.47
	5	(5%)	0.97	0.72	0.16	0.00	0.18
200	20	(10%)	0.96	0.67	0.13	0.00	0.57
	10	(5%)	0.95	0.59	0.41	0.00	0.16
	5	(2.5%)	0.95	0.66	0.54	0.00	0.03
500	50	(10%)	0.93	0.46	0.78	0.99	0.99
	25	(5%)	0.95	0.45	0.80	0.88	0.99
	10	(2%)	0.95	0.51	0.56	0.01	0.83
	5	(1%)	0.92	0.60	0.41	0.71	0.06
1000	100	(10%)	0.91	0.41	0.92	1.00	0.72
	50	(5%)	0.91	0.43	0.77	1.00	0.85
	20	(2%)	0.95	0.49	0.51	0.99	0.98
	10	(1%)	0.92	0.53	0.33	0.50	0.30
	5	(0.5%)	0.93	0.51	0.19	0.62	0.05

Note: MCC stands for MicroCellClust.

Bold results indicate which method outperforms the others for each experiment.

of MicroCellClust to reidentify important marker genes when resampling the target subpopulation (5 GARP+ Tregs are randomly sampled from the 96 available ones for each run). A high occurrence is also observed when considering all the runs (table 4.2b), thus including those where only a subset of the Tregs form the solution and/or some CD8 T cells are also selected.

4.2.2 Identification of rare Jurkat cells among 293T cells

We further assess the performance of MicroCellClust while varying data sizes (from 50 to 1000 cells) and rare cell proportions using a dataset containing two human immortalized cell lines: 293T and Jurkat cells (Zheng et al., 2017) (see appendix A.2). These two cell types were mixed in vitro and a single-nucleotide variant has been used to determine the lineage of each cell, resulting in a total

of 1540 293T cells and 1694 Jurkat cells. We design a controlled experiment by sampling at random rare populations of Jurkat cells (from 0.5% to 10% of the data, with a minimum of 5 cells) and abundant populations (up to 1000 cells) of 293T cells and ≈ 15000 genes. MicroCellClust uses ≈ 9500 genes expressed in at most 25% of the cells. For each combination, 10 independent runs are performed. For this experiment, the κ parameter of MicroCellClust has been chosen as $\kappa = \frac{100}{|C|}$ which produces results with ≈ 20 marker genes, regardless of the size of the number of rare cells in the identified bicluster.

Table 4.3 reports the F_1 score of each method averaged over 10 runs. MicroCellClust achieves a high F_1 score ($\geq 90\%$) in each case (the top-1 result always produces a Jurkat related bicluster during this experiment). This is not the case for the other methods. The methods designed for large-scale datasets perform poorly whenever the total number of cells is less or equal to 200. For larger datasets, with at least 500 cells, they may perform well but their performance significantly drops when the subpopulation to identify only contain a few cells (5 to 10). These results illustrate that the performance of MicroCellClust is stable with respect to the total number of cells. It is clearly the best performing method to identify a particularly rare subpopulation of cells. This is consistent with the objective to identify rare patterns in data of limited availability.

4.2.3 Rareness versus distinctiveness

The previous sections illustrate the effectiveness of MicroCellClust at identifying target rare subpopulations of cells in actual scRNA-seq data. It consistently achieves a high performance when varying the rareness of the subpopulation to be identified, outperforming other rare cell identification methods. Given the known limitation of generic (bi)clustering at identifying rare cells (Grün et al., 2015; Jiang et al., 2016), such methods were not included in these analyses. Yet, the latter did not evaluate how the distinctiveness of the rare subpopulations (i.e. the amount of out-of-cluster expression of its corresponding genes) affects clustering performance. One may suppose that whenever a rare subpopulation is very distinct, regular (bi)clustering algorithms could be able to detect them.

In order to compare MicroCellClust to generic (bi)clustering algorithms, in terms of both rareness and distinctiveness, several artificial datasets containing 1000 cells and 1000 genes are generated according to the following protocol. The matrices are initially filled with -1 values. A predefined number of columns (5, 20 or 50) are arbitrarily selected to represent the rare subpopulation to be identified. Similarly, 30 rows are selected to be marker genes for this target subpopulation,

and positive expression values will be added in such a way they will show differential expression within the target cells.

For each gene, a range of positive expression is defined by randomly selecting a lower and upper bound within the $[0, 4]$ interval. Assigning a positive expression value to a particular cell will be done by randomly choosing a real value from this interval. First, positive expression values are randomly added to the genes that are not related to the target subpopulation (see appendix A.3 for further details). Then, positive expression values are added to the 30 marker genes in the following way. All the target cells are assigned a positive value, except for a few dropouts (generated with a probability between 10% and 20%, depending on the gene) for which the values remain -1 . For the out-of-cluster cells, a positive expression is only assigned according to some small probability, meaning most values will remain negative. For each gene, a different probability value is chosen within a predefined range, such that some genes will present more out-of-cluster expression than others.

In order to vary the distinctiveness of the target subpopulation, a different range of probability is defined for each dataset. By doing so, one can monitor the amount of out-of-cluster expression, and therefore how distinctive the target subpopulation is from the remaining cells. In the first case, this range goes from 0.1% to 1%, meaning all genes have few out-of-cluster expression and the rare subpopulation is very distinct (figure 4.5a). In the next 3 cases, the upper bound is increased to 10%, 20% and 30%, respectively (figure 4.5b). This means the subpopulation becomes less and less distinct, but some very distinct genes might still exist in the data (even though their number is expected to decrease each time, as the number of marker genes, 30, stays constant). In the last case, the lower bound is increased to 10% (keeping an upper bound of 30%), meaning there are no longer any highly differentiated genes (figure 4.5c).

MicroCellClust is here compared to 4 generic (bi)clustering algorithms. First, it is compared to k -means (MacQueen, 1967), a classical clustering algorithm. Then, it is compared to SC3 (Kiselev et al., 2017), a consensus clustering method specifically developed for scRNA-seq data (also used in the motivation example of section 2.3.1). Finally, a hierarchical clustering and a biclustering algorithm are evaluated. Agglomerative hierarchical clustering with WGCNA (Langfelder and Horvath, 2008) and Spectral biclustering (Kluger et al., 2003) were chosen as they are top performing methods in their respective categories (Saelens et al., 2018). For k -means, SC3, and Spectral, the parameter indicating the number of clusters to identify was set to two. For WGCNA, a threshold was set at a depth of 5: all

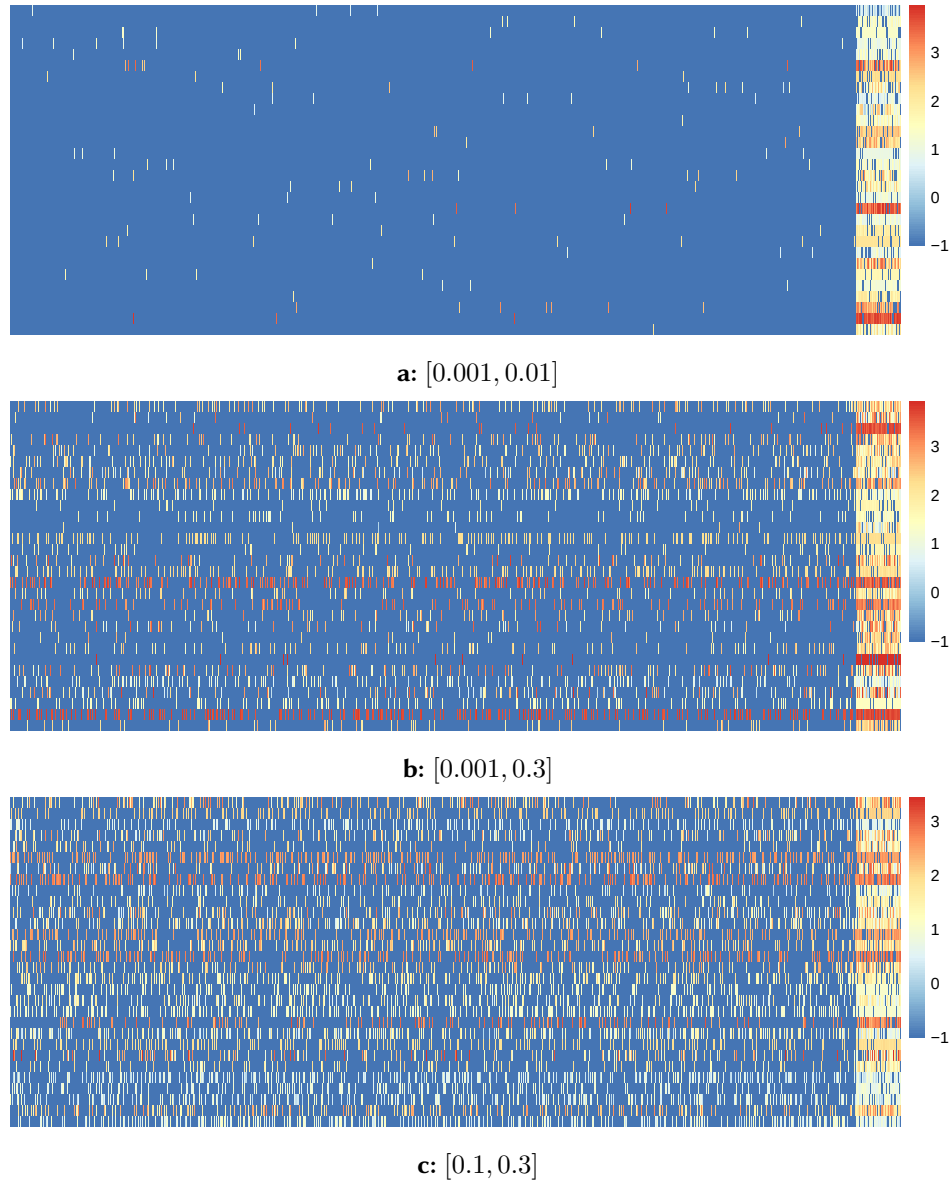


Figure 4.5: Impact of the range of probability on the distinctiveness of the target.
a The very restrictive range of $[0.001, 0.01]$ means all genes are have high differential expression. **b** As the upper bound of the range is increased to 0.3, the target is less distinct. Yet, some very differentiated genes remain. **c** When raising the lower bound to 0.1, there are no more highly differentiated genes.

Table 4.4: Analysis of rareness versus distinctiveness. F_1 scores on artificial data containing 1000 cells and 1000 genes, 30 of which correspond to a target rare subpopulation. Each dataset differs in terms of rareness of the subpopulation to be identified (first column), and the distinctiveness w.r.t. the other cells in the data (second column).

Rare cells	O.o.c. exp. range	MCC	k -means	SC3	WGCNA	Spectral
50 (5%)	[0.001, 0.01]	1.00	0.18	1.00	0.97	1.00
	[0.001, 0.1]	1.00	0.10	1.00	0.97	1.00
	[0.001, 0.2]	1.00	0.16	1.00	0.87	0.10
	[0.001, 0.3]	0.99	0.32	1.00	0.99	0.90
	[0.1, 0.3]	0.85	0.09	0.11	0.61	0.10
20 (2%)	[0.001, 0.01]	1.00	0.10	0.10	1.00	1.00
	[0.001, 0.1]	1.00	0.10	0.11	0.90	0.03
	[0.001, 0.2]	0.98	0.11	0.11	0.92	0.06
	[0.001, 0.3]	0.95	0.10	0.10	0.70	0.04
	[0.1, 0.3]	0.89	0.11	0.13	0.15	0.04
5 (0.5%)	[0.001, 0.01]	1.00	0.11	0.11	0.91	1.00
	[0.001, 0.1]	1.00	0.11	0.11	0.91	1.00
	[0.001, 0.2]	1.00	0.10	0.16	1.00	0.01
	[0.001, 0.3]	0.91	0.10	0.10	0.32	0.14
	[0.1, 0.3]	0.05	0.10	0.10	0.01	0.01

Note: MCC stands for MicroCellClust.

Bold results indicate which method outperforms the others for each experiment.

nodes in the tree up to this depth (included) were considered as potential solutions, and the one containing the highest proportion of target cells was selected. As before, a F_1 score is computed to measure how accurately the target cells are clustered.

The results are reported in table 4.4. First of all, one observes k -means clustering is ineffective at this task, regardless of the proportion of rare cells present in the data and their distinctiveness. SC3 is able to identify the relatively large subpopulations of 50 cells whenever there are at least a few genes that are very differentially expressed. Interestingly, if these genes are absent (last of the 5 cases), it does not detect the target subpopulation, and splits the data into two large clusters based on one differentially expressed gene. SC3 also fails to detect smaller

subpopulations. WGCNA is overall the best performing generic method, as it identifies with a relatively high accuracy rare subpopulations of different sizes when they are relatively differentiated. However, it loses in efficiency when the target is less differentiated, and tends to scatter these cells in different parts of the tree. Finally, Spectral, while being able to perfectly detect targets of different sizes when they are very differentiated, completely misses them in most of the cases.

MicroCellClust, on the other hand, achieves a high (if not perfect) performance in almost all the cases, regardless of the proportion of rare cells. In the case there are no highly differentiated genes, for 50 and 20 cells, MicroCellClust reports a small drop in performance (as a few extra cells are included in the solution), but still clearly outperforms the alternatives. It is only in the rather complicated case where the subpopulation contains only 5 cells with few differentiation that MicroCellClust, as well as the alternatives, fails to detect the target. These results further confirm the relevance of MicroCellClust to identify rare subpopulations of cells.

4.3 Case studies on scRNA-seq data

We further illustrate the potential of MicroCellClust to highlight small subpopulations among cells of interest, by analyzing three scRNA-seq datasets. We report how MicroCellClust highlights a small subpopulation with a specific phenotype inside CD4 T cells from breast tumor samples, as well as a small subpopulation linked to cell cycle activity among activated Tregs. Finally, we report how MicroCellClust identifies several relevant subpopulations in mouse embryonic stem cells.

4.3.1 Identification of a T regulatory related subpopulation within CD4 T cells

MicroCellClust is first used to analyze a set of 182 CD4 T cells from the breast sample under study in section 4.2.1 (see details in appendix A.1). It contains 93 tumor and 89 healthy cells (after quality control) and 16,737 genes expressed in at most 25% of the cells. The objective is to check whether any subpopulation with a specific function can be identified among these CD4 T cells, and to link it to the immune response against breast cancer. The healthy cells can serve here as a control to see whether the identified subpopulation only contains tumor cells, i.e. with a function specific to a tumor.

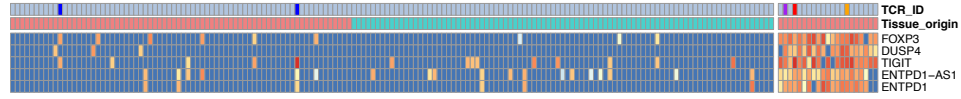
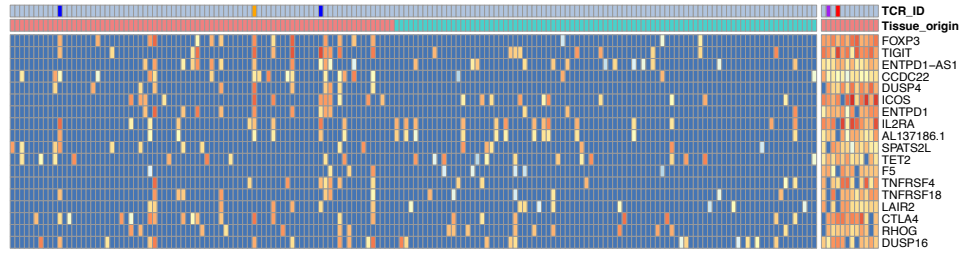
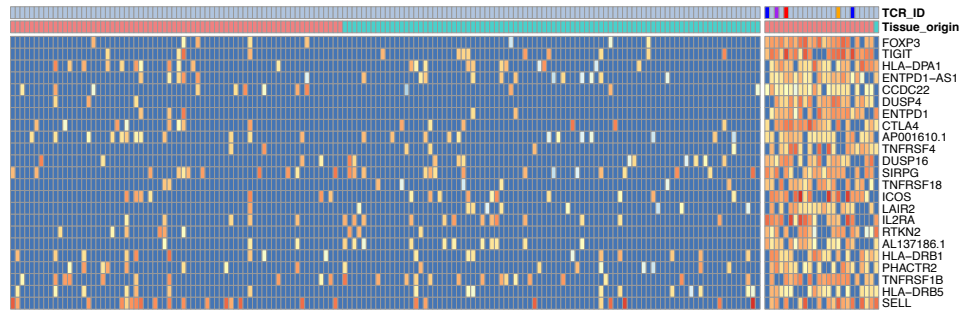
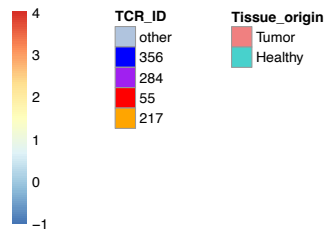
a: $\kappa = 0.5, \mu = 10\%$ b: $\kappa = 0.3, \mu = 10\%$ c: $\kappa = 0.3, \mu = 30\%$ 

Figure 4.6: Treg related subpopulations (cells on the right) identified among CD4 lymphocytes, obtained when changing parameters κ and μ . The marker genes identified by MicroCellClust indicate that these cell-clusters are related to GARP⁺ Tregs. Cells sharing a TCR sequence with a GARP⁺ Treg are reported (see first row on top of each heatmap) further confirming this hypothesis.

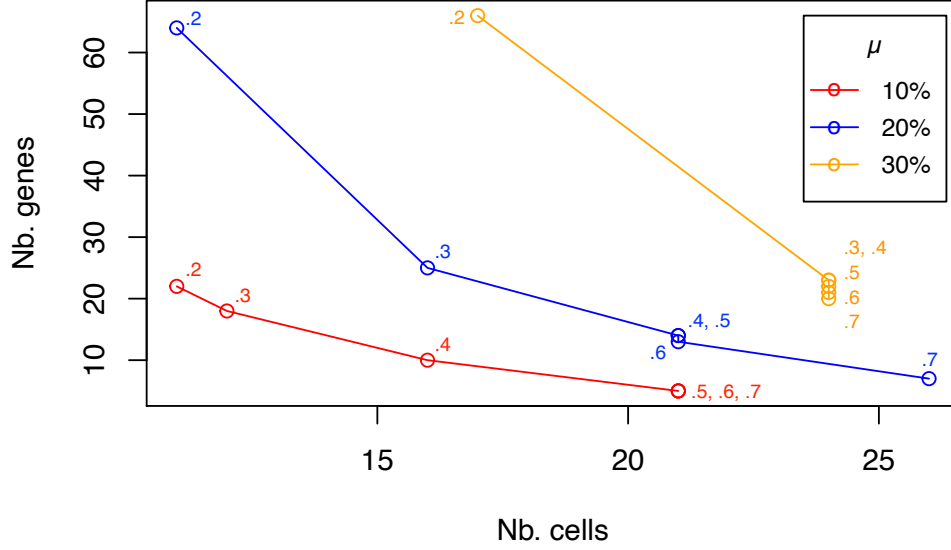


Figure 4.7: Number of cells and genes identified by MicroCellClust in the CD4 T cell data in function of the κ parameter. The values of κ are indicated next to the dots. The results are reported for three different values of the μ parameter.

MicroCellClust identifies a cluster of 23 cells originating from the tumor, characterized by the expression of Treg associated genes such as FOXP3 and IL2RA (figure 4.6c). Interestingly, these cells were not annotated as GARP+ Tregs by the flow cytometry sorting since they did not express GARP. However, their gene expression clearly indicates Treg-related functions. This is confirmed by a Gene Ontology analysis (table 4.5). We conjecture that GARP is transiently expressed after Treg activation, and that the identified cells might be related to the GARP+ Tregs by being resting or not recently activated Tregs. Indeed, 5 cells from the identified cluster appear to have the same T cell receptor (TCR) sequences than GARP+ Tregs, which supports our hypothesis. To sum up, MicroCellClust enables us to identify a set of resting or not recently activated Tregs among CD4 T cells, a subpopulation that had not been sorted out by flow cytometry.

Parameter tuning

This previous result has been obtained by slightly tuning the κ and μ parameters from their default values. Indeed, MicroCellClust returns a bicluster with 5 marker genes when using $\kappa = 0.5 \approx \frac{100}{|C|}$ and $\mu = 10\%$, including FOXP3 which already suggests a Treg related function (figure 4.6a). This relatively low

Table 4.5: GO functions for the Treg related subpopulation within CD4 T cells.

This analysis has been performed using the `clusterProfiler` R package (Yu et al., 2012), using the genes returned by `MicroCellClust` with $\kappa = 0.3$ and $\mu = 30\%$. The 20 most significantly enriched GO terms are reported.

	ID	Description	p.adjust	GeneRatio
1	GO:0042110	T cell activation	3.1138e-09	10/20
2	GO:0050863	regulation of T cell activation	3.1138e-09	9/20
3	GO:0051249	regulation of lymphocyte activation	3.1138e-09	10/20
4	GO:0007159	leukocyte cell-cell adhesion	1.3524e-07	8/20
5	GO:1903037	regulation of leukocyte cell-cell adhesion	1.8311e-06	7/20
6	GO:0042098	T cell proliferation	2.1847e-06	6/20
7	GO:0045589	regulation of regulatory T cell differentiation	2.1847e-06	4/20
8	GO:0051251	positive regulation of lymphocyte activation	2.1847e-06	7/20
9	GO:0045066	regulatory T cell differentiation	2.4163e-06	4/20
10	GO:0050870	positive regulation of T cell activation	2.5437e-06	6/20
11	GO:0050670	regulation of lymphocyte proliferation	2.5953e-06	6/20
12	GO:0032944	regulation of mononuclear cell proliferation	2.5953e-06	6/20
13	GO:1903039	positive regulation of leukocyte cell-cell adhesion	2.9698e-06	6/20
14	GO:0002696	positive regulation of leukocyte activation	2.9698e-06	7/20
15	GO:0070663	regulation of leukocyte proliferation	2.9698e-06	6/20
16	GO:0050867	positive regulation of cell activation	3.3784e-06	7/20
17	GO:0022407	regulation of cell-cell adhesion	3.5021e-06	7/20
18	GO:0045785	positive regulation of cell adhesion	3.5021e-06	7/20
19	GO:0002517	T cell tolerance induction	4.5396e-06	3/20
20	GO:0022409	positive regulation of cell-cell adhesion	5.0522e-06	6/20

number of marker genes suggests to decrease κ to 0.3 in order to decrease the out-of-cluster penalty, allowing more genes to be included in the solution. This provides a solution with 12 cells and 18 genes, including IL2RA which further supports the Treg interpretation (figure 4.6b).

Decreasing κ also tends to select fewer cells, as can be seen in figure 4.7. Only 12 of the 21 previously identified cells are selected with $\kappa = 0.3$ and $\mu = 10\%$. Some cells express a significant proportion of these marker genes but are not included in this bicluster. Such a behavior can probably be explained by the presence of some dropouts in the scRNA-seq data, and the transient nature of Treg activation. This suggests to increase the μ parameter to 30% to relax the constraint imposing a maximal proportion of negative entries in the bicluster (figure 4.6c). Figure 4.7 also shows that with $\mu = 30\%$, the cell subpopulation is the most stable with respect to κ , as the same 24 cells are selected for $\kappa \in [0.3, 0.7]$. The identified subpopulation is *a posteriori* validated by the fact that the bicluster found includes 5 cells sharing a TCR sequence with a GARP+ Treg.

4.3.2 Identification of a cell-cycle related subpopulation in tumor GARP+ Tregs

Our next example illustrates the use of MicroCellClust to search for a potential subpopulation inside a set of GARP+ Tregs among breast tumor samples from two different patients. The scRNA-seq data represents here 168 cells and 13,772 genes expressed in at most 25% of the cells.

MicroCellClust, with default values $\kappa = 0.6 \approx \frac{100}{|C|}$ and $\mu = 10\%$, identifies a subpopulation of 21 cells which forms a specific cluster associated to numerous cell division related genes (figure 4.8). A GO analysis indeed highlights cell division functions (table 4.6). Such an interpretation makes sense since GARP+ Tregs represent a population of recently activated Tregs, and T cell activation usually leads to cell proliferation. MicroCellClust thus reveals the existence of dividing Tregs inside breast tumor tissues, an observation definitely worth further investigation.

Parameter analysis

On this data, the default values for μ and κ give a result with 33 genes, which is appropriate to perform a biological interpretation. Moreover, the resulting bicluster is stable with respect to the κ parameter. When increasing κ , the exact same cell subpopulation is identified, while the number of genes decreases (from

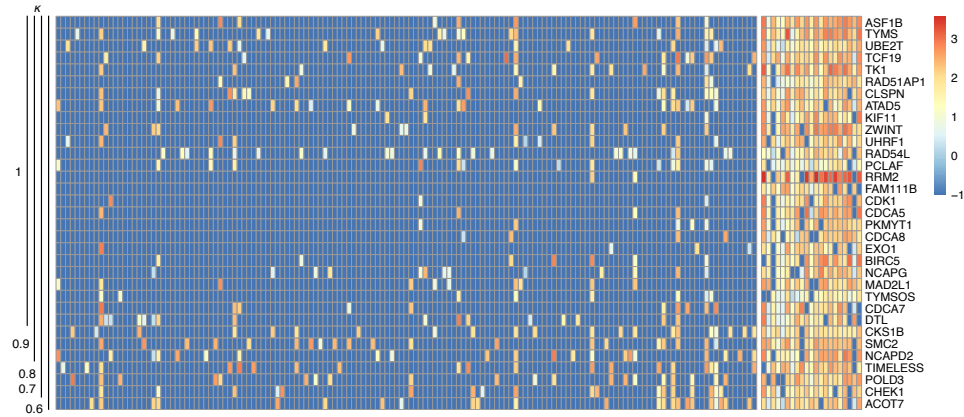


Figure 4.8: Cell-cycle related subpopulation among GARP+ Tregs (21 cells on the right). The identified marker genes are mostly related to cell division. Vertical bars on the left of the heatmap indicate which marker genes are selected when changing the value of κ from 0.6 to 1.

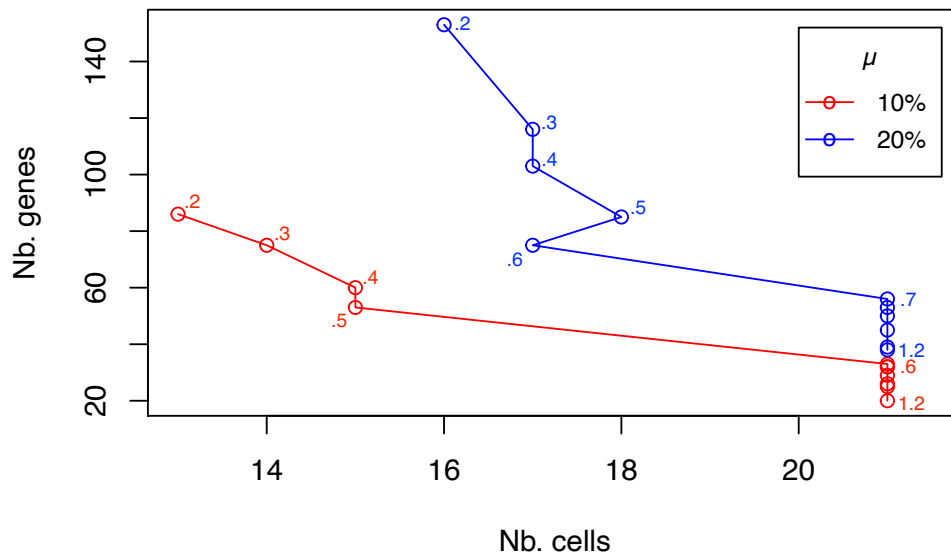


Figure 4.9: Number of cells and genes identified by MicroCellClust in the GARP+ Treg data in function of the κ parameter. The results are reported for two different values of the μ parameter.

Table 4.6: GO functions for the cell-cycle subpopulation in GARP+ Tregs. This analysis has been performed using the `clusterProfiler` R package (Yu et al., 2012). The 20 most significantly enriched GO terms are reported.

	ID	Description	p.adjust	GeneRatio
1	GO:0140014	mitotic nuclear division	1.6551e-10	11/31
2	GO:0000280	nuclear division	3.2562e-10	12/31
3	GO:0048285	organelle fission	6.8814e-10	12/31
4	GO:0006260	DNA replication	2.2744e-09	10/31
5	GO:0000070	mitotic sister chromatid segregation	6.0179e-07	7/31
6	GO:0030261	chromosome condensation	1.4243e-06	5/31
7	GO:0007076	mitotic chromosome condensation	1.4243e-06	4/31
8	GO:0000819	sister chromatid segregation	1.7845e-06	7/31
9	GO:0007059	chromosome segregation	2.9742e-06	8/31
10	GO:0000075	cell cycle checkpoint	3.5763e-06	7/31
11	GO:0031572	G2 DNA damage checkpoint	1.1120e-05	4/31
12	GO:0098813	nuclear chromosome segregation	1.1120e-05	7/31
13	GO:0071103	DNA conformation change	4.5714e-05	7/31
14	GO:0006323	DNA packaging	5.3637e-05	6/31
15	GO:0045787	positive regulation of cell cycle	1.2577e-04	7/31
16	GO:0044839	cell cycle G2/M phase transition	1.8450e-04	6/31
17	GO:0031570	DNA integrity checkpoint	2.2603e-04	5/31
18	GO:0007088	regulation of mitotic nuclear division	2.6408e-04	5/31
19	GO:0006310	DNA recombination	2.6518e-05	6/31
20	GO:1901987	regulation of cell cycle phase transition	3.7422e-04	7/31

33 for $\kappa = 0.6$ down to 26 for $\kappa = 1$) since out-of-cluster expression is more penalized with a higher κ value. This results in only keeping the most differentially expressed genes (check the left part of figure 4.8).

As the results in figure 4.8 correspond to our suggestion of selecting 10 to 30 genes, the value $\kappa = 0.6$ seems appropriate. With a lower value (*e.g.* $\kappa = 0.5$), fewer cells would be selected (see figure 4.9) while several out-of-cluster cells would express a lot of marker genes in such a solution. This further confirms that the choice of $\kappa = 0.6 \approx \frac{100}{|C|}$ is suitable. Increasing μ to *e.g.* 20% is not relevant here, since most κ values produce a bicluster containing also 21 cells (see figure 4.7). In such cases, one tends to prefer the most restrictive μ value (*i.e.* 10%).

4.3.3 Identification of rare subpopulations in mouse embryonic stem cells

This last case study shows the result of MicroCellClust on a dataset of mouse embryonic stem cells 4 days after leukemia inhibitory factor withdrawal (Klein et al., 2015a) (see appendix A.4). It contains 683 cells and 19 749 genes expressed in at most 25% of the cells. Three executions of MicroCellClust, with $\kappa = 0.1$ and $\mu = 10\%$, are performed to identify three distinct subpopulations (figure 4.10). The first bicluster is composed of 5 cells and 18 marker genes, including genes from the Zscan4 family and Tcstv1, which have been associated to 2C-like cells (Macfarlan et al., 2012). To evaluate the reproducibility of this very small subpopulation ($< 1\%$ of the cells) when resampling data, 500 cells are drawn at random while forcing the 5 identified cells to be included. This subpopulation is correctly identified again over 10 independent runs. Moreover, it is stable with respect to the choice of κ value, as the same bicluster is identified for κ between 0.08 and 0.2.

The second bicluster returned by MicroCellClust contains 14 cells and 8 marker genes, among which primitive endoderm markers Col4a1, Col4a2 and Lama1. Finally, a subpopulation of 61 cells with three highly expressed marker genes is identified, including maternally imprinted genes Rhox6 and Rhox9. The 3 subpopulations identified by MicroCellClust on this dataset have been identified in the original publication using additional *prior* knowledge. This confirms the ability of MicroCellClust to identify subpopulations with biologically relevant gene combinations.

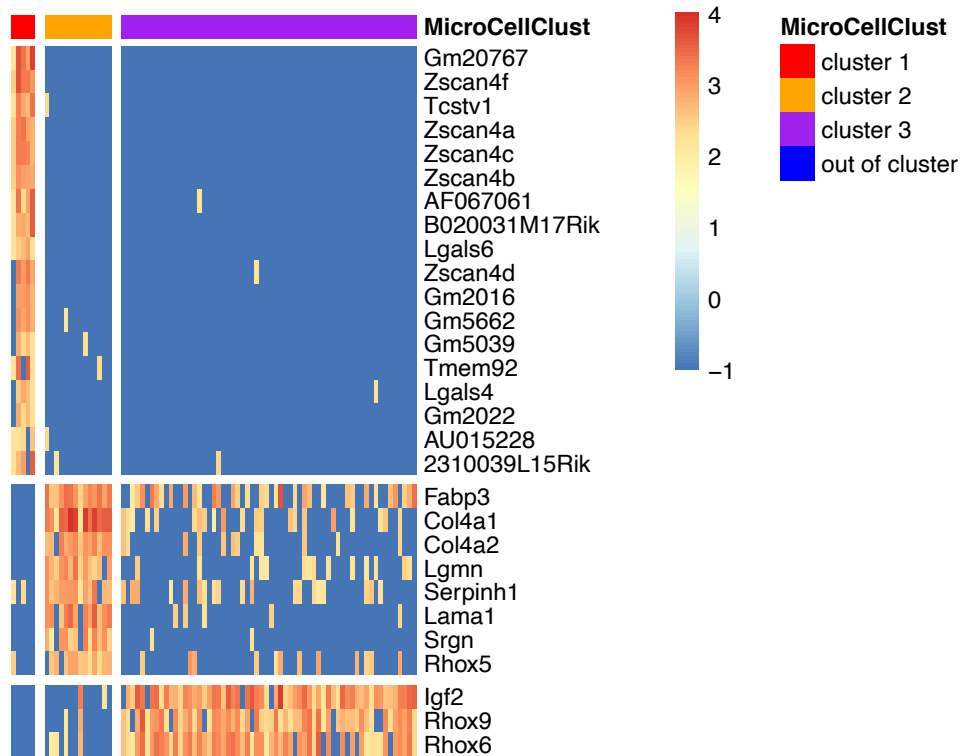


Figure 4.10: Three subpopulations identified by MicroCellClust in mouse embryonic stem cells. The reported biclusters are identified with $\kappa = 0.1$ and $\mu = 10\%$. Out of cluster cells are not displayed.

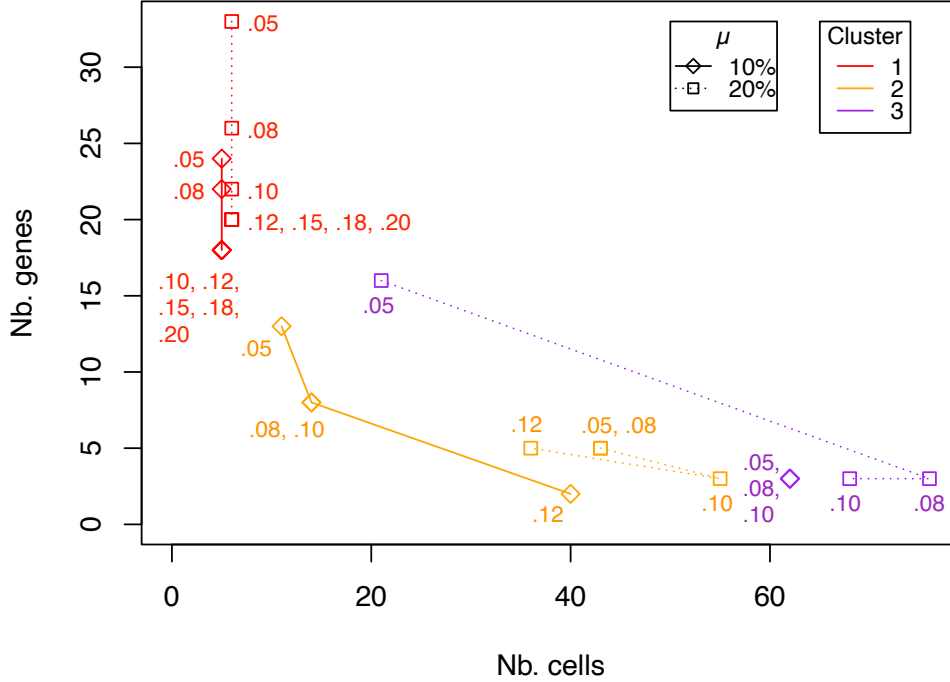


Figure 4.11: Number of cells and genes in function of the κ parameter for the 3 subpopulations identified by MicroCellClust in the mouse stem cell data. The results are reported for two different values of the μ parameter.

Parameter analysis

Figure 4.10 illustrates the 3 subpopulations identified by MicroCellClust in the mouse stem cell data with $\kappa = 0.1 \approx \frac{100}{|C|}$ and $\mu = 10\%$. The expression of the marker genes of cluster 1 is highly differentiated. It is therefore unsurprising that the same 5 cells are selected for a large set of κ values around 0.1 (see figure 4.11). When relaxing μ to 20%, an extra cell, which expresses about half of the marker genes, is added (all marker genes of the $\mu = 10\%$ solution are marker genes for the $\mu = 20\%$ result; see figure 4.12). In particular, it expresses 4 genes of the Zscan4 family, so this extra cell could also be of interest even though it does not express a large part of the marker genes.

The number of cells and genes assigned to cluster 2 is more sensitive to variations of κ , which probably comes from the fact that the marker genes are more expressed in out-of-cluster cells. In particular, when κ gets too high, e.g. $\kappa = 0.12$, the penalty is too strict and only 2 genes are selected. Such a value is thus not

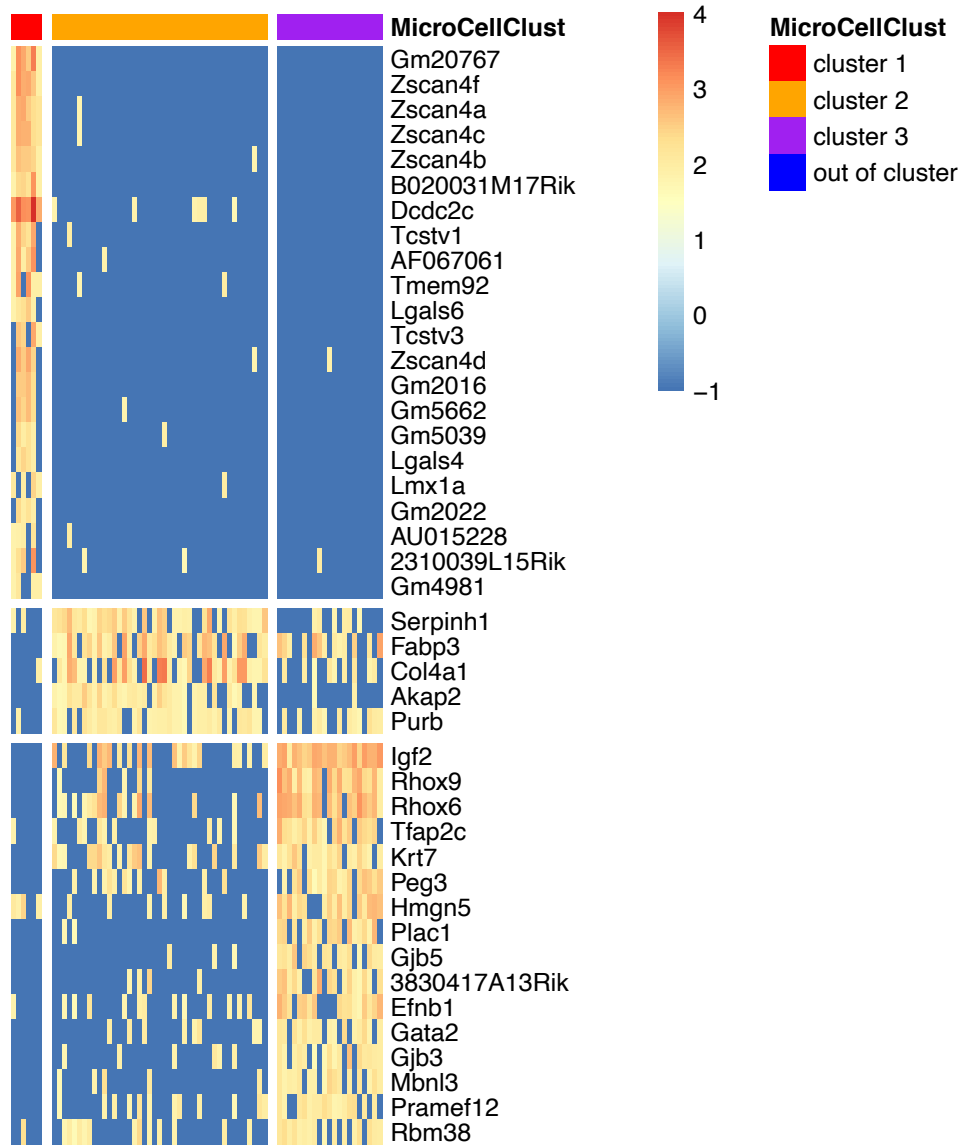


Figure 4.12: The 3 subpopulations identified in mouse embryonic stem cells when using $\mu = 20\%$. For clusters 2 and 3, $\kappa = 0.05$; for cluster 1 the marker genes identified with $\kappa = 1$ are reported (which is a subset of the genes obtained with $\kappa = 0.05$).

adapted for this particular subpopulation. The result with $\mu = 20\%$ is slightly less interesting but reported in figure 4.12 for completeness: a larger number of cells is selected but to some relevant marker genes (such as Col4a2) no longer belong to the solution.

MicroCellClust identifies a third bicluster made of 3 marker genes (Igf2, Rhox9 and Rhox6) that are expressed in a relatively significant number of cells (9% of the data). These 3 genes are also identified with different combinations of parameter values (in a varying number of cells depending on these values). Interestingly, one particular combination, $\kappa = 0.05$ and $\mu = 20\%$ identifies these genes together with 13 other genes, expressed in 21 of the 62 previously identified cells. These cells could be seen as a subcluster within the 62 cells, as they coexpress other marker genes besides Igf2, Rhox9 and Rhox6 (see cluster 3 in figure 4.12).

4.4 Discussion

The results reported in this chapter show that MicroCellClust is effective at identifying *small* subpopulations of cells with *specific* gene expression in small and middle scale scRNA-seq data. It outperforms existing rare cell identification methods at identifying target rare subpopulations, especially when they contain only a few cells. MicroCellClust also highlights several subpopulations of interest in T cells extracted from breast tumor samples, as well as in mouse embryonic stem cells. Moreover, MicroCellClust is stable with respect to changes in the data. For instance, the same marker genes are identified when performing multiple experiments where different Treg cells are sampled to form a rare subpopulation. This shows its usefulness to identify biomarkers, as opposed to traditional feature selection which can be highly unstable with respect to changes in the data (Hamer, 2022).

Ensuring the identification of rare cells

Paradoxically, there actually is no mechanism that specifically targets rare cells in the problem definition (4.1)(4.2) itself. Indeed, the maximization induced by the global sum criterion somewhat contrasts with the objective of identifying *small* subpopulations of cells. One execution of MicroCellClust on the original could therefore first identify a large subpopulation of cells, as it will have a large global sum (i.e. maximizing the first term of equation (4.1)). Yet, by performing multiple executions (see section 4.1.3), MicroCellClust will be able to detect other,

potentially rare, subpopulations. We therefore argue that the MicroCellClust optimization problem has the *capacity* to detect rare cells.

In practice, MicroCellClust highlights rare subpopulations of cells thanks to the initial filtering of genes, whereby genes expressed in more than 25% (by default) of the cells are left out. Indeed, genes expressed in many cells may lead to a trivial solution (in the limit a solution where $J = C$). Instead of this filtering, one could in principle use the multiple execution approach to identify such a large bicluster in the first result, and then filter out these genes to perform an additional execution to search for a rare subpopulation of cells. In the limit, one could relax the constraint on the proportion of negative values allowed in the bicluster by setting $\mu = 100\%$ in this first execution to identify the largest possible bicluster. This would boil down to the max-sum submatrix approach (see section 3.1), whose purpose is precisely to identify large biclusters of cells and genes.

This approach has been evaluated on the GARP+ Tregs case study from section 4.3.2. It yields a bicluster containing 3909 genes and all cells. Performing a second execution of MicroCellClust (with default $\mu = 10\%$) on the remaining 15 771 genes returns the same subpopulation of 21 cells, with 47 genes (instead of 33 when using the 25% cutoff). It turns out this corresponds to removing genes expressed in more than 33% of the cells: roughly the same genes are kept (see figure 4.13), leading to the same MicroCellClust solution. Interestingly, even though the number of genes in the solution increases by almost half, there is only a 10% increase of the objective value. This makes sense as these genes will have more out-of-cluster expression, which will be penalized by the second term of equation (4.1) and lead to a low contribution to the objective value. Changing the filtering threshold therefore has little influence on the identified cell subpopulation.

Given the computational overhead of performing multiple executions, we favor the pragmatic approach of defining a fixed threshold. Simply removing the genes expressed in more than 25% of the cells when preprocessing the data gives good results in all experiments. As discussed in sections 3.2.1 and 4.1.1, alternative data preprocessing could also be considered to search for other kind of patterns, rather than simply ignoring these genes. One could for example keep all genes but invert the sign for those expressed in more than half of the cells. This would lead to select genes that are underexpressed in the selected cells. However, such genes have not been observed in any of the performed experiments.

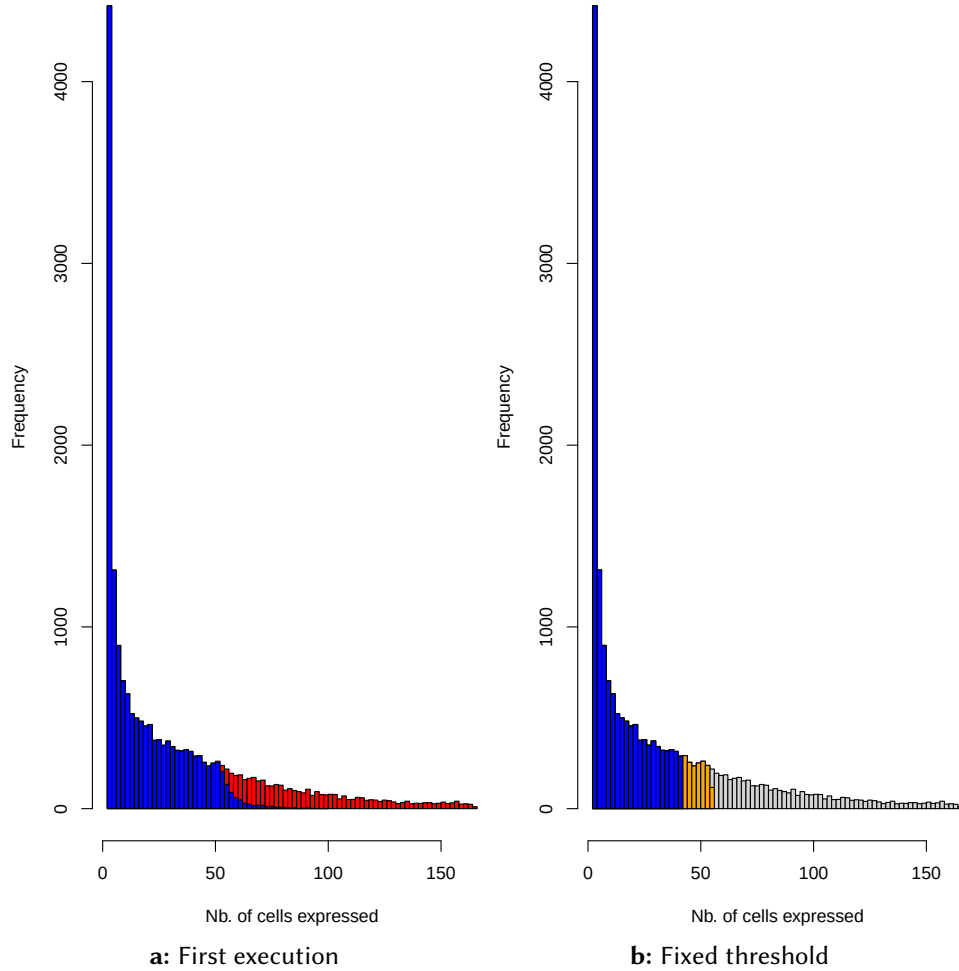


Figure 4.13: Histogram of the genes, in function of the number of cells they are expressed in, for the GARP+ Tregs data. Only genes expressed in at least two different cells are considered. **a** In red, the genes forming part of the MicroCellClust solution with $\mu = 100\%$, which would be removed in subsequent executions to identify rare subpopulations of cells. **b** Genes kept after a 25% (blue) and 33% (orange) threshold.

Influence of the parameters

The proposed optimization problem relies on two parameters: κ and μ . The reported experiments suggest to choose $\kappa = \frac{100}{|C|}$ and $\mu = 10\%$ by default. These parameters may be further tuned in order to refine the solution, typically based on the number of returned marker genes. Opting for approximately 20 to 30 such genes is good rule of thumb in this regard in order to conduct a subsequent biological interpretation. In all experiments, such a tuning has been easily conducted without the need of any additional prior knowledge.

The general trend that can be observed in figures 4.7, 4.9, and 4.11, is that the number of selected genes is inversely correlated to the value of κ . Indeed, whenever κ is increased, the number of genes in the solution decreases. This makes sense given the definition of the MicroCellClust optimization problem, as it inflicts a larger penalty on out-of-cluster expression and therefore only keeps the most differentially expressed genes. One should note that tuning this κ parameter didn't change the biological interpretation of the biclusters in any of the experiments. One might therefore choose the value of κ based on the number of genes that is deemed appropriate for visualization purposes and/or biological interpretation (e.g. using a Gene Ontology analysis).

The μ parameter controls the maximum proportion of negative values that is allowed within the bicluster. By default, it is set to 10%. Increasing this parameter (for instance to 20%) naturally leads to more cells and genes being included in the solution. Indeed, the number of genes increases as genes that contain more negative values (e.g. have higher dropout rates) can now be included. We therefore suggest to increase the value of μ whenever the number of genes returned with $\mu = 10\%$ is very low. Increasing μ also leads to an increase in the number of cells, as out-of-cluster cells that express a significant part of the genes identified with $\mu = 10\%$ may then be included. We therefore suggest to increase μ whenever a visual inspection of the results shows that some out-of-cluster cells express a significant part (for instance 2/3) of the marker genes.

Scope of MicroCellClust

The case studies reported in this chapter consider small and middle-scale data (typically ≤ 1000 cells). Indeed, the primary motivation for the MicroCellClust approach is to identify small subpopulations in a cell population that has been isolated as relatively homogeneous (e.g. intra-tumoral Tregs isolated with FACS). MicroCellClust thus focuses on studies for which the cells to be analyzed are

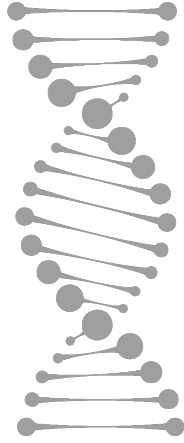
scarce (either due to the limited availability of samples and/or their limited abundance in the considered tissue). In such data, rare cell types only represent a handful of cells, which make them particularly difficult to detect (compared to large-scale data where such rare cells types could be represented by hundreds of cells). This is precisely the situation in which MicroCellClust performs particularly well (see table 4.3). The beam search solver (see section 5.2) used to produce these results easily scales to such data.

Yet, it is also possible to use MicroCellClust on large-scale datasets (typically order of 100 000 cells) thanks to the use of rareness scores as heuristics (see chapter 6). Such data is typically generated to perform a general characterization of abundant and easily accessible material, i.e. identify all cell types present in a very heterogeneous sample. The case studies reported in sections 6.2.1 and 6.2.2 show that MicroCellClust 2 is also able to identify rare and biologically relevant groups of cells in such a context.

MicroCellClust takes as input a single data matrix representing all gene expression values for a population of cells typically coming from a single biological sample. Using the same procedure, one can easily analyze the data coming from a few samples simply put together. A natural extension would consider the sample (or donor) identity as a third dimension, beyond genes and cells. It would indeed be interesting to find specific subpopulations of cells and their associated marker genes that would also be common across different donors or conditions. Chapter 7 adapts the optimization problem introduced here to address this generalized objective.

Efficient solving of the optimization problem

5



The previous chapter formalizes the goal of rare cell mining as an optimization problem, the solving of which produces a cell/gene bicluster characterizing a rare subpopulation present in the data. For the biologist, it can in principle be seen as a black box, i.e. a mathematical problem with a unique solution, which he then can interpret in function of the cells and genes returned. Yet, from a computational point of view, producing the optimal solution is all but trivial. This chapter thus focuses on the “computer science part”, i.e. how can such a result be efficiently found.

Section 5.1 first states the implicit search space of MicroCellClust, reducing the search space to consider to only the subsets of cells. Section 5.2 defines a search strategy to solve small to middle scale instances, i.e. search for unexpected patterns in data that is often limited in availability, which is the prime focus of MicroCellClust. A beam search approach is presented, which produces high-quality results in a limited amount of time.

5.1 Implicit search space

A naive approach to solving the MicroCellClust problem would be to enumerate all possible combinations of genes and cells, i.e. evaluating $2^{|\mathcal{G}|+|\mathcal{C}|}$ potential biclusters. Yet, it turns out that whenever one dimension is fixed (typically the cells), the optimal solution along the other dimension (the genes) can be found in linear time. Indeed, the objective function (4.1) of MicroCellClust:

$$(I^*, J^*) = \underset{\substack{I \subseteq \mathcal{G} \\ J \subseteq \mathcal{C}}}{\operatorname{argmax}} \sum_{i \in I} \left(\sum_{j \in J} m_{ij} - \kappa \sum_{k \in \mathcal{C} \setminus J} \max\{0, m_{ik}\} \right) \quad (5.1)$$

can be formulated as a sum of individual gene contributions ω_i :

$$(I^*, J^*) = \operatorname{argmax}_{\substack{I \subseteq \mathcal{G} \\ J \subseteq \mathcal{C}}} \sum_{i \in I} \omega_i(J) \quad (5.2)$$

with:

$$\omega_i(J) = \sum_{j \in J} m_{ij} - \kappa \sum_{k \in \mathcal{C} \setminus J} \max\{0, m_{ik}\} \quad (5.3)$$

For a given cell cluster $J \subseteq \mathcal{C}$, the corresponding set of genes $I_J^* \subseteq \mathcal{G}$ that maximizes objective (4.1) contains all genes that contribute positively to this objective, i.e. have positive $\omega_i(J)$ values:

$$I_J^* = \{i \in \mathcal{G} \mid \omega_i(J) \geq 0\} \quad (5.4)$$

This set can be evaluated in linear time, reducing the search space to consider to $\mathcal{O}(|\mathcal{G}| \cdot 2^{|\mathcal{C}|})$.

5.1.1 Constraint on negative values

While this property holds for unconstrained optimization, it is not strictly the case when considering the global constraint (4.2):

$$\frac{|\{(i, j) \mid i \in I, j \in J, m_{ij} < 0\}|}{|I| \cdot |J|} \leq \mu \quad (5.5)$$

Indeed, simply selecting all genes with positive contributions could violate this constraint, since some might have many negative values within the bicluster, leading to a global proportion larger than μ . Once all such genes are identified, one still needs to select a subset of them which respects this constraint, while maximizing the objective.

A straightforward linear approach would be to greedily add genes one by one, by decreasing order of ω value, until the inclusion of a new gene violates the constraint. Such an approach is not guaranteed to select the optimal set of genes, as for instance a gene with two negative values could be replaced by two genes, each having only one negative value, which jointly lead to a higher contribution.

A better procedure can be defined by observing that a large fraction of the genes with $\omega_i \geq 0$ can automatically be included, meaning a branching need only be performed on a subset of them. Intuitively, one can order the genes by their number of negative values within the selected cells. Genes with no such

values can of course be directly included. Also, genes with few negative values, i.e. less than $\mu|J|$, can also be included as they will only slightly increase the global proportion of negative values. It is only for genes whose number of negative values are above this threshold that choices need to be made.

Formally, by posing $n_i^\ominus(J)$ as the number of negative values of gene i within the selected cells J , constraint (5.5) can be rewritten as:

$$\frac{\sum_{i \in I} n_i^\ominus(J)}{|I| \cdot |J|} \leq \mu \quad (5.6)$$

$$\iff \sum_{i \in I} \left(\frac{n_i^\ominus(J)}{|J|} - \mu \right) \leq 0 \quad (5.7)$$

Hence, we can directly include all genes for which the contribution to this sum is negative, i.e. genes whose proportion of negative values within the selected cells is smaller than the μ threshold:

$$I_J^\ominus = \left\{ i \in I_J^* \mid n_i^\ominus(J) \leq \mu|J| \right\} \quad (5.8)$$

The decision that remains is to choose the genes in I_J^* that positively contribute to the sum in constraint (5.7), denoted $I_J^\oplus$, in order to maximize the objective function.

Let C be the remaining capacity, i.e. the proportion of negative values that can still be added in the bicluster before violating constraint (5.7):

$$C = \sum_{i \in I_J^\ominus} \left(\mu - \frac{n_i^\ominus(J)}{|J|} \right) \quad (5.9)$$

Choosing the genes to add to the bicluster corresponds to the *knapsack problem* (Dantzig, 1957; Martello and Toth, 1987):

$$\text{maximize } \sum_{i \in I_J'} \omega_i(J) \quad (5.10)$$

$$\text{such that } \sum_{i \in I_J'} \left(\frac{n_i^\ominus(J)}{|J|} - \mu \right) \leq C \quad (5.11)$$

The knapsack problem is a known NP-complete problem (Crescenzi et al., 1995). Yet, a very close approximation can in practice be found in linear time by greedily

selecting genes from $I_J^\oplus$ in increasing order of their negative values (and breaking ties by ω values). Indeed, the only suboptimal choice that could be performed in this case is one unselected gene, with n negative values, having a larger contribution than multiple selected genes which together have n negative values. Such a situation is however highly unlikely, as that gene would need very high expression values to have a larger objective contribution than multiple genes combined.

Moreover, such a situation cannot occur in the current implementation of MicroCellClust which also uses individual constraints for each gene with the same μ value (see equation 4.4 in section 4.1.2). Since $\mu|J|$ is rounded up in these constraints, they allow the inclusion of genes with *one* more negative value than those in $I_J^\ominus$. In other words, all genes in $I_J^\oplus$ have in this case the same number of negative values. Adding them greedily by decreasing order of ω in the knapsack therefore necessarily produces an optimal result.

5.1.2 Objective value computation

Using this procedure, finding the optimal assignment of genes for a certain group of cells is done in $\mathcal{O}(|\mathcal{G}| \cdot |\mathcal{C}|)$,¹ since computing each gene contribution ω_i according to equation (5.3) is done by summing (or subtracting) the expression values of all cells, both inside and outside the bicluster. One can improve the computational efficiency by precomputing the cumulative positive expression of each gene across all cells before starting the search:

$$cpe_i = \sum_{c \in \mathcal{C}} \max\{0, m_{ic}\} \quad (5.12)$$

One can then compute the objective contribution of each gene by subtracting the precomputed cpe value, then adding only the values of the selected cells, and compensating for the fact these cells also appear in cpe :

$$\omega_i(J) = \sum_{j \in J} (m_{ij} + \kappa \cdot \max\{0, m_{ij}\}) - \kappa \cdot cpe_i \quad (5.13)$$

The computational complexity thus becomes $\mathcal{O}(|\mathcal{G}| \cdot |J|)$. Given that searching for rare cell clusters implies $|J| \ll |\mathcal{C}|$, this gives a substantial computational speedup, as this operation is repeated many times during the search. Moreover, one can ignore the number of genes in the overall complexity, as they can be efficiently filtered.

¹Note that sorting the genes is in theory a $\mathcal{O}(|\mathcal{G}| \log |\mathcal{G}|)$ operation. Yet, since the genes are sorted by number of negative values, i.e. a discrete quantity, this operation can be performed in linear time using a bin sort.

Gene filtering

As discussed in section 4.1.2, defining a constraint for each gene allows to efficiently filter the $\mathcal{G}$ domain before computing the objective value of a bicluster, i.e. leaving out genes which one knows beforehand cannot satisfy its constraint.

A naive implementation to compute the optimal assignment of genes for a given set of cells J would compute the ω_i values for all genes in $\mathcal{G}$, and then filter out those for which the constraint (4.4) on the maximal proportion of negative values is violated. Given there are generally order of 10 000 genes in a scRNA-seq dataset, and each cell only express a small fraction of them, such an approach would be extremely inefficient.

Before starting the search for an optimal solution, one can precompute an expression dictionary which contains, for each cell, the list of genes that are expressed. When computing the objective value for a set of cells J , the domain of genes to be considered reduces to the intersection of the gene expression lists of each cell in J . Moreover, one can directly count the number of cells that don't express a particular gene using this dictionary. The genes violating constraint (4.4) can thus be filtered out before computing the ω_i values. This efficient domain filtering makes the number of genes negligible in the overall execution time. For the remainder of this thesis, we will ignore the number of genes and report the complexities only in terms of the number of cells.

5.2 Solving small scale instances

This section discusses how the NP-hard optimization problem (4.1)(4.2) of MicroCellClust can be efficiently solved for small to middle scale data (up to order of 1000 cells). This is motivated by the search for rare subpopulations in data that is often limited in availability (e.g. intra-tumoral Tregs), and for which rare cells are consequently present in very small amounts (a task at which MicroCellClust performs better than typical alternatives, as shown in section 4.2).

5.2.1 Branch and bound approach

A straightforward approach is to adapt the CPGC algorithm (see section 3.1.2) to the MicroCellClust objective. Indeed, it turns out the upper bound in equation (3.3) is also an upper bound of the MicroCellClust objective (equation (4.1)). However, this bound is much less tight for this objective than for the max-sum one. This results from the fact that the constrained optimization problem is de-

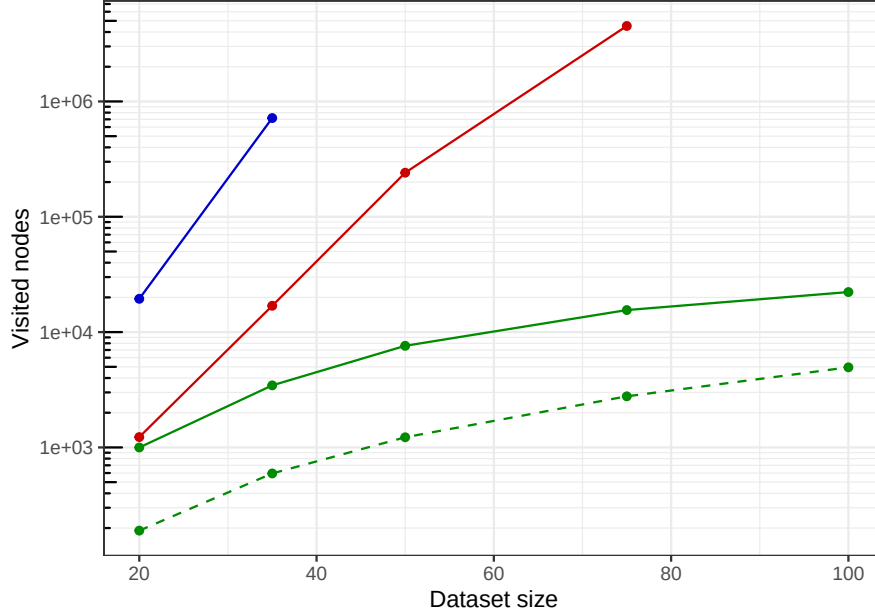


Figure 5.1: Number of biclusters (in logarithmic scale) evaluated for one of the runs of the GARP+/CD8+ experiment with $\mu = 0\%$. Blue: during a full search; red: branch-and-bound; and green: beam search with $t = 20$. The dashed line represents the number of pairs.

signed here to search for biclusters with a *small* number of cells, while the max-sum objective could lead to include many cells. Moreover, this bound is overly optimistic as the second term of equation (4.1), which subtracts the out-of-cluster expression, decreases the objective value compared to the max-sum objective for the same (I, J) bicluster. Consequently, there is often an important gap between the actual objective value and this (optimistic) upper bound, leading to a less efficient filtering.

For instance, the upper bound for the node corresponding to the 5 Tregs, i.e. the optimal solution of the GARP+ versus CD8 experiment, is 20 times higher than its objective value. This means the search will continue expanding a large part of the subtree under this node, even though the optimal solution has already been found. Indeed, figure 5.1 shows the number of visited nodes during such a search still follows an exponential trend with respect to the number of cells.

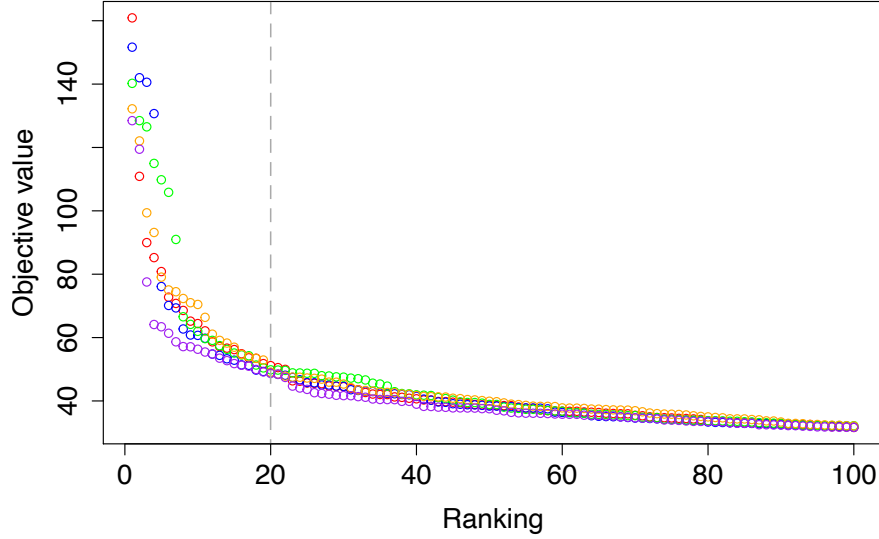


Figure 5.2: Distributions of the objective values of the 100 best pairs of cells for 5 runs of the GARP+ vs CD8+ controlled experiment. The distributions follow a power law the slope of which becomes small around rank 20.

5.2.2 Beam search

To find a more efficient way to search for the MicroCellClust solution, one can take advantage of the expected similarity between the different selected cells to evaluate, at an early stage of the search, which combinations of cells are most likely to produce an optimal solution. Indeed, due to the high specificity of the optimal bicluster induced by constraint (4.2), any subset of the selected cells is expected to form a promising solution, the objective value of which is above average compared to a random subset of cells of the same size.

When plotting the objective values of all possible pairs of cells in decreasing order, one indeed observes this distribution follows a power law. Figure 5.2 shows the distribution of pairs for 5 resamplings of the GARP+/CD8+ experiment (see section 4.2.1). A visual inspection of this plot illustrates that the ≈ 20 first biclusters have an objective value above average (past this point the slope of the distribution becomes small, corresponding to the long tail of the power law). It is assumed that at least one of these pairs must be included in the cell subset of the optimal solution. This suggests to only consider these pairs to continue the search, and ignore the long tail of the distribution.

This motivates solving the MicroCellClust optimization problem by using a *beam search*, which is a heuristic search strategy that explores the search space by only expanding the most promising solutions of a certain level of the search tree. It follows a *breadth-first* strategy, where at each level of the search the solver selects the b biclusters with highest objective values to continue the search. Only supersets of the set of cells they include are evaluated at the next level (see algorithm 5.1). It thus first evaluates all possible pairs of cells, selects the b ones with highest objective to generate supersets of 3 cells (by adding each possible cell, one at a time, to these pairs), then evaluates them and selects the b best ones to generate candidates with 4 cells, and so forth until no improvement in objective value is found during several successive search levels (figure 5.3).²

Beam size

Such a beam search algorithm has no guarantee of finding an optimal solution to problem (4.1)(4.2). Indeed, if the beam size b is too small, one could miss the optimal solution if none of its subsets is selected during one of the search levels. Since beam search follows a greedy strategy that never reevaluates previously made choices, the result will be suboptimal. However an optimal solution can be found with high probability whenever b is appropriately chosen. Choosing the beam size as the rank at which the long tail of the distribution of pairs seems to start (see figures 5.2 and 5.4) proves able to identify biologically relevant solutions on small and middle-scale data, as shown in sections 4.2 and 4.3 (table 5.1 summarizes the values of b used for each of these experiments).

We use the GARP+/CD8+ experiment to compare the performance of the beam search with an exact procedure. Figure 5.1 shows the number of visited nodes during branch-and-bound (exact search) and beam search for one run when varying the number of cells. The metaparameter μ is set here equal to 0%, which implies that such a strict non negativity constraint already eliminates a lot of candidate solutions. Yet, branch-and-bound still has an exponential complexity as a function of the number of cells. In contrast, the beam search has a polynomial complexity. Its computational bottleneck is the evaluation of all pairs at level 2, i.e. $\frac{|\mathcal{C}|^2 - |\mathcal{C}|}{2} \in \mathcal{O}(|\mathcal{C}|^2)$ (dashed line on Fig. 5.1). Moreover, the beam search proves

²A naive implementation to choose the b best solutions at each level of the beam search would store all solutions and sort them. We use a more efficient approach using a priority queue to store a fixed number (b) of current best solutions in memory. Inserting a new one is done linearly with respect to the size of the queue, and the worst one is discarded in constant time to keep a size of b . This behavior is represented in algorithm 5.1 by the `b_best` function.

Algorithm 5.1: MicroCellClust beam search

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ an expression matrix

Input: $\kappa \in \mathbb{R}_+$, $\mu \in [0, 1]$ the parameters of problem (4.1)(4.2)

Input: $b \in \mathbb{N}$ a beam size

Output: $I^* \subseteq \mathcal{G}$, $J^* \subseteq \mathcal{C}$, $\Omega^* \in \mathbb{R}$ sets of genes and cells maximizing problem (4.1)(4.2), and the corresponding objective value

$I^*, J^*, \Omega^* \leftarrow \emptyset, \emptyset, 0$

// Evaluate pairs of cells (search level 2)

$beam_{\ell=2} \leftarrow \emptyset$

for $j \in \mathcal{C}$ **do**

for $j' \in \{j' \in \mathcal{C} \mid j < j'\}$ **do**

$J \leftarrow \{j, j'\}$

$I, \Omega \leftarrow \text{obj}(\mathbf{M}, J, \kappa, \mu)$

$beam_{\ell=2} \leftarrow b_best(beam_{\ell=2} \cup \{J\})$

if $\Omega \geq \Omega^*$ **then** $I^*, J^*, \Omega^* \leftarrow I, J, \Omega$

└┐

// Evaluate next search levels

for $\ell \in \{3, 4, \dots\}$ **do**

$beam_{\ell} \leftarrow \emptyset$

for $J^0 \in beam_{\ell-1}$ **do**

for $j \in \mathcal{C} \setminus J^0$ **do**

$J \leftarrow J^0 \cup \{j\}$

$I, \Omega \leftarrow \text{obj}(\mathbf{M}, J, \kappa, \mu)$

$beam_{\ell} \leftarrow b_best(beam_{\ell} \cup \{J\})$

if $\Omega \geq \Omega^*$ **then** $I^*, J^*, \Omega^* \leftarrow I, J, \Omega$

└┐

└┐

return I^*, J^*, Ω^*

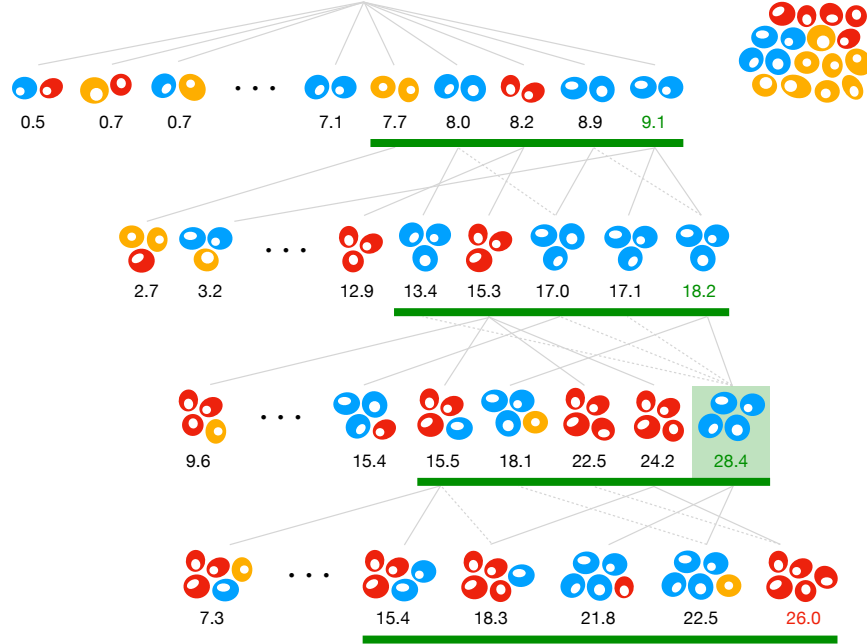
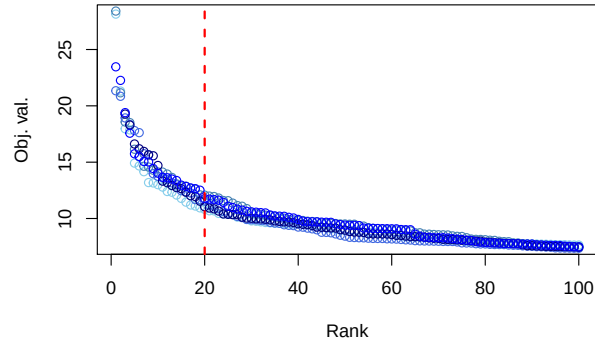


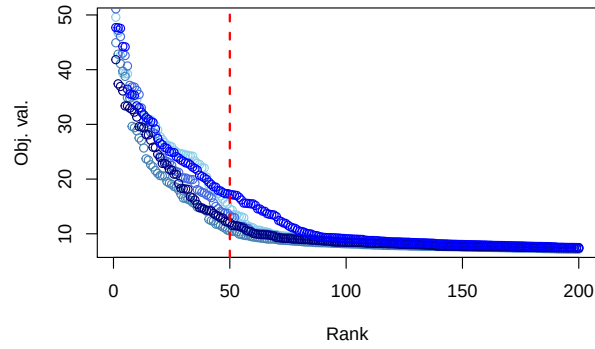
Figure 5.3: Graphical representation of the execution of a beam search. This illustration represents the search tree corresponding to a fictional dataset containing 17 cells, of 3 different types. The solutions underlined in green are those selected at each level of the beam search, with $b = 5$. Only children of these solutions are generated at the next level of the search. An optimal solution is here found at level $\ell = 4$, as no higher value solution is found in subsequent levels (only one is represented here).

to be accurate for this experiment. The beam search with $b = 20$ produces the same results as the exhaustive search for each of the 100 independent runs reported in section 4.2.1. With a lower value such as $b = 10$, the optimum solution is missed only for 15 out of 100 runs. Yet, the solutions returned in these case are close in terms of both objective value and genes selected, i.e. with no impact in terms of interpretation of the result.

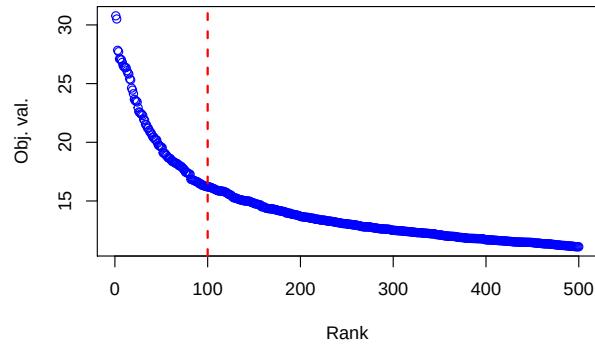
The impact of the beam size is further evaluated on the breast cancer CD4+ dataset (section 4.3.1), for which a value of $b = 100$ was chosen based on a visual inspection of the distribution of pairs (figure 5.4c). Such a value seems appropriate as even a much higher one of $b = 500$ returns the exact same solution with 24 cells and 23 genes. It turns out $b = 100$ is actually a very safe choice as one needs to go as low as $b = 2$ to get a suboptimal result. Even in this very extreme



a: 5 runs of the Jurkat vs 293T experiment with 10 rare cells among total of 100 cells ($\kappa = 1$)

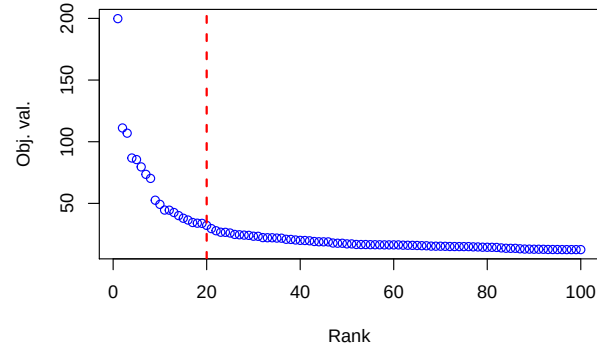


b: 5 runs of the Jurkat vs 293T experiment with 10 rare cells among total of 1,000 cells ($\kappa = 0.1$)

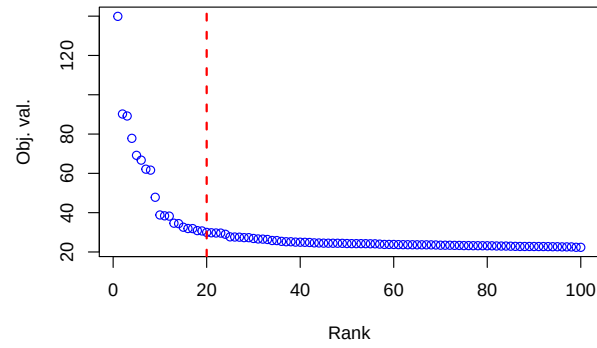


c: Breast cancer CD4+ ($\kappa = 0.5$)

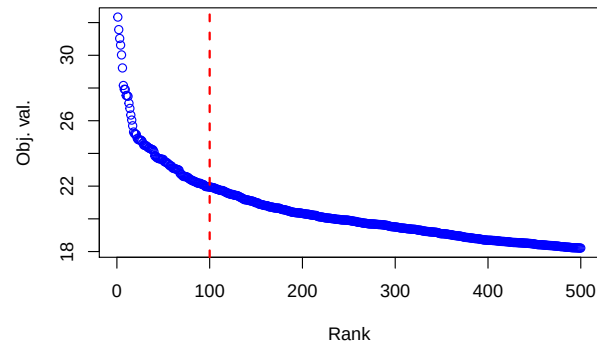
Figure 5.4: Distributions of the objective values of pairs of cells. The dashed red lines indicate the chosen values for the beam size b , i.e. the rank where the long tail starts. Figure continues on page 102.



d: Breast cancer GARP+ ($\kappa = 0.6$)



e: Mouse stem cell ($\kappa = 0.1$)



f: Mouse stem cell ($\kappa = 0.1$) after removing the cells of cluster 1. This has a large impact on the shape of the distribution, and motivates a higher value of $t = 100$ to find the two next biclusters.

Figure 5.4: Distributions of the objective values of pairs of cells. The dashed red lines indicate the chosen values for the beam size b , i.e. the rank where the long tail starts. Figure starts on page 101.

Dataset	Section	Figure	b
GARP+ vs CD8+	4.2.1	5.2	20
Jurkat vs 293T (100 cells)	4.2.2	5.4a	20
Jurkat vs 293T (1,000 cells)	4.2.2	5.4b	50
Breast cancer CD4+	4.3.1	5.4c	100
Breast cancer GARP+	4.3.2	5.4d	20
Mouse stem cell (cl. 1)	4.3.3	5.4e	20
Mouse stem cell (cl. 2 & 3)	4.3.3	5.4f	100

Table 5.1: Values for the beam size b used for the different case studies. The 2nd column refers to the related section. The 3rd column refers to the figures from which the values of b are inferred.

case, the result still contains 19 out of the 23 genes from the optimal solution, meaning there is no impact in terms of biological interpretation (indeed, similar GO functions are returned in both cases). This tendency is confirmed with the GARP+ Treg dataset (section 4.3.2), where a low value of $b = 4$ still produces the same result as $b = 20$ (chosen based on figure 5.4d). In conclusion, the value of the beam size parameter b is not critical to the final result of the beam search, and has no impact on the biological interpretation of the result.

Efficient solution generation

As the CPGC algorithm uses a depth-first approach, it efficiently computes the ω_i values (i.e. the contributions of each gene to the objective; see equation (5.13)) by using a reversible variable $psums_i$ for each gene. Indeed, a new assignment of cells only differs from the previous one by one cell. Computing an ω_i value simply amounts to adding the expression value of the new cell to $psums_i$, which is done in constant time. The reversible variable maintains previous states on a stack, enabling it to return to a previous value when backtracking in the search.

Such an approach can be directly implemented in the local search algorithm used in the second part of the MicroCellClust 2 solver. However, it is not readily applicable in a breadth-first setting, as one needs to evaluate all children nodes of all previous solutions before going deeper into the search. Yet, one can mimic such an approach by observing large groups of solutions of a search level share the same parent.

Indeed, new solutions are generated by adding, one at a time, all remaining cells to solutions from the previous search level. As computing the ω_i values is performed in $\mathcal{O}(|J|)$ (see equation (5.13); we here ignore the complexity in terms of number of genes), a naive approach would, for each previous solution in the beam, perform $\mathcal{O}(|\mathcal{C}|)$ times this $\mathcal{O}(|J|)$ computation (see algorithm 5.2).

Alternatively, one can compute the ω_i values of the parent once, and then add in constant time the expression values of each new cell (see algorithm 5.3). The complexity becomes:

$$\mathcal{O}(|J - 1|) + \mathcal{O}(|\mathcal{C}|) \cdot \mathcal{O}(1) \quad (5.14)$$

which gives a substantial gain in execution time.

This procedure has the added advantage that, when precomputing the ω_i values for all genes, one can filter out those that are guaranteed to violate constraint (4.4) on the maximal proportion of negative values, further reducing the execution time.

5.2.3 Execution time and memory analysis

The computational cost of the beam search solver is analyzed using the Jurkat versus 293T controlled experiment described in section 4.2.2. These experiments are executed on a MacBook Pro laptop.³ The beam search stops the search whenever no improvement of the current best objective value is found during, by default, 25 search levels after the best solution. Here, we consider datasets with 10% of rare cells, i.e. the worst case scenario in terms of execution time, and use $b = 20$ as beam size.

Table 5.2 and figure 5.5 show that the execution time follows a quadratic distribution in terms of the number of cells. A quadratic regression is performed to infer the execution time for larger datasets. This model predicts an execution time of 22 minutes for a dataset of 1,500 cells. We measured 20.24 minutes when running such an instance of the Jurkat vs 293T data, confirming the quadratic assumption. The data reported in table 5.2 also indicate that the memory load increases linearly with the number of cells.

³Mac OS 10.15.7; 2.7 GHz Intel Core i7 CPU; 16 GB RAM.

Algorithm 5.2: Computation of the solutions: naive approach

```

for  $J^0 \in \text{beam}_{\ell-1}$  do
  for  $j \in \mathcal{C} \setminus J^0$  do                                     //  $\mathcal{O}(|\mathcal{C}|)$ 
     $J \leftarrow J^0 \cup \{j\}$ 
     $I \leftarrow \emptyset$ 
    for  $i \in \mathcal{G}$  do
      Compute  $\omega_i(J)$                                        //  $\mathcal{O}(|J|)$ 
      if  $\omega_i \geq 0$  then  $I \leftarrow I \cup \{i\}$ 

```

Algorithm 5.3: Computation of the solutions: efficient approach

```

for  $J^0 \in \text{beam}_{\ell-1}$  do
  // Precompute the contributions of each gene
  for  $i \in \mathcal{G}$  do
    Compute  $\omega_i^0(J^0)$                                        //  $\mathcal{O}(|J|)$ 
  for  $j \in \mathcal{C} \setminus J^0$  do                                     //  $\mathcal{O}(|\mathcal{C}|)$ 
     $J \leftarrow J^0 \cup \{j\}$ 
     $I \leftarrow \emptyset$ 
    // Add the expression values of the new cell
    for  $i \in \mathcal{G}$  do
       $\omega_i \leftarrow \omega_i^0 + m_{ij}$                          //  $\mathcal{O}(1)$ 
      if  $m_{ij} \geq 0$  then  $\omega_i \leftarrow \omega_i + \kappa \cdot m_{ij}$ 
      if  $\omega_i \geq 0$  then  $I \leftarrow I \cup \{i\}$ 

```

Table 5.2: Execution time and memory load during the Jurkat vs 293T experiment. The observed values are averaged over 10 runs (with 10% of rare cells). A quadratic regression is performed to infer the execution time for larger datasets.

a: Observed time and memory			b: Quadratic regression	
Number cells	Exe. time [min]	Memory [MB]	Number cells	Exe. time
50	0.03	140	1,500	22 min
100	0.16	180	2,000	39 min
200	0.49	333	2,500	1 h
500	2.50	705	5,000	4 h
1,000	9.87	1770	10,000	16 h
			20,000	5 days

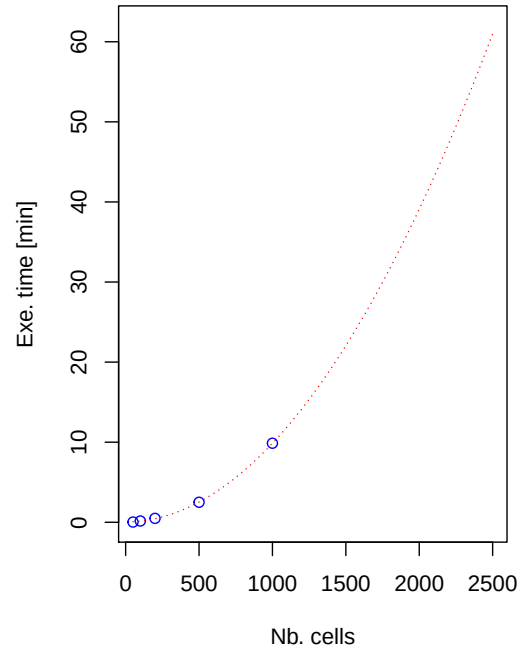


Figure 5.5: Evolution of execution time during the Jurkat vs 293T experiment. The observed values (average of 10 runs with 10% of rare cells) are represented by the blue dots. The dashed red line represents a quadratic regression.

5.3 Discussion

In conclusion, the proposed beam search solver is effective to solve the small to middle-scale instances presented in chapter 4. This choice is motivated by the fact that this application doesn't require an optimal solution, in the sense that the objective value to be maximized is not directly linked to a real-world utility (contrary to problems minimizing distances, energy consumption, etc.). The actual utility will be measured by the subsequent biological interpretation (for instance in terms of a GO analysis). As such, a domain expert will probably favor a high-quality approximation fast, rather than having to perform days of computation on a supercomputer.

The choice of a beam search is further relevant given the fact it is especially well suited for the identification of rare subpopulations. Indeed, whenever multiple rare subpopulations coexist in the data, it sometimes happens the beam search first returns the smallest one, even if it has a lower objective value than another one. Such behavior is explained by the selection of the best biclusters in the first levels of the beam search. The assumption is that any subset of a (locally) optimal solution is expected to form a promising solution the objective value of which is above average. While this is indeed observed in practice, the objective values of these small biclusters (e.g. pairs of cells) are penalized by the expression of the selected genes by the other cells from that subpopulation, which are still out-of-cluster at this point. Whenever two or more rare subpopulations of relatively different size coexist in the data, pairs of cells coming from the smallest one will logically be less penalized as there is less out-of-cluster expression. The beam might thus be filled by these solutions during the first levels of the search, meaning the beam search will focus on that subpopulation and no longer consider the other ones at later stages of the search (even if they turn out to have a larger objective value at that point).

A smaller solution being returned first is observed in the mouse embryonic stem cell case study (see section 4.3.3). The 8-cell and 18-cell biclusters identified during the two first executions of MicroCellClust have actually a much lower objective value than the 61-cell bicluster identified in the third execution (respectively 46% and 35%). Yet, the 8-cell bicluster is first identified as the pairs of cells it contains have a more than 4 times higher objective value than the pairs from the 61-cell subpopulation, meaning only these pairs are selected for further expansion. Interestingly, while the pairs from the second (18-cell) and third (61-cell) bicluster have roughly the same objective value, only the former improves

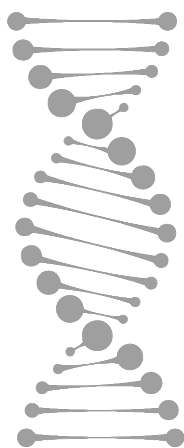
the objective value during the first search levels, meaning subsets of the 61-cell subpopulation progressively disappear from the beam.

We argue that such an unintended behaviour is actually not undesirable, as the rarest subpopulations are the most difficult ones to detect, and this is precisely the strength of MicroCellClust compared to typical alternatives (see section 4.2.2). Indeed, as each local optimum of the optimization problem is *a priori* of interest for the subsequent biological analysis, as it represents a specific subpopulation, it can be beneficial to start with the ones that are most easily missed.

This is especially the case for large-scale data, containing tens to hundreds of thousands of cells, as rare cell types may come in varying proportions. There could be clusters of a few tens up to multiple thousands of cells that could be considered as rare. Given that larger rare subpopulations are relatively easily detected, there would be little benefit to using MicroCellClust to identify them. Its interest would reside in identifying very small subpopulations, for instance containing a few tens of cells representing less than 1% of the total. However, the proposed beam search solver would fall short when applied on such data due to its quadratic complexity. Chapter 6 therefore presents a hybrid approach with extra heuristics in order to scale MicroCellClust to such data.

Multivariate rare cell mining in large-scale single-cell data

6



The primary motivation for the MicroCellClust approach is to identify small subpopulations of cells in small and middle-scale data (typically ≤ 1000 cells), i.e. when rare cell types are generally represented by only a handful of cells. The quadratic complexity of the beam search solver easily scales to such data, but would be less practical for a significantly larger number of cells. Extra heuristics are used to tackle large-scale datasets (order of 100 000 cells), which are now commonly generated. Consequently, MicroCellClust 2¹ can be used for the task of rare cell type characterization in abundant data, with a multivariate approach that contrasts with typical alternatives designed to scale to large data.

Section 6.1 proposes a hybrid approach, that relies on a priori computed rareness score to efficiently filter out cells and solve the MicroCellClust optimization problem with a linear time complexity. Section 6.2 shows the application of MicroCellClust 2 on two large-scale scRNA-seq datasets, and its ability to highlight biologically relevant subpopulations with a linear time complexity.

¹*Related publication:* Gerniers, A., and Dupont, P. (2022). MicroCellClust 2: a hybrid approach for multivariate rare cell mining in large-scale single-cell data. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 148–153.

6.1 MicroCellClust 2: a hybrid approach

The quadratic complexity of the beam search comes from the first step, i.e. the generation and evaluation of all possible pairs of cells. The goal is to identify pairs of similar cells, which will consequently yield a higher objective value than a random pair. In order to use the beam search for large-scale data, one must find these best pairs while generating only a subset of all possible combinations.

Let us assume a *rareness score* which assigns a high value to cells that present similarity with few other cells, i.e. are rare, and a low value to abundant ones. One can reasonably assume that two similar cells (rare or not) should also have relatively similar rareness scores, and therefore restrict the evaluation to pairs of cells that are similar within this score distribution instead of all $\mathcal{O}(|\mathcal{C}|^2)$ pairs. The FiRE (Jindal et al., 2018) or DoRC (Chen et al., 2019) methods precisely compute such a score (see section 2.3.3). Finding the best pairs is then done in linear time by pairing each cell to the n closest ones according to the rareness score ($n \ll |\mathcal{C}|$; typically 100).

Pairing cells according to a rareness score reduces the complexity of the beam search from quadratic to linear.² Additional heuristics are used to further improve computational speed when considering a large number of cells. Indeed, most of them have a low rareness score, and will unlikely be part of any rare subpopulation. One could preselect cells that have a high rareness score, and consider only those as variables of the optimization problem (i.e. only these cells can be selected in J). In other words, the search is restricted to the submatrix $\mathbf{M}_{\mathcal{G}C_{rare}}$ of the original matrix $\mathbf{M}$,³ where C_{rare} denotes the set of cells that are designated as rare by FiRE or DoRC. Both of them consider a cell as rare if its score is $\geq q_3 + 1.5 \cdot IQR$, where q_3 denotes the third quartile of the distribution and IQR the interquartile range. Note that this results form a collection of rare cells, but not necessarily *one* rare subpopulation. Since FiRE or DoRC do not contain any clustering mechanism, these cells might come from several different subpopulations.

²To produce this pairing, the cells must be sorted by their rareness score, which is $\mathcal{O}(n \log n)$ in general. Yet, linear sorting algorithms exist provided the rareness score is rounded to a fixed precision (which is sufficient for the proposed approach). In practice, this sorting operation is negligible in the overall runtime.

³Note that the cells not selected as rare should still be considered in the second term of objective (4.1) that penalizes the out-of-cluster expression. Yet, this is simply done using the precomputed sum of expression, as shown in equations (5.12) and (5.13), meaning only the submatrix of rare cells is needed to compute the objective values and perform the search.

Using such heuristics greatly reduces the search space (typically less than 10% of the cells remain under consideration), but relies on FiRE/DoRC having a high recall, i.e. most interesting cells must be preselected (possibly among others). Results described in section 6.2.2 show this recall is not always high enough: some cells might not be included in the MicroCellClust solution as they failed the rareness threshold, even though they express a large part of the genes selected in the bicluster (and are therefore probably part of the same cell subpopulation). Yet, this can easily be detected so as to extend the search to these cells (see steps 3 and 4 below). We therefore propose a hybrid procedure to run MicroCellClust on large-scale data (the pseudocode of which is given in algorithm 6.1):

1. Compute the rareness score for each cell using the FiRE (or alternatively DoRC) method, and select the cells passing the rareness threshold.
2. Run the beam search algorithm with only those cells as variables of the optimization problem. The objective is to identify at least part of the subpopulation of interest, with an initial set of marker genes.
3. Create a new search space containing cells that express part of the genes found in step 2 (whether or not they were previously candidates after step 1). Selecting cells expressing at least 50% of these genes seems appropriate according to our experiments to ensure a wide enough selection of candidate cells.
4. Take these cells as variables and search for an optimal solution by performing a local search starting from the solution of step 2.

This hybrid strategy efficiently explores the search space and produces results with high objective values in linear time, allowing the analysis of large-scale datasets. Indeed, the beam search using FiRE/DoRC is particularly effective at identifying a good approximate solution, i.e. it quickly focuses on the zone of interest. On the other hand, a local search algorithm thoroughly explores this zone and therefore yields solutions with higher objective value than the beam search.

Beam search

The pseudo-code of the beam search used by MicroCellClust 2 is given in algorithm 6.2. The number n of pairs to form for each cell, as well as the beam size b (i.e. the number of solutions to keep after each level), have been chosen empirically as their values are not critical to the end result (since one only looks for an approximate solution at this point). A value of 100 produces good results on all

Algorithm 6.1: MicroCellClust 2

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ an expression matrix
Input: $\mathbf{r} \in \mathbb{R}^{|\mathcal{C}|}$ a rareness score distribution for the cells
Input: $\kappa \in \mathbb{R}_+$, $\mu \in [0, 1]$ the parameters of problem (4.1)(4.2)
Input: $n_r \in \mathbb{N}$, $n_i \in \mathbb{N}$, $T : \mathbb{N} \rightarrow \mathbb{R}_+$ the parameters of the simulated annealing
Output: $I^* \subseteq \mathcal{G}$, $J^* \subseteq \mathcal{C}$, $\Omega^* \in \mathbb{R}$ sets of genes and cells maximizing problem (4.1)(4.2), and the corresponding objective value

```
// 1) Preselect rare cells
 $C_{\text{rare}} = \{j \in \mathcal{C} \mid r_j \geq q_3(\mathbf{r}) + 1.5 \cdot IQR(\mathbf{r})\}$ 

// 2) Beam search (see algorithm 6.2)
 $I^\sim, J^\sim, \Omega^\sim = \text{beam\_search}(\mathbf{M}_{\mathcal{G}C_{\text{rare}}}, \mathbf{r}, \kappa, \mu)$ 

// 3) Cells expressing genes of 2)
 $C_{\text{cand}} = \{j \in \mathcal{C} \mid |\{i \in I^\sim : m_{ij} \geq 0\}| \geq 0.5 \cdot |I^\sim|\}$ 

// 4) Local search (see algorithm 6.3)
 $I^*, J^*, \Omega^* = \text{simul\_annealing}(\mathbf{M}_{\mathcal{G}C_{\text{cand}}}, I^\sim, J^\sim, \Omega^\sim, \kappa, \mu, n_r, n_i, T)$ 
```

(large and small scale) datasets considered. We therefore use $n = b = 100$ for all experiments.

A naive implementation to choose the 100 best solutions at each level of the beam search would store all solutions and sort them. We use a more efficient approach using a priority queue to store a fixed number (100) of current best solutions in memory. Inserting a new one is done in log-time with respect to the size of the queue (i.e. negligible compared to $|\mathcal{C}|$), and the worst one is discarded in constant time to keep a size of 100 (this behavior is represented in algorithm 6.2 by the `100_best` function).

The efficiency of the beam search is further improved by comparing the similarity in terms of selected genes between all solutions selected at a certain search level. Whenever this similarity is higher than 90%, one can consider the beam

Algorithm 6.2: MicroCellClust 2 – beam_search

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ an expression matrix
Input: $\mathbf{r} \in \mathbb{R}^{|\mathcal{C}|}$ a rareness score distribution for the cells
Input: $\kappa \in \mathbb{R}_+$, $\mu \in [0, 1]$ the parameters of problem (4.1)(4.2)
Output: $I^\sim \subseteq \mathcal{G}$, $J^\sim \subseteq \mathcal{C}$, $\Omega^\sim \in \mathbb{R}$ sets of genes and cells
approximating (4.1)(4.2), and corresponding objective value

$I^\sim, J^\sim, \Omega^\sim \leftarrow \emptyset, \emptyset, 0$

// Evaluate pairs of cells (search level 2)

$beam_{\ell=2} \leftarrow \emptyset$

for $j \in \mathcal{C}$ **do**

for $j' \in 100_closest(\mathbf{r}, r_j)$ **do**

$J \leftarrow \{j, j'\}$

$I, \Omega \leftarrow \text{obj}(\mathbf{M}, J, \kappa, \mu)$

$beam_{\ell=2} \leftarrow 100_best(beam_{\ell=2} \cup \{J\})$

if $\Omega \geq \Omega^\sim$ **then** $I^\sim, J^\sim, \Omega^\sim \leftarrow I, J, \Omega$

// Evaluate next search levels

for $\ell \in \{3, 4, \dots\}$ **do**

$beam_\ell \leftarrow \emptyset$

for $J^0 \in beam_{\ell-1}$ **do**

for $j \in \mathcal{C} \setminus J^0$ **do**

$J \leftarrow J^0 \cup \{j\}$

$I, \Omega \leftarrow \text{obj}(\mathbf{M}, J, \kappa, \mu)$

$beam_\ell \leftarrow 100_best(beam_\ell \cup \{J\})$

if $\Omega \geq \Omega^\sim$ **then** $I^\sim, J^\sim, \Omega^\sim \leftarrow I, J, \Omega$

return $I^\sim, J^\sim, \Omega^\sim$

search has converged to a specific subpopulation, as all partial solutions in the beam express roughly the same genes. The search can thus be restricted to the cells that express at least part of these genes (by default 30%). Cells that fail this threshold are removed from the domain as they are highly unlikely to be part of that particular subpopulation.

Local search

For the second search (step 4) of the MicroCellClust 2 procedure, one could in principle reuse the beam search on the new search space defined in step 3. Yet, this approach is inefficient as it would reevaluate biclusters of 2 cells, then 3, etc., leading to many unneeded computations. A far more efficient approach is to start from the approximate solution obtained at step 2, and to explore the search space around it to look for a better solution (which is precisely the purpose of local search). We implemented a local search based on simulated annealing (Van Laarhoven and Aarts, 1987) (see algorithm 6.3), which has a theoretical guarantee to find the optimal solution (under some assumptions such as infinite time), leading to very good approximations in practice (Granville et al., 1994). Using this approach for the second search in MicroCellClust 2 indeed leads to solutions with a higher objective value than when reusing the beam search.

Simulated annealing explores the search space by stochastically visiting neighbors of the current solution, obtained by adding or removing a cell from the current assignment. A neighbor is always accepted if it improves the current objective value (i.e. the value of equation (4.1) given the current assignment of variables). If a neighbor has a lower value, it might still be accepted according to a certain probability to avoid getting stuck in a local optima. Multiple restarts are also performed to further diversify the search.

The simulated annealing algorithm has a number of parameters. n_r is the number of restarts (the more, the higher the probability of approaching the global optimum; a value of 20 is an appropriate trade off with respect to the execution time). n_i is the number of iterations within one restart, i.e. the number of neighbors generated ($100 \cdot |\mathcal{C}|$ works well on all datasets). T defines a “temperature” function decreasing with time, e.g. $T(t) = T_0 \cdot \alpha^t$ with $\alpha \in]0, 1[$, which ensures a relatively high probability to accept a worse neighbor at the beginning of a restart, so as to diversify the search, which gradually vanishes towards the end. Additionally, cells are ordered by their expression over the initial genes at the beginning of the search. During the generation of a neighbor, the random selection is weighted so as to slightly favor promising cells: the unselected (resp.

Algorithm 6.3: MicroCellClust 2 – simulated_annealing

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ an expression matrix

Input: $I^0 \subseteq \mathcal{G}$, $J^0 \subseteq \mathcal{C}$, $\Omega^0 \in \mathbb{R}$ an initial solution to start the local search

Input: $\kappa \in \mathbb{R}_+$, $\mu \in [0, 1]$ the parameters of (4.1)(4.2)

Input: $n_r \in \mathbb{N}$, $n_i \in \mathbb{N}$ the number of restarts and the number of iterations within each restart

Input: $T : \mathbb{N} \rightarrow \mathbb{R}_+$ a decreasing “temperature” function

Output: $I^* \subseteq \mathcal{G}$, $J^* \subseteq \mathcal{C}$, $\Omega^* \in \mathbb{R}$ sets of genes and cells maximizing (4.1)(4.2), and corresponding objective value

$I^*, J^*, \Omega^* \leftarrow I^0, J^0, \Omega^0$

repeat n_r times

$J_{\text{curr}}, \Omega_{\text{curr}} \leftarrow J^0, \Omega^0$ // Reset current

for $t \in \{0, \dots, n_i\}$ **do**

/* Generate neighbor: randomly add or remove one cell */

$j \leftarrow \text{random}(\mathcal{C})$

if $j \in J_{\text{curr}}$ **then** $J \leftarrow J_{\text{curr}} \setminus \{j\}$

else $J \leftarrow J_{\text{curr}} \cup \{j\}$

/* Evaluate the neighbor */

$I, \Omega \leftarrow \text{obj}(\mathbf{M}, J, \kappa, \mu)$

/* Update current if improves objective value; else update according to a certain probability */

if $\Omega \geq \Omega_{\text{curr}}$ **or** $\text{random}([0, 1]) < e^{\frac{-(\Omega_{\text{curr}} - \Omega)}{T(t)}}$ **then**

$J_{\text{curr}}, \Omega_{\text{curr}} \leftarrow J, \Omega$

if $\Omega \geq \Omega^*$ **then** $I^*, J^*, \Omega^* \leftarrow I, J, \Omega$

return I^*, J^*, Ω^*

selected) cells that express these genes most (resp. less) have a higher probability to be added (resp. removed).

6.2 Case studies on large-scale scRNA-seq data

Two scRNA-seq datasets are used to evaluate the ability of MicroCellClust 2 to identify rare cells in large-scale data. We show that MicroCellClust 2 identifies rare megakaryocytes within 68k peripheral blood mononuclear cells (PBMCs) (Zheng et al., 2017). MicroCellClust 2 also manages to find rare ependymal cells within 160k mouse brain cells (Zeisel et al., 2018). This case study demonstrates that MicroCellClust 2 is more effective at identifying the entire subpopulation than FiRE or DoRC. For both experiments, normalized read counts are $\log_{10}(x + 0.1)$ transformed, and genes expressed in more than 25% of the cells are left out (respectively 2% and 7% of the genes).

6.2.1 Identification of rare megakaryocytes within 68k PBMCs

Our first case study contains 68 579 peripheral blood mononuclear cells (PBMCs) sampled from a healthy donor (Zheng et al., 2017) (see appendix A.5). FiRE is used to obtain a rareness score for each cell.⁴ MicroCellClust 2 executes as follows (see algorithm 6.1):

1. 4238 cells are marked as rare using the FiRE threshold.
2. These cells are used as candidate variables for the beam search (algorithm 6.2), which returns an initial approximation containing 174 cells and 21 genes.
3. 226 cells are found to express more than 50% of these genes and are selected as candidates for the local search.
4. The simulated annealing (algorithm 6.3) starts from the solution of step 2, and finds a new solution inside the search space defined in step 3 (which improves the objective value by 15%). This final solution contains 202 cells and 22 genes.

Figure 6.1 shows the results. The presence of PF4 among the marker genes suggests MicroCellClust 2 identifies a rare subpopulation of megakaryocytes (0.3%

⁴Using the same protocol as in (Jindal et al., 2018), including a $\log_2(x + 1)$ normalization. The FiRE results might differ according to the normalization used, but with limited (if any) impact for on MicroCellClust 2 end result.

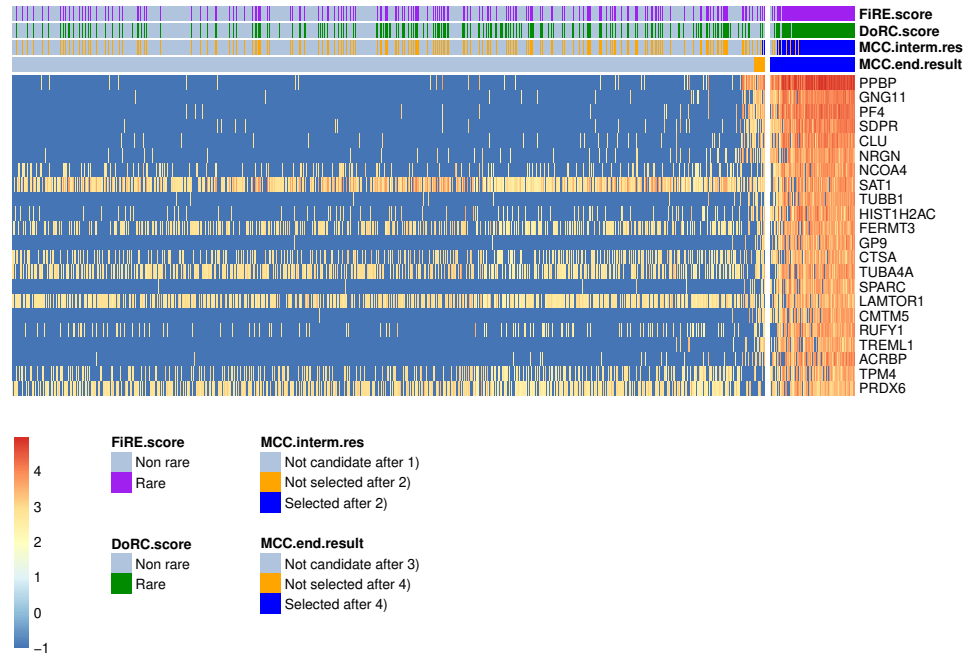


Figure 6.1: Megakaryocyte subpopulation identified by MicroCellClust 2 within the 68k PBMCs. For the clarity of the visualization, cells are ordered increasingly by the sum of their expression over the selected genes and only the 2000 cells with highest sum are displayed (3% of the data). The MCC.interm.res annotation indicates the result of the beam search using only the cells passing the FIRE rareness threshold (indicated by the FIRE.score annotation; the DoRC result is also indicated by DoRC.score). The MCC.end.result annotation indicates the result of the local search, i.e. the end result of MicroCellClust 2. It contains 202 cells that highly express the 22 marker genes, which are lowly expressed in the remaining cells (note that only a small fraction of them are displayed: it may appear that genes such as SAT1 are also highly expressed in the out-of-cluster cells, but this is not the case when considering all 66k of them).

of the cells), which is consistent with previous analyses of this data (Jindal et al., 2018; Zheng et al., 2017).

Alternatively, one could use DoRC as rareness score, in which case 3951 cells are marked as rare (2898 of which are also part of the FiRE result). Interestingly, the exact same solution is returned by the beam search at step 2, and consequently the same subpopulation is identified as end result. This illustrates the stability of MicroCellClust 2 regarding the rareness score chosen as heuristics.

6.2.2 Identification of rare ependymal cells within 160k mouse brain cells

Our second case study contains 160 796 mouse brain cells, which have been sampled to create a detailed census of cell types in the mouse nervous system (Zeisel et al., 2018) (see appendix A.6), which may include rare ones. Figure 6.3 shows the result after using MicroCellClust 2 on this data:

1. 1333 cells are marked as rare using the FiRE threshold.
2. The beam search (algorithm 6.2) using these cells as candidates identifies a subpopulation of 392 cells and 64 genes. These genes are lowly expressed among the cells that were candidates after step 1 but not selected during the beam search. Interestingly, around 500 cells that were not candidates after step 1 (they failed the FiRE threshold and could therefore not be selected during the beam search) appear to highly express the identified genes (figure 6.2).
3. By selecting all cells expressing at least 50% of the genes returned at step 2, a new search space is defined containing 1144 cells.
4. The local search (algorithm 6.3) identifies a much larger subpopulation of 912 cells (0.6% of the data), characterized by 58 genes, which improves the objective value by 155%.

This subpopulation expresses many genes that have been linked to ependymal cells, including among others *Rsph1*, *Dynlrb2*, *Tmem212*, *Foxj1*, *Riad1*, *Ccdc153* (Zeisel et al., 2018), which further confirms the ability of MicroCellClust 2 to identify biologically coherent rare subpopulations in large-scale data.

As seen in figure. 6.3, a majority of the selected cells were not labeled as rare by FiRE or DoRC, despite 26 out of the 58 marker genes being part of the 1000 most variable genes used to compute the rareness scores. This illustrates the effectiveness of selecting both the cells and genes at the same time to iden-

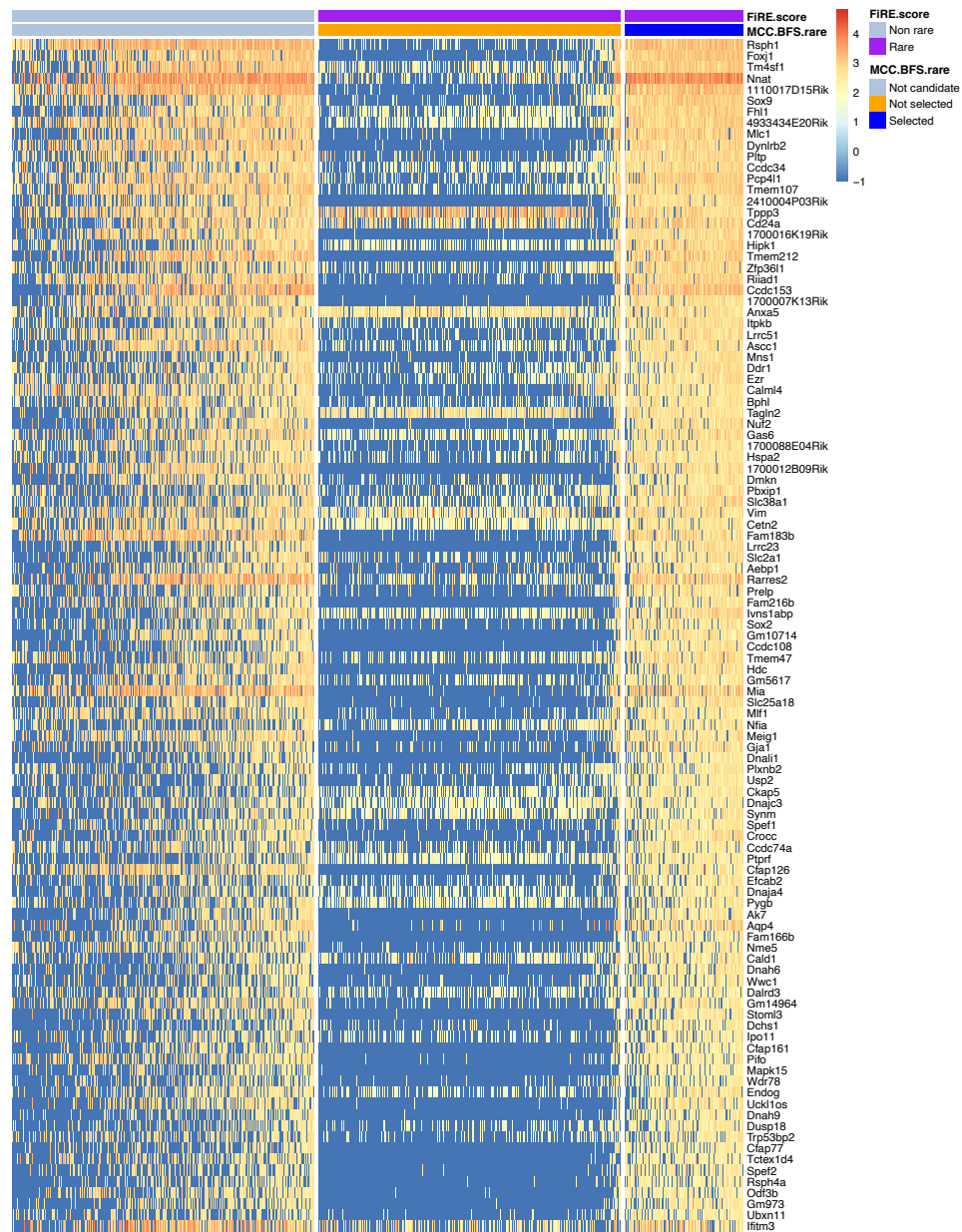


Figure 6.2: Ependymal subpopulation identified within the 160k mouse brain cells marked as rare by FiRE. Result of the breadth-first algorithm on the 160k mouse brain cells when using only the cells passing FiRE's rareness threshold as candidates (for each group of the MCC.BFS.rare annotation, only the 1000 cells with highest sum of expression are displayed).

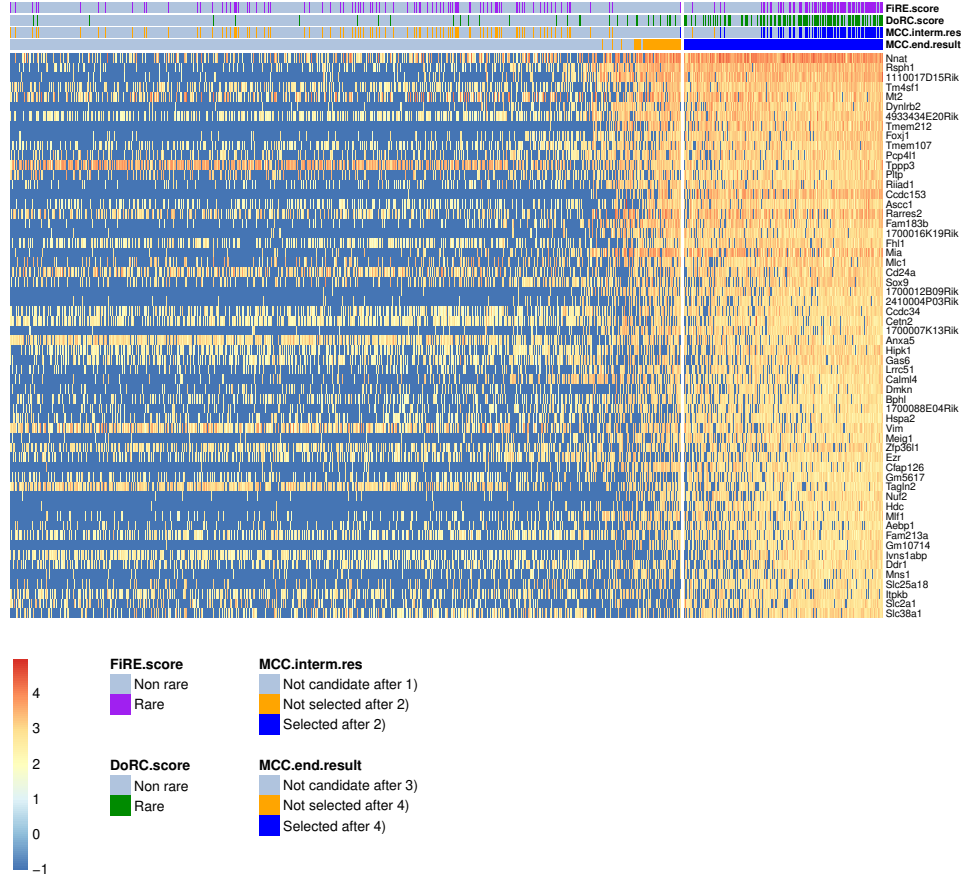


Figure 6.3: Ependymal subpopulation identified by MicroCellClust 2 within the 160k mouse brain cells. Only 4000 cells (those with the highest sum of expression over the selected genes) out of the 160k ones are displayed (2.5% of the data). The 58 marker genes are highly expressed among the 912 cells selected by MicroCellClust 2. Both FiRE and DoRC only mark a fraction of these cells (respectively 398 and 447) as rare (see the FiRE.score and DoRC.score annotations), missing the majority of these cells. MicroCellClust 2 is therefore more effective at identifying the subpopulation as a whole. Note that the objective function provides an ordering of the cells (based on their sum of expression values over the selected genes), which allows to identify cells that are closely related even though they do not express all the selected genes (tuning the κ and μ parameters might slightly change which genes and cells are selected, but this doesn't change the underlying biological interpretation in any of the reported experiments).

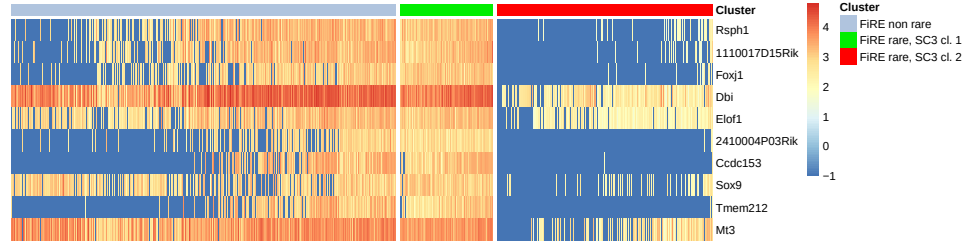


Figure 6.4: Result of SC3 applied on the cells marked as rare by FiRE. The first cluster, containing 399 cells, is characterized by 10 marker genes (displayed on the figure), 6 of which are present in the MicroCellClust 2 solution and can be linked to ependymal cells. The cells that were not considered rare by FiRE which most express these genes are also displayed. A relatively significant part of them highly express all the marker genes of cluster 1, showing the low recall of FiRE on this data as it failed to indicate a large part of the ependymal cells as rare.

tify a subpopulation as a whole. Indeed, FiRE and DoRC don't select genes when computing the rareness score, instead relying on an *a posteriori* clustering or differential expression analysis to identify marker genes. Such an approach would in this case only be able to identify half of the identified ependymal subpopulation as only half of them were labeled as rare (as an example, figure 6.4 shows the result when using SC3 (Kiselev et al., 2017) to cluster the FiRE rare cells).

6.2.3 Scalability

The use of a beam search algorithm to find a first approximation among rare cells, combined with local search to refine the solution, makes MicroCellClust 2 scalable to large volumes of data. Random subsamplings of both datasets are drawn to evaluate the execution time of the solver in function of the number of cells. Subsamplings of 10k up to 60k cells are drawn from the 68k PBMCs, and sets of 25k to 150k cells are drawn from the 160k mouse brain cells, while ensuring the same relative proportion of megakaryocytes (resp. ependymal cells) is kept. Figure 6.5 shows MicroCellClust 2 has a linearly execution time with respect to the number of cells. It is also robust regarding variations of the data in the sense that a megakaryocyte (resp. ependymal) subpopulation is identified within each subset of the PBMC (resp. mouse brain) data.

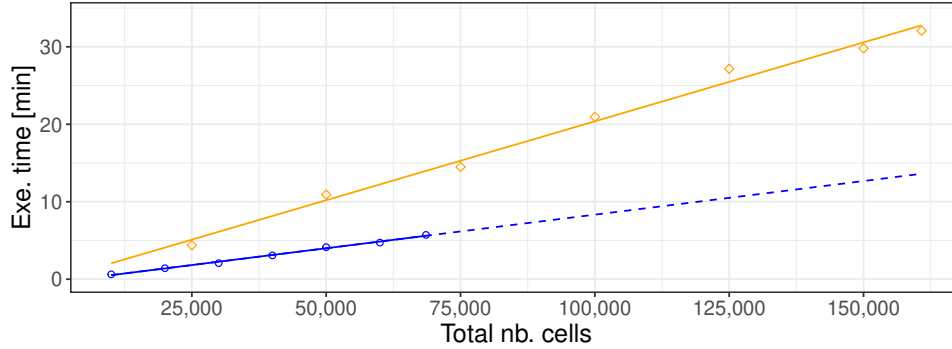


Figure 6.5: Evolution of the runtime of MicroCellClust 2 for subsamplings of different size for (◐) the 68k PBMCs dataset and (◑) the 160k mouse brain cells. These results have been obtained using a MacBook Pro laptop (Mac OS 11.6.5; 2.7 GHz Intel Core i7 CPU; 16 GB RAM).

6.3 Additional features

This section presents additional features that can be added to the MicroCellClust 2 solver. First, an automated procedure to tune the κ and μ parameters is discussed. This ensures ensures a good solution, in terms of interpretability, is presented to the user when running MicroCellClust 2 “out of the box”. Secondly, an adaptation of the FiRE method, which computes a coherent score for groups of cells, is discussed. This allow to generalize the heuristics used in the beam search of MicroCellClust 2 to all search levels.

6.3.1 Automated parameter tuning

As illustrated in section 4.3, the two parameters κ and μ of the constrained optimization problem (4.1)(4.2) can be easily tuned without any need for prior knowledge. They behave in the same way using the MicroCellClust 2 solver, and can be easily interpreted by a domain expert:

- κ controls the strength of the differentiation between in-cluster and out-of-cluster expression. A higher value tends to select less genes, as only those with few out-of-cluster expression contribute positively to the objective. A lower value implies a smaller penalty for out-of-cluster expression, which naturally leads to more genes being selected.
- μ controls the proportion of negative values tolerated inside the bicluster, as false negative values can be caused by to the dropout phenomenon.

It makes sense to raise the value of μ (from its default 10%) whenever a group of cells expresses a significant part of the genes in the bicluster but is not selected due to a strict constraint (4.2). Increasing μ naturally leads to selecting more cells, and also tends to select more genes.

Their values can be tuned to obtain typically around 10 to 50 genes, which can then be used to perform a biological analysis.

The latest version of the solver includes some mechanisms to automate this tuning. During the beam search (algorithm 6.2), the value of κ is automatically decreased (resp. increased) whenever too few (resp. too many) genes are selected in the best solution of each level:

- At the beginning of the beam search, we require the result to contain at least 20 cells. Indeed, the number of genes initially tends to decrease when more cells are added (e.g. 3 cells naturally tend to have less genes in common than 2 cells, etc.), so a too strict value of κ in these early stages of the search could lead to solutions with very few (or no) genes down the line. In case the best result at level 2 contains less than 20 genes, the value of κ is interactively decreased until a solution with more than 20 genes is obtained. In the subsequent search levels, the best result of that level is tested with the original κ value: whenever it results in a bicluster with more than 20 genes, the solver switches back to the original κ value.
- During the entire beam search, whenever the best solution at a certain level contains more than 200 genes, the value of κ is automatically increased. This is motivated by the fact that having so many genes is generally not necessary to perform a biological analysis of the result, and it has a substantial impact on the execution time.

To automate the tuning of μ , a new value is estimated from the result returned by the beam search: a test bicluster containing the selected genes and all cells expressing over 70% of these genes is formed, and a new value of μ is computed from the proportion of negative values within this bicluster.

The results in figures 6.1 and 6.3 are obtained using this protocol:

- $\kappa = 0.0015$ (default, i.e. $\frac{100}{|C|}$) and $\mu = 15\%$ ($\nearrow$) for the 68k PBMCs dataset;
- $\kappa = 0.004$ ($\nearrow$) and $\mu = 20\%$ ($\nearrow$) for the 160k mouse brain dataset.

Note that the parameter values define a hard threshold between selected and non-selected cells. In practice, the difference is often more gradual. This can be especially be seen in figure 6.3, where some non-selected cells express part of the

genes in the bicluster. The objective function provides an ordering of the cells (namely the sum of expression values over the selected genes), highlighting cells that look similar to those inside the bicluster, but missed the cut as they express only part of the selected genes. Changing the parameters can affect this threshold but does not change the underlying biological interpretation in any of the reported experiments.

6.3.2 Generalization of the heuristics

Since rareness scores such as FiRE and DoRC are only defined for individual cells, their usage as heuristics within the beam search is limited to the first step of the algorithm, corresponding to the evaluation of pairs of cells. In subsequent levels of the search, all remaining individual cells are added to the solutions from the previous level. It could potentially be beneficial, both in terms of execution time and quality of the solution, to only add cells that are close in terms of rareness score.

A naive approach to assign a rareness score to a group of cells would be to simply average the individual FiRE scores of each cell. Yet, such a procedure would not take into account any relationships between these cells in terms of gene expression. Indeed, one would expect a group of cells with high score to represent one particular cell type, i.e. cells that are both rare and share a common pattern of expression, as one would obtain with regular clustering. By simply aggregating FiRE scores, one could end up with a group of cells which are indeed rare, but due to different rare patterns of expression.

Yet, even though FiRE produces individual scores, one can take advantage of its algorithm to identify cells that are grouped together during this process (Dubuisson, 2024). Indeed, during each run of the FiRE algorithm (see algorithm 2.5 on page 35), cells are grouped into buckets based on their expression within a random subset of genes (see section 2.3.3). Cells that are hashed in the same bucket therefore express the genes selected for this run in a similar way. Overall, one can define a similarity matrix between cells by counting the number of runs where two cells end up in the same bucket (i.e. are assigned the same hash code).

Dubuisson (2024) generalized FiRE to produce rareness scores for predefined groups cells by defining a *homogeneity* metric. For a group of cells J , it is defined as the proportion of pairs within this group which fall in the same bucket during

at least one of the FiRE runs:

$$h(J) = \frac{|\{(j, j') \in J^2 \mid \exists n : hash_{jn} = hash_{j'n}\}|}{\frac{|J|^2 - |J|}{2}} \quad (6.1)$$

An *overall rareness* score for the group is defined based on the individual scores of the cells in J . Dubuisson (2024) defines it as the smallest FiRE score among all cells in J :

$$r(J) = \min_{j \in J} \text{FiRE}(j) \quad (6.2)$$

This strict criterion ensures all cells in J are effectively rare for the group to (have a chance to) be considered rare. Since the beam search of MicroCellClust 2 is only executed on the cells marked as rare by FiRE, such a strict criterion makes sense. Yet, it might be useful for other applications to define an alternative $r(J)$ metric, e.g. the mean of the individual rareness scores, to allow some exceptions. For instance, a group containing one cell failing the rareness threshold but which has a high homogeneity could still be of interest and have a relatively high final score.

The *final score* of the group corresponds to its overall rareness score, but penalized in case of a lack of homogeneity. Whenever the homogeneity score of J falls below some threshold α (by default, $\alpha = 70\%$), a penalty is applied to the rareness score. The final score becomes:

$$g(J) = \begin{cases} r(J) & \text{if } h(J) \geq \alpha \\ r(J) \cdot h(J) & \text{otherwise} \end{cases} \quad (6.3)$$

This generalized score can be integrated as heuristics in the MicroCellClust 2 beam search (algorithm 6.4). For each search level $\ell \geq 3$, the rareness score of each solution in $beam_{\ell-1}$ is computed. Instead of producing all children solutions, i.e. adding all remaining cells, only the 100 cells with closest rareness scores to the group score are considered.

On the Jurkat v. 293T experiment (see section 4.2.2), this leads to a $\approx 25\%$ speed up in execution time. Moreover, this modified beam search outputs solutions with slightly higher objective values than the original one. This shows the new heuristic pruning actually helps to keep the best candidates in the beam, allowing the identification of a better end solution, the parents of which would otherwise be eliminated at some point in the beam search.

Algorithm 6.4: MicroCellClust 2 – beam_search_alt

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ an expression matrix

Input: $\mathbf{r} \in \mathbb{R}^{|\mathcal{C}|}$ a rareness score distribution for the cells

Input: $g : \mathcal{C} \rightarrow \mathbb{R}$ a function computing the rareness score for groups of cells

Input: $\kappa \in \mathbb{R}_+$, $\mu \in [0, 1]$ the parameters of problem (4.1)(4.2)

Output: $I^\sim \subseteq \mathcal{G}$, $J^\sim \subseteq \mathcal{C}$, $\Omega^\sim \in \mathbb{R}$ sets of genes and cells approximating (4.1)(4.2), and corresponding objective value

$I^\sim, J^\sim, \Omega^\sim \leftarrow \emptyset, \emptyset, 0$

$beam_{\ell=2} \leftarrow \emptyset$

for $j \in \mathcal{C}$ **do**

for $j' \in 100_closest(\mathbf{r}, r_j)$ **do**
Evaluate $J \leftarrow \{j, j'\}$ // See algorithm 6.2

for $\ell \in \{3, 4, \dots\}$ **do**

$beam_\ell \leftarrow \emptyset$
for $J^0 \in beam_{\ell-1}$ **do**

for $j \in 100_closest(\mathbf{r}, g(J^0))$ **do**
Evaluate $J \leftarrow J^0 \cup \{j\}$ // See algorithm 6.2

return $I^\sim, J^\sim, \Omega^\sim$

6.4 Discussion

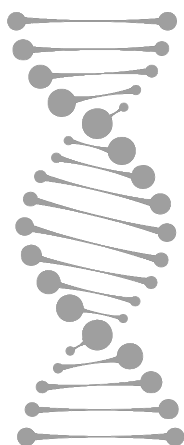
This chapter describes a new approach to solve the optimization problem of MicroCellClust which scales to large single-cell dataset. It is based on a hybrid approach which first uses a rareness score as heuristics to efficiently find a good approximated solution in a limited search space, before using a local search algorithm to enlarge the search space in order to approach the optimal solution. This method has a linear time complexity in terms of the number of cells, which is confirmed by the reported experiments on scRNA-seq data. Consequently, MicroCellClust 2 easily scales to datasets containing hundreds of thousands of cells to identify rare subpopulations with highly specific genes.

These experiments show the ability of MicroCellClust 2 to extract relevant subpopulations from large-scale data, together with specific marker genes. The joint selection of cells and genes ensures the globality of a subpopulation is highlighted, whereas methods that rely on *a posteriori* gene selection tend to miss part of the relevant cells. This further confirms the relevance of the multivariate approach used by MicroCellClust (2), even in large-scale single-cell data where identifying rare cell types is comparatively easier.

Such large-scale datasets can be obtained thanks to the high throughput achieved by droplet-base sequencing technologies. Another way to obtain such data is by aggregating multiple single-cell samples, which can come from different donors, and even be sequenced using different platforms. The next chapter proposes an adaptation of MicroCellClust that considers these samples as a third dimension, beyond genes and cells, in order to find specific subpopulations of cells and their associated marker genes that would also be common across different biological samples.

Searching for rare cells across multiple samples

7



The previous chapters demonstrate the usefulness of the global sum criterion to analyze scRNA-seq data, in particular its robustness to high technical and biological variability. These effects are exacerbated when dealing with multiple single-cell samples, which often exhibit batch effects. The traditional approach to deal with such aggregated data consists in first integrating the data into one sample, and then performing a regular clustering (thus ignoring the partition into multiple samples).

The global sum criterion, on the other hand, is likely insensitive to these batch effects, as shifts in expression values between samples have little influence as long as the global value of the bicluster remains high. This chapter presents a new optimization problem, based on this criterion, to explicitly search for rare subpopulations shared across single-cell samples.¹

Section 7.1 motivates the benefits of an explicit search across samples. Section 7.2 presents scCross, a modified version of the MicroCellClust optimization problem targeted at cross-sample results. Section 7.3 evaluates this new approach on multi-sample scRNA-seq data. Section 7.4 presents a modified solver to efficiently solve the new optimization problem.

7.1 Motivation

At present, the field of single-cell RNA sequencing has reached a high maturity: many datasets are publicly available, and sequencing new biological samples has

¹*Related publication:* Gerniers, A., Nijssen, S., and Dupont, P. scCross: efficient search for rare subpopulations across multiple single-cell samples. *Bioinformatics*, 40(6):btac371.

become relatively easy and cheap thanks to the widespread availability of high throughput sequencing machines. It is now common to aggregate multiple single-cell samples (e.g. coming from different donors) into one experiment in order to get a better understanding of the pathology under study. Yet, as discussed in section 2.2.4, aggregating multiple samples of single-cell data usually comes at the price of *batch effects*, due to technical and biological variability between samples. To prevent clustering algorithms from picking up this unwanted variation, rather than biologically relevant information, several *data integration* techniques have been proposed (Argelaguet et al., 2021; Luecken et al., 2022) (see section 2.2.4 for details on some of these methods). They aim at combining the different samples into one consistent dataset, upon which a regular clustering algorithm can be applied.

While data integration is often necessary to accurately identify cell types in multi-sample single-cell data, improper integration may lead to overcorrection, whereby genuine biological variability is erased along with the batch effects (Büttner et al., 2019). This is especially problematic when dealing with rare cells, as improper data integration may make it impossible to detect them (Guo et al., 2022). Some methods, such as MNN (Haghverdi et al., 2018) and Seurat v3 (Stuart et al., 2019), might also be sensitive to the order in which the samples are integrated, which can lead to spurious alignment between disparate cell types (Hie et al., 2019). Moreover, integrating large datasets often requires a high computational cost, both in terms of time and memory, especially for methods performing integration in the original genes $\times$ cells space, making a differential expression analysis impossible. Methods integrating cells after projection to lower-dimensional spaces tend to be computationally efficient, but at the cost of losing the interpretation in terms of genes and/or cells concerned. Being able to identify rare cell expression patterns across samples without needing data integration would therefore be useful to analyze the heterogeneity of single-cell data.

As shown in chapter 4, MicroCellClust proves particularly robust with respect to the high technical and biological variability of single-cell data, and is therefore likely insensitive to batch effects. Even though it has been tested in a single sample setting, it could, in principle, be applied to multiple samples concatenated in a larger expression matrix. Doing so would however not guarantee that MicroCellClust could identify a subpopulation of cells actually shared among different samples (e.g. multiple donors in a cancer study). By design of this ap-

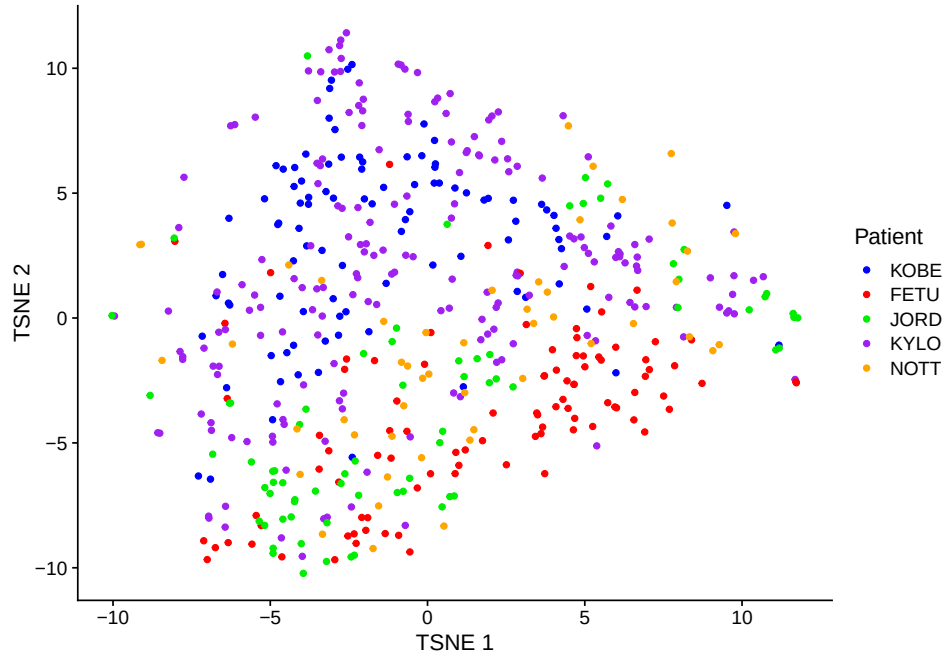


Figure 7.1: Dimensionality reduction using t -SNE of breast cancer GARP+ Tregs from 5 donors. No distinct clusters emerge, which indicates this data seems homogeneous and does not suffer from the presence of significant batch effects.

proach, it could simply return a specific subpopulation of cells that would also be restricted to a single donor.

Such a behavior can be observed when considering the breast tumor GARP+ Tregs, in which MicroCellClust revealed a subpopulation with marker genes linked to cell division functions (see section 4.3.2), this time aggregating the data from 5 different donors (for a total of 505 cells, and 19 722 genes expressed in at most 25% of the cells; see appendix A.1). Interestingly, a t -SNE dimensionality reduction (figure 7.1) does not show any evidence of significant batch effects between these 5 samples. This data therefore constitutes a best-case scenario, where identifying cross-sample subpopulations should be relatively easy (compared to data with strong batch effects).

Fig. 7.2 shows the result of multiple MicroCellClust executions, whereby cells identified in previous executions are left out in order to search for a new bicluster in the remaining cells (see section 4.1.3). MicroCellClust first identifies a series

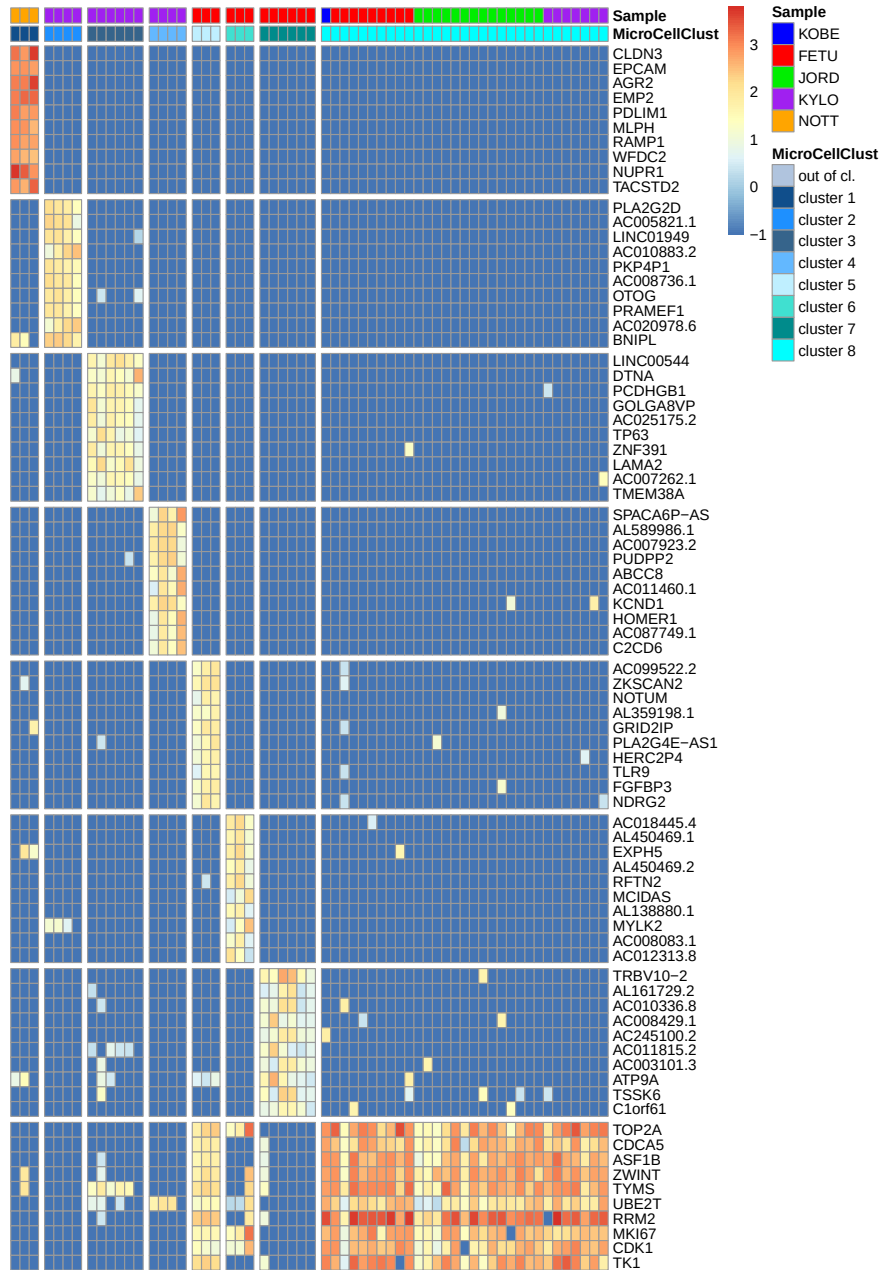


Figure 7.2: Result of eight executions of MicroCellClust on breast cancer GARP+ Tregs from 5 donors. The 10 genes with highest contribution to the objective are shown for each cluster (out-of-cluster cells are not displayed). The genes returned for cluster 8, which correspond to Treg-related functions, are also highly expressed in cluster 5, which has been identified earlier and grouped to sample-specific genes.

of small, donor-specific, subpopulations. It is only during the 8th execution that it identifies a subpopulation with 31 cells shared between multiple donors, which exhibits 55 cell-cycle related genes. Moreover, the cells from cluster 5 seem related to this subpopulation as they also express these same genes. Yet, they were clustered beforehand based on a series of genes specific to a single donor, for which a Gene Ontology (GO) analysis (Ashburner et al., 2000; The Gene Ontology Consortium, 2019) did not yield any function (with a significance threshold of 0.05 including FDR correction).

This shows that cells might be classified based on sample-specific artifacts in the absence of an objective function targeted to cross-sample results. The next section presents scCross, an extension of MicroCellClust, in order to identify subpopulations of cells with specific marker genes that are effectively present across multiple samples of single-cell data.

7.2 scCross definition

Let $\mathcal{S}$ be a set of single-cell samples (for instance corresponding to different donors). Sample s can be represented by a matrix $\mathbf{M}_s \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}_s|}$, with $\mathcal{G}$ the set of rows associated to the genes (assumed identical for each sample²) and $\mathcal{C}_s$ the set of columns associated to the cells. This data can be aggregated in one matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$, with $\mathcal{C} = \bigcup_{s \in \mathcal{S}} \mathcal{C}_s$, by concatenating the data of each sample:

$$\mathbf{M} = (\mathbf{M}_1 \cdots \mathbf{M}_{|\mathcal{S}|}) \quad (7.1)$$

As before (see section 3.2.1), an entry m_{ij} of this matrix is here assumed positive only if cell j expresses gene i , and negative otherwise.

As for the MicroCellClust objective (see section 4.1.1), the goal remains to select a subset of genes $I \subseteq \mathcal{G}$ and a subset of cells $J \subseteq \mathcal{C}$ representative of a small subpopulation of cells with specific genes. A gene i is considered specific whenever it is highly expressed for the cells in J , and has low expression in the cells not included in J . Considering one sample s , the specificity of gene i w.r.t. the selected cells within this sample is measured by:³

$$\omega_{is}(J) = \sum_{j \in \mathcal{C}_s \cap J} m_{ij} - \kappa \sum_{k \in \mathcal{C}_s \setminus J} \max\{0, m_{ik}\} \quad (7.2)$$

²In case of differences between genesets, for instance due to the integration of samples sequenced by different technologies, one simply takes the intersection between all genesets.

³This corresponds exactly to the ω_i values defined in equation (5.3) on page 92, but here specific to each sample (hence the extra s index).

The first term sums up the expression values corresponding to the selected cells within this sample ($j \in \mathcal{C}_s \cap J$) to assess whether they highly express gene i . The second term penalizes any positive expression in the remaining cells from that same sample ($k \in \mathcal{C}_s \setminus J$), to assess whether expression of gene i is indeed specific to the selected cells. The relative influence of this penalty is controlled by the κ parameter (default: $\kappa = \frac{100}{|\mathcal{C}|}$). The higher this value, the fewer genes are selected in I as only the most differentiated ones have a positive ω value.

The purpose of scCross, compared to MicroCellClust, is not only to identify a small subpopulation of cells with highly expressed genes, but in addition to ensure that this subpopulation is present across the different samples (i.e. making sure that J contains cells from as many samples as possible). In this context, computing an optimal bicluster (I^*, J^*) is performed through solving the following optimization problem:

$$(I^*, J^*) = \operatorname{argmax}_{\substack{I \subseteq \mathcal{G} \\ J \subseteq \mathcal{C}}} \sum_{i \in I} \delta_i(J) \sum_{s \in \mathcal{S}} \omega_{is}(J) \quad (7.3)$$

$$\text{such that } \frac{|\{(i, j) \mid i \in I, j \in J, m_{ij} < 0\}|}{|I| \cdot |J|} \leq \mu \quad (7.4)$$

The objective function in the equation (7.3) sums the ω values for each gene and each sample, i.e. computes the global sum. As before (see section 4.1.2), the parameter μ in the constraint (7.4) controls the proportion of negative values allowed in the solution (by default 10%) to ensure that the selected genes are sufficiently often expressed across the bicluster.

The $\delta_i(J)$ factor is intended to evaluate the similarity between the expression of a gene i across the different samples (both inside and outside the selected bicluster). Without it, i.e. setting it to 1, the objective (7.3) is equivalent to the original MicroCellClust optimization problem (see section 4.1). Indeed, simply summing the gene contributions of each sample ($\sum_i \sum_s \omega_{is}$) amounts to computing the gene contributions over the entire dataset ($\sum_i \omega_i$), thus ignoring the partition into different samples. One could therefore identify a subpopulation that is specific to one sample, but not observed in others.

The $\delta_i(J)$ term penalizes genes that are not expressed similarly among all samples, i.e. with large differences between the ω_{is} of gene i . This factor is defined as the ratio between the geometric mean (i.e. the n -th root of the product

of n quantities⁴) and the arithmetic mean:

$$\delta_i(J) = \frac{\left(\prod_{s \in \mathcal{S}} \max \{ \varepsilon, \omega_{is}(J) \} \right)^{\frac{1}{|\mathcal{S}|}}}{\frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \max \{ \varepsilon, \omega_{is}(J) \}} \quad (7.5)$$

The geometric mean has a high value, close to the arithmetic mean, when gene i is expressed similarly across all samples (if all ω_{is} tend to be equal then δ_i tends to 1 and no penalty is applied), but would be lower than the arithmetic mean in case of differences between the ω_{is} values. In the later case, δ_i induces an effective penalty and lowers the influence of the gene i in the global objective to be maximized. This weighting thus gives a comparative advantage to genes that are present across all (or most) samples, and cancels out those specific to only one (or a few) sample(s).

The choice of the proposed δ metric is further motivated by the fact that the geometric mean is more influenced by higher values. A gene with high expression values in all but a few samples would only be slightly penalized. In contrast, a gene that is highly expressed in only a few samples gets a heavy penalty (i.e. a low δ value), even though alternative metrics such as the variance may be equal in both cases.

In summary, scCross improves upon MicroCellClust to analyze multi-sample data by evaluating each sample separately ($\sum_{s \in \mathcal{S}}$) and by introducing a regularization factor between them (δ_i). This new optimization problem can be efficiently solved using an adaptation of the MicroCellClust 2 solver (see section 7.4 for details on how this solver was adapted to this specific problem).

Figure 7.3 shows the evaluation of both objectives on a toy example. The purple bicluster has the maximal value under the MicroCellClust objective function. In contrast, its value vanishes according to the scCross objective as indeed the selected bicluster is restricted only to cells from the third sample. The green bicluster is the scCross optimal solution and includes some cells from all samples.

7.3 Results on scRNA-seq data

This section shows through multiple experiments on scRNA-seq data that scCross is capable of identifying common subpopulations across multiple samples,

⁴Negative ω_{is} are replaced by some small positive constant $\varepsilon > 0$, e.g. 0.1, to stay in the real domain even when n is pair.

	s1						s2						s3						MCC		scCross	
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18				
g1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	3	2	-1	5	4	3	15	-8	1.5	-8
g2	-1	1	-1	-1	-1	1	-1	-1	-1	2	-1	-1	3	-1	4	2	1	3	8	-9	1.0	-9
g3	-1	-1	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	2	6	3	4	1	1	14	7	1.4	2.0
g4	1	-1	-1	-1	1	-1	1	-1	1	-1	1	-1	4	3	5	-1	6	3	15	-11	1.3	-11
g5	-1	2	1	2	-1	-1	-1	1	3	-1	-1	-1	-1	2	1	4	-1	-1	-5	15	-1.2	14.2
g6	-1	4	2	1	-1	1	-1	1	2	-1	1	-1	-1	-1	4	1	0	-1	-10	14	-3.7	13.7
g7	-1	2	3	-1	-1	-1	1	3	1	1	1	-1	-1	2	6	2	-1	-1	-5	17	-0.8	14.8
																			52	46	5.2	42.7

Figure 7.3: Toy example comparing the results of MicroCellClust (MCC) and scCross for two distinct subpopulations. The purple bicluster is the optimal MicroCellClust solution (with $\kappa = 1$ and $\mu = 0.1$). It corresponds to the subpopulation $J_1 = \{13, 14, 15, 16, 17, 18\}$, which is specific to the third sample. The green bicluster is the optimal scCross solution (with same κ and μ values). It corresponds to the subpopulation $J_2 = \{2, 3, 4, 8, 9, 11, 14, 15, 16\}$, distributed over the three samples. The numbers on the right indicate the contribution of each gene under the two different optimization objectives, with the final objective value of the bicluster below. Numbers in gray indicate that the corresponding genes are not selected, either because their contribution is negative or because they violate the constraint (7.4).

without the need for data integration. First, it shows that scCross increases the quality of the solution found by MicroCellClust in section 7.1. Then, it shows that scCross outperforms typical alternatives in a controlled experiment with artificial batch effects, where a specific rare cell type serves as target. The robustness of scCross with respect to data integration is further shown as it identifies (i) a cilium subpopulation within bone marrow sequenced from 7 donors, and (ii) several subpopulations from 8 pancreas samples sequenced using two different technologies. Finally, the last case study shows that scCross can be used in a different setting, namely, to search for rare cells across different organs, here tested on mouse data.

7.3.1 Identification of cell-cycle genes across GARP+ Treg samples

As shown in section 7.1, MicroCellClust, which is not targeted to cross-sample results, first classifies the GARP+ Treg data from 5 donors based on sample-specific artifacts (see figure 7.2). It is only on the 8th execution that it identifies a cross-sample result, with cell cycle related genes.

This is not the case with scCross as it identifies this subpopulation directly (figure 7.4), including the three cells from cluster 5 (for a total of 31 cells and 60

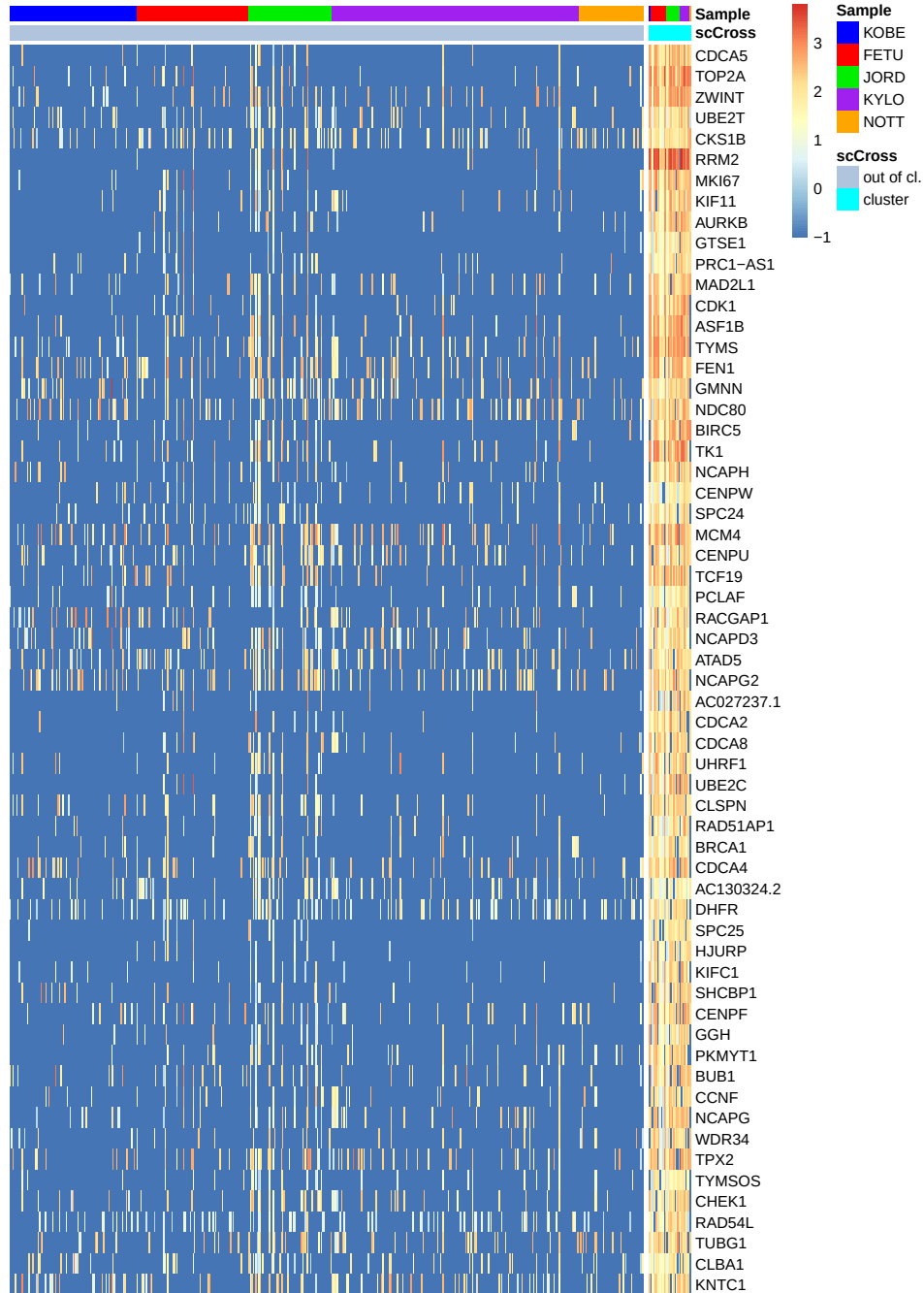


Figure 7.4: Result of scCross on the breast cancer GARP+ Tregs from 5 donors. The cluster 8 of MicroCellClust is identified during the first execution, and includes the 3 cells from cluster 5.

genes). Interestingly, this new cell cluster has a 10% higher value under the MicroCellClust objective than cluster 8, showing that successive executions can prevent MicroCellClust from reaching the local optimum. Moreover, scCross identifies a cell expressing these cell-cycle genes within the NOTT sample. This is not identified with MicroCellClust, showing scCross is able to identify related cells across samples even if present in very small amounts. This illustrates that scCross outperforms MicroCellClust at identifying subpopulations shared across multiple samples.

7.3.2 Identification of rare Jurkat cells with artificial batch effects

To assess the performance of scCross as an alternative to the usual data integration + clustering approach, we designed a controlled experiment using a dataset containing two human immortalized cell lines: 293T and Jurkat cells (Zheng et al., 2017) (see appendix A.2). Several rare populations of Jurkat cells (containing 6, 12 or 20 cells; with 10 independent sampling runs for each size) are sampled at random from the 1694 available ones. For each one, an abundant population of 293T cells is sampled from the 1540 available ones to obtain mixed datasets of 1000 cells, with $\approx 15\,000$ genes ($\approx 10\,800$ of which are expressed in at most 25% of the cells and are used by scCross).

To simulate batch effects, each dataset is randomly divided in two samples of 500 cells (each one containing the same number of rare Jurkat cells). The first sample is left untouched, whereas the expression values of the second one are multiplied by some value to artificially create batch effects. 3 different scenarios are used:

- “Global” roughly divides the expression values of each gene by two (i.e. each gene is multiplied by a random value generated by a Gaussian distribution with $\mu = 0.5$ and $\sigma = 0.02$).
- “Gene-specific” multiplies all values of the same gene by a random constant (generated by a uniform distribution in $[0.2, 1.8]$, which differs for each gene).
- “Value-specific” multiplies each expression value independently by a different constant (generated by a uniform distribution in $[0.5, 1.5]$).

The ability to identify the rare Jurkat cells is measured by the F_1 score, i.e. the harmonic mean between precision (the proportion of Jurkat cells within the solution) and recall (the proportion of Jurkat cells to be found actually included in the solution).

Table 7.1: Identification results of rare Jurkat cells among 293T cells with artificially created batch effects. Average F_1 score over 10 independent runs.

Nb. rare	Batch type	scCross	MNN			Seurat v3			Scanorama		
			R.ID3	Gini3	scAIDE	R.ID3	Gini3	scAIDE	R.ID3	Gini3	scAIDE
20	Global	0.95	0.79	0.83	0.14	0.71	1.00	0.10	0.63	0.22	0.53
	Gene-specific	0.97	0.98	0.46	0.23	0.52	0.47	0.16	0.84	0.43	0.51
	Value-specific	0.96	0.98	0.87	0.07	0.16	0.31	1.00	0.35	0.66	0.39
12	Global	0.92	0.74	0.22	0.07	0.63	0.68	0.06	0.37	0.05	0.14
	Gene-specific	0.90	0.72	0.32	0.21	0.47	0.34	0.08	0.46	0.10	0.26
	Value-specific	0.92	0.82	0.81	0.05	0.12	0.47	0.95	0.37	0.08	0.14
6	Global	0.92	0.50	0.07	0.03	0.55	0.59	0.09	0.39	0.12	0.03
	Gene-specific	0.92	0.56	0.14	0.06	0.60	0.62	0.09	0.23	0.04	0.05
	Value-specific	0.93	0.80	0.26	0.02	0.60	0.59	0.20	0.35	0.08	0.05

Note: R.ID3 stands for RaceID3, and Gini3 for GiniClust3.
Bold results indicate which method outperforms the others for each experiment according to a paired t -test with significance threshold of 0.05 (multiple bold values for the same experiment indicates there is no significant difference between them).

scCross is compared to 3 rare cell clustering methods: RaceID3, GiniClust3, and scAIDE (see section 2.3 for details on these methods). For each of these methods, the data is first integrated using MNN, Seurat v3, or Scanorama (see section 2.2.4 for details on these methods). This choice of methods is motivated by the fact they return corrected counts, i.e. do not embed the data in lower dimensional spaces but conserve the gene $\times$ cell space, and do not use any additional prior information such as cell type labels, making them usable in the unsupervised setting that is (bi)clustering. In particular, Scanorama is the only top performing method from the Luecken et al. (2022) benchmark that satisfies these two requirements.

Table 7.1 reports the average F_1 over 10 runs for scCross and each combination of data integration + clustering. scCross consistently achieves a high F_1 score ($\geq 90\%$), making it the best performing approach in six out of the nine cases. Moreover, the result of scCross seems robust w.r.t. the different batch effects, as the difference in F_1 between the 3 scenarios for the same number of rare cells is at most 2%.

This is not the case for the other approaches. First, one observes the F_1 score of the different clustering methods are dependent on the integration method: RaceID3 generally achieves a higher F_1 with MNN, whereas GiniClust3 overall performs better with Seurat v3. scAIDE, on the other hand, reaches a higher F_1 with Seurat v3 in half of the cases, and with Scanorama in the other half.

Moreover, when comparing the result of a given clustering method between the 3 scenarios, one observes large differences in F_1 . This shows that data integration methods, while aiming at correcting the batch effects, are not always able to reconstitute the original data. Consequently, the differences in expression values in the resulting datasets lead to varying clustering performances. The most striking example is given by scAIDE, which achieves relatively high F_1 scores of 0.98 and 0.76 on the original data with 20 and 12 rare cells, respectively. Yet, this performance vanishes in nearly all cases once batch effects are introduced and subsequently corrected by the data integration methods. This case study thus shows a concrete example where data integration fails to accurately compensate for batch effects.

One further observes that whenever a clustering method outperforms scCross for a particular type of batch effect, its performance is lower on the same data transformed differently. This is for instance the case for scAIDE, which achieves a perfect performance when applied, after integration with Seurat v3, on the 20-jurkat datasets with “value-specific” perturbations. Yet, its performance

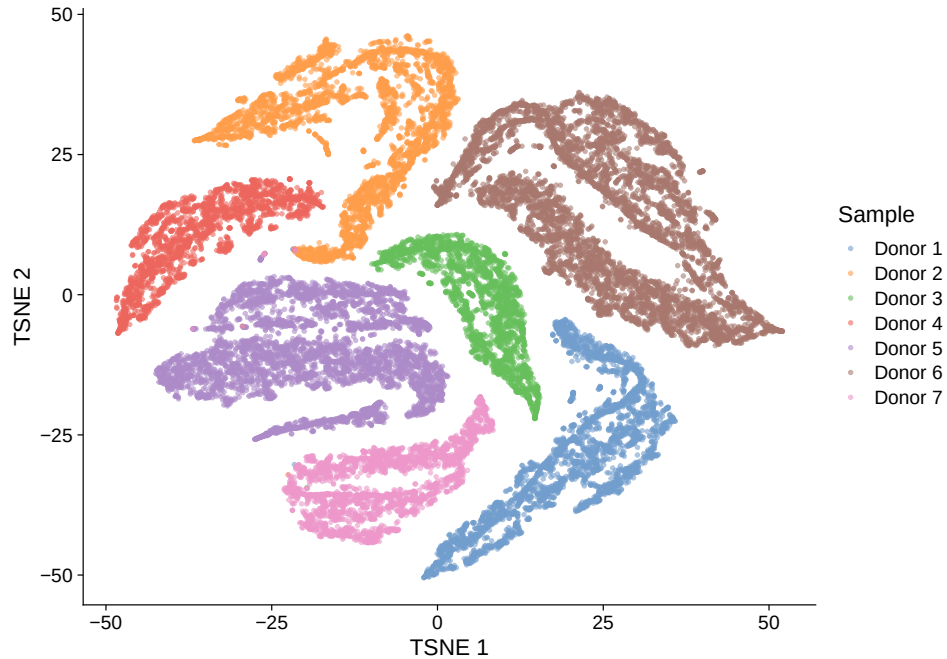


Figure 7.5: Dimensionality reduction of the NSCLC DTCs dataset using t -SNE. Several clusters are visible, but each one contains only cells coming from one single donor. t -SNE evidently captures variability between batches rather than grouping cells together by specific expression patterns.

crashes when used on the same data but with different types of batch effects. In a truly unsupervised single-cell experiment, there would be no way to monitor in which situation, and with which data integration method, a particular clustering method would perform well. This demonstrates the benefit of using scCross, which consistently identifies the same subpopulation of cells regardless of the nature of the batch effects present in the data.

7.3.3 Identification of a cilium subpopulation across different donors

Next, scCross is used to analyze a dataset with 26 027 Non-Small Cell Lung Cancer (NSCLC) Dissociated Tumor Cells (DTCs) coming from 7 human donors (see appendix A.7 for details on this data). This data contains 31 094 genes, 27 452 of which are expressed in at most than 25% of the cells. Strong batch effects are

observed between the 7 donors when projecting this data in a reduced dimension (figure 7.5).

The scCross result on this data consists of 61 cells, coming from 6 out of the 7 donors (figure 7.6). This subpopulation is characterized by 77 genes.⁵ A GO analysis (table 7.2) links them to the cilium, an organelle whose dysfunction causes various genetic diseases. Moreover, many of these genes have been identified in CiliaCarta (Van Dam et al., 2019), a compendium of ciliary genes (table 7.3). Yet, 33 of the genes identified by scCross are not listed in CiliaCarta, even though the scCross result shows they appear to be strongly related to know ciliary genes. Since several hundred proteins are thought to be involved in ciliary functions and identifying new ones is still ongoing, this observation is definitively worth further investigation.

Interestingly, typical rare cell clustering methods do not identify this cilium subpopulation. RaceID3 scatters these cells in 13 different clusters. To evaluate whether using data integration could enable RaceID3 to identify this cluster, we use MNN to integrate the samples of the 7 donors. Yet, the cilium subpopulation is still divided among several (here, 15) clusters.

GiniClust3 clusters cells by first identifying *high Gini genes*. Yet, none of the genes identified by scCross are labeled as such. These cells are thus missed by GiniClust3 and grouped within a large cluster of 2018 cells. Using the MNN-integrated data does not improve this result, since again none of these genes are labeled as *high Gini* and the identified cells are scattered in 14 different clusters.

Finally, it appears that the lower dimension projection used by scAIDE is not effective to capture the cilium related genes, as the corresponding cells are scattered in different larger clusters (both with the original and integrated data). These comparisons further illustrate the ability of scCross to identify specific subpopulations that are missed by typical alternatives, even when used in conjunction with data integration.

scCross has also been applied on integrated data to evaluate its robustness with respect to batch effects (figure 7.7). This results in a subpopulation of 61 cells (57 in common with the previous result, i.e. 93% of the identified cells) and 79 genes (73 in common with the result without data integration). This illustrates that scCross identifies the same cilium linked subpopulation whether or not the data is integrated.

⁵Using $\kappa = 0.2$ and $\mu = 0.1$. The value of κ has been raised from its default value of 0.004 so as to keep only the most differentiated genes.

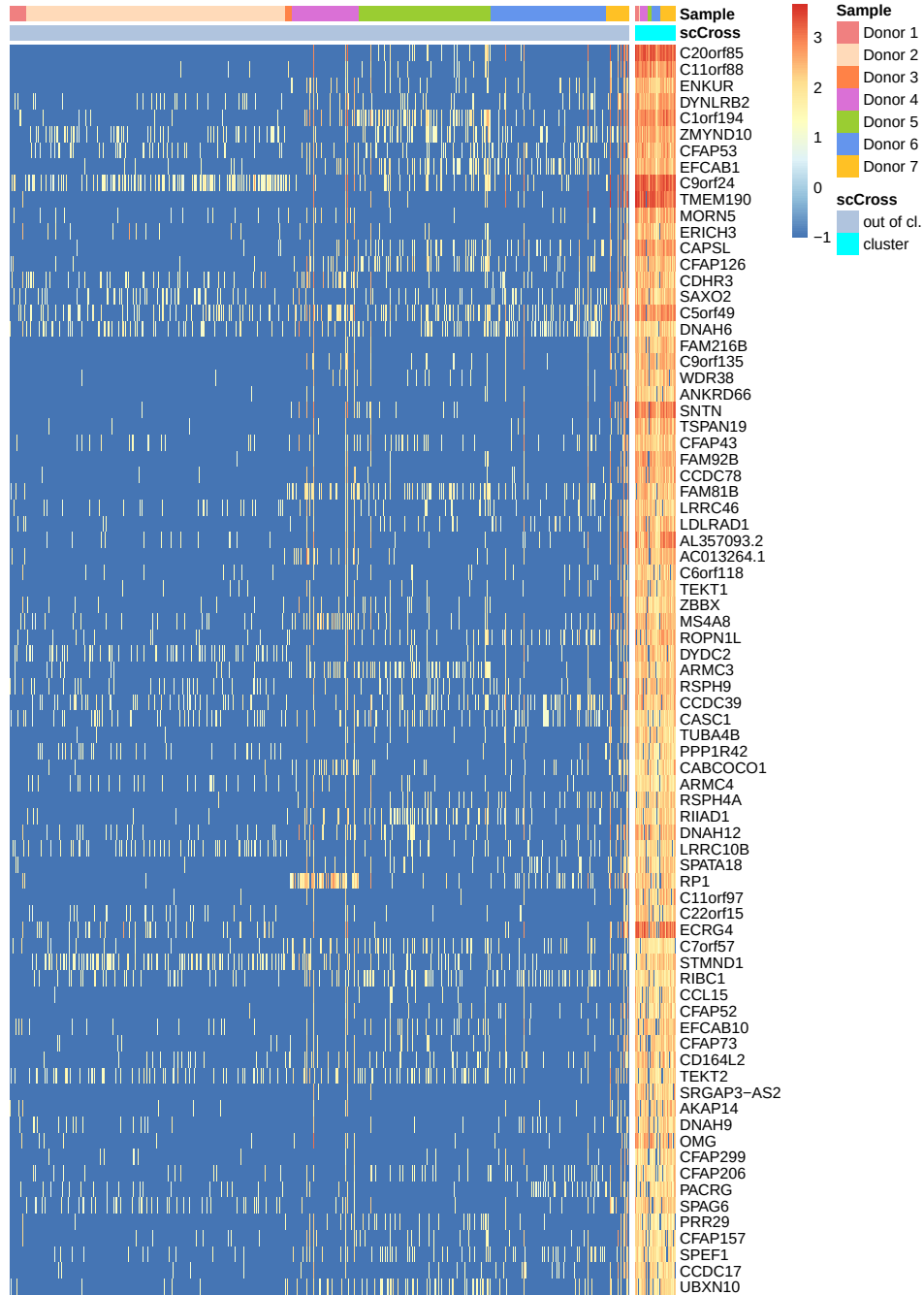


Figure 7.6: scCross result on the NSCLC DTCs dataset. Only the 1000 cells with highest objective value are displayed (out of 26 027).

Table 7.2: GO functions for the subpopulation identified by scCross within the NSCLC DTCs data. This analysis has been performed using the `clusterProfiler` R package (Yu et al., 2012). The 20 most significantly enriched GO terms are reported.

	ID	Description	p.adjust	GeneRatio
1	GO:0003341	cilium movement	5.0524e-22	17/38
2	GO:0007018	microtubule-based movement	1.7572e-21	20/38
3	GO:0035082	axoneme assembly	2.5309e-17	12/38
4	GO:0001578	microtubule bundle formation	5.5063e-16	12/38
5	GO:0060294	cilium movement involved in cell motility	3.6587e-15	12/38
6	GO:0001539	cilium or flagellum-dependent cell motility	5.8895e-15	12/38
7	GO:0060285	cilium-dependent cell motility	5.8895e-15	12/38
8	GO:0044782	cilium organization	1.8517e-14	15/38
9	GO:0060271	cilium assembly	1.7246e-13	14/38
10	GO:0030317	flagellated sperm motility	9.3987e-11	9/38
11	GO:0097722	sperm motility	9.3987e-11	9/38
12	GO:0044458	motile cilium assembly	9.2067e-10	7/38
13	GO:0003351	epithelial cilium movement involved in extracellular fluid movement	3.6916e-09	6/38
14	GO:0006858	extracellular transport	5.3579e-09	6/38
15	GO:0036159	inner dynein arm assembly	5.4005e-07	4/38
16	GO:0007286	spermatid development	1.8768e-06	7/38
17	GO:0048515	spermatid differentiation	2.2408e-06	7/38
18	GO:0007288	sperm axoneme assembly	3.8099e-06	4/38
19	GO:0120316	sperm flagellum assembly	1.2823e-05	4/38
20	GO:0003352	regulation of cilium movement	1.4323e-05	4/38

Table 7.3: Cilia Carta data for the subpopulation identified by scCross in the NSCLC DTCs data. This table continues on page 146.

Gene name	Cilia Carta rank	Cilia Carta score	Gene Onth.	Gold std.
C20orf85	122	4.348476959		
C11orf88	212	2.943095083		
ENKUR	54	6.472108308	True	
DYNLRB2	18	7.475605752	True	
C1orf194	131	4.348476959		
ZMYND10	140	4.063228988	True	
CFAP53				
EFCAB1	118	4.348476959		
C9orf24	325	1.939597639		
TMEM190				
MORN5	334	1.939597639		
ERICH3				
CAPSL	208	2.943095083		
CFAP126				
CDHR3	206	2.943095083		
SAXO2				
C5orf49	132	4.348476959		
DNAH6	109	4.63427188	True	True
FAM216B	333	1.939597639		
C9orf135	338	1.939597639		
WDR38				
ANKRD66				
SNTN	336	1.939597639	True	
TSPAN19				
CFAP43				
FAM92B	3702	-3.974226523	True	
CCDC78	322	1.939597639		
FAM81B	209	2.943095083		
LRRC46	125	4.348476959		
LDLRAD1				
AL357093.2				
AC013264.1				
C6orf118	120	4.348476959		
TEKT1	326	1.939597639	True	
ZBBX	328	1.939597639		
MS4A8				
ROPN1L	144	4.063228988	True	True
DYDC2	123	4.348476959		

Table 7.3: Cilia Carta data for the subpopulation identified by scCross in the NSCLC DTCs data. This table starts on page [145](#).

Gene name	Cilia Carta rank	Cilia Carta score	Gene Onth.	Gold std.
ARMC3	72	5.351974403		
RSPH9	87	5.066726433	True	True
CCDC39	595	0.144718304	True	True
CASC1	313	1.939597639		
TUBA4B				
PPP1R42				
CABCOCO1				
ARMC4	539	0.49832414	True	
RSPH4A	173	3.673731529	True	True
RIIAD1				
DNAH12	112	4.63427188	True	
LRRC10B				
SPATA18				
RP1	504	0.74932291	True	True
C11orf97				
C22orf15				
ECRG4				
C7orf57	324	1.939597639		
STMND1				
RIBC1	44	7.0431512		
CCL15	321	1.939597639		
CFAP52				
EFCAB10				
CFAP73				
CD164L2				
TEKT2	93	4.919519851	True	True
SRGAP3-AS2				
AKAP14	335	1.939597639	True	
DNAH9	14	7.475605752	True	
OMG				
CFAP299				
CFAP206				
PACRG	39	7.0431512	True	True
SPAG6	81	5.066726433	True	True
PRR29				
CFAP157				
SPEF1	83	5.066726433	True	
CCDC17	320	1.939597639		
UBXN10	159	3.916022407	True	

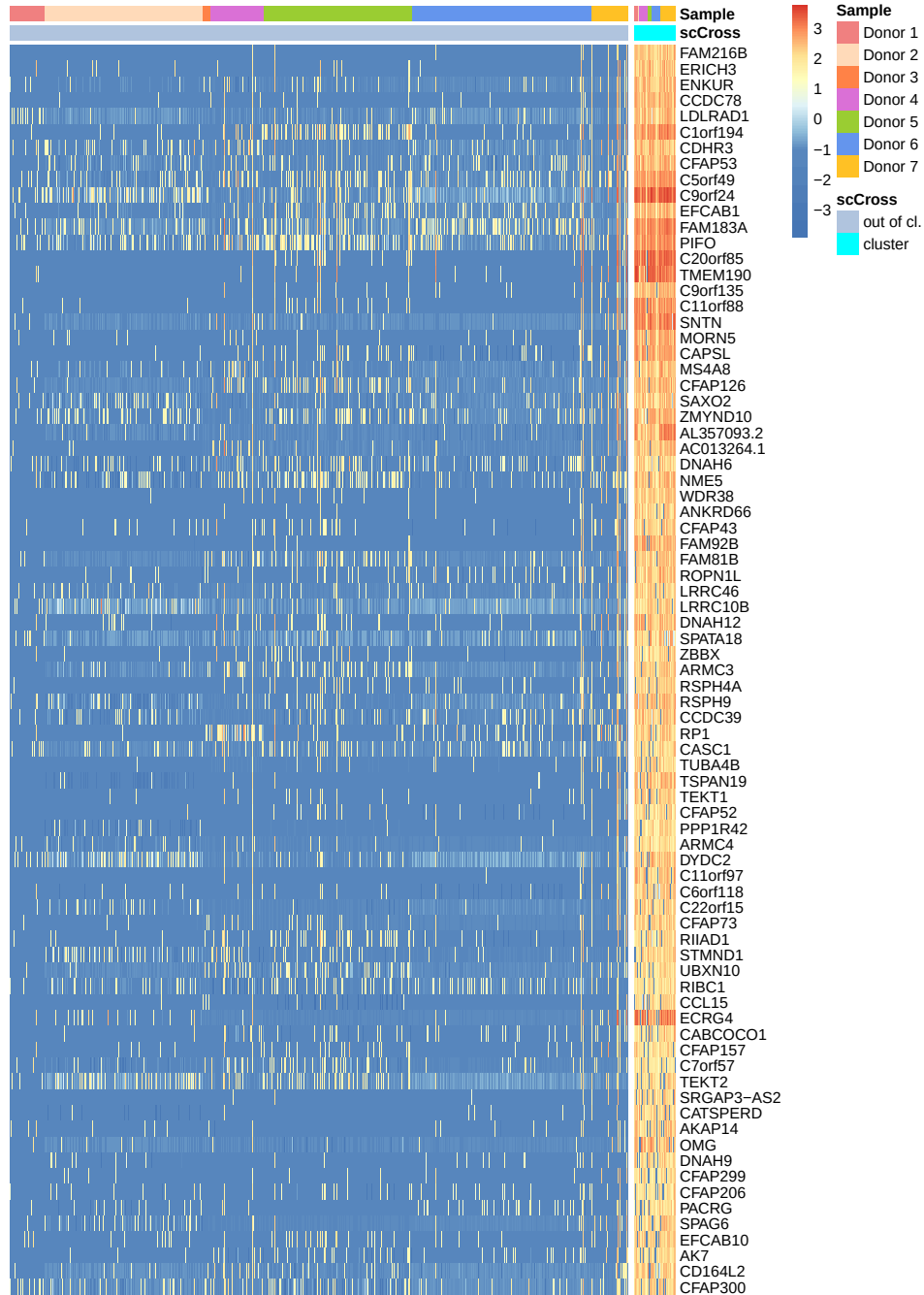


Figure 7.7: scCross results on the NSCLC DTCs dataset after data integration. The result closely resembles the one obtained on the original unintegrated data (figure 7.6), which indicates scCross doesn't suffer from the presence of batch effects.

7.3.4 Identification of subpopulations in pancreas data from two studies

This section investigates the effectiveness of scCross on data coming from different studies, and sequenced by different technologies. We aggregate eight samples of the human pancreas from cadaveric donors, coming from two studies. [Baron et al. \(2016\)](#) have sequenced four samples containing respectively 1937, 1724, 3605 and 1303 cells using the inDrop protocol ([Klein et al., 2015b](#)). [Muraro et al. \(2016\)](#) have sequenced four samples of 768 cells each with the CEL-seq2 protocol ([Hashimshony et al., 2016](#)). There are 16 251 genes in common between the two studies (respectively 82% and 86% of the genes returned by inDrop and CEL-seq2), 13 055 of them are used by scCross as they are expressed in at most 25% of the cells.

A t -SNE dimensional reduction (figure 7.8) reveals that there are strong batch effects between the two studies, as all clusters exclusively contain cells from the same study. We conjecture that, among possible other factors, the observed batch effects come from differences in gene detection rates between the two sequencing technologies. Yet, one expects to find shared expression patterns across samples, corresponding to cell types linked to the function of the pancreas in the human body, beyond sample-specific patterns coming from these batch effects.

Figure 7.9 shows the result of scCross. First, it identifies a subpopulation of 53 cells characterized by 22 genes linked to an immune response (according to a GO analysis, see table 7.4). Interestingly, these cells are well distributed across the 8 donors even though there is a visible difference in expression between the samples of the two studies. This visible difference is probably due to the differences in the sequencing pipeline between the two studies. When performing further executions of scCross, additional subpopulations shared by all 8 donors are exhibited. They consist of a tiny subpopulation of 14 cells with 37 cell cycle genes and a subpopulation of 526 cells with 17 genes involved in collagen and structural organization. These results demonstrate the ability of scCross to effectively group related cells together despite strong differences in expression levels induced by batch effects.

7.3.5 Identification of rare subpopulations across different mouse organs

Rather than searching for shared rare expression patterns across different donors, scCross can also be used to find rare subpopulations present across other types

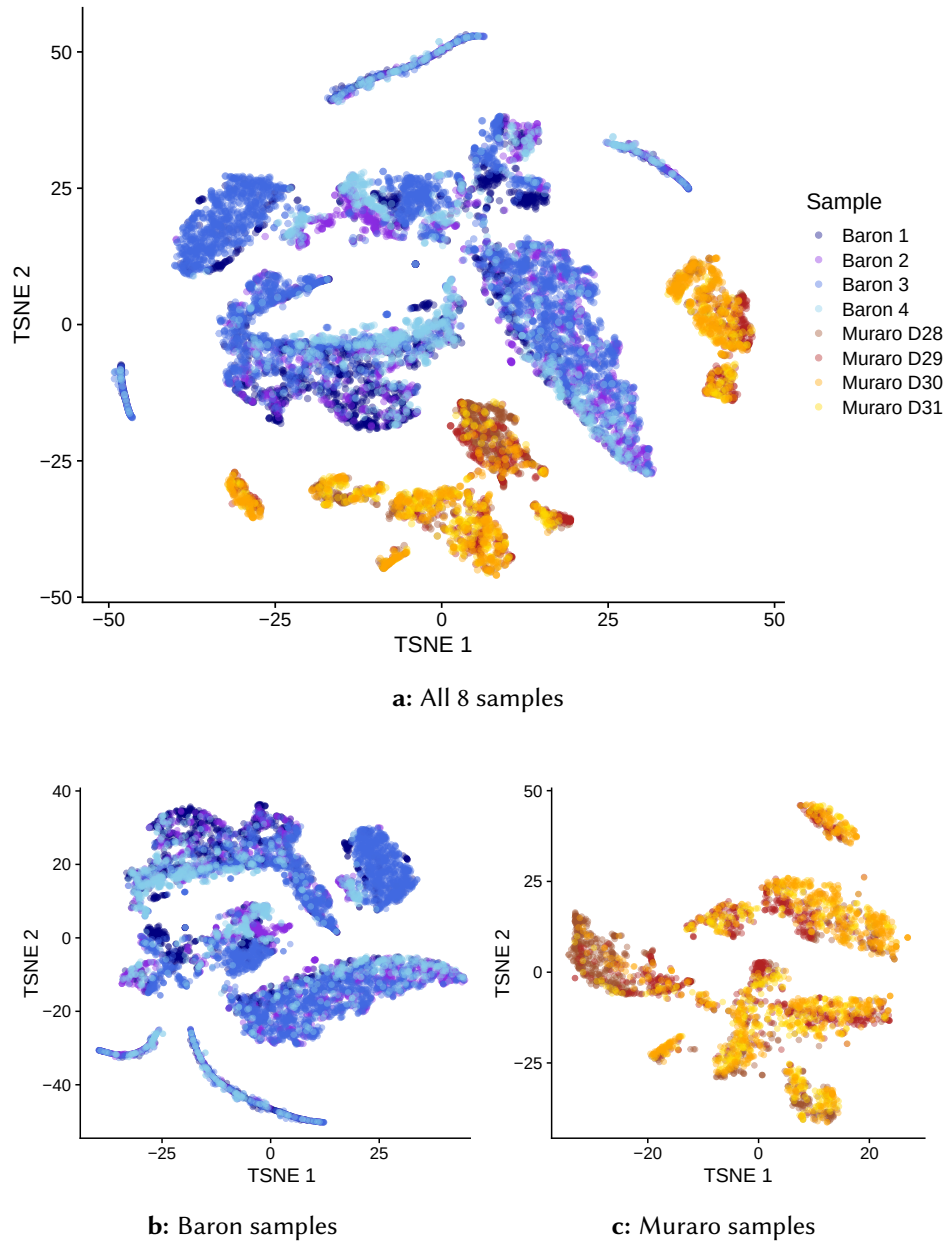


Figure 7.8: Dimensionality reduction of the pancreas data using t -SNE. **a** Reduced space generated with all 8 samples. One observes strong batch effects between the data from the two different studies. **b, c** Reduced spaces for the samples from each study separately. The cells from the different donors are relatively mixed, although some clusters contain more cells from a specific donors (which might indicate slight biological batch effects).

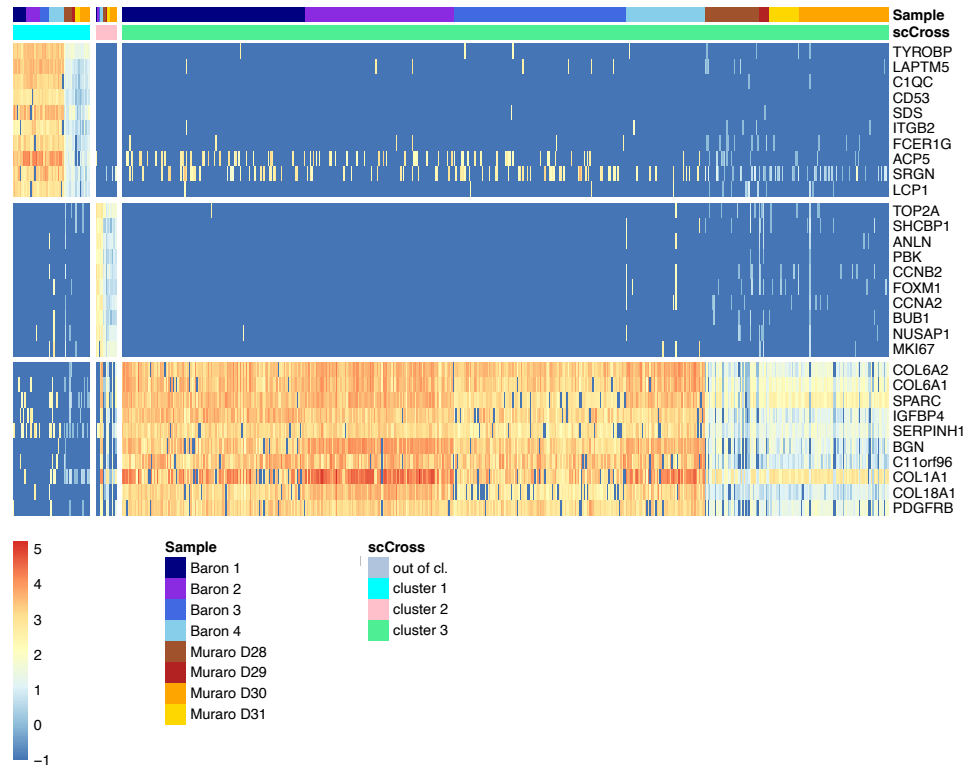


Figure 7.9: Summary of the results of scCross on the pancreas case study. For each cluster, 10 genes are displayed (out-of-cluster cells, which represent 96% of the total, are not displayed). One observes a significant shift in expression levels between the cells of the two studies. Yet, the global sum criterion used in scCross ensures related cells are grouped into the same cluster. Figures 7.10, 7.11 and 7.12 give the complete result for each cluster.

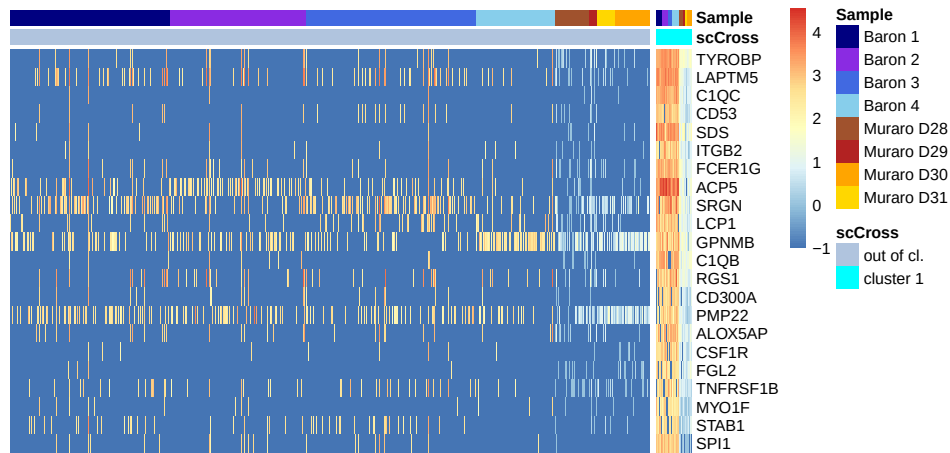


Figure 7.10: scCross result on the pancreas data. Only the 1000 cells with highest objective value are displayed (out of 11 641).

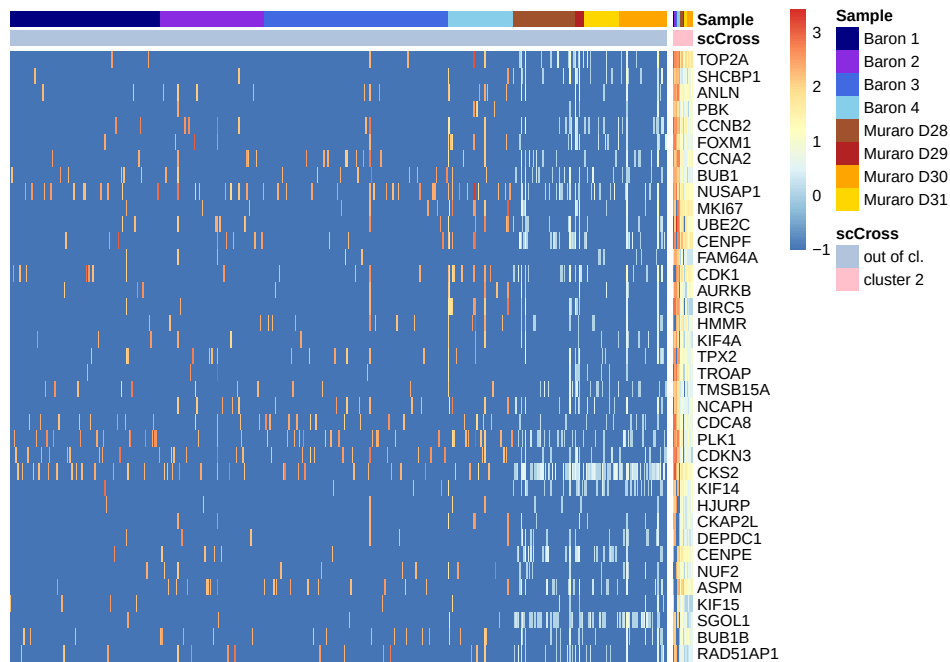


Figure 7.11: Second result of scCross on the pancreas data. Only the 500 cells with highest objective value are displayed (out of 11 641).

Table 7.4: GO terms for the first subpopulation identified in the pancreas data.
The 20 most significantly enriched GO terms are reported.

	ID	Description	p.adjust	GeneRatio
1	GO:0042119	neutrophil activation	1.0262e-06	5/22
2	GO:0036230	granulocyte activation	1.0281e-06	5/22
3	GO:0002283	neutrophil activation involved in immune response	1.6384e-06	4/22
4	GO:0002697	regulation of immune effector process	1.6611e-06	8/22
5	GO:0002443	leukocyte mediated immunity	1.8636e-06	8/22
6	GO:0002366	leukocyte activation involved in immune response	4.5741e-06	7/22
7	GO:0002263	cell activation involved in immune response	4.5741e-06	7/22
8	GO:0002683	negative regulation of immune system process	5.2637e-06	8/22
9	GO:0002275	myeloid cell activation involved in immune response	8.5849e-06	5/22
10	GO:0002695	negative regulation of leukocyte activation	8.5849e-06	6/22
11	GO:0050866	negative regulation of cell activation	1.4766e-05	6/22
12	GO:1902563	regulation of neutrophil activation	4.6081e-05	3/22
13	GO:0002886	regulation of myeloid leukocyte mediated immunity	4.6081e-05	4/22
14	GO:0051250	negative regulation of lymphocyte activation	8.2199e-05	5/22
15	GO:0002888	positive regulation of myeloid leukocyte mediated immunity	1.0990e-04	3/22
16	GO:0032928	regulation of superoxide anion generation	1.1921e-04	3/22
17	GO:0022408	negative regulation of cell-cell adhesion	1.7998e-04	5/22
18	GO:0002274	myeloid leukocyte activation	3.2344e-04	5/22
19	GO:0097529	myeloid leukocyte migration	3.2344e-04	5/22
20	GO:0002444	myeloid leukocyte mediated immunity	3.2344e-04	4/22

Table 7.5: GO terms for the second subpopulation identified in the pancreas data. The 20 most significantly enriched GO terms are reported.

	ID	Description	p.adjust	GeneRatio
1	GO:0000280	nuclear division	5.6920e-27	23/34
2	GO:0048285	organelle fission	2.8744e-26	23/34
3	GO:0007059	chromosome segregation	5.4023e-24	21/34
4	GO:0098813	nuclear chromosome segregation	1.3084e-22	19/34
5	GO:0140014	mitotic nuclear division	3.1803e-22	18/34
6	GO:0000819	sister chromatid segregation	1.8671e-21	17/34
7	GO:0000070	mitotic sister chromatid segregation	2.3947e-19	15/34
8	GO:0044772	mitotic cell cycle phase transition	1.4433e-18	18/34
9	GO:0051983	regulation of chromosome segregation	3.1870e-16	12/34
10	GO:0030071	regulation of mitotic metaphase/anaphase transition	3.7732e-14	10/34
11	GO:1902099	regulation of metaphase/anaphase transition of cell cycle	4.8086e-14	10/34
12	GO:0007091	metaphase/anaphase transition of mitotic cell cycle	4.9206e-14	10/34
13	GO:0051783	regulation of nuclear division	5.5664e-14	11/34
14	GO:0044784	metaphase/anaphase transition of cell cycle	5.8252e-14	10/34
15	GO:0033045	regulation of sister chromatid segregation	1.0061e-13	10/34
16	GO:1905818	regulation of chromosome separation	2.6983e-13	9/34
17	GO:0007088	regulation of mitotic nuclear division	2.9875e-13	10/34
18	GO:0033044	regulation of chromosome organization	4.0758e-13	12/34
19	GO:0007094	mitotic spindle assembly checkpoint signaling	4.0758e-13	8/34
20	GO:0071173	spindle assembly checkpoint signaling	4.0758e-13	8/34

Table 7.6: GO terms for the third subpopulation identified in the pancreas data.
The 20 most significantly enriched GO terms are reported.

	ID	Description	p.adjust	GeneRatio
1	GO:0030199	collagen fibril organization	6.3798e-09	6/16
2	GO:0030198	extracellular matrix organization	6.1434e-07	7/16
3	GO:0043062	extracellular structure organization	6.1434e-07	7/16
4	GO:0045229	external encapsulating structure organization	6.1434e-07	7/16
5	GO:0071230	cellular response to amino acid stimulus	6.8507e-07	5/16
6	GO:0071229	cellular response to acid chemical	9.8134e-07	5/16
7	GO:0043200	response to amino acid	3.8033e-06	5/16
8	GO:0001101	response to acid chemical	6.2356e-06	5/16
9	GO:0060348	bone development	8.3899e-05	5/16
10	GO:0032963	collagen metabolic process	1.1646e-04	4/16
11	GO:0061448	connective tissue development	1.4692e-04	5/16
12	GO:0071711	basement membrane organization	1.4692e-04	3/16
13	GO:0085029	extracellular matrix assembly	2.8402e-04	3/16
14	GO:0032964	collagen biosynthetic process	4.7617e-04	3/16
15	GO:0060350	endochondral bone morphogenesis	7.2685e-04	3/16
16	GO:0051216	cartilage development	8.8808e-04	4/16
17	GO:0038063	collagen-activated tyrosine kinase receptor signaling pathway	1.4056e-03	2/16
18	GO:0043589	skin morphogenesis	1.5922e-03	2/16
19	GO:0060349	bone morphogenesis	2.5769e-03	3/16
20	GO:0038065	collagen-activated signaling pathway	2.6001e-03	2/16

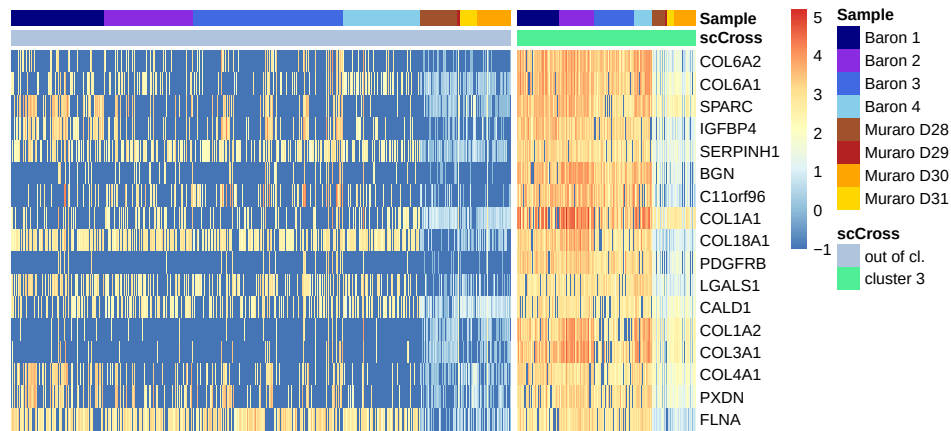


Figure 7.12: Third result of scCross on the pancreas data. Only the 2000 cells with highest objective value are displayed (out of 11 641).

of samples, such as different organs. The *Tabula Muris* ([Tabula Muris Consortium, 2018](#)) lets us compare gene expression in cell types that are shared between distinct tissues. More specifically, scCross has been used to analyze 2251 T cells sequenced across 6 organs using the Smart-seq2 protocol, with 15 024 genes expressed in at most 25% of the cells.

The first execution results in a subpopulation of 31 cells characterized by 61 genes (figures 7.13 and 7.14). A GO analysis links these genes to leukocyte activation (table 7.7). Unsurprisingly, the majority of the cells come from the bone marrow, where leukocytes are formed from stem cells. Yet, 7 cells also come from the fat, as well as a few cells in other tissues, which could constitute a sign of inflammation.

A second execution of scCross (figures 7.13 and 7.15) finds a cluster of 91 cells with 57 genes indicating proliferation (including *Top2a*, *Rrm2*, *Kif11* and *Pbk*). A part of this cluster has been identified in the original publication related to this data ([Tabula Muris Consortium, 2018](#)), but it was interpreted as a cluster of thymic T cells. Even though the large majority of the cells indeed come from the thymus, scCross shows proliferation also appears in the bone marrow and fat, as well as a few cells in the lung, muscle and spleen. This illustrates that scCross can be used to analyze rare cell types across organs.

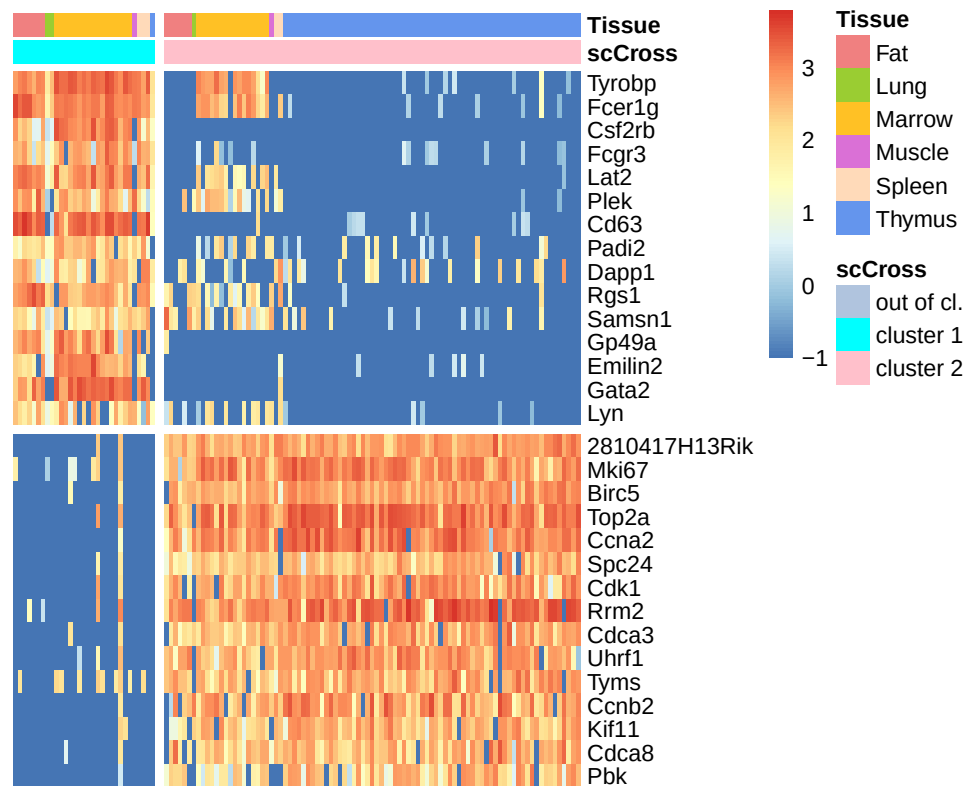


Figure 7.13: Summary of the result of scCross on the *Tabula Muris* data. For each cluster, the 15 best genes are displayed (out-of-cluster cells, i.e. 96% of the total, are not displayed). The full results of each cluster are presented in figures 7.14 and 7.15.



Figure 7.14: scCross result on the *Tabula Muris* data. Only the 1000 cells with highest objective value are displayed (out of 2251). The value of μ was set to 20%, as the default value of 10% yields a solution with only two genes (Tyrobp and Fcer1g), which motivates the fact to relax the constraint on the maximum number of negatives values allowed within the bicluster.

Table 7.7: GO terms for the first subpopulation identified in the *tabula muris* data. The 20 most significantly enriched GO terms are reported.

	ID	Description	p.adjust	GeneRatio
1	GO:0002443	leukocyte mediated immunity	3.0556e-11	15/56
2	GO:0002444	myeloid leukocyte mediated immunity	3.5094e-11	10/56
3	GO:0002274	myeloid leukocyte activation	7.1583e-11	12/56
4	GO:0002275	myeloid cell activation involved in immune response	3.3488e-10	9/56
5	GO:0043299	leukocyte degranulation	2.0241e-09	8/56
6	GO:0045576	mast cell activation	2.0419e-09	8/56
7	GO:0043303	mast cell degranulation	5.0292e-09	7/56
8	GO:0002366	leukocyte activation involved in immune response	5.0292e-09	11/56
9	GO:0002448	mast cell mediated immunity	5.0292e-09	7/56
10	GO:0002263	cell activation involved in immune response	5.0292e-09	11/56
11	GO:0002279	mast cell activation involved in immune response	5.0292e-09	7/56
12	GO:0045055	regulated exocytosis	9.7862e-09	10/56
13	GO:0002886	regulation of myeloid leukocyte mediated immunity	2.0849e-08	7/56
14	GO:0006887	exocytosis	2.4006e-08	11/56
15	GO:0032418	lysosome localization	6.1318e-08	7/56
16	GO:1990849	vacuolar localization	6.1318e-08	7/56
17	GO:0050727	regulation of inflammatory response	3.1718e-07	10/56
18	GO:0002703	regulation of leukocyte mediated immunity	9.1296e-07	9/56
19	GO:0043304	regulation of mast cell degranulation	1.1063e-06	5/56
20	GO:0050866	negative regulation of cell activation	1.3120e-06	8/56

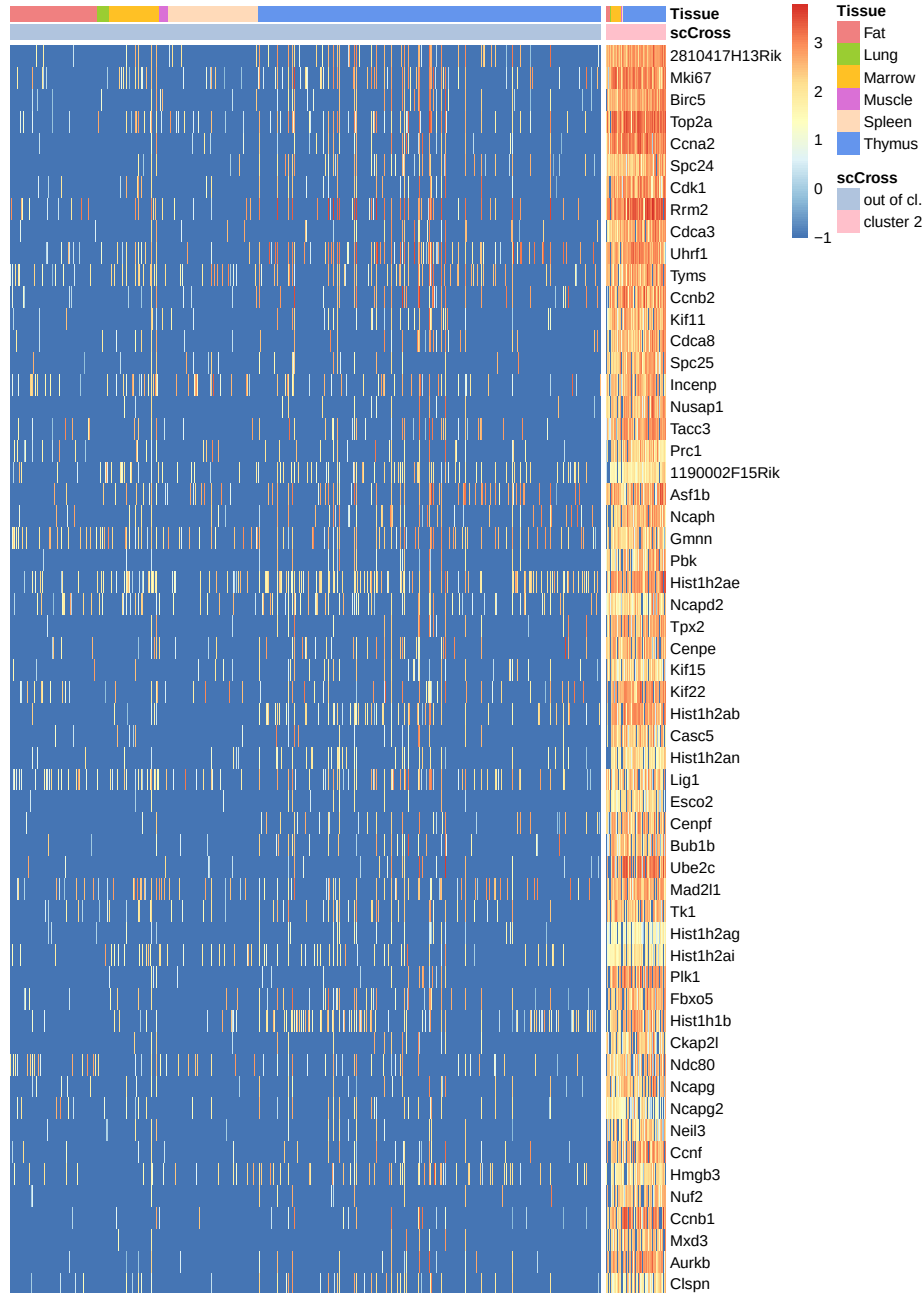


Figure 7.15: Second result of scCross on the *Tabula Muris* data. The result with $\kappa = 0.5$ is displayed, which leads to a stricter selection of genes (62 instead of 173 with the original $\kappa = 0.044$ value), which in turn leads to the inclusion of 20 previously out of cluster cells (which expressed a significant part of the 173 genes with $\kappa = 0.044$).

Table 7.8: GO terms for the second subpopulation identified in the *tabula muris* data. The 20 most significantly enriched GO terms are reported.

	ID	Description	p.adjust	GeneRatio
1	GO:0007059	chromosome segregation	9.5400e-50	35/52
2	GO:0000819	sister chromatid segregation	2.0976e-48	30/52
3	GO:0098813	nuclear chromosome segregation	5.3012e-48	32/52
4	GO:0000070	mitotic sister chromatid segregation	7.3412e-47	28/52
5	GO:0140014	mitotic nuclear division	5.2846e-46	30/52
6	GO:0000280	nuclear division	4.9782e-43	32/52
7	GO:0051983	regulation of chromosome segregation	2.7251e-41	24/52
8	GO:0051304	chromosome separation	7.0365e-39	21/52
9	GO:1905818	regulation of chromosome separation	1.0509e-37	20/52
10	GO:0007088	regulation of mitotic nuclear division	4.8092e-31	19/52
11	GO:0033044	regulation of chromosome organization	1.2085e-30	22/52
12	GO:0033047	regulation of mitotic sister chromatid segregation	3.8918e-30	16/52
13	GO:0033045	regulation of sister chromatid segregation	5.9684e-30	18/52
14	GO:0010965	regulation of mitotic sister chromatid separation	5.9684e-30	16/52
15	GO:0044772	mitotic cell cycle phase transition	5.9684e-30	25/52
16	GO:0051306	mitotic sister chromatid separation	1.3296e-29	16/52
17	GO:0051783	regulation of nuclear division	4.5944e-29	19/52
18	GO:0033046	negative regulation of sister chromatid segregation	6.9632e-29	15/52
19	GO:0033048	negative regulation of mitotic sister chromatid segregation	6.9632e-29	15/52
20	GO:0045841	negative regulation of mitotic metaphase/anaphase transition	6.9632e-29	15/52

7.4 Multiple beam solver

The scCross solver is based on the same principle as the MicroCellClust 2 solver (see section 6.1), i.e. combines a beam search, to efficiently find a first approximate solution, with a local search algorithm (the rareness score heuristics to scale to large datasets are also applicable here). Yet, simply using the MicroCellClust 2 solver as such doesn't produce the desired result. Indeed, an optimal solution to the scCross optimization problem is expected to contain many cells from different samples. However, the beam search starts from clusters containing only a few cells (and therefore representing few different samples). Therefore, the δ_i factor gives a large penalty to all solutions, canceling out its effect during this early phase. One frequently observes that the beam search is drawn to a region of the search space relative to only one sample, and gets stuck there.

The beam search is therefore modified to take advantage of the sample identity of each cell (contained in a vector s , where s_j indicates from which sample cell j comes), so as to ensure diversity in the search (see algorithm 7.1).

The first adaptation consists in maintaining separate beams for each sample, in order to avoid filling the beam only with solutions related to one sample. At each search level, $sbeams_s$ keeps track of the 100 best solutions containing at least one cell from sample s . At the next search level, the algorithm expands the union of all solutions contained in the sample beams. This ensures a larger diversity of solutions if the search is temporarily dominated by single-sample ones. If the 100 best solutions all contain cells from one same sample, this strategy ensures other solutions, containing cells from other samples, are also selected for further expansion. In the long run, cross-sample solutions will be favored thanks to the effect of δ_i , and all beams will tend to contain the same solutions (thus going back to a regular beam search).

A second adaptation consists in adding an extra initialization step (algorithm 7.2), which aims at finding a small solution present in as many samples as possible by combining pairs of cells from different samples. It starts by selecting the 100 best pairs of each sample, which form new surrogate variables for a search where only pairs from different samples are combined (i.e. solutions of this initialization step contain at most 2 cells from the same sample). The best solutions from this initialization step form the starting point of the actual beam search, which explores the search space level by level by adding one cell to solutions from the previous level.

Algorithm 7.1: scCross – beam_search

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ an expression matrix

Input: $\mathbf{sid} \in \mathbb{N}^{|\mathcal{C}|}$ a vector indicating to which sample each cell belongs

Input: $\kappa \in \mathbb{R}_+$, $\mu \in [0, 1]$ the parameters of problem (7.3)(7.4)

Output: $I^\sim \subseteq \mathcal{G}$, $J^\sim \subseteq \mathcal{C}$, $\Omega^\sim \in \mathbb{R}$ sets of genes and cells approximating (7.3)(7.4), and corresponding objective value

```

/* Get an initial set of solutions with cells from
   at least two different samples (algorithm 7.2) */
beamℓstart, ℓstart ← starting_phase(M, sid,  $\kappa$ ,  $\mu$ )
I~, J~, Ω~ ← best(beamℓstart)

// Continue beam search
for ℓ ∈ {ℓstart + 1, ...} do
    beams ← ∅|S| // One beam for each sample
    for J0 ∈ beamℓ-1 do
        for j ∈ C \ J0 do
            J ← J0 ∪ {j}
            I, Ω ← obj(M, sid, J,  $\kappa$ ,  $\mu$ )
            /* Add to beams of each sample represented
               in J */
            for s ∈ {sidj | j ∈ J} do
                beamss ← 100_best(beamss ∪ {J})
            if Ω ≥ Ω~ then I~, J~, Ω~ ← I, J, Ω

    // Combine all beams for next search level
    beamℓ ← ⋃s ∈ S beamss

return I~, J~, Ω~

```

Algorithm 7.2: scCross – starting_phase

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ an expression matrix
Input: $\mathbf{sid} \in \mathcal{S}^{|\mathcal{C}|}$ a vector indicating to which sample each cell belongs
Input: $\kappa \in \mathbb{R}_+$, $\mu \in [0, 1]$ the parameters of problem (7.3)(7.4)
Output: $res \subseteq \mathcal{P}(\mathcal{C})$, $\ell \in \mathbb{N}$ a set of cell-clusters of size ℓ , each containing at most 2 cells from the same sample

// Generate pairs of cells from the same sample

for $s \in \mathcal{S}$ **do**

$pairs_s \leftarrow \emptyset$

for $j \in \mathcal{C}_s$; $j' \in \{j' \in \mathcal{C}_s \mid j < j'\}$ **do**

$J \leftarrow \{j, j'\}$; $I, \Omega \leftarrow \text{obj}(\mathbf{M}, \mathbf{sid}, J, \kappa, \mu)$

$pairs_s \leftarrow 100_best(pairs_s \cup \{J\})$

$beam_{\ell=1} \leftarrow \bigcup_{s \in \mathcal{S}} pairs_s$

// Beam search with pairs as variables

$res \leftarrow \emptyset$; $\ell_{res} \leftarrow 0$

for $\ell \in \{2, \dots, |\mathcal{S}|\}$ **do**

$sbeams \leftarrow \emptyset^{|\mathcal{S}|}$ // One beam for each sample

for $J^0 \in beam_{\ell-1}$ **do**

for $s' \in \{s \in \mathcal{S} \mid s \neq sid_j \forall j \in J^0\}$ **do** // Only add pairs

for $P \in pairs_{s'}$ **do** // from $\neq$ samples

$J \leftarrow J^0 \cup P$; $I, \Omega \leftarrow \text{obj}(\mathbf{M}, \mathbf{sid}, J, \kappa, \mu)$

for $s \in \{sid_j \mid j \in J\}$ **do**

$sbeams_s \leftarrow 100_best(sbeams_s \cup \{J\})$

$beam_\ell \leftarrow \bigcup_{s \in \mathcal{S}} sbeams_s$

if $\text{best}(beam_\ell) \geq \text{best}(res)$ **then** $res \leftarrow beam_\ell$; $\ell_{res} \leftarrow \ell$

return res, ℓ_{res}

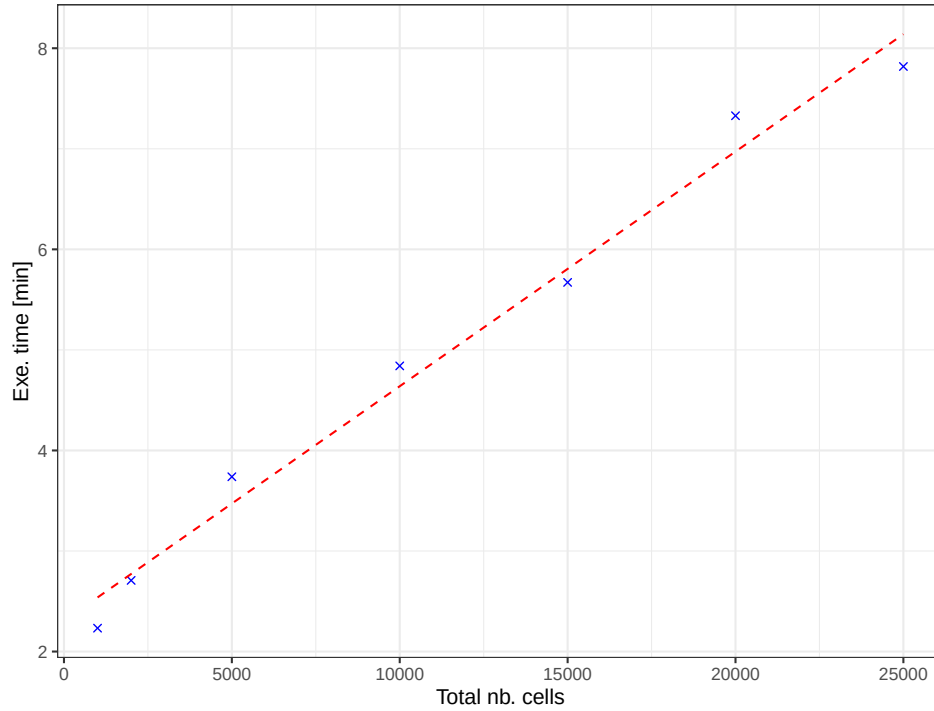


Figure 7.16: Execution time of scCross when sampling datasets of various sizes from the NSCLC DTCs dataset. This experiment was conducted on a MacBook Pro laptop (Mac OS 14.3.1, Apple M1 pro CPU, 16GB RAM).

Finally, as for the MicroCellClust 2 solver, the solution of the beam search serves as initialization for a simulated annealing algorithm, which will stochastically explore its neighborhood in order to improve the quality of the solution and produce the final scCross result. Figure 6.5 shows the execution time of scCross evolves linearly with respect to the number of cells (when using the rareness score heuristics, see section 6.1).

7.5 Discussion

This chapter presents scCross, a method for identifying small subpopulations of cells with specific genes across multiple single-cell samples. It relies on a global sum criterion, which proves insensitive to the batch effects typically present in multi-sample data. Moreover, it explicitly searches for shared expression pat-

terns across samples, which contrasts with the traditional two-step approach of first integrating datasets (i.e. transforming the data into one sample) and then using traditional clustering algorithms (thus ignoring the partition into multiple samples).

The reported experiments show scCross effectively groups related cells together, even with strong batch effects resulting from the use of different sequencing technologies. These experiments also show the ability of scCross to jointly identify specific genes for each subpopulation. In particular, a case study on lung cancer cells shows that it identifies a subpopulation characterized by known ciliary genes, together with other genes that have not yet been linked to the cilium. This particular subpopulation is missed by typical alternatives, even when they run on integrated data. Moreover, scCross shows a higher stability than alternative methods on a controlled experiment where different kind of batch effects are artificially created. Such results show the effectiveness of scCross for a joint identification of cells and genes.

Modifying the optimization problem

The goal of scCross is to identify rare subpopulations of cells that are present most or all of the different samples contained in the aggregated single-cell data. This necessitates changing the optimization problem of MicroCellClust to include this additional objective.

There are several approaches that can be explored, one of which is extending the optimization problem with new constraints to enforce this behavior. For instance, one can require the solution to contain at least one cell from each sample:

$$\exists j \in J : j \in \mathcal{C}_s \quad \forall s \in \mathcal{S} \quad (7.6)$$

or a relaxed version requiring a minimum number of samples from which the cells must come. Yet this approach suffers from two drawbacks. First, it necessitates a strict criterion (i.e. the number of samples present in the final solution) and therefore doesn't allow any flexibility. Second, only adding constraints without changing the objective function can lead to "borderline" solutions, whereby some loosely related cells are added to an existing solution in order to satisfy the constraints, rather than searching for an actual cross-sample result. This motivates the approach described in section 7.2, which modifies the objective function such that the optimization naturally tends to maximize the number of samples present in the solution. This can actually be seen as a sort of *soft constraint*: there is no strict pass/fail condition, but genes violating the requirements will be penalized.

It is possible to further extend the objective function of scCross as one might also consider relaxing some of the constraints, in particular the per-gene constraints (equation (4.4)). Indeed, these constraints impose a maximum proportion μ of negative values for each gene within the solution. Yet, the purpose of the modified objective function of scCross is to allow a gene to not be expressed within a few of the samples (albeit with a penalty). Such a gene might however still be canceled out if the negative values in these samples exceed the proportion allowed by the constraint. It might therefore make sense to soften these constraints.

One way of doing so is to count, for each gene, the number of samples that satisfy the constraint. Let $\lambda_{is}(J)$ be a boolean value indicated whether the maximum proportion of negative values of gene i is below μ within the selected cells of sample s :

$$\lambda_{is}(J) = \begin{cases} 1 & \text{if } - \sum_{j \in \mathcal{C}_s \cap J} \min\{0, \text{sign}(m_{ij})\} \leq \mu \cdot |\mathcal{C}_s \cap J| \\ 0 & \text{otherwise} \end{cases} \quad (7.7)$$

We can add a soft constraint to the objective function by multiplying the contribution of gene i by the proportion of samples for which it satisfies the constraint:

$$(I^*, J^*) = \underset{\substack{I \subseteq \mathcal{G} \\ J \subseteq \mathcal{C}}}{\text{argmax}} \sum_{i \in I} \left(\delta_i(J) \sum_{s \in \mathcal{S}} \omega_{is}(J) \cdot \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \lambda_{is}(J) \right) \quad (7.8)$$

$$\text{such that} \quad - \sum_{i \in I} \sum_{j \in J} \min\{0, \text{sign}(m_{ij})\} \leq \mu |I| |J| \quad (7.9)$$

The contribution of gene i is in this case linearly penalized by the number of samples violating the constraint.

However, when evaluating this strategy on some scRNA-seq case-studies, it does not appear to fundamentally change the result. Indeed, the additional genes that are now included contribute very lowly to the objective value of the bicluster, and therefore has no impact on the nature of the identified subpopulation. On the other hand, it has a significant impact on the execution time, as the solver can no longer filter out genes that do not satisfy the constraint. Given its limited impact on the solution and its high computational cost, these soft constraints were not retained in the scCross model.

Even so, the new optimization problem of scCross is highly non-linear, and more complicated to solve than the original MicroCellClust problem. Indeed, a

classical beam search algorithm, which is very effective at identifying approximate solutions in the MicroCellClust space, struggles to find cross-sample solutions. Yet, performing an explicit search across samples enables the solver to use this information to diversify the search using multiple beams. Such a procedure ensures different parts of the search space are visited, effectively converging to a solution that contain cells from multiple samples.

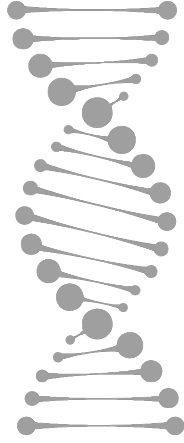
Perspectives

The main focus of scCross is to identify subpopulations shared by different donors in single-cell data, but it can be used in other research settings. We show that it can also be used, for example, to identify genes patterns shared across different organs. Moreover, the underlying optimization problem is generic: it only takes as input a gene $\times$ cell data matrix, and a vector indicating to which sample each cell belongs. It therefore makes no assumption on how the different samples are generated, and could even be applied to other data types than scRNA-seq.

This chapter focuses on data coming from different samples where the same modality is sequenced (here scRNA-seq), also called *horizontal* integration. The following chapter evaluates how the approaches developed during this thesis can be used to deal with *vertical* integration, whereby several modalities are sequenced for the same sample of cells, in order to find subpopulations with strong links between the different modalities.

Perspectives for multi-modal analysis

8



Recent advances in multi-omics sequencing technologies allows the sequencing of multiple modalities inside the same single-cells. For instance, one can sequence both the transcriptome and the epigenome of individual cells. This raises an interesting research question: how are these modalities linked together, and how do they influence the cell function. This chapter investigates how methods based on the global sum criterion can be used in this context.

Section 8.1 presents a general framework to adapt MicroCell-Clust (or scCross) to multi-omics data. Section 8.2 gives a concrete example on data combining scRNA-seq and scATAC-seq. The goal is to identify subpopulations of cells that coexpress both a set of genes and a series of open chromatin regions, which allows the study of relationships between enhancer (co)activity and gene expression.

8.1 General framework

Let $\mathcal{X}$ be a set of modalities. A single-cell dataset sequenced by some modality x can be represented by a matrix $\mathbf{M}_x \in \mathbb{R}^{|\mathcal{F}_x| \times |\mathcal{C}|}$, with $\mathcal{C}$ the set of cells, and $\mathcal{F}_x$ the set of features sequenced by that modality (e.g. genes in case of scRNA-seq). Multiple modalities, sequenced in the same cells $\mathcal{C}$, can be aggregated in a single matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{C}|}$, with $\mathcal{F} = \bigcup_{x \in \mathcal{X}} \mathcal{F}_x$, by concatenating the rows of the corresponding data matrices:

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_{|\mathcal{X}|} \end{pmatrix} \quad (8.1)$$

In principle, the MicroCellClust optimization problem (or scCross in case one also aggregates multiple biological samples) could simply be applied to this matrix. Yet, it makes little sense to use the same κ and μ values for different modalities. For instance, one could be more subject to dropouts during the sequencing phase. Using a shared value of μ would in this case lead to selecting less features of that particular modality compared to the others.

The MicroCellClust optimization problem can be simply modified to introduce separate κ_x and μ_x values for each modality:

$$(I^*, J^*) = \operatorname{argmax}_{\substack{I \subseteq \mathcal{F} \\ J \subseteq \mathcal{C}}} \sum_{x \in \mathcal{X}} \sum_{i \in I_x} \left(\sum_{j \in J} m_{ij} - \kappa_x \sum_{k \in \mathcal{C} \setminus J} \max\{0, m_{ik}\} \right) \quad (8.2)$$

$$\text{such that } \frac{|\{(i, j) \mid i \in I_x, j \in J, m_{ij} < 0\}|}{|I_x| \cdot |J|} \leq \mu_x \quad \forall x \in \mathcal{X} \quad (8.3)$$

From a computational point of view, this introduces little changes when computing the optimal assignment of features I_J^* given a set of cells J (see section 5.1): one simply computes the optimal assignment of features for each modality separately, and sum the result. The MicroCellClust 2 solver (see section 6.1) can therefore be reused directly by replacing the calls to `obj` by algorithm 8.1.

Algorithm 8.1: `multiomic_obj`

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$ an expression matrix

Input: $J \subseteq \mathcal{C}$ the assignment of cells for which to compute the optimal assignment of genes

Input: $\kappa \in \mathbb{R}_+^{|\mathcal{X}|}$, $\mu \in [0, 1]^{|\mathcal{X}|}$ the parameters of problem (8.2)(8.3)

Output: $I_J^* \subseteq \mathcal{G}$, $\omega \in \mathbb{R}$ the optimal assignment of genes given J , and corresponding objective value

for $x \in \mathcal{X}$ **do**

$I_x, \Omega_x \leftarrow \text{obj}(\mathbf{M}_x, J, \kappa_x, \mu_x)$

return $\bigcup_{x \in \mathcal{X}} I_x, \sum_{x \in \mathcal{X}} \Omega_x$

8.2 Combining transcriptomics and chromatin accessibility

As a concrete example of applying the methods developed during this thesis on multiomic data, this section presents the application of MicroCellClust on data combining single-cell transcriptomics and epigenomics.

8.2.1 Single-cell epigenomics

Epigenomics is the study of chemical changes occurring in the DNA of a cell that affects when, where, and how the DNA is expressed. These modifications do not change the DNA sequence itself but affect gene activity and expression. It can result in changes to the structure of chromatin and changes to the function of the genome (Bernstein et al., 2007). Single-cell epigenomics allows the study of the epigenetic modifications in individual cells (Kelsey et al., 2017).

One type of single-cell epigenomics techniques is *single-cell ATAC-seq* (Assay for Transposase-Accessible Chromatin using sequencing). It identifies regions of open chromatin, which enables the identification of regulatory elements such as promoters and enhancers. Both are non-coding sequences that regulate gene expression. *Promoters* initiate transcription, and are therefore located near the beginning of a gene. *Enhancers*, on the other hand, fine-tune gene expression levels and can be located far away from the genes they regulate, sometimes even a million of base pairs away (Panigrahi and O'Malley, 2021; Schoenfelder and Fraser, 2019). These two interact together through *chromatin looping*, which physically brings them closer and enhances transcriptional activation (figure 8.1). Indeed, while they can be far away in terms of linear distance, they may be spatially close due to the way chromatin is folded within the nucleus, which allows distant enhancers to influence gene transcription.

Recent advances allow the sequencing of scATAC-seq and of scRNA-seq in the same single cells. This is a powerful way to gain deeper insights into cellular heterogeneity, gene regulation, and cell state transitions. Indeed, a cell expressing specific marker genes might also exhibit open chromatin regions (assessed through scATAC-seq) associated with those genes. As scATAC-seq enables the identification of promoters and enhancers, integrating this information with scRNA-seq data allows to link gene expression levels to specific regulatory elements, and study how chromatin accessibility influences gene expression. By examining co-accessibility patterns (i.e. regions that tend to be open together), one can predict potential enhancer-promoter interactions (Carter and Zhao, 2021). Finally, changes in chromatin accessibility indicate transitions between cell states

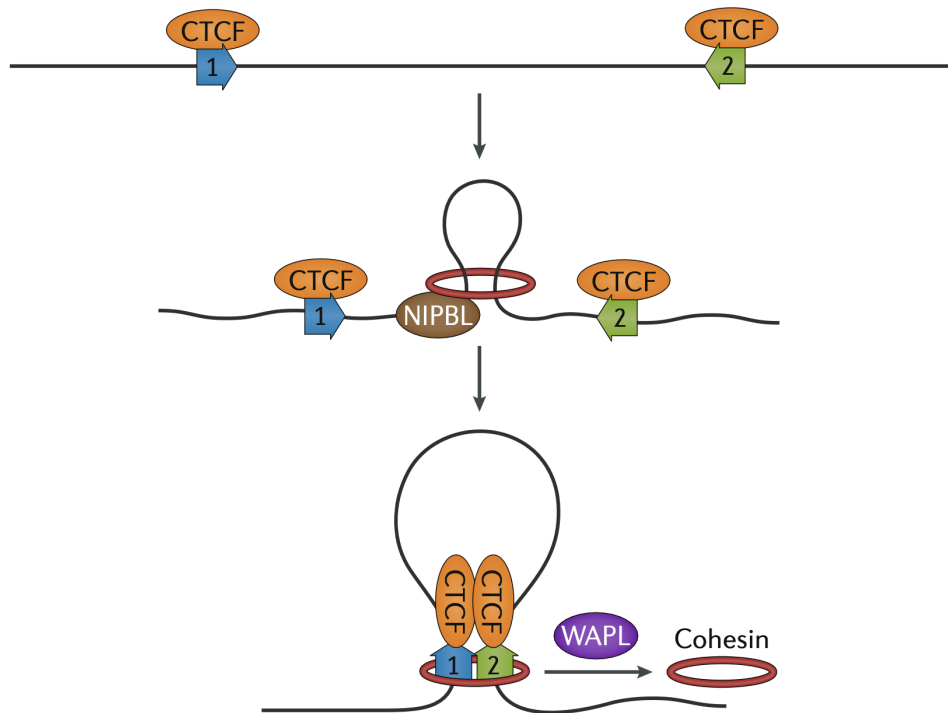


Figure 8.1: Chromatin looping formed by extrusion (Schoenfelder and Fraser, 2019). Cohesin, a protein complex with a ring-shaped structure, progressively extrudes chromatin to form a growing loop. The extrusion stops whenever the cohesin encounters a CTCF protein, which are located near promoters and enhancers. The resulting physical proximity allows the enhancer to fine tune the expression level of a gene located near the promoter.

(e.g. differentiation, activation, or response to stimuli), whereas scRNA-seq only provides a snapshot of cell states (Kelsey et al., 2017).

8.2.2 Enhancer co-activity

Enhancers are often cell-type specific: they activate genes only in specific cell types or under certain conditions. This ensures precise gene regulation tailored to each cell's function, contributing to cellular diversity. Moreover, enhancers can regulate multiple genes simultaneously, allowing coordinated control of gene expression. Conversely, multiple enhancers may target the same gene. This increases the robustness of gene regulation, and contributes to cell-specific gene expression patterns (Kvon et al., 2021; Osterwalder et al., 2018). Indeed, cells often rely on an intricate network of enhancers to regulate gene expression. In particular, *shadow enhancers* are groups of enhancers that regulate the same gene with overlapping activity in space and time. This provides robustness to genetic variation, such as loss-of-function mutations in individual enhancers, and environmental stress, guaranteeing stable and precise gene transcription.

While several studies have identified gene-enhancer associations, Ziyani et al. (2023) investigated the association between multiple enhancers at a large-scale using multimodal scRNA-seq and scATAC-seq, by exploring the co-activity between enhancers associated with a certain gene. Indeed, it was unknown whether shadow enhancers are active at the same time. By having gene expression and enhancer activity measurements in the same cells, they observed sets of enhancers active upon gene expression, suggesting that enhancer co-activity occurs pervasively across genes.

Yet, such an approach relied on the *a priori* selection of target genes, for which the coactivity between associated enhancers is analyzed. Such an analysis is thus only multivariate regarding the enhancers, but doesn't explore how the coactivity between certain enhancers could affect the expression of other genes. Moreover, only the enhancers located within a certain vicinity (1 Mb around gene TSS) were considered. Given that enhancers can be located far away from the genes they regulate (they even need not be located on the same chromosome), such a semi-automatic procedure could miss potential associations.

Using a fully automatic and multivariate approach makes therefore sense to study the relationships between enhancers and gene expression. MicroCellClust (or an adaptation thereof) can be used to identify sets of genes and peaks that are jointly and specifically expressed in a (rare) subpopulation of cells, which can serve as basis to investigate how these elements interact with each other.

8.2.3 Applying MicroCellClust on 10X Multiome

The 10X Multiome platform, or Chromium Single Cell Multiome ATAC + Gene Expression, is a sequencing platform using single-cell barcodes to directly link chromatin accessibility data to gene expression data. This platform gives a data matrix $\mathbf{M}$ with $\mathcal{X} = \{\text{RNA}, \text{ATAC}\}$. The measured features can be represented by $\mathcal{F} = \mathcal{G} \cup \mathcal{P}$, where $\mathcal{G}$ denotes the set of *genes* sequenced through scRNA-seq, and $\mathcal{P}$ the set of *peaks* (i.e. DNA regions representing an area of open chromatin) sequenced by scATAC-seq.

MicroCellClust is applied on a dataset containing 10 032 human PBMCs (see appendix A.10 for details on this data), with 23 246 genes and 4634 peaks expressed in at most 25% of the cells. With default parameters $\kappa = 0.01$ and $\mu = 10\%$, the result is a bicluster containing 102 cells and 49 features (figure 8.2). It corresponds to a rare subpopulation of plasmacytoid dendritic cells (102 out of the 109 cells previously labeled “plasmacytoid DC” are selected).

While this subpopulation is clearly defined by 47 specific genes, the default MicroCellClust result only identified 2 specific peaks. When looking closely at the expression of these peaks within the bicluster, one observes they contain 10% of negative values, and therefore barely make the cut with $\mu = 10\%$. This contrasts with the 8 genes expressed in all selected cells, and the additional 20 genes with less than 5% of negative values. This higher proportion of negative values for the scATAC-seq modality could be explained by multiple factors, for instance a higher dropout rate during sequencing, and/or the fact that multiple enhancers have overlapping functions (i.e. target the same genes) to provide robust gene expression. This motivates to split the optimization problem by modality (equations (8.2) and (8.3)) and relax μ_{ATAC} compared to μ_{RNA} .

Figure 8.3 shows the result of MicroCellClust with $\mu_{\text{ATAC}} = 20\% = 2 \cdot \mu_{\text{RNA}}$. The relaxed constraint leads to the selection of 13 peaks. This result can now serve as basis to investigate the relationships between gene expression and enhancers in this rare plasmacytoid DC subpopulation. Interestingly, looking at the expression of genes and peaks in the out-of-cluster cells can also give some clues. For instance, some genes (such as, among others, *AFF3*, *NIBAN3*, and *COBLL1*) are also highly and differentially expressed in the memory and naive B cells. Such a specific pattern can also be seen for a few peaks, which might indicate some kind of relationship between them, which could be further analyzed biologically.

Here, only the influence of the parameters κ and μ has been considered. Yet, other parameters could be taken into account to finetune the result. For instance,



Figure 8.2: Result of MicroCellClust on the 10k PBMCs multiome data. Only the 2000 cells with highest objective value (out of the 10k ones) are displayed. MicroCellClust identifies a rare subpopulation of plasmacytoid dendritic cells. It is characterized by 47 genes, but only 2 peaks (bottom rows) are selected.

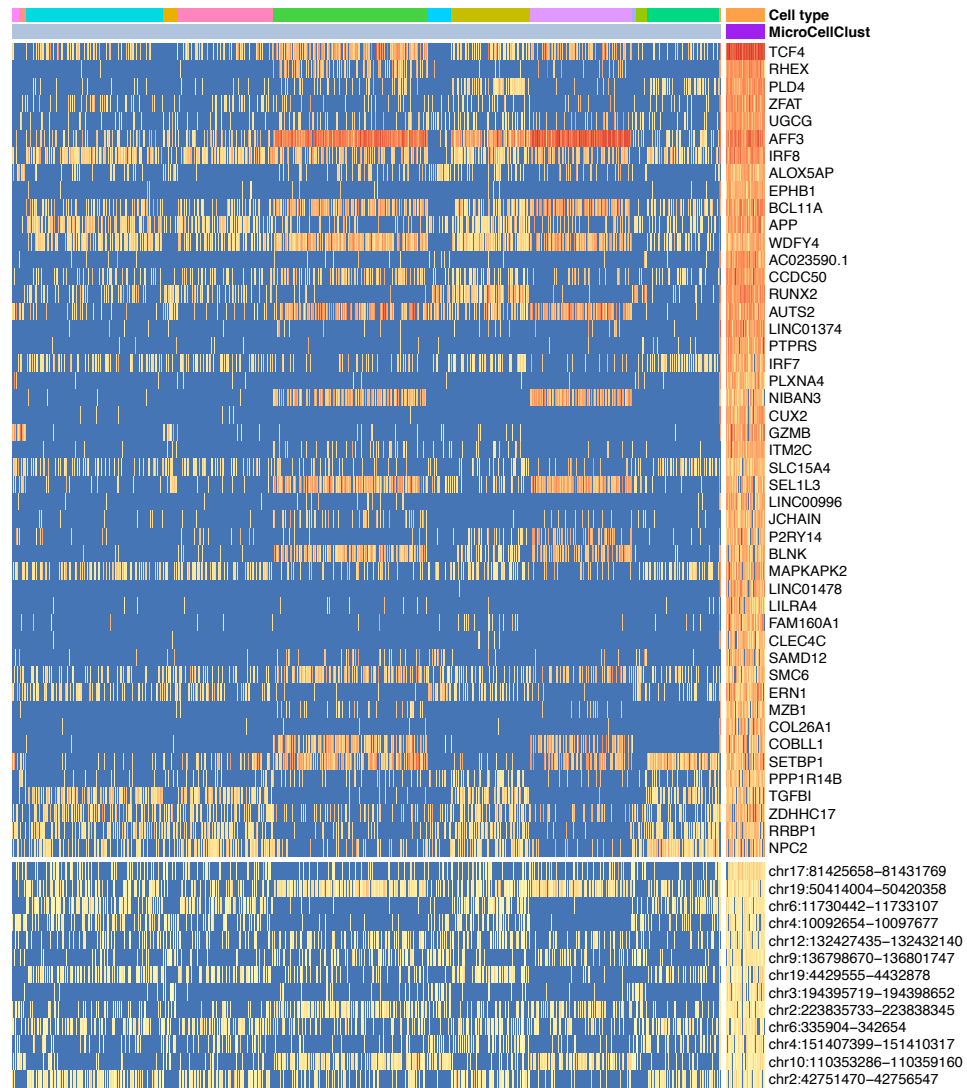


Figure 8.3: Result of MicroCellClust on the 10k PBMCs multiome data with $\mu_{\text{ATAC}} = 20\% = 2 \cdot \mu_{\text{RNA}}$. Only the 2000 cells with highest objective value are displayed. The same subpopulation is identified, and 11 additional peaks are now selected.

the initial filtering of features (by default keeping only those expressed in at most 25% of the cells) could be applied differently for the genes than for the ATAC peaks. Similarly, one observes the expression levels of scATAC-seq are in general lower than that of scRNA-seq. One might therefore consider different normalizations of the data for each modality. Yet, such a shift in expression does not constitute a problem for MicroCellClust (as was also observed in section 7.3.4), and as such constitutes one of its advantages to analyze multi-omic data.

8.3 Discussion

In summary, the methods developed during this thesis can easily be applied to multiomic data, in the case several modalities have been sequenced in the same single-cells. Using MicroCellClust on this data allows to link patterns of expression between several modalities in an unsupervised and multivariate way, by jointly identifying them with a (rare) subpopulation of cells. The analysis can be refined by specifying separate parameter values for each modality, without additional cost in terms of solving the optimization problem. Other aspects could also be taken into account, such as specific normalizations of the data for each modality. Yet, the stability of MicroCellClust with regards to shifts of expression (whether due to batch effects in case of horizontal integration, or due to different omic types in case of vertical integration) makes it particularly appropriate to analyze multi-modal data.

This approach has here been applied on multimodal scRNA-seq and scATAC-seq data, aiming at linking regions of open chromatin, which could correspond to enhancers, with gene expression. The use of a fully automated and unsupervised approach, such as MicroCellClust makes sense as enhancers can be located far away from the genes they regulate. The experiment conducted on PBMCs indeed identifies regions of open chromatin that are jointly expressed with a set of genes characteristic of plasmacytoid dendritic cells. Beyond the scope of this thesis, it would be worth further investigating the relationships between these identified features, and apply the same approach on other multimodal datasets.

Conclusions and perspectives

9

Single-cell data is a cornerstone of today's medical research. Gene expression data such as scRNA-seq allows to get a precise view on the composition of complex cell tissues. In particular, the presence of *rare cell types*, representing a small fraction of the population, has proven difficult to detect, yet critical to gain a good understanding of different pathologies. This necessitates the development of dedicated methods beyond generic (bi)clustering.

This thesis explores the use of a *multivariate* criterion to search for rare subpopulations of cells, and corresponding specific genes, in single-cell data. Unlike univariate approaches that consider individual genes in isolation, the proposed approaches evaluate the expression profiles of multiple genes simultaneously to identify rare patterns of gene expression. Specifically, they use a *global sum* criterion, which computes the utility of a gene/cell bicluster as a whole by summing the corresponding expression values, rather than relying on pairwise comparisons of values (as is often the case in traditional biclustering). This choice proves particularly appropriate for scRNA-seq data, which is subject to high technical and biological variability. Indeed, variations of expression levels, for instance due to transcriptional bursting or dropouts, will have little influence as long as the value of the bicluster as a whole remains high. Similarly, this criterion is robust with respect to batch effects when multiple biological samples are aggregated.

As such, the global sum criterion alone, as used in the max-sum submatrix problem (Branders, 2021), is not suited to identify *highly specific* subpopulations. Indeed, it tends to identify very large and generic biclusters when applied on scRNA-seq data. Applying a high threshold on the expression values to be considered significant could in principle lead to smaller biclusters, but at the cost of loosing many genes which are detected at lower transcript levels. Moreover, it ignores how the selected genes are expressed outside the bicluster, and is thus not suited to search for differentially expressed genes. The MicroCellClust approach

presented in chapter 3 therefore refines the optimization problem to search for genes that are highly and consistently expressed across the selected cells, and lowly expressed in the remaining “out-of-cluster” cells. This approach allows to effectively detect rare patterns of expression within single-cell data.

Moreover, MicroCellClust turns out to be especially effective when rare cell types are only represented by a *handful of individuals*. Indeed, the performance of typical alternatives is shown to drop whenever the number of cells forming a rare subpopulation decreases. The performance of MicroCellClust, on the other hand, stays constant when varying the proportion of rare cells to be detected. This shows the benefit of using MicroCellClust on data which is of limited availability, for instances specific cell types isolated using FACS. Several case studies show MicroCellClust, thanks to its multivariate nature, is able to find specific patterns of expression in data that would otherwise appear as relatively homogeneous.

In larger scale scRNA-seq data, the task of identifying rare cell types is comparatively easier, as they are represented by more individuals. Yet, the scale of this data makes this task *computationally* challenging, especially for clustering approaches which often have a quadratic complexity. One approach has been to develop methods assigning *rareness scores* to individual cells. These procedures are computationally efficient, but at the cost of losing an automated detection of *groups* of cells representing a certain type. MicroCellClust 2 hybridizes with these methods to efficiently recover rare biclusters in large-scale single-cell data. The rationale behind this approach is that rareness scores are able to efficiently identify a zone of interest in the search space, inside which MicroCellClust can effectively group cells together based on specific gene expression patterns. This allows the identification of rare cells in a multivariate way in data containing up to hundreds of thousands of cells.

The large availability of single-cell data also means that *multiple biological samples* are frequently aggregated. Such data allows to search for rare patterns of expression that are also common across different donors or conditions. Such data however also comes with *batch effects*, due to technical artifacts during the sequencing pipeline and/or biological variability between individuals. This can prevent clustering methods from detecting expression differences between cell types, requiring to first *integrate* the data into one coherent sample. Yet, the global sum criterion proves insensitive to these batch effects. scCross adapts the MicroCellClust optimization problem to search for rare subpopulations of cells present in multiple samples which express the same specific genes. This explicit search across samples contrasts with the traditional two-step approach of using data in-

tegration (i.e. transforming the data into one sample) followed by clustering (thus ignoring the partition into multiple samples). Several case studies show scCross is able to group related cells together, even in the presence of strong batch effects (for instance due to the use of different sequencing technologies).

Perspectives

The methods developed during this thesis have been applied to multiple case studies on scRNA-seq data. They show MicroCellClust (2) and scCross are able to identify coherent rare cell subpopulations, which have been biologically interpreted. Yet, (variations of) these methods could also be applied to a broader array of settings.

As explored in chapter 8, these methods can be applied to *multiomic* data, in order to search for links between different kinds of features. A modified version of MicroCellClust has been applied on scRNA-seq and scATAC-seq data, as recent advancements enable the sequencing of these modalities in the same cells, thus allowing a direct comparison of gene expression and open chromatin regions. The reported case study highlights a set of genes and a set of peaks which are both highly expressed within a certain cell type. An analysis of the relationships between these two groups would be of interest, beyond the scope of this thesis. With the constant evolution of sequencing platforms, such analyses could be applied to other combinations of omics in the future. As such, the focus need not stay on rare cells specifically, as the global sum criterion could also be used to find other kind of patterns linking several modalities.

MicroCellClust and scCross could also easily be adapted to be used in a *supervised* setting, in order to compare gene expression between different treatments or conditions. Such a setting has been used in section 4.3.1, where cells from the tumor and healthy cells of the same patient have been combined into one dataset. The healthy cells serve there for *a posteriori* validation, as one observes the identified bicluster only contains tumor cells, and the marker genes are lowly expressed in the healthy cells. Yet, the optimization problems can be adapted to specifically discriminate between conditions. One could require selected cells to be part of the same condition (or even only consider the cells from one target condition as variables), and only compute the out-of-cluster penalty over the remaining conditions. This would allow the identification of (not necessarily rare) patterns of expression that are specific to one condition and not observed in others.

This thesis focused on biomarker identification to gain new insights in certain pathologies. The long-term goal is to identify sets of genes (or other biomarkers)

which have a specific function in the immune response to a pathology in order to develop new treatments using immunotherapy. Yet, the identified genes could also serve down the line in a *prognostic or diagnostic* perspective. One could adapt the optimization problem of MicroCellClust to specifically search for the expression of (a subset of) some target genes within a group of cells from the patient to be diagnosed. One would here typically consider the genes as variables in the reduced space (see section 5.1), and need to define parameters and thresholds to determine the presence or absence of a specific pattern.

Combinatorial optimization

Applying the global sum criterion to new situations might require changing the optimization problem to better fit the desired goal. As discussed in section 7.5, there are two possible approaches to do so: adding constraints or refining the objective.

The first one has the advantage of simplicity, as it is easy to add new constraints to an existing model. It also has the added benefit that extra constraints might eliminate more solutions, which improves the computational efficiency. Yet, this often requires to define hard thresholds, and one must take care that the additional constraints effectively reach the goal. In some cases, extra constraints only slightly alters the solutions: enough to strictly satisfy the constraints, but without fundamentally changing the identified subpopulations.

When extra constraints do not seem appropriate, one should instead opt to modify the objective function directly. This can for instance be done by “softening” an existing constraint, i.e. defining a metric that evaluates how well a solution satisfies the constraint (instead of a strict pass/fail threshold) and penalizing the objective value accordingly. This has the advantage that the new goal is directly included in the function to optimize, meaning that the search will naturally be drawn to high quality solutions. The drawback of such an approach is that these soft constraints can no longer be used to filter out solutions, and therefore tend to increase the computation time of solvers. Yet, this does not as such constitute an issue if it significantly increases the quality of the results (in terms of reaching the desired goal and/or biological interpretation). One should therefore carefully evaluate the costs and benefits to decide whether it is beneficial to expand the objective function.

From a computer science perspective, the methods developed during this thesis are formalized as *combinatorial optimization problems*. Solving such problems constitutes a field of itself within computer science and applied mathemat-

ics, and modifying the optimization problems to fit new applications might require to reevaluate which approach is more efficient. The MicroCellClust and scCross problems have been solved using adaptations of well-known approaches, i.e. beam search and simulated annealing, with problem-specific heuristics. This choice is motivated by the fact that they produce good quality approximations in a reasonable amount of time. Yet, there is no guarantee that these solvers will remain efficient when adding extra constraints and/or refining the objective function. This might require to modify these solvers, or even evaluate completely different approaches. It could also be of interest to evaluate whether such other approaches can improve the performance of MicroCellClust and scCross with respect to the existing solvers, both in terms of solution quality as of speed.

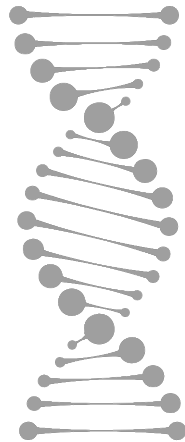
From a software perspective, the existing solvers could be improved to automatically identify multiple biclusters. Indeed, the current implementations only identifies one subpopulation at a time. The user needs to manually launch subsequent executions, until he feels he has extracted all the information. To automate this, one could imagine several beam searches operating in parallel: once one starts to converge to a particular zone of the search space, a new one could be launched excluding this zone, etc. Once multiple candidate biclusters are obtained, a local search could be performed on the combination of these solutions to try to solve the generalized K -biclusters optimization problem (4.5)–(4.7). Yet, such an automated approach might not be straightforward, as sensible values of the κ and μ parameters for one bicluster might not be the best choice for another. A carefully crafted automated procedure will be required to produce an optimal result for the end user.

Other perspectives

This thesis highlights the benefits of using multivariate approaches to identify biomarkers in single-cell data. The choice was made to rely on the global sum criterion, as it is well suited to the high variability of single-cell data. Yet, other approaches could be investigated, such as graph-based ones. Cells would typically be connected in a network based on their gene expression, and community detection algorithms could be applied to identify cell types. In particular, the identification of rare cells could be related to the task of *anomaly detection* in graphs (Akoglu et al., 2015).

Even though this thesis focuses on biological applications, the methods developed in its context are *generic* biclustering methods. They can therefore easily be used on other data sources where one wishes to identify *rare patterns*. As an ex-

ample, one could think of customer data indicating which programs are watched by whom on a streaming platform. Indeed, one could probably find many biclusters in such data, probably corresponding to different genres. Some would be very large, corresponding to blockbuster films and series, while others might be rare and corresponding to specific *niche* genres. It might be of interest to identify these programs, and the customers who frequently watch them, in order to perform targeted advertisement.



Bibliography

- Abraham, V., Taylor, D. L., and Haskins, J. (2004). High content screening applied to large-scale cell biology. *Trends in biotechnology*, 22(1):15–22.
- Adan, A., Alizada, G., Kiraz, Y., Baran, Y., and Nalbant, A. (2017). Flow cytometry: basic principles and applications. *Critical reviews in biotechnology*, 37(2):163–176.
- Akoglu, L., Tong, H., and Koutra, D. (2015). Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29:626–688.
- Argelaguet, R., Cuomo, A., Stegle, O., and Marioni, J. (2021). Computational principles and challenges in single-cell data integration. *Nature biotechnology*, 39(10):1202–1215.
- Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.
- Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J. M., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al. (2000). Gene Ontology: tool for the unification of biology. *Nature genetics*, 25:25–29.
- Bandura, D., Baranov, V., Ornatsky, O., Antonov, A., Kinach, R., Lou, X., Pavlov, S., Vorobiev, S., Dick, J., and Tanner, S. (2009). Mass cytometry: technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry. *Analytical chemistry*, 81(16):6813–6822.
- Baron, M., Veres, A., Wolock, S., Faust, A., Gaujoux, R., Vetere, A., Ryu, J., Wagner, B., Shen-Orr, S., Klein, A., et al. (2016). A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360.
- Bellman, R. (1957). *Dynamic Programming*. Rand Corporation research study. Princeton University Press.

- Bentley, J. (1984). Programming pearls: algorithm design techniques. *Communications of the ACM*, 27(9):865–873.
- Bergmann, S., Ihmels, J., and Barkai, N. (2003). Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical review E*, 67(3):031902.
- Bernstein, B. E., Meissner, A., and Lander, E. S. (2007). The mammalian epigenome. *Cell*, 128(4):669–681.
- Bonner, W., Hulett, H., Sweet, R., and Herzenberg, L. (1972). Fluorescence activated cell sorting. *Review of Scientific Instruments*, 43(3):404–409.
- Branders, V. (2021). *Finding submatrices of maximal sum: applications to the analysis of gene expression data*. PhD thesis, UCLouvain.
- Branders, V., Schaus, P., and Dupont, P. (2018). Combinatorial optimization algorithms to mine a sub-matrix of maximal sum. In Appice, A., Loglisci, C., Manco, G., Masciari, E., and Ras, Z., editors, *New Frontiers in Mining Complex Patterns*, pages 65–79. Springer International Publishing, Cham.
- Branders, V., Schaus, P., and Dupont, P. (2019). Identifying gene-specific subgroups: an alternative to biclustering. *BMC Bioinformatics*, 20:625.
- Brennecke, P., Anders, S., Kim, J. K., Kołodziejczyk, A., Zhang, X., Proserpio, V., Baying, B., Benes, V., Teichmann, S., Marioni, J., et al. (2013). Accounting for technical noise in single-cell RNA-seq experiments. *Nature methods*, 10(11):1093–1095.
- Bricard, O. (2019). Index hopping correction for scRNA-seq. <https://rdr.io/github/obricard/cleanhop/>.
- Büttner, M., Miao, Z., Wolf, F. A., Teichmann, S., and Theis, F. (2019). A test metric for assessing single-cell RNA-seq batch correction. *Nature methods*, 16(1):43–49.
- Carter, B. and Zhao, K. (2021). The epigenetic basis of cellular heterogeneity. *Nature Reviews Genetics*, 22(4):235–250.
- Chen, X., Wu, F.-X., Chen, J., and Li, M. (2019). DoRC: Discovery of rare cells from ultra-large scRNA-seq data. In *2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM '19*, pages 111–116.

- Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Ismb*, volume 8, pages 93–103.
- Choudhary, S. and Satija, R. (2022). Comparison and evaluation of statistical error models for scRNA-seq. *Genome biology*, 23(1):27.
- Cox, T. and Cox, M. (2000). *Multidimensional scaling*. CRC press, Boca Raton, 2nd edition.
- Crescenzi, P., Kann, V., and Halldórsson, M. (1995). A compendium of NP optimization problems.
- Dantzig, G. (1957). Discrete-variable extremum problems. *Operations research*, 5(2):266–288.
- Dawande, M., Keskinocak, P., and Tayur, S. (1996). On the biclique problem in bipartite graphs.
- Dong, R. and Yuan, G.-C. (2020). GiniClust3: a fast and memory-efficient tool for rare cell type identification. *BMC bioinformatics*, 21:158.
- Dubuisson, E. (2024). Rareness quantification of groups of cells and application to large-scale single-cell (bi)clustering. Master’s thesis, École polytechnique de Louvain, UCLouvain.
- Eckenrode, K., Righelli, D., Ramos, M., Argelaguet, R., Vanderaa, C., Geistlinger, L., Culhane, A., Gatto, L., Carey, V., Morgan, M., et al. (2023). Curated single cell multimodal landmark datasets for R/Bioconductor. *PLOS Computational Biology*, 19(8):e1011324.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Second International Conference on Knowledge Discovery and Data Mining*, KDD-96, pages 226–231.
- Franklin, C., Livingstone, E., Roesch, A., Schilling, B., and Schadendorf, D. (2017). Immunotherapy in melanoma: Recent advances and future directions. *European Journal of Surgical Oncology*, 43(3):604–611.
- Fukaya, T., Lim, B., and Levine, M. (2016). Enhancer control of transcriptional bursting. *Cell*, 166(2):358–368.

- Gawad, C., Koh, W., and Quake, S. (2016). Single-cell genome sequencing: current state of the science. *Nature Reviews Genetics*, 17(3):175–188.
- Gerniers, A., Bricard, O., and Dupont, P. (2021). MicroCellClust: mining rare and highly specific subpopulations from single-cell expression data. *Bioinformatics*, 37(19):3220–3227.
- Gerniers, A. and Dupont, P. (2022). MicroCellClust 2: a hybrid approach for multi-variate rare cell mining in large-scale single-cell data. In *2022 IEEE International Conference on Bioinformatics and Biomedicine, BIBM '22*, pages 148–153.
- Gerniers, A., Nijssen, S., and Dupont, P. (2024). scCross: efficient search for rare subpopulations across multiple single-cell samples. *Bioinformatics*, 40(6):btac371.
- Giesen, C., Wang, H., Schapiro, D., Zivanovic, N., Jacobs, A., Hattendorf, B., Schüffler, P., Grolimund, D., Buhmann, J., Brandt, S., et al. (2014). Highly multiplexed imaging of tumor tissues with subcellular resolution by mass cytometry. *Nature methods*, 11(4):417–422.
- Gini, C. (1912). Variabilità e mutabilità. In Pizetti, E. and Salvemini, T., editors, *Memorie di metodologica statistica*. Libreria Eredi Virgilio Veschi, Roma.
- Gionis, A., Indyk, P., Motwani, R., et al. (1999). Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, pages 518–529.
- Granville, V., Krivanek, M., and Rasson, J.-P. (1994). Simulated annealing: a proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):652–656.
- Grün, D., Kester, L., and Van Oudenaarden, A. (2014a). Validation of noise models for single-cell transcriptomics. *Nature methods*, 11(6):637–640.
- Grün, D., Kester, L., and Van Oudenaarden, A. (2014b). Validation of noise models for single-cell transcriptomics. *Nature methods*, 11(6):637–640.
- Grün, D., Muraro, M., Boisset, J.-C., Wiebrands, K., Lyubimova, A., Dharmadhikari, G., van den Born, M., Van Es, J., Jansen, E., Clevers, H., et al. (2016). De novo prediction of stem cell identity using single-cell transcriptome data. *Cell stem cell*, 19(2):266–277.

- Grün, D., Lyubimova, A., Kester, L., Wiebrands, K., Basak, O., Sasaki, N., Clevers, H., and van Oudenaarden, A. (2015). Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, 525:251–255.
- Guo, T., Chen, Y., Shi, M., Li, X., and Zhang, M. (2022). Integration of single cell data by disentangled representation learning. *Nucleic Acids Research*, 50(2):e8–e8.
- Haghverdi, L., Lun, A., Morgan, M., and Marioni, J. (2018). Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–427.
- Hamer, V. (2022). *Estimation and optimization of feature selection stability with applications to omics data*. PhD thesis, UCLouvain.
- Hariri, S. and Kind, M. C. (2018). Batch and online anomaly detection for scientific applications in a kubernetes environment. In *Proceedings of the 9th Workshop on Scientific Cloud Computing*, ScienceCloud ’18.
- Hashimshony, T., Senderovich, N., Avital, G., Klochender, A., De Leeuw, Y., Anavy, L., Gennert, D., Li, S., Livak, K., Rozenblatt-Rosen, O., et al. (2016). CEL-Seq2: sensitive highly-multiplexed single-cell RNA-seq. *Genome biology*, 17(1):1–7.
- Helleputte, T. (2010). *Inductive biases for stable feature selection in high dimensional spaces : applications to gene profiling and diagnosis from DNA microarrays*. PhD thesis, UCLouvain.
- Herman, J., Sagar, and Grün, D. (2018). FateID infers cell fate bias in multipotent progenitors from single-cell RNA-seq data. *Nature methods*, 15(5):379–386.
- Hie, B., Bryson, B., and Berger, B. (2019). Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nature biotechnology*, 37(6):685–691.
- Islam, S., Zeisel, A., Joost, S., La Manno, G., Zajac, P., Kasper, M., Lönnerberg, P., and Linnarsson, S. (2014). Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature methods*, 11(2):163–166.
- Jiang, L., Chen, H., Pinello, L., and Yuan, G.-C. (2016). GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome Biology*, 17:144.

- Jindal, A., Gupta, P., Jayadeva, and Sengupta, D. (2018). Discovery of rare cells from voluminous single cell expression data. *Nature communications*, 9:4719.
- Jordan, C., Guzman, M., and Noble, M. (2006). Cancer stem cells. *New England Journal of Medicine*, 355(12):1253–1261.
- Kaufman, L. (1990). Partitioning around medoids (program pam). *Finding groups in data*, 344:68–125.
- Kelly, R. (2020). Single-cell proteomics: progress and prospects. *Molecular & Cellular Proteomics*, 19(11):1739–1748.
- Kelsey, G., Stegle, O., and Reik, W. (2017). Single-cell epigenomics: Recording the past and predicting the future. *Science*, 358(6359):69–75.
- Kiselev, V., Andrews, T., and Hemberg, M. (2019). Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282.
- Kiselev, V., Kirschner, K., Schaub, M., Andrews, T., Yiu, A., Chandra, T., Natarajan, K., Reik, W., Barahona, M., Green, A., et al. (2017). SC3: consensus clustering of single-cell RNA-seq data. *Nature methods*, 14(5):483–486.
- Kivioja, T., Vähärautio, A., Karlsson, K., Bonke, M., Enge, M., Linnarsson, S., and Taipale, J. (2012). Counting absolute numbers of molecules using unique molecular identifiers. *Nature methods*, 9(1):72–74.
- Klein, A., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D., and Kirschner, M. (2015a). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201.
- Klein, A., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D., and Kirschner, M. (2015b). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201.
- Kluger, Y., Basri, R., Chang, J., and Gerstein, M. (2003). Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, 13(4):703–716.
- Kolodziejczyk, A., Kim, J., Svensson, V., Marioni, J., and Teichmann, S. (2015). The technology and biology of single-cell RNA sequencing. *Molecular cell*, 58(4):610–620.

- Kvon, E., Waymack, R., Gad, M., and Wunderlich, Z. (2021). Enhancer redundancy in development and disease. *Nature Reviews Genetics*, 22(5):324–336.
- Land, A. and Doig, A. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 18(3):497–520.
- Langfelder, P. and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. *BMC bioinformatics*, 9:1–13.
- Lazzeroni, L. and Owen, A. (2002). Plaid models for gene expression data. *Statistica sinica*, pages 61–86.
- Le Van, T., Van Leeuwen, M., Carolina Fierro, A., De Maeyer, D., Van den Eynden, J., Verbeke, L., De Raedt, L., Marchal, K., and Nijssen, S. (2016). Simultaneous discovery of cancer subtypes and subtype features by molecular data integration. *Bioinformatics*, 32(17):i445–i454.
- Le Van, T., Van Leeuwen, M., Nijssen, S., Fierro, A., Marchal, K., and De Raedt, L. (2014). Ranked tiling. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014. Proceedings, Part II*, pages 98–113. Springer.
- Lei, T., Chen, R., Zhang, S., and Chen, Y. (2023). Self-supervised deep clustering of single-cell RNA-seq data to hierarchically detect rare cell populations. *Briefings in bioinformatics*, 24(6):bbad335.
- Li, G., Ma, Q., Tang, H., Paterson, A., and Xu, Y. (2009). QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic acids research*, 37(15):e101–e101.
- Luecken, M., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Müller, M., Strobl, D., Zappia, L., Dugas, M., Colomé-Tatché, M., et al. (2022). Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50.
- Luecken, M. and Theis, F. (2019). Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular systems biology*, 15(6):e8746.
- 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. *F1000Research*, 5.

- Lv, Q., Josephson, W., Wang, Z., Charikar, M., and Li, K. (2006). Ferret: A toolkit for content-based similarity search of feature-rich data. In *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, EuroSys '06, page 317–330.
- Macfarlan, T., Gifford, W., Driscoll, S., Lettieri, K., Rowe, H., Bonanomi, D., Firth, A., Singer, O., Trono, D., and Pfaff, S. (2012). Embryonic stem cell potency fluctuates with endogenous retrovirus activity. *Nature*, 487:57–63.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.
- Madeira, S. and Oliveira, A. (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM transactions on computational biology and bioinformatics*, 1(1):24–45.
- Martello, S. and Toth, P. (1987). Algorithms for knapsack problems. *North-Holland Mathematics Studies*, 132:213–257.
- Mellman, I., Coukos, G., and Dranoff, G. (2011). Cancer immunotherapy comes of age. *Nature*, 480(7378):480–489.
- Murali, T. and Kasif, S. (2002). Extracting conserved gene expression motifs from gene expression data. In *Biocomputing 2003*, pages 77–88. World Scientific.
- Muraro, M., Dharmadhikari, G., Grün, D., Groen, N., Dielen, T., Jansen, E., Van Gurp, L., Engelse, M., Carlotti, F., De Koning, E., et al. (2016). A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4):385–394.
- Osterwalder, M., Barozzi, I., Tissières, V., Fukuda-Yuzawa, Y., Mannion, B., Afzal, S., Lee, E., Zhu, Y., Plajzer-Frick, I., Pickle, C., et al. (2018). Enhancer redundancy provides phenotypic robustness in mammalian development. *Nature*, 554(7691):239–243.
- Panigrahi, A. and O’Malley, B. (2021). Mechanisms of enhancer action: the known and the unknown. *Genome biology*, 22(1):108.
- Picelli, S., Faridani, O., Björklund, Å., Winberg, G., Sagasser, S., and Sandberg, R. (2014). Full-length RNA-seq from single cells using Smart-seq2. *Nature Protocols*, 9:171–181.

- Pierson, E. and Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology*, 16(1):1–10.
- Plitas, G., Konopacki, C., Wu, K., Bos, P., Morrow, M., Putintseva, E., Chudakov, D., and Rudensky, A. (2016). Regulatory T cells exhibit distinct features in human breast cancer. *Immunity*, 45(5):1122–1134.
- Pontes, B., Giráldez, R., and Aguilar-Ruiz, J. S. (2015). Biclustering on expression data: A review. *Journal of biomedical informatics*, 57:163–180.
- Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature communications*, 9(1):284.
- Sadeghi, M. and Armanfard, N. (2023). Deep clustering with self-supervision using pairwise data similarities. *Authorea Preprints*.
- Saelens, W., Cannoodt, R., and Saeys, Y. (2018). A comprehensive evaluation of module detection methods for gene expression data. *Nature communications*, 9(1):1090.
- Satija, R., Farrell, J., Gennert, D., Schier, A., and Regev, A. (2015). Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502.
- Schoenfelder, S. and Fraser, P. (2019). Long-range enhancer–promoter contacts in gene expression control. *Nature Reviews Genetics*, 20(8):437–455.
- Stegle, O., Teichmann, S., and Marioni, J. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145.
- Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W., Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.
- Susto, G. A., Beghi, A., and McLoone, S. (2017). Anomaly detection through on-line isolation forest: An application to plasma etching. In *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference, ASMC '17*, pages 89–94.
- Tabula Muris Consortium (2018). Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, 562(7727):367–372.

- The Gene Ontology Consortium (2019). The Gene Ontology resource: 20 years and still GOing strong. *Nucleic Acids Research*, 47(D1):D330–D338.
- Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.
- Todorov, H. and Saeys, Y. (2019). Computational approaches for high-throughput single-cell data analysis. *FEBS journal*, 286(8):1451–1467.
- Tsoucas, D. and Yuan, G.-C. (2018). GiniClust2: a cluster-aware, weighted ensemble clustering method for cell-type detection. *Genome biology*, 19:1–13.
- Tunnacliffe, E. and Chubb, J. R. (2020). What is a transcriptional burst? *Trends in Genetics*, 36(4):288–297.
- Vallejos, C., Risso, D., Scialdone, A., Dudoit, S., and Marioni, J. (2017). Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nature methods*, 14(6):565–571.
- Van Dam, T., Kennedy, J., van der Lee, R., De Vrieze, E., Wunderlich, K., Rix, S., Dougherty, G., Lambacher, N., Li, C., Jensen, V., et al. (2019). CiliaCarta: An integrated and validated compendium of ciliary genes. *PloS one*, 14(5):e0216705.
- van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(86):2579–2605.
- Van Laarhoven, P. and Aarts, E. (1987). *Simulated annealing: Theory and applications*. Springer, Dordrecht.
- Wagner, A., Regev, A., and Yosef, N. (2016). Revealing the vectors of cellular identity with single-cell genomics. *Nature biotechnology*, 34(11):1145–1160.
- Wang, X., He, Y., Zhang, Q., Ren, X., and Zhang, Z. (2021). Direct comparative analyses of 10X Genomics Chromium and Smart-seq2. *Genomics, Proteomics & Bioinformatics*, 19(2):253–266.
- Wang, Z., Dong, W., Josephson, W., Lv, Q., Charikar, M., and Li, K. (2007). Sizing sketches: A rank-based analysis for similarity search. In *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '07, page 157–168.

- Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.
- Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1–3):37–52.
- Xie, J., Ma, A., Fennell, A., Ma, Q., and Zhao, J. (2018). It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data. *Briefings in Bioinformatics*, 20(4):1450–1465.
- Xie, K., Huang, Y., Zeng, F., Liu, Z., and Chen, T. (2020). scAIDE: clustering of large-scale single-cell RNA-seq data reveals putative and rare cell types. *NAR Genomics and Bioinformatics*, 2(4):lqaa082.
- Yang, P., Huang, H., and Liu, C. (2021). Feature selection revisited in the single-cell era. *Genome Biology*, 22:1–17.
- Yu, G., Wang, L.-G., Han, Y., and He, Q.-Y. (2012). clusterProfiler: an R package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology*, 16(5):284–287.
- Yu, M., Stott, S., Toner, M., Maheswaran, S., and Haber, D. (2011). Circulating tumor cells: approaches to isolation and characterization. *Journal of Cell Biology*, 192(3):373–382.
- Zeisel, A., Hochgerner, H., Lönnerberg, P., Johnsson, A., Memic, F., et al. (2018). Molecular architecture of the mouse nervous system. *Cell*, 174(4):999–1014.e22.
- Zheng, G., Terry, J., Belgrader, P., Ryvkin, P., Bent, Z., Wilson, R., Ziraldo, S., Wheeler, T., McDermott, G., Zhu, J., et al. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8:14049.
- Ziegenhain, C., Vieth, B., Parekh, S., Reinius, B., Guillaumet-Adkins, A., Smets, M., Leonhardt, H., Heyn, H., Hellmann, I., and Enard, W. (2017). Comparative analysis of single-cell RNA sequencing methods. *Molecular cell*, 65(4):631–643.
- Ziyani, C., Delaneau, O., and Ribeiro, D. (2023). Multimodal single cell analysis infers widespread enhancer co-activity in a lymphoblastoid cell line. *Communications Biology*, 6(1):563.

Appendix

Data description and availability



This appendix contains details and links to the different datasets used throughout this thesis.

A.1 Breast cancer data

The data used in sections 2.3, 4.2.1, 4.3.1 and 4.3.2 of this dissertation is composed of T lymphocytes extracted from a human breast tumor and a healthy breast tissue. These cells are sorted in a 384-well plate by flow cytometry according to three distinct phenotypes: CD8 T cells (CD3+ CD8+), CD4 T cells (CD3+ CD4+), and activated regulatory T cells (CD3+ CD4+ CD25+ GARP+). A total of 96 cells from each phenotype and tissue origin combination are processed following the Smart-seq2 protocol (Picelli et al., 2014) to generate scRNA-seq data on the Illumina HiSeq4000 platform. 72 GARP+ tumorous cells of another donor are processed in the same way. A summary is given in table A.1. The `cleanhop` algorithm (Bricard, 2019) is used to remove potential misassigned reads due to index hopping. It subtracts to the number of reads, for each cell and for each gene, a percentage (0.5% by default) of the sum of the reads associated with the gene among all the cells sharing the same i7 index (column), and the sum of all the reads associated with the gene among all the cells sharing the same i5 index (row). Finally, read counts are normalized by counts per million.

This data is available on Zenodo, at:

<https://doi.org/10.5281/zenodo.4580332>

Sections 7.1 and 7.3.1 use an extension of this breast cancer data, specifically GARP+ Tregs from the tumor of 5 different patients (KOBE: 96 cells; JORD: 72 cells; FETU: 96 cells; KYLO: 192 cells; NOTT: 49 cells). This data has been processed in the same way, and is available on Zenodo, at:

<https://doi.org/10.5281/zenodo.10471064>

Donor	Tissue_origin	FACS_phenotype	Nb. cells	2.3	4.2.1	4.3.1	4.3.2
KOBÉ	ExtraTumor	CD3+_CD4+	89				
	Tumor	CD3+_CD4+	93				
		CD3+_CD8+	83				
		CD3+_CD4+_CD25+_GARP+	96				
JORD			72				

Table A.1: Annotations of the breast cancer data. The 4th column indicates the number of cells that were processed (after quality control) for each data subset; the 3 last columns indicate in which experiments they are used (number of the section in the main manuscript).

A.2 Jurkat and 293T mixture

50%:50% *Jurkat:293T Cell Mixture* (Zheng et al., 2017). This data has been downloaded from the 10X Genomics website, <https://www.10xgenomics.com/resources/datasets>, and is also available on the Sequence Read Archive, with accession code SRX1723923.

The two cell lines were mixed *in vitro*, and sequenced on the Illumina HiSeq2500 Rapid Run V2 platform with $\approx 33\,000$ reads per cell. The resulting data was processed by the Cell Ranger 1.1 pipeline.

A.3 Artificial data for rareness versus distinctiveness

Several matrices of $1000 \text{ cells} \times 1000 \text{ genes}$ are generated, with default values of -1 . For each gene i , a positive expression range $[v_i^{\min}, v_i^{\max}]$ is defined, with randomly chosen $v_i^{\max} \in [1, 4]$ and $v_i^{\min} \in [0, v_i^{\max}]$.

Out of the 1000 genes, 970 are filled with random values by selecting, for each one of them, an expression probability p_i chosen from $U(0.01, 0.50)$. Each value in the expression vector of gene i will have a probability p_i to be assigned a positive value randomly chosen in $U(v_i^{\min}, v_i^{\max})$.

The 30 remaining genes will be attributed to the target rare subpopulation of cells, which is characterized by $n_{\text{rare}} \in \{5, 20, 50\}$ cells. For each gene i , a dropout probability d_i is chosen from $U(0.10, 0.20)$. Each rare cell is assigned a positive expression value from $U(v_i^{\min}, v_i^{\max})$ with a probability of $1 - d_i$.

The distinctiveness of the target subpopulation is defined by an out-of-cluster expression range $[p_{\min}^{\text{oc}}, p_{\max}^{\text{oc}}]$. For each of the 30 marker genes, a probability p_i^{oc}

is chosen from $U(p_{\min}^{\text{oc}}, p_{\max}^{\text{oc}})$, and the out-of-cluster cells are assigned a positive value from $U(v_i^{\min}, v_i^{\max})$ according to this probability.

A.4 Mouse embryonic stem cells

Droplet barcoding for single cell transcriptomics applied to embryonic stem cells (Klein et al., 2015a). This data has been downloaded from the Gene Expression Omnibus, with accession code GSE65525.

A.5 68k PBMCs

Fresh 68k PBMCs (Donor A) (Zheng et al., 2017). This data has been downloaded from the 10X Genomics website, <https://www.10xgenomics.com/resources/datasets>, and is also available on the Sequence Read Archive, with accession code SRX1723926.

The PBMCs were retrieved from a healthy donor, and sequenced on the Illumina NextSeq 500 High Output platform with $\approx 20\,000$ reads per cell. The resulting data was processed by the Cell Ranger 1.1 pipeline.

A.6 160k mouse brain cells

Atlas of the Adolescent Mouse Brain (Zeisel et al., 2018). This data has been downloaded from <http://mousebrain.org/adolescent/>, and is also available on the Sequence Read Archive, with accession code SRP135960.

A.7 Non-small cell lung cancer dissociated tumor cells

40k Mixture of NSCLC DTCs from 7 donors (v3.1 Chemistry). This data has been downloaded from the 10X Genomics website, <https://www.10xgenomics.com/resources/datasets>

Each donor was CellPlexed and pooled at equal proportions. Gene expression and CellPlex libraries were generated using the Chromium X. Libraries were sequenced on an Illumina NovaSeq 6000 to a read depth of approximately 50 000 mean reads per cell for gene expression and 35 000 mean reads per cell for CellPlex.

A.8 8 pancreas samples

This dataset aggregates 8 human pancreas samples from two different studies:

- 4 samples from [Baron et al. \(2016\)](#) sequenced with inDrop protocol, downloaded from the Gene Expression Omnibus, with accession code GSE84133.
- 4 samples from [Muraro et al. \(2016\)](#) sequenced with CEL-seq2 protocol, downloaded from the Gene Expression Omnibus, with accession code GSE85241.

A.9 *Tabula muris*

Tabula Muris – Single-cell RNA-seq data from Smart-seq2 sequencing of FACS sorted cells (v2). This data has been downloaded from Figshare, at <https://doi.org/10.6084/m9.figshare.5829687.v7>

A.10 10X Multiome PBMCs

10k Human PBMCs. This data has been accessed through the SingleCell-MultiModal R package ([Eckenrode et al., 2023](#)), available on Bioconductor at <https://doi.org/doi:10.18129/B9.bioc.SingleCellMultiModal>