UCLouvain

epl

École polytechnique de Louvain

Rareness quantification of groups of cells and application to large-scale single-cell (bi)clustering

Author: **Elliott Dubuisson**
Supervisors: **Pierre Dupont, Siegfried Nijssen**
Readers: **Alexander Gerniers, Benoît Ronval**
Academic year 2023–2024
Master [120] in Computer Science

Abstract

Identifying rare subpopulations of cells is critical in the field of medicine. It enhances our understanding of diseases, enables more accurate diagnostics, and facilitates earlier detection of illnesses. In the past decade, single-cell RNA sequencing techniques (scRNA-seq) have emerged. They provide a measurement of the gene expression profile of individual cells, information which was previously only observable for a bulk sample of cells.

Researchers have used scRNA-seq data to develop algorithms that analyse the rareness of cells based on their respective gene expression compared to a given population of cells. *Finder of rare entities* (FiRE) is a method that assigns a rareness score to each cell by comparing its gene expression against the expression profile of the rest of the population. *MicroCellClust 2* (MCC2) is a beam search-based algorithm that returns a small subpopulation of cells that express highly specific genes. MCC2 uses the results of FiRE to prune the cells to consider, in order to make the algorithm more efficient.

In this thesis, we propose FiRE-n, an algorithm that expands the FiRE methodology by enabling it to assign a rareness score to a group of 1 to n cells. This score is based both on the rareness of the cells forming the group and on their relative homogeneity. The results show that FiRE-n can identify homogeneous subpopulations of rare cells correctly. Following the development of FiRE-n, we introduce MCC2$\star$, a version of the MCC2 algorithm that uses FiRE-n to prune the groups at each level of the beam search. MCC2$\star$ returns slightly better solutions than MCC2 with a runtime reduced by 20% to 25%. Finally, we propose a novel method that retrieves homogeneous groups made of rare cells. This method is called *retriever of critical clusters* (ReCC). Results show that ReCC returns rare and homogeneous clusters effectively.

Acknowledgements

First, I would like to thank my supervisors, Prof. Pierre Dupont and Prof. Siegfried Nijssen, for their time and valuable guidance. I am grateful to Prof. Dupont for proposing this topic and to Prof. Nijssen for agreeing to supervise my thesis during Prof. Dupont's absence.

Then, I would like to thank Alexander Gerniers for the time he spent assisting me during my whole research. It has been a pleasure to work and discuss with him about potential leads and solutions regarding problems I was facing. His help and insights have been fundamental to my work.

Finally, I would like to thank my parents for always being supportive throughout the whole process of my thesis. I am grateful to my dad for his comments and sound advice regarding my redaction and presentations.

Acknowledgement of AI Assistance

In the preparation of this master thesis, artificial intelligence tools have been employed to enhance the quality and clarity of the text. OpenAI's ChatGPT has been used for the reformulation of some sentences to improve coherence and readability. Additionally, Grammarly has been used to check and refine the grammar of the text.

Contents

Chapter 1

Introduction

1.1 Problem Statement and Context

In recent years, the emergence of Single-Cell RNA Sequencing (scRNA-seq) has revolutionised the ability to dissect the complexities of cellular heterogeneity with unprecedented detail [1]. Unlike traditional bulk RNA-seq techniques, which provide a global gene expression profile for a given population of cells, scRNA-seq enables the characterisation of gene expression patterns at the individual cell level. This groundbreaking technology has opened new frontiers in biological research, allowing scientists to explore the diversity of cell types.

The versatility of scRNA-seq is one of its main strengths, as it can be used across various biological disciplines [1] such as developmental biology [2], immunology, cancer biology [3], neurobiology, regenerative medicine, and others. For instance, the evolution of cancer can be detected by observing the accumulation of heterogeneity in a population of cells and scRNA-seq has offered valuable insights to researchers about minor treatment-resistant cell subpopulations within complex tumours. These insights can guide the selection of therapies depending on the tumour's characteristics, thereby enabling more precise treatment tailored to each individual patient [4].

In a given population of cells, rare cells typically refer to cells that express genes that are different from the ones expressed by the rest of the population. Identifying such rare subpopulations of cells can be crucial in detecting certain diseases. For instance, regulatory T (Tregs) cells who are usually located in lymphoid organs and barrier tissues have been discovered to be found in human cancer and contribute to the progression of cancer [5]. Detecting these subpopulations of Treg cells could help diagnose cancer at earlier stages and prevent its development. However,

classical clustering approaches usually fail to detect these rare subpopulations as they tend to form larger clusters rather than small ones [6].

With the objective of detecting such rare cells and following the recent development of scRNA-seq, researchers of the Indian Institute of Technology Delhi have developed Finder Of Rare entities (FiRE). FiRE is an algorithm that assigns, in just a few seconds, a rareness score to every cell in a population based on its individual gene expression profile [7]. As such, FiRE does not perform clustering of the data, yet these rareness scores can be helpful for bioinformaticians to focus their analysis on a fragment of cells within a large scRNA-seq dataset. These rareness scores are used by MicroCellClust (MCC) [8], a method that identifies rare subpopulations of cells using a multivariate approach, to scale up to large data.

1.2 Objective

The objective of this thesis encompasses two main goals. Firstly, it endeavours to expand the functionality of the FiRE method, allowing it to provide a rareness score for groups ranging from individual cells to those containing up to n cells. Secondly, it aims to refine the MCC algorithm by incorporating this expanded FiRE method in it, following a heuristic already used at some steps of the MCC algorithm. The intention is to enhance the runtime efficiency of MCC and improve its results in terms of objective value by leveraging the generalised FiRE method to narrow down its search space. The challenge is to manage to generalise the FiRE method to provide the same information as FiRE does while keeping its scalability property to improve the runtime of MCC.

1.3 Overview

This thesis presents a comprehensive exploration of scRNA-seq data analysis, focusing on the identification of rare cell populations. After this introduction, chapter 2 proposes a background review, explaining the functioning of scRNA-seq and delving into how the FiRE and MicroCellClust algorithms work. Detailed insights are provided into their methodologies, algorithms, and characteristics, laying the foundation for the subsequent analyses.

Following the background review, chapter 3 proposes a generalisation of the FiRE scoring method to groups of size n. First, the experimental setup is described, including the evaluation criteria that will be used to assess the performance of

the generalised algorithm. The concepts of rareness and group homogeneity are discussed along with their impact on the group score. A baseline method is defined, against which the results of the generalised FiRE algorithm, called FiRE-n, will be assessed. One last section discusses the results of the FiRE-n method.

Following the description and the analysis of the results of this generalised approach, the impact of integrating the FiRE-n method in the MicroCellClust algorithm on the performance and on the outcome is assessed in chapter 4.

Finally, chapter 5 develops an alternative method to identify critical groups of cells based on a clustering approach, instead of the classification approach used in FiRE. Instead of assigning scores to groups of cells, this method aims at retrieving the most interesting clusters based on the criteria described in chapter 3. The results of this method are also analysed and discussed.

The conclusion provides insights into the effectiveness and limitations of the various strategies deployed and by giving potential leads on other approaches, beyond the scope of this thesis, that could be pursued.

Chapter 2

Background

This chapter first introduces scRNA-seq data and then presents the two principal methods used in the following chapters, namely Finder of Rare Entities (FiRE) and MicroCellClust (MCC). FiRE assigns a rareness score to each cell of a dataset based on their gene expression. The results of the FiRE algorithm are used by MCC to improve its efficiency. MicroCellClust's purpose is to identify a group of rare cells as well as a corresponding subset of genes from scRNA-seq data. The identified group of cells can in turn be used for additional scientific research.

2.1 Single cell RNA sequencing

The algorithms detailed in section 2.2 and section 2.3 operate on scRNA-seq datasets. These datasets are typically structured as matrices where each row corresponds to a gene and each column corresponds to an individual cell. In this thesis, the set of cells and the set of genes of the scRNA-seq matrices are denoted as $\mathcal{C}$ and $\mathcal{G}$ respectively. The entries in the matrix represent gene expression levels, which quantify the activity of specific genes within individual cells [9]. The ability to measure gene expression at single-cell resolution allows for a detailed characterisation of cellular heterogeneity, providing insights into complex biological processes that would be obscured in bulk RNA sequencing analysis. Indeed, bulk RNA sequencing techniques allow to measure the expression level of certain genes over large populations of cells [10] but do not allow for individual cell analysis.

Obtaining these gene expression levels for individual cells involves several stages such as cell isolation, library preparation, sequencing and computational analysis [11]. First, the cells of the population under study are isolated from each other to ensure that they can be analysed in an independent manner without being influenced by neighbouring cells [12]. Then, the RNA of the individual cells is converted

into a library of complementary DNA (cDNA) [13] that will be sequenced to provide a gene expression profile for each individual cell [14]. Computational analysis is performed on this sequencing data to obtain a meaningful gene expression matrix.

The gene expression data resulting from these different operations is crucial to understand the functional state of each cell and can help infer cell types within a heterogeneous population. Dimensionality reduction techniques such as t-distributed stochastic neighbour embedding (t-SNE) [15] can help visualise and cluster similar cells together, further analysis of the marker genes expressed by each of these clusters allows biologists to annotate these clusters.

Algorithms such as FiRE and MicroCellClust make use of gene expression matrices to identify cells that express different genes from the rest of the population and that can thus be considered to be a specific cell type. These algorithms specifically focus on rare cell types, typically representing less than 5% of the data, which are often missed by regular clustering approaches. These algorithms are presented in the next sections.

2.2 Finder of Rare Entities

FiRE is a method implemented by Jindal,A et al. [7] whose purpose is to identify rare cells by assigning a rareness score to every cell in the dataset based on their gene expression. In contrast to other pre-existing algorithms such as RaceID [16] and GiniClust [17], FiRE excludes any clustering as part of the algorithm since it is known to be particularly inefficient when density varies between different data points. The lack of clustering allows FiRE to give a rareness score to each cell in just a few seconds whereas the previously mentioned methods take an unbearable amount of time to finish their clustering when the dataset exceeds a few tens of thousands of cells.

The FiRE algorithm is motivated by the fact that computing the rareness of a cell is like computing the density around that cell, but in reverse. FiRE leverages the Sketching technique [18], an influential approach for efficiently encoding a vast collection of data points within a reduced dimensional space. Sketching is a method developed to retrieve feature-rich data objects resembling a given query data object using compact data structures. This approach is characterised by the use of *sketches*, which are compact representations of the different data points. These sketches have two advantages: firstly, they are low-dimensional data structures, and secondly, the distance between two feature vectors can be calculated from the sketch alone. FiRE makes use of this sketching technique to derive which cells are

different from the rest of the population.

2.2.1 Algorithm

The FiRE method consists of three main steps: preprocessing, sketching and scoring.

Preprocessing

Before being executed, the FiRE algorithm requires some preprocessing of the scRNA-seq matrix given as input. The preprocessing consists in selecting the 1000 most fluctuating genes from the scRNA-seq dataset, keeping only the genes that are the most influential, to help detect rare cells from abundant ones. The selected genes are those that show the greatest differences in expression levels compared to other genes with similar average expression levels.

Sketching

This step combines sketching and hashing. The FiRE algorithm uses the sketching technique [18] to reduce the dimensionality of the cells. It derives a *sketch* for each cell from their vector of gene expression. This is done by randomly selecting M genes and for each of them, generating a random threshold ranging between the minimum and the maximum value observed in the dataset for that gene. Then a bit-vector is computed for each cell as the result of the application of the thresholds on the corresponding gene expression values of the cells. If for a given gene, the expression value of a cell is larger than the randomly generated threshold then the value of the bit-vector for that gene is 1 and the gene is considered as expressed by that cell, if not, the value is 0. These bit-vectors are called *sketches*.
For each of these sketches, a hashcode is computed by using the modulo hashing technique on the result of the dot product between the sketch and a randomly generated weight vector w. The idea behind this is that it is expected that two similar cells (i.e. of the same type) have very close, if not identical, sketches and thus would share their hashcode. These hashcodes can be seen as *buckets* in which similar cells are placed.
These operations are repeated L times, thus, assigning L buckets to each cell, one for each iteration l.

Scoring

Once the sketching and hashing stage has been completed, the final step consists in assigning a rareness score to each cell based on the density of the buckets they

have been placed in. Indeed, since it is expected that similar cells are placed in the same bucket during one iteration, cells that have frequently been placed in buckets that are heavily populated are more likely to be abundant.
To compute these rareness scores, L bucket density estimators p_{il} are computed for each cell, each of which denotes the probability of a randomly selected cell to be placed in the same bucket (i.e. have the same hash) as the cell i at iteration l. The density estimator of the bucket of a cell i coming from a population $\mathcal{C}$ at iteration l is computed as :

$$p_{il} = \frac{|\text{Bucket of cell } i|}{|\mathcal{C}|} \tag{2.1}$$

The second step consists in reducing the variance of the density estimators p_{il} to increase the precision of the scores. This is done by bringing together the L density estimates for each cell, yielding a FiRE score with the following formula :

$$\text{FiRE score}_i = -2 * \sum_{l=1}^{L} \log_e(p_{il}) \tag{2.2}$$

This formula slightly resembles Shannon's entropy formula [19]: the smaller the density estimator (i.e. the least crowded the bucket is), the higher the rareness score, thus assigning higher scores to cells that are not similar to the rest of the population.

Parameters

The algorithm takes several variables as input :

- L is the number of times the sketching is performed. The larger the value of L, the lower the variance of the density estimators will tend to be. The common use is to have L equal to 100.
- M is the number of genes randomly selected to perform the sketching. A low value for M means that a larger value for L will be needed to compensate for the lack of information while a too large value for M might make the scores too sensitive to noisy measurements. Therefore, it is crucial to choose an appropriate value for M for the rareness scores to have a good balance between accuracy and sensitivity. The authors of the paper determined 50 to be a reasonably good default value for M.
- H is the size of the hashing table. H should be a prime number that is sufficiently large to prevent hash collisions. It is advised to use a value of H at least 10 times greater than the number of cells in the dataset.

Pseudo-code

Algorithm 1 Adapted FiRE Algorithm

Input: $\mathbf{m} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$, a gene expression matrix
Input: **M**, the number of genes to randomly select for each sketching iteration
Input: **L**, the number of sketching iterations to perform
Input: **H**, the size of the hashing table
Output: **scores**, the FiRE scores of each of the cells $\in \mathcal{C}$
▷ *Performing the sketching of every cell L times* ◁
for $l = 1, \ldots, L$ **do**
    $genes \leftarrow set\ of\ M\ genes\ randomly\ selected\ from\ \mathcal{G}$
    $t \leftarrow vector\ of\ M\ thresholds\ randomly\ generated$
    $w \leftarrow vector\ of\ M\ weights\ randomly\ generated$
    for all $g \in genes$ **do**
        for all $i \in \mathcal{C}$ **do**
            if $m_{gi} \geq t[g]$ **then**
                $sketch[g] = 1$
            else
                $sketch[g] = 0$
        $bucket[l][i] = (sketch \cdot w)\ \%\ H$

▷ *Computing the FiRE score of every cell* ◁
for all $i \in \mathcal{C}$ **do**
    $scores[i] \leftarrow \sum_{l=0}^{\mathbf{L}} \log(\frac{|bucket[l][i]|}{|\mathcal{C}|})$
return *scores*

2.2.2 Rareness threshold

A thresholding approach is introduced by the authors of FiRE to interpret the FiRE scores and determine which cells should be considered rare and which are to be viewed as abundant. In their methodology, the rareness of a cell is determined

by comparing its rareness score to the distribution of the rest of the FiRE scores of the population. A cell is marked as *rare* by FiRE if :

$$FiRE\ score \geq q3 + 1.5 \times IQR \tag{2.3}$$

As denoted in (2.3), FiRE considers a cell to be rare if its score is greater or equal to the value of the third quartile of the distribution of FiRE scores across all cells added to 1.5 times the interquartile range, calculated as the difference between the third and the first quartile of the FiRE scores in the population. This threshold is called the *rareness threshold*.

2.2.3 Strengths and weaknesses of FiRE

The major strength of the FiRE algorithm resides in its speed and scalability. As previously mentioned, the already existing methods were inefficient when confronted with large datasets and would take a very long time to finish with datasets exceeding a few tens of thousands of cells, that is if they would even terminate. Thanks to the sketching technique, FiRE allows for rapidly obtaining an individual rareness score for each of the cells in the dataset, even if the number of cells in the dataset gets quite large. Where GiniClust and RaceID use clustering methods that run in $\mathcal{O}(|\mathcal{C}|^2)$, FiRE assigns a rareness score to every cell in $\mathcal{O}(|\mathcal{C}|)$.

Another benefit of FiRE that stands out is its robustness against noise. Experiments showed that the algorithm's results are significantly better when there are at least 20 differentially expressed genes (i.e. genes that differ between different types of cells) in the dataset. Therefore, if the gene expression between two cells only differs on a few genes, FiRE will not consider them as highly different.
However, FiRE only allows us to get rareness scores on individual cells. As it stands, the algorithm cannot be used to find a subpopulation of rare cells, i.e. FiRE identifies a collection of rare cells, not *one rare subpopulation of cells*. Expanding the FiRE method for it to be able to assign a rareness score to a group of cells will be the task touched upon in chapter 3.

2.3 MicroCellClust

MicroCellClust is an algorithm developed by Gerniers, A. et al. [8] whose goal is to identify, within a cell population, a group of rare cells along with the subset of genes that makes this subpopulation rare. These subpopulations are to be retrieved from single-cell gene expression data. This task is transcribed in the form of a constrained optimisation problem, formalised as a variant of the max-sum submatrix problem [20], which is known to be NP-hard.

2.3.1 Preprocessing

The input of the MCC algorithm is a preprocessed scRNA-seq dataset. The preprocessing applied to the input scRNA-seq dataset transforms the gene expression matrix into a matrix where the genes are represented by the rows and the columns are representing the cells, and where the entry m_{ij} of the matrix is positive if the gene i is expressed in cell j and negative otherwise. To do so, a log-scaling $log_{10}(m_{ij} + 0.1)$ is applied to all the elements of the original matrix. This function returns a positive value when the expression value is over 0.9 and returns a negative value otherwise. Thus, 0.9 is considered as the expression threshold above which a gene is considered to be expressed in a cell.

2.3.2 Adaptation of the max-sum submatrix problem

The MicroCellClust method is formalised as an adaptation of the max-sum submatrix problem [20]. The objective is the following : find a subset I of genes (i.e. rows) and a subset J of cells (i.e. columns) that maximise the sum of gene expression values. However, a new nuance is introduced to the original problem; here, a small population of cells characterised by highly specific genes is being sought. In other words, the objective is to return a group of cells with a high expression of particular genes that are rarely expressed in other cells.

The MicroCellClust objective function is defined as follows:

$$(I^*, J^*) = \underset{\substack{I \subseteq \mathcal{G} \\ J \subseteq \mathcal{C}}}{\operatorname{argmax}} \sum_{i \in I} \Big(\sum_{j \in J} m_{ij} - \kappa \sum_{k \in \mathcal{C} \setminus J} \max\{0, m_{ik}\} \Big) \tag{2.4}$$

$$\text{such that } \frac{\Big| \Big\{ (i,j) \,\Big|\, i \in I, j \in J, m_{ij} < 0 \Big\} \Big|}{|I| \cdot |J|} \leq \mu \tag{2.5}$$

Where $\mathcal{G}$ and $\mathcal{C}$ represent the sets of genes and cells respectively and m_{ij} represents the elements at row i and column j of the preprocessed gene expression matrix m.

The first term of the sum of (2.4) represents the classic max-sum submatrix's definition as it computes the sum of the selected genes' expression in the selected cells. MCC introduces a second term to the original definition of the problem, called the out-of-cluster penalty. This out-of-cluster penalty enforces the resulting set of genes to contain highly specific genes. Indeed, it penalises the objective value

for any positive expression of the genes in I in cells that are not in the cluster J. To control the proportion of negative gene expression values (i.e. genes present in I that are not expressed by the cells in J), a constraint is introduced (see (2.5)). That constraint limits the proportion of negative gene expression values to remain under a threshold μ, typically fixed to 10%.

2.3.3 Beam Search

To solve the adapted max-sum submatrix problem in a computationally efficient way, a beam search based approximation algorithm is used by MCC.
A beam search algorithm is a search algorithm that usually aims to find the most likely sequence of output tokens given an input sequence. It explores different paths in a probabilistic model to generate the most probable output sequence. Such an algorithm starts with an initial input and generates a set of possible output tokens. The beam search maintains a fixed-width "beam" that contains the most promising candidates. It prunes the less likely candidates to only focus on the best ones.
At each step, the algorithm expands the beam by generating new candidate sequences by appending tokens from the current candidates. The new candidates are assigned probabilities based on the model's predictions. The beam is pruned to retain the top candidates, typically based on their cumulative probabilities. This process continues until a maximum length or a predefined stopping criterion is reached.

2.3.4 MicroCellClust's generic approach

The initial step involves forming the set of all possible pairs of cells. Subsequently, the algorithm proceeds to calculate the objective value for each pair and from this pool of pairs, only the k best pairs with the best objective value are retained. Computing the objective value (i.e. selecting the set of genes that maximises the objective function for the given cells) for a group of cells can be done in linear time. Once the set of pairs of cells with the highest objective value has been retrieved, the same step is repeated but now with groups of three cells. In other words, triplets of cells are generated from the selected pairs, the objective value of these triplets is computed and the ones with the smaller objective values are pruned before repeating the step once again with clusters of four cells and so on. The algorithm stops when no improvement is observed in terms of objective value for several levels in a row.

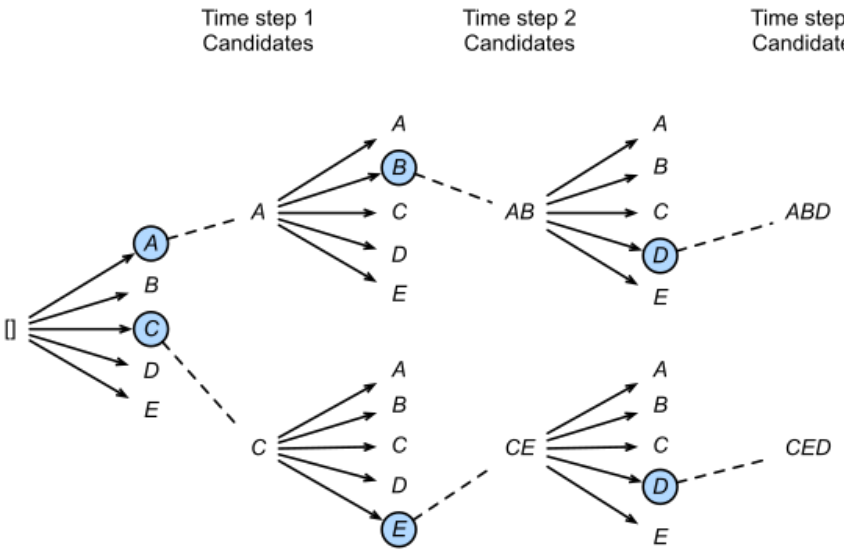

Figure 2.1: Illustration of the process of a beam Search with a beam width of 2. [21]

2.3.5 Introduction of a heuristic and the role of FiRE in MicroCellClust

Since the first search step consists in generating all possible pairs of cells, it cannot be solved in linear time but in quadratic time. The complexity of this first level represents a serious bottleneck for datasets containing more than 10.000 cells. The following steps can be solved in linear time since there is a pruning mechanism. Therefore, in order to improve the efficiency of the first step and linearise its complexity, a scalable version of the MCC algorithm called MicroCellClust 2 (MCC2) [22] was introduced. MCC2 implements two heuristics based on the rareness scores of cells to make the algorithm more efficient with larger datasets.

First, when the input dataset contains a large number of cells, MCC2 is going to select the cells that are the most likely to be part of the final solution as input for the beam search. Typically, MCC2 is going to be executed exclusively on the cells which are considered rare by FiRE, since these are the cells that are the most likely to express highly specific genes.
The second heuristic reduces the search space into a smaller one during the first level of the beam search. Based on the assumption that relatively similar cells are expected to have a similar FiRE rareness score, MCC2 only pairs each cell with the n cells that have the closest FiRE score. Finding the n closest FiRE scores can be done in linear time and, thus, solves the bottleneck issue.

MCC2 also introduces a local search based on simulated annealing [23]. This local search explores neighbours of the solution returned by the beam search (i.e. slightly modified version of the current solution, obtained by adding or removing one cell from the cluster).

This thesis aims to define a method to obtain a rareness score for groups of cells, to allow the use of the second heuristic at each level of the beam search. With such a scoring method, at each level, groups could be linked with the n cells that have the closest rareness score.

2.3.6 Parameters of microCellClust

The MicroCellClust solver takes several parameters in input :

- μ is the percentage of negative value that is accepted in the final solution. It helps select genes that are expressed in most of the cells in the cluster. The usual value for μ is 10%.
- κ is a real number between 0 and 1 and controls how specific the genes included in the solution are. The higher κ will be, the more non-specific genes will penalise the objective value. A good value for κ is $\frac{100}{|C|}$
- $nPair$ is the number of cells that each cell is going to be paired with during the first beam search level, based on their FiRE score. The higher $nPair$, the larger the search space and thus the better the objective value will tend to be as more candidates will be evaluated. However, as mentioned previously, if too many pairs are generated, the algorithm might not be able to terminate due to time complexity problems. A value of 100 is generally chosen as it provides a good trade-off between complexity and reducing the search space.
- The variable $nKeep$ represents the number of candidates that are retained at each step of the search process (i.e. the clusters giving the best objective values).
- The variable $stopNoImprove$ denotes the number of consecutive steps where no improvement in objective value is observed. This parameter determines when the algorithm should halt its execution.

Chapter 3

A classification approach: scoring cell clusters

This chapter studies different approaches to adapting the FiRE methodology to classify groups of cells as rare or non-rare. This classification is done by assigning a rareness score to groups of cells based on the rareness and homogeneity of the cells composing these groups. The first section presents the methodological aspects and the experimental setup, the next section introduces a baseline approach and the last section analyses an adaptation of the FiRE algorithm called FiRE-n and its results.

3.1 Methodology and experimental set-up

3.1.1 Assessment criteria

What are the criteria against which the group scoring methods should be evaluated? In other words, what characterises the rareness of a cluster of cells? Two criteria will be discussed next: the rareness of the cells forming the group and the homogeneity of the group.

Rareness of the cells

The first, and perhaps the most evident, criterion is the rareness of the cells constituting the group. In this thesis, we decided that cell rareness would be determined according to the FiRE methodology (see section 2.2) and that for a subpopulation of cells to be considered rare, all cells part of this group should be considered rare themselves.
The FiRE score is used as the rareness criterion to evaluate how cells should be considered during the experiments for several reasons. First, FiRE has proven to be

an efficient method to retrieve rare cells [7]. Secondly, opting for FiRE's rareness threshold as a criterion allows the detection of rare cells even within abundant cell types. Indeed, even populations of cells labelled with the same cell type can contain subpopulations of specific cells that could be interesting to observe.
Therefore, using FiRE scores to determine cells as rare or abundant makes sense as it will retrieve cells that express different genes than the majority of the population, without taking their type into account as unsupervised data is taken in input.

Homogeneity of the group

Having a group composed only of rare cells is not sufficient for it to be considered *one* rare subpopulation. The cells that it contains should also be rare for similar reasons. Rare cells coming from different cell types should not be clustered together. Indeed, the objective is here to identify homogeneous subpopulations of cells so that further analysis can be run on them. In the following, a pair of cells is considered homogeneous if both cells are of the same type (i.e. have similar differentially expressed genes), and a cluster of cells is deemed homogeneous if all the pairs of cells composing the group are homogeneous.

Clusters that meet both the rareness and homogeneity criteria will be labelled as *critical* in the rest of this thesis.

3.1.2 Experimental set-up

Dataset

To evaluate the performance of different approaches a dataset consisting of 1000 cells originating from two distinct types is used. The data used to construct these datasets comes from a dataset containing 1607 cells of type 293T and 1781 cells of type Jurkat that is publicly available from the 10xgenomics website.
It was arbitrarily decided to use the 293T cells as an abundant type and to consider the Jurkat cells as the rare type. Unless otherwise specified, in the following, experimental datasets are built by randomly selecting 950 293T cells and 50 Jurkat cells from the original dataset, giving us a subpopulation of 5% of cells coming from the rare type.

Types of cells From this dataset and the FiRE output, four different categories of cells can be identified :

- The 293T cells considered abundant by FiRE (i.e. whose FiRE score is below the rareness threshold).

- The 293T cells considered rare by FiRE (i.e. whose FiRE score is above the rareness threshold). Indeed, there could be different subtypes of 293T cells, which could potentially be rare.
- The Jurkat cells considered abundant by FiRE. In this context, they can be seen as a false negative because FiRE failed to identify them as rare.
- The Jurkat cells considered rare by FiRE.

Group labels From these categories of cells, grouping cells together can lead to 4 different categories of groups:

- **Abundant homogeneous**: Groups composed of cells that share the same type and that are all considered abundant by FiRE.
- **Abundant heterogeneous**: Groups that are partly composed of 293T cells considered common by FiRE but also of Jurkat cells. This kind of group is labelled as abundant because it contains some cells that are not considered rare by FiRE, therefore not respecting the rareness criterion issued in section 3.1.1.
- **Rare homogeneous / Critical**: Groups composed entirely of cells sharing the same type and who are all said to be rare by FiRE.
- **Rare heterogeneous**: Groups consisting entirely of cells deemed rare by FiRE and originating from the two distinct types, therefore not respecting the homogeneity criterion issued in section 3.1.1.

Objective

The scoring methods should be able to correctly assign a label to a group of cells according to the criteria introduced. Only rare homogeneous groups should be labelled as critical. A drop in score should be observed for heterogeneous groups for their lack of homogeneity.

Evaluation Indicators

To evaluate the effectiveness of each method several indicators are observed.

Criticality (Adequate labels) First, for each group, the label (i.e. critical or non-critical) assigned by the method is checked. A method is considered effective if it only designates the *rare homogeneous groups* as *critical* since each of the other categories of groups does not respect at least one of the rareness or homogeneity criteria.

Score Difference (Group score vs cell score) Then, in order to gain insights into the distribution of the scores assigned, the average difference between the individual FiRE scores of the cells composing a group and the score assigned to the group is looked at. The score difference of a group G is computed as follows:

$$\text{Score difference(G)} = \frac{\sum_{i \in G} abs(Score(G) - FiRE_i)}{|G|} \tag{3.1}$$

The goal is to observe how the composition of a group impacts the score assigned to it. Ideally, one expects the score of a homogeneous group to resemble the scores of the cells composing it (the score difference is smaller) while it would be desirable for scores of heterogeneous groups to suffer a drop from the individual rareness scores of the cells for lacking homogeneity (the score difference is larger).

Score range The third indicator is whether the group score falls within the range of the FiRE scores of its cells. For further uses, it would be preferable for the scores of groups to remain comparable to the individual FiRE scores. Therefore, the scores of homogeneous groups are desired to be between the maximum and minimum FiRE scores of their cells. On the other hand, the score of heterogeneous groups should be lower than the lowest of the FiRE scores of the cells composing the group to show the lack of homogeneity.

F_1 score The last indicator used to evaluate scoring methods is the F_1 score. The F_1 score is a metric often used to assess the performance of a classification model. It combines both precision and recall into a single value, providing a balanced measure of a model's accuracy. *Precision* represents the ratio of correctly predicted positive observations to the total predicted positives, while *recall* represents the ratio of correctly predicted positive observations to the actual positives in the dataset. In the current context, *positive observations* represent groups of rare and homogeneous cells, i.e. cells of the same type, whose FiRE score exceeds the FiRE rareness threshold. The F_1 score is computed as the harmonic mean of precision and recall, ensuring that both aspects are equally weighted.

3.2 Baseline approach

A straightforward approach to assign a rareness score to a group of n cells could be to simply derive a score solely from the FiRE scores of the cells forming that group. One could imagine many metrics that could be applied to the set of FiRE scores of a group of cells, for this baseline approach we chose to opt for the minimum.

3.2.1 Description of the method

Scoring

The metric chosen in this baseline approach is the *minimum of the individual FiRE scores* of the cells in the group. Using this method, the rareness score assigned to a group G is the lowest FiRE score among the cells composing that group.

$$\text{Score of group} = \min_{i \in G} \text{FiRE}_i \tag{3.2}$$

The choice of this metric is motivated by the rareness criterion described in section 3.1.1. If a group is to be considered rare only if all the cells in the group are considered rare by FiRE then it makes sense to evaluate the rareness of the group through the rareness score of its most abundant cell.

Classification Criterion

To determine whether a group is to be considered critical according to this method, the rareness threshold used in the original FiRE algorithm will be employed (see subsection 2.2.2). Therefore, a group is considered critical according to this method only if its most common cell has a FiRE score above the rareness threshold, enforcing all the cells composing a rare group to be considered rare by FiRE.
Defining a new rareness threshold for this specific method would not make sense as the distribution of FiRE scores remains identical.

3.2.2 Results

In order to assess the baseline approach, we designed an experiment to observe its effectiveness on the four different categories of groups described in section 3.1.2. Table 3.1 shows the average of each evaluation indicator (see section 3.1.2) on 100 independent runs where for each run, a new dataset containing 5% of rare cells was sampled. At each run, one group of size 3, 5 and 10 was randomly sampled for each category. The distribution of each type of cell for the heterogeneous groups was the following :

- Groups of size 3: two Jurkat cells and one 293T cell.

	Criticality (% critical)	***Difference***	***Range*** (% inside)
Rare homogeneous (Jurkat cells)			
Desired result	critical (=100%)	smaller	inside range (=100%)
N=3	100%	29.074	100%
N=5	100%	38.259	100%
N=10	100%	43.13	100%
Abundant homogeneous (293T cells)			
Desired result	not critical (=0%)	smaller	inside range (=100%)
N=3	0%	62.739	100%
N=5	0%	72.972	100%
N=10	0%	91.005	100%
Rare heterogeneous (Jurkat and rare 293T cells)			
Desired result	not critical (=0%)	larger	outside range (=0%)
N=3	100%	28.162	100%
N=5	100%	34.252	100%
N=10	100%	45.686	100%
Abundant heterogeneous (at least one non-rare 293T cell)			
Desired result	not critical (=0%)	larger	outside range (=0%)
N=3	0%	183.022	100%
N=5	0%	200.248	100%
N=10	0%	250.253	100%

Table 3.1: Percentage of groups of each category that are considered critical using the baseline method, the average difference between the group score and the FiRE score of the cells in the groups and percentage of scores falling within the range of cell scores for each category of groups.

- Groups of size 5: three Jurkat cells and two 293T cells.
- Groups of size 10: seven Jurkat cells and three 293T cells.

To identify to which category each cell belonged, the FiRE algorithm was run with values of 1017881, 100, and 50 for parameters H, L and M respectively.

Table 3.1 shows that this approach is not able to discriminate heterogeneous from homogeneous groups, and thus assigns a high score to both homogeneous and heterogeneous groups of rare cells, constantly misclassifying the latter. Indeed, since all the cells composing these groups are sampled from cells that are considered rare by FiRE, it is straightforward that the minimum FiRE score of these cells is above the FiRE rareness threshold. Therefore, this baseline method already

exhibits one obvious drawback: it is only taking one of the two criteria into account; only the rareness of the cells is considered and the method fails to account for the homogeneity criterion.
The second column shows that the average difference between group score and individual FiRE score is larger for *abundant heterogeneous groups* than for *abundant homogeneous groups*, in line with the desired outcome. This larger difference is due to the *heterogeneous abundant group* being made of rare cells of the jurkat type (with thus, on average, a higher FiRE score) and of abundant cells of the 293T type. However, there is no significant difference between *homogeneous and heterogeneous rare groups* since they are composed of cells with similar rareness scores, albeit from different types. The baseline method thus also fails in this respect.

The third column assesses if the group score falls within the desired range, i.e. inside the range of cell scores for groups of rare cells and lower for groups of abundant cells. In this approach, this indicator is trivial and fails for heterogeneous groups, since, by design, the group score is the lowest cell score and cannot be lower.

Additionally, the results are consistent whatever the size of the group.

To conclude, simply combining the FiRE scores is not sufficient to identify homogeneous groups of rare cells. This motivates adapting the FiRE method itself, to take advantage of the sketching technique to evaluate the homogeneity of cells. This baseline approach will be used to benchmark the rest of the methods for identifying critical groups.

3.3 Generalising FiRE

In the previous sections of this chapter, the experimental setup and a baseline method to attribute a rareness score to groups of cells were discussed. This section introduces another classification method which consists of an attempt at generalising the FiRE algorithm, adapting it so that it is able to assign scores to groups of cells of any given size n, hence its name FiRE-n. The naive approach developed in section 3.2 can retrieve clusters of rare cells but never evaluates whether these cells are linked based on their gene expression. Therefore, FiRE-n is introduced as it differs from the baseline approach insofar as it includes a measure of group homogeneity in the scoring process. The following subsections address the way homogeneity is measured and how FiRE-n is implemented. Then the results of the experiments are discussed and compared with those of the baseline approach.

3.3.1 Homogeneity metric

As explained in section 2.2.1, the FiRE algorithm works by repeating the process of placing cells in buckets based on their gene expression for a given set of M genes and then by assessing the rareness of the cells according to the relative size of the buckets they were stored in. At a given iteration of the FiRE process, two cells are placed in the same bucket if they have a similar expression on the M genes randomly selected at that iteration. Therefore, one could expect two cells that fall often in the same bucket over the whole process to be of the same type. Two approaches are investigated: measuring homogeneity at each iteration of the FiRE process and measuring homogeneity once the FiRE process is over.

Measuring homogeneity at each iteration of the FiRE process

Since similar cells are expected to fall in the same bucket at each iteration, using the distribution of cells in the buckets as information on homogeneity immediately comes to mind. A pair of cells could be considered homogeneous at a given iteration if placed in the same bucket. That way, a homogeneity score for a group at a given iteration could be derived by observing the number of homogeneous pairs of cells it contains. Finally, the global homogeneity of a group could be measured by combining its homogeneity scores of every iteration, in the same way it is done to compute the original FiRE scores.

This approach has been implemented into the FiRE algorithm but has not led to usable results.

To understand why this approach failed, it is important to dive into the dynamics of bucket allocation. Parameter H, the size of the hashing table, determines the amount of buckets available for cells to be sorted into. The authors of FiRE advise choosing a prime larger than 10 times the size of the dataset as a value for H because it needs to be sufficiently large to avoid collision during the hashing of the sketch so that buckets adequately discriminate cell types. Figure 3.1 shows that, for a population of 1000 cells with a proportion of 95% of cells of one type and of 5% of cells of another type, when H is larger than 1000, more than 91.7% of buckets are homogeneous and filled with one type of cell. Therefore, two cells in the same bucket have a high chance of being of the same type if H is large enough. This proportion stays constant until less than 1000 buckets are available, the drop in bucket homogeneity is probably caused by hash collisions.

Nevertheless, as the parameter H increases, the number of buckets correspondingly expands, potentially resulting in decreased population density within each bucket. There is a trade-off between group discrimination (a larger H) and bucket

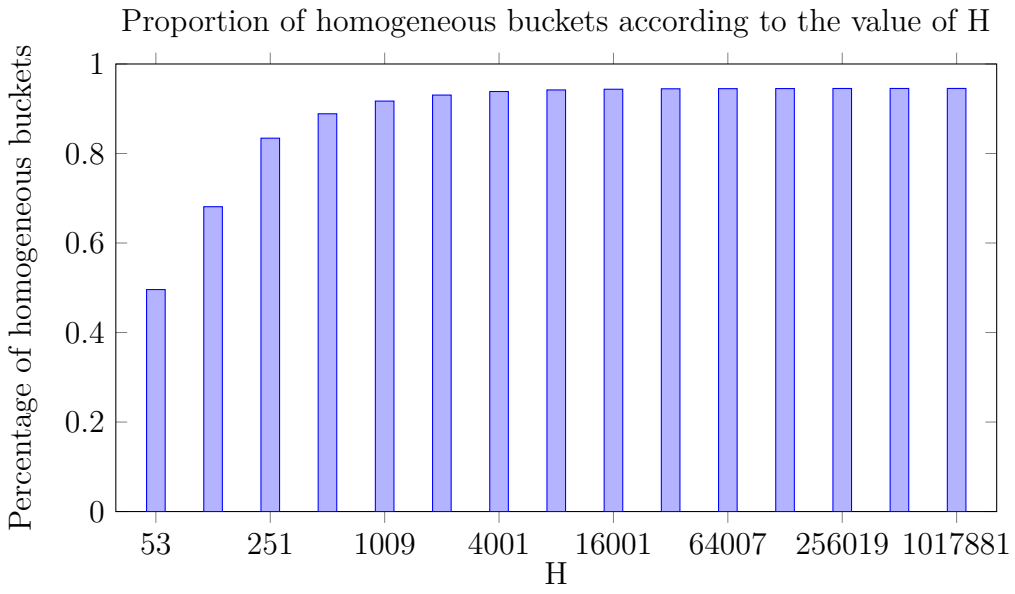

Figure 3.1: Percentage of buckets containing only one type of cell (i.e. either only Jurkat cells or only 293T cells) according to the number of available buckets

density (a smaller H). Ideally, to effectively gauge the homogeneity of a group as outlined earlier, it is desirable to have buckets that are predominantly composed of cells of a single type while representing a substantial portion of that type's population. However, as depicted in Figure 3.2, regardless of the value of H, buckets containing Jurkat cells typically contain fewer than 5 cells on average, whereas buckets containing 293T cells encompass an average of approximately 40 cells.

Moreover, Figure 3.3 shows that for a reasonable value of H, the 50 rare Jurkat cells are distributed into 37 buckets while the population of 950 abundant 293T cells is sorted into around 250 buckets, making it unlikely for two cells of the same type to be regularly in the same bucket. Moreover, the observed probability of a cell being alone in its bucket is also much higher for Jurkat cells (the rare ones) than for 293T cells (the abundant ones) as shown in Figure 3.4. Jurkat cells have almost a 50% chance of ending up alone in a bucket for a given iteration of the FiRE process.

Despite the high percentage of homogeneous buckets for sufficiently large values of H, measuring the homogeneity of clusters of cells as a combination of the homogeneity observed at each iteration of the process is assessed as presenting little added value for the following reasons. First, a rare cell is going to be alone in its bucket in 50% of the iterations of the process; meaning that in half of the iterations, it will be considered homogeneous to no other cells. Second, on average, cells share

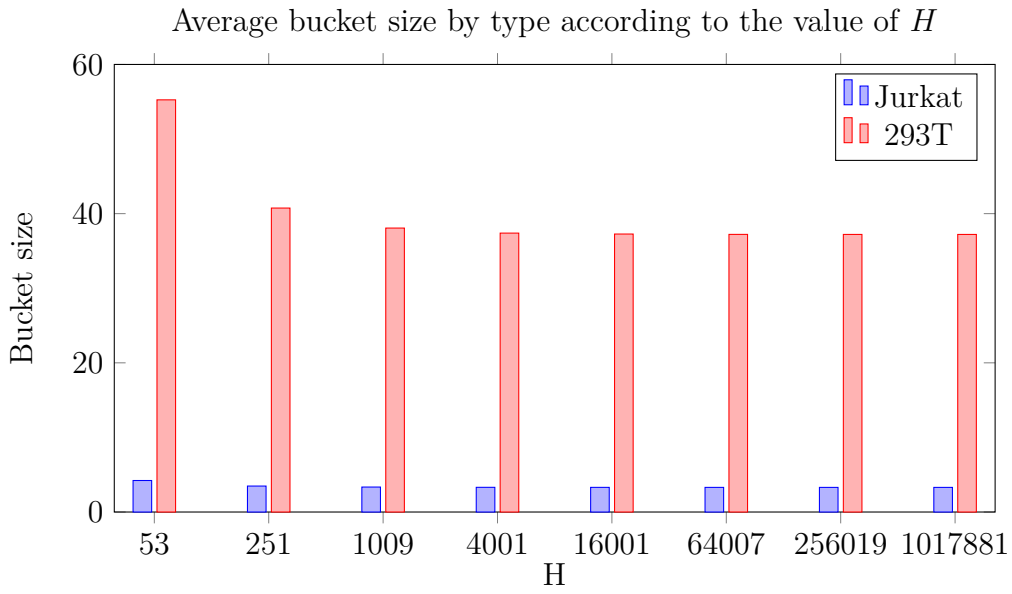

Figure 3.2: Average size of buckets containing at least one cell of that type, categorised by the value of parameter H

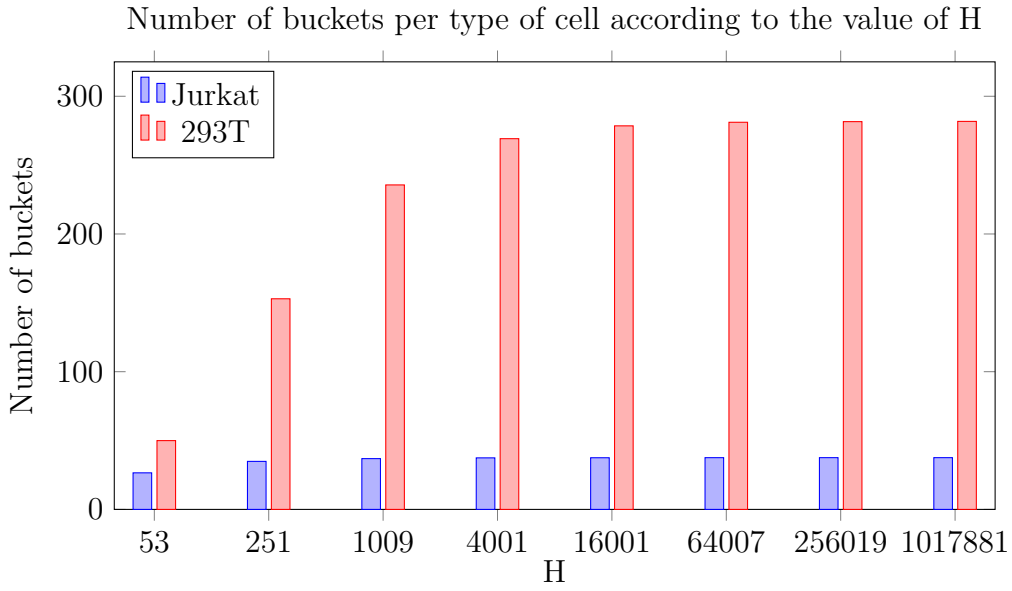

Figure 3.3: Average number of buckets in which each type of cell is distributed at each iteration according to the value of H

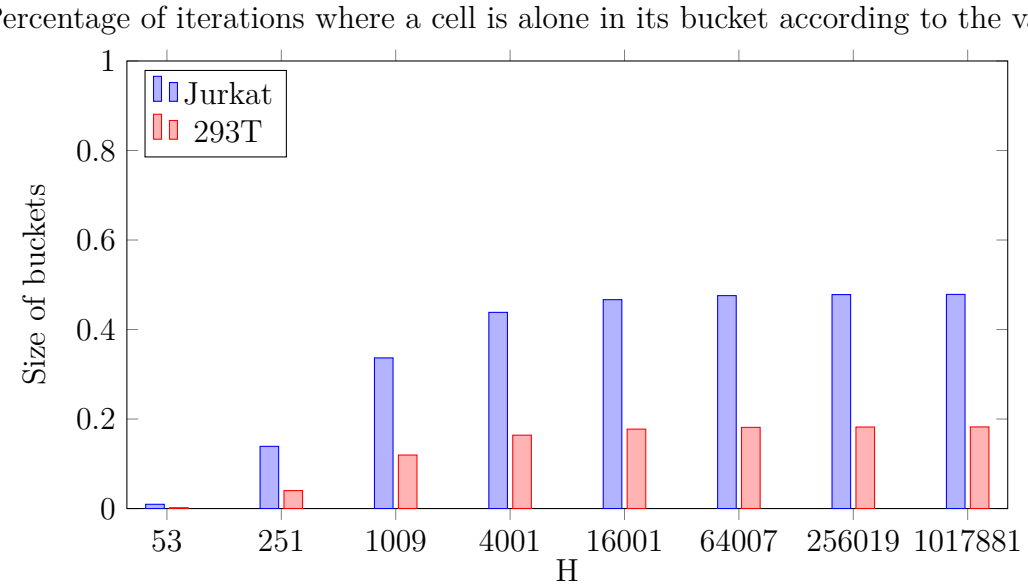

Figure 3.4: Percentage of iterations where a cell is alone in its bucket according to the value of H

their bucket with less than 10% of their respective population, making it unlikely for two cells of the same type to actually fall in the same bucket on a regular basis. Third, the rare population of cells is split into 37 different buckets on average at each iteration. Therefore, for all those reasons combined, the probability of two rare cells to be falling into the same bucket on a significant number of iterations, and thus of being considered homogeneous at the end of the process, is extremely low which leads to the conclusion that this method to measure homogeneity can not provide the expected information. Another approach has thus been investigated.

Measuring homogeneity once the FiRE process is over

To circumvent the fact that cells often end up alone at one given iteration, homogeneity could be measured by checking if two cells fell in the same bucket during any iteration of the whole FiRE process rather than at each iteration. To observe the average number of times two cells fall in the same bucket over the whole FiRE algorithm, 10 runs of FiRE were executed resampling the data for each run and the average number of times a pair of cells fell in the same bucket was stored. Figure 3.5 clearly shows two regions of homogeneous pairs (i.e. top right representing 293T pairs and bottom left Jurkat pairs). Cells of different types do not tend to fall into the same bucket during the whole process. Therefore, this information seems

exploitable to measure the homogeneity of a group of cells and to be incorporated into the FiRE algorithm to try to improve the baseline method.

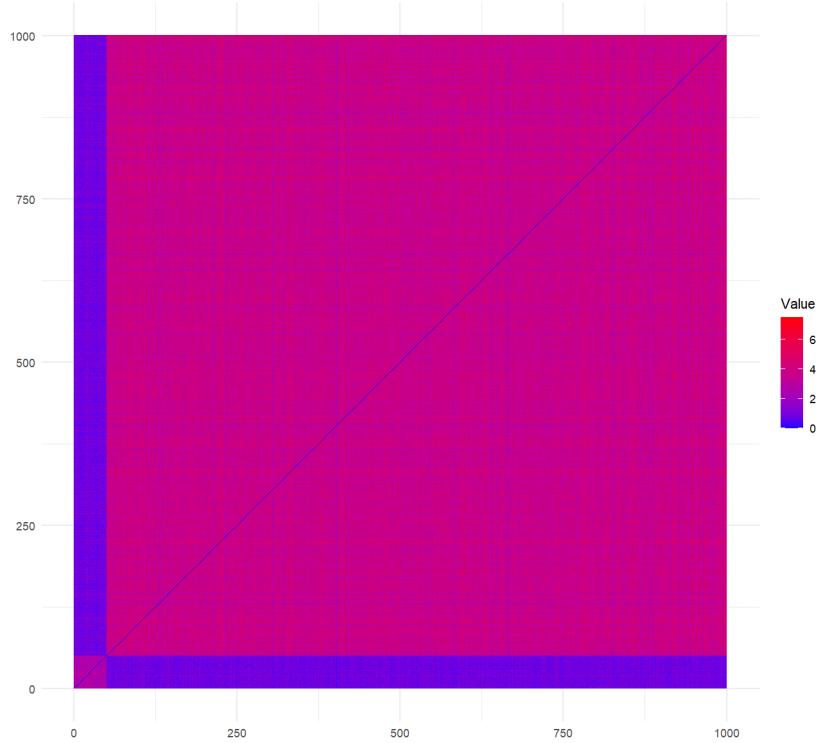

Figure 3.5: A N x N heatmap where each element represents the average number of times two cells ended up in the same bucket during the FiRE algorithm

3.3.2 FiRE-n

Scoring

The goal of this adapted method is thus to inject a measure of homogeneity into the group scoring process. Analysis of the baseline approach revealed that employing the minimum FiRE score of the cells constituting a group (section 3.2.1) effectively distinguished groups containing solely rare cells from others. Consequently, the decision was made to persist with this metric but with the addition of a penalty for groups lacking homogeneity.
To evaluate the homogeneity of a group of cells, a homogeneity score is computed based on the homogeneity property of the pairs of cells composing the group. From

the observations made in section 3.3.1, we consider a pair of cells as homogeneous if they fall in the same bucket during at least one iteration of the FiRE process. The homogeneity score of a group is defined as the *proportion of homogeneous pairs in the total number of pairs*. Thus, the rareness and homogeneity scores of a group G are computed as:

$$rareness(G) = \min_{c \in G} FiRE\ score(c) \tag{3.3}$$

$$homogeneity(G) = \frac{number\ of\ homogeneous\ pairs(G)}{\frac{|G| \times (|G|-1)}{2}} \tag{3.4}$$

This homogeneity score is then used to penalise the scores of clusters that lack homogeneity. If the proportion of homogeneous pairs in a group is below the threshold α, with $\alpha \in [0, 1]$, its rareness score will be multiplied by the homogeneity score. Otherwise, the rareness score remains unchanged.
Thus, the score assigned to a group G by the FiRE-n method is computed as :

$$FiRE\text{-}n(G) = \begin{cases} rareness(G) & \text{if } homogeneity(G) > \alpha \\ rareness(G) \times homogeneity(G) & \text{otherwise} \end{cases} \tag{3.5}$$

Optimisation of the threshold α

In order to determine the optimal value of the α threshold, the sensitivity of the F_1 indicator was studied. Different values of α ranging from 0.5 to 1 were experimented with, a value of 0.7, meaning that 70% of the pairs composing the group need to be homogeneous for the group to be considered homogeneous and to keep its score non-penalised.

According to the results presented in Figure 3.6, a value of α between 0.7 and 0.8 presents good results in terms of F_1 scores. When α increases from 0.7 to 0.8, precision increases but recall decreases. Since recall is the proportion of true positives caught by the algorithm, for the same F_1 performance, it is preferable to maximise recall rather than precision to avoid missing more true positives. Hence, a value of 0.7 was chosen for α.

Classification criterion

The criterion used by FiRE-n to classify groups (i.e. critical or non-critical) is the same as the one used in the baseline method. A group is considered *critical* with this method if its FiRE-n score exceeds that of the FiRE rareness threshold (see subsection 2.2.2) computed on the original population of cells. This choice is motivated by the observation that, using this classification criterion, the baseline

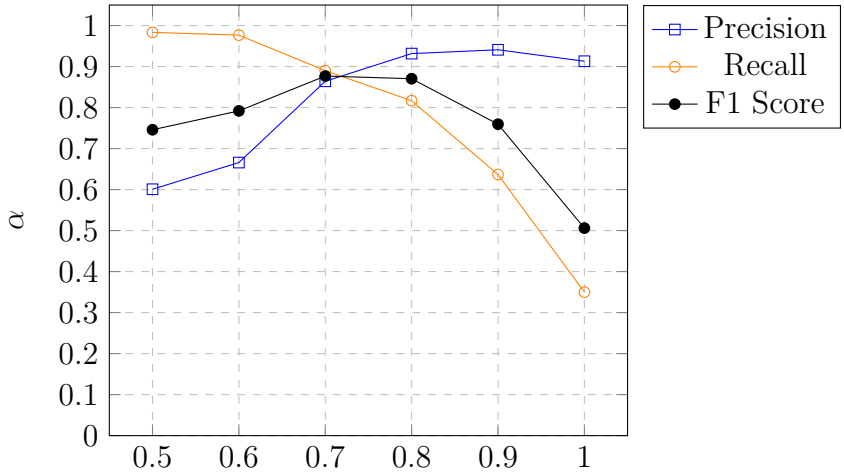

Figure 3.6: Precision, Recall and F_1 score for the homogeneity of groups according to values of α

method correctly identifies groups composed of only rare cells. Therefore, with the hypothesis that the score of heterogeneous groups is penalised by FiRE-n, it is expected that this classification criterion would effectively discern rare homogeneous groups from others.

Additional inputs

In addition to the L, M, and H parameters inherent in the original FiRE algorithm as well as the preprocessed scRNA-seq dataset, FiRE-n requires two additional inputs:

- The enumeration of the groups for which the scoring is desired
- The α threshold that determines the proportion of rare pairs required for a cluster to be considered homogeneous.

Therefore, FiRE-n requires the user to specify the different groups that need to be classified by the method.

Pseudo code

Algorithm 2 FiRE-n Algorithm

Input: $\mathbf{m} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{C}|}$, a gene expression matrix
Input: $\mathbf{L}, \mathbf{M}, \mathbf{H}$, the FiRE parameters
Input: $\alpha \in [0, 1]$, the homogeneity threshold
Input: **Groups**, the list of groups to score
Output: **Groups_Scores**, the FiRE-n score of each group $\in$ **Groups**
$FiRE\_Scores = FiRE(L, M, H, m)$ ▷ *Running the original FiRE algorithm*
▷ *Update homogeneity matrix for cells composing the groups* ◁
for all $l \in \{1, \ldots, \mathbf{L}\}$ **do**
    for all $g \in$ **Groups do**
        for all $i \in \{1, \ldots, |g|\}$ **do**
            $bucket \leftarrow buckets[l][g[i]]$ ▷ *Retrieve bucket of cell at iteration l*
            for all $j \in bucket$ **do**
                $homogeneity[g[i]][j] \leftarrow 1$
▷ *Assign a score to each group according to its rareness and to its homogeneity* ◁
for all $g \in$ **Groups do**
    $Rareness\_Score \leftarrow \min_{i \in g} FiRE\_Scores_i$
    $Homogeneous\_pairs \leftarrow \sum_{i \in g} \sum_{j \in g \setminus \{i\}} homogeneity[i][j]$
    $Number\_Of\_Pairs \leftarrow \frac{|g| \times (|g|-1)}{2}$
    $Homogeneity\_Score \leftarrow \frac{Homogeneous\_pairs}{Number\_Of\_Pairs}$
    if $Homogeneity\_Score < \alpha$ **then**
        $Group\_Scores[group] = Homogeneity\_Score \times Rareness\_Score$
    else
        $Group\_Scores[group] = Rareness\_Score$
return $Group\_Scores$

	Criticality (% critical)		***Difference***		***Range*** (% inside)	
Rare homogeneous (Jurkat cells)						
Desired result	critical (=100%)		smaller		inside range (=100%)	
Method used	Baseline	FiRE-n	Baseline	FiRE-n	Baseline	FiRE-n
N=3	100%	74%	29.074	149.315	100%	74%
N=5	100%	93%	38.259	73.251	100%	93%
N=10	100%	100%	43.13	43.13	100%	100%
Abundant homogeneous (293T cells)						
Desired result	not critical (=0%)		smaller		inside range (=100%)	
Method used	Baseline	FiRE-n	Baseline	FiRE-n	Baseline	FiRE-n
N=3	0%	0%	62.739	105.913	100%	85%
N=5	0%	0%	72.972	72.972	100%	100%
N=10	0%	0%	91.005	91.005	100%	100%
Rare heterogeneous (Jurkat and rare 293T cells)						
Desired result	not critical (=0%)		larger		outside range (=0%)	
Method used	Baseline	FiRE-n	Baseline	FiRE-n	Baseline	FiRE-n
N=3	100%	10%	28.162	439.575	100%	10%
N=5	100%	20%	34.252	547.49	100%	20%
N=10	100%	12%	45.686	461.441	100%	12%
Abundant heterogeneous (at least one non-rare 293T cell)						
Desired result	not critical (=0%)		larger		outside range (=0%)	
Method used	Baseline	FiRE-n	Baseline	FiRE-n	Baseline	FiRE-n
N=3	0%	0%	183.022	439.575	100%	36%
N=5	0%	0%	200.248	334.013	100%	68%
N=10	0%	0%	250.253	338.591	100%	71%

Table 3.2: Comparison of the two group scoring methods in terms of percentage of groups that are considered critical, average difference between the group score and the FiRE score of the cells in the groups and percentage of scores falling within the range of cell scores.

	Precision	*Recall*	F_1 *score*
Baseline	0.5	1	0.667
FiRE-n	0.864	0.89	0.877

Table 3.3: Comparison of the two group scoring methods in terms of precision, recall and F_1 score. These metrics are computed on the classification of the groups sampled for the experiments whose results are displayed in Table 3.2

3.3.3 Results

Table 3.2 compares the performance of the baseline method with the FiRE-n method and helps assessing whether FiRE-n can be considered satisfactory for scoring groups of cells.

A first observation of Table 3.2 is that FiRE-n is more effective than the baseline method at discerning *critical* groups from others. On average, most rare homogeneous groups are adequately identified as *critical* while most rare heterogeneous groups are being rejected. This is, thus, in line with the intended results.
It should be noted, however, that FiRE-n exhibits a small drop in performance for the identification of rare homogeneous groups, since, for smaller group sizes of 3 and 5 cells, respectively 26% and 7% of those groups are misclassified and not identified as *critical* (false negatives). For groups of 10 cells, all critical groups are adequately identified. The homogeneity penalty of FiRE-n can explain the positive rate of false negatives at lower group sizes. Indeed, for a group of size 3 to suffer from the penalty, only one of its three pairs needs to be considered heterogeneous (i.e. the two cells of the group never fall in the same bucket during the FiRE process). In that case, the homogeneity score of the group would be 0.666, below the 0.70 threshold. In other words, for a group of size 3 to avoid the homogeneity penalty, all cell pairs must have fallen at least once in the same bucket during the FiRE process. When the group size increases, the number of cell pairs allowed to never fall into the same bucket also increases, moving from 0 (size 3) to 2 or 20% of pairs (size 5) to 13 or 29% of pairs (size 10).
Additionally, while the baseline method misclassifies 100% of the rare heterogeneous groups as *critical* (false positives), FiRE-n inadequately identified only 14% of those groups on average (10%, 20%, 12% of groups of size 3, 5, 10 respectively).

As expected, the performance of the baseline approach in rejecting groups of abundant cells is maintained: no group containing non-rare cells was considered *critical*.

Furthermore, Table 3.2 shows that in contrast to the baseline method, FiRE-n also performs as expected with regard to the score difference indicator. It is smaller for homogeneous groups and larger for heterogeneous groups. This shows that the heterogeneity penalty is adequately applied, preventing heterogeneous groups from being misclassified in most cases.
Finally, Table 3.3 shows that the FiRE-n method outperforms the baseline method in terms of classification thanks to the large precision improvement. By penalising heterogeneous groups, FiRE-n reduces the number of rare heterogeneous groups that were considered *critical* which increases precision (less false positives). Despite

the slight drop in recall that translates to the slight increase of misclassified rare homogeneous clusters (more false negatives), the F_1 score of the FiRE-n method largely outperforms that of the baseline approach.

3.4 Conclusion

The objective of this chapter was to identify a method for giving a rareness score to groups of cells. After having presented an experimental set-up and tested a baseline approach, this chapter has developed the generalised FiRE-n method to assign a rareness score to groups of cells while adequately taking group heterogeneity into account through the application of a group heterogeneity penalty. By doing so, FiRE-n addresses the gaps of the baseline approach and manages to classify each category of groups defined in section 3.1.2, with satisfactory accuracy. Furthermore, the more populated the clusters are, the more accurate the classification is. However, FiRE-n requires a list of targeted cell clusters as input, which limits the applicability of the approach to scenarios where the clusters are known in advance. chapter 5 discusses an approach that intends to lift this limitation.

Chapter 4

Using FiRE-n as heuristics

In chapter 3 a generalised approach of the FiRE method called FiRE-n was introduced and evaluated. This method allows to score a group of cells based on the rareness of the cells that compose it and on their relative homogeneity; this approach appeared to be effective at identifying critical groups of cells.
The present chapter assesses the integration of FiRE-n as heuristics to improve the search within the MicroCellClust 2 (MCC2) algorithm [22] in terms of temporal execution and objective value. FiRE-n is used to prune candidates during the beam search implemented in MCC2.

4.1 Integration of FiRE-n

4.1.1 Rationale

As explained in subsection 2.3.5, the original MCC2 algorithm makes use of the FiRE scores to reduce the search space. Under the assumption that two similar cells should have similar rareness scores, MCC2 pairs each cell to the n cells having the closest FiRE score and uses these pairs for the first level of the beam search.
The idea behind the integration of FiRE-n is to use the same heuristic at each level of the beam search for the construction of groups. FiRE-n allows us to obtain a criticality score for a given group, making it possible to reduce the search space at every beam search level by only associating groups to cells with a FiRE score similar to the group's FiRE-n score.
In the original MCC2 paper, the use of this heuristic to improve the algorithm's time execution is motivated by the fact that similar cells are expected to have rather similar scores. However, it is not straightforward that this assumption still holds true when comparing the original FiRE scores with FiRE-n scores. As a reminder, a group's FiRE-n score is equal to the minimum of the FiRE scores of the cells in

the group, to which a penalty can be applied according to the homogeneity of the group. Therefore, if a group is considered to be homogeneous by FiRE-n, the FiRE and the FiRE-n scores remain comparable. Indeed, the FiRE-n score will be equal to the individual FiRE score of one of the cells composing the group. Therefore, since the group was deemed homogeneous, by the assumption that similar cells should have similar rareness scores, the FiRE-n score should be representative of the group's FiRE scores. We can, thus, expect the FiRE and FiRE-n scores to be comparable for homogeneous groups. On the other hand, if a group is not deemed homogeneous, a penalty is applied to the rareness score of the group which removes the comparability property of the two scores. However, groups that are considered heterogeneous are not likely to be great candidates for the MicroCellClust solution as cells that are not of the same type will rarely express the same highly specific genes. Therefore, linking non-similar cells to these heterogeneous groups should not hurt the final objective value too much as they are unlikely to be one of the top solutions that will be considered at the next level. The MCC2 algorithm with the FiRE-n integration will be called MicroCellClust 2$\star$ (MCC2$\star$) in the rest of this thesis.

4.1.2 Implementation

The pseudo-code for the implementation of the beam search of MCC2$\star$ is detailed below in Figure 4.1. Two additional inputs are required in comparison to the original MCC2 algorithm: the similarity matrix of the input cells and the α parameter. The similarity matrix is a matrix that denotes the number of times that two cells fell in the same bucket during the FiRE process and the α parameter represents the proportion of pairs of cells that are needed to be homogeneous for a group of cells to be considered as such (see section 3.3.2). Both of these parameters are necessary to compute the FiRE-n scores of the groups of cells at each level of the beam search.
The generation of pairs remains unchanged compared to the original MCC2 algorithm. Each cell is linked to the cells that have the closest FiRE score; the number of cells each cell is paired with depends on the value of the parameter $nPair$ (see subsection 2.3.6). The beam search of MCC2$\star$ starts to differ after the evaluation of pairs of cells. From level 3 until the end of the beam search, the $nKeep$ groups that returned the highest objective value at the previous level will be selected and the FiRE-n score of each of these groups will be computed. Then, as for the second level, candidate groups will be formed by linking each of the best groups of the previous level to the $nPair$ cells whose FiRE score is the closest to the group's FiRE-n score. The beam search will then be continued as per the original algorithm.

Algorithm 3 MCC2⋆ beam search

Input: $\mathbf{M} \in \mathbb{R}^{|\mathcal{G}|\times|\mathcal{C}|}$ an expression matrix
Input: $\mathbf{r} \in \mathbb{R}^{|\mathcal{C}|}$ the FiRE score distribution for the cells
Input: **simMatrix** $\in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$ the similarity matrix of the input cells
Input: α the similarity threshold parameter of the FiRE-n method
Output: $I^{\sim} \subseteq \mathcal{G}, J^{\sim} \subseteq \mathcal{C}, \Omega^{\sim} \subseteq \mathbb{R}$, the sets of gene and cells part of the bi-cluster solution and the corresponding objective value
$I^{\sim}, J^{\sim}, \Omega^{\sim} \leftarrow \emptyset, \emptyset, 0$
▷ *Evaluating pairs ($l = 2$)* ◁
for $j \in \mathcal{C}$ **do**
    for $j' \in 100\_closest(\mathbf{r}, j)$ **do**
        $J \leftarrow \{j, j'\}$
        $I, \Omega \leftarrow obj(\mathbf{M}, J)$
        if $\Omega \geq \Omega^{\sim}$ **then**
            $I^{\sim}, J^{\sim}, \Omega^{\sim} \leftarrow I, J, \Omega$
for $l \in \{3, 4, \dots\}$ **do**
    $Groups\_scores \leftarrow FiRE - n\_score(100\_best(l-1),$ **simMatrix**$, \alpha,$ **r**$)$
    for $J^0 \in 100\_best(l-1)$ **do**
        $J^0\_score \leftarrow Groups\_scores(J^0)$
        for $j \in 100\_closest(\mathbf{r}, J^0\_score, J^0)$ **do**
            $J \leftarrow J^0 \cup \{j\}$
            $I, \Omega \leftarrow obj(\mathbf{M}, J)$
            if $\Omega \geq \Omega^{\sim}$ **then**
                $I^{\sim}, J^{\sim}, \Omega^{\sim} \leftarrow I, J, \Omega$
Return $I^{\sim}, J^{\sim}, \Omega^{\sim}$

Figure 4.1: Pseudo-code for the adapted version of the beam search presented in [22], assuming a value of 100 for both *nKeep* and *nPair* (respectively the number of top-solutions to consider for expansion at the next level and the number of candidates to form from of the top solution of the previous level)

4.2 Results

	Objective value		Beam Search execution time [sec.]	
Proportion of rare cells	*MCC2*	*MCC2⋆*	*MCC2*	*MCC2⋆*
1%	197.4	200.557	80.227	62.684
2.5%	296.536	303.739	85.66	65.87
5%	456.167	468.393	96.053	69.883
10%	676.505	685.374	119.687	83.403

Table 4.1: Comparison of the objective value returned by MCC2 and MCC2⋆ and of the time needed by each method to complete the beam search. Experiments were conducted on datasets composed of 1000 Jurkat and 293T cells with Jurkat cells representing 1%, 2.5%, 5% or 10% of the total population. Results are the average of 100 independent runs. (Windows 10 Home; 1.6GHz Intel Core i5 CPU; 8GB RAM)

Table 4.1 displays the comparison between MCC2 and MCC2⋆. The objective value and the time execution of the beam search of both methods are observed in order to measure the impact of the integration of FiRE-n in the original MCC2 algorithm. The experiments are performed on datasets containing 1000 cells coming from Jurkat and 293T types with a proportion of Jurkat cells varying from 1% to 10%. Both methods are executed with the default parameters that are $nKeep$ and $nPair$ set to 100, μ fixed to 0.1, κ with a value of 1 and $stopNoImprove$ set to 25 (see subsection 2.3.6). A value of 0.7 was chosen for the α parameter during the execution of the MCC2⋆ algorithm, meaning that groups with less than 70% of homogeneous pairs would see their FiRE-n score penalised. This value for α is motivated by the experiments of section 3.3.2. The results displayed in Table 4.1 are the average of the results of the execution of both algorithms on 100 different populations of cells.

The first observation that can be made is that, on average, MCC2⋆ outperforms MCC2 both in objective value and in time execution in all the studied cases; the execution time of MCC2⋆'s beam search is shorter on average and yields an average objective value greater than the one returned by MCC2.
Table 4.1 reveals that the objective value returned by MCC2⋆ slightly exceeds on average the one returned by the original algorithm. That shows that, despite the pruning applied during MCC2⋆'s beam search, the use of the heuristic helps keeping the best candidates in the beam. That heuristic even allows the selection

of good candidates that are being eliminated by MCC2, yielding slightly better results in terms of objective value.
Table 4.1 also shows that MMC2⋆ provides a speed up of the beam search of 30-40% compared to MCC2. That speed-up of the beam search translates to an improvement of 20-25% of the total runtime of the algorithm.

One of MCC2's strengths is its scalability as it is able to run on datasets of tens of thousands of cells. To ensure that the improvement brought to MCC2⋆ did not affect this property, the objective value and the total time execution for both methods are compared on populations made of 10.000 to 50.000 cells sampled from a dataset containing 68.000 peripheral blood mononuclear cells (PCMCs).

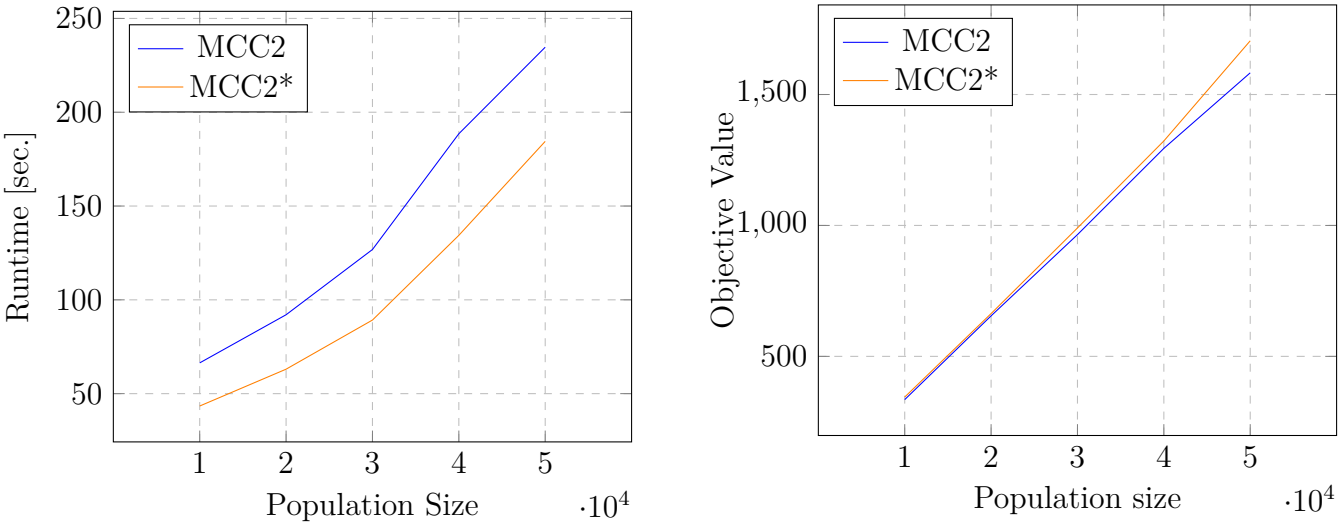

Figure 4.2: Evolution of the runtime and of the objective value of MCC2 and MCC2⋆ for populations of 10k to 50k PBMCs. The results are an average of 10 independent runs. (Windows 10 Home; 1.6GHz Intel Core i5 CPU; 8GB RAM)

Figure 4.2 shows that MCC2⋆ does not harm the scalability of MCC2. Moreover, the improvement in runtime observed on smaller populations on Table 4.1 seems to be greater on large-scale populations. Indeed, the larger the populations are, the more candidate groups are going to be evaluated by MCC2. Therefore, the pruning taking place in MCC2⋆ results in greater runtime improvements. The right-hand part of Figure 4.2 shows that the objective value returned by MCC2⋆ is still roughly similar to the one returned by MCC2 but that the slight improvement observed in Table 4.1 gets larger as the population increases. This reinforces the idea that the heuristic implemented in MCC2⋆ selects better candidates at each search level.

4.3 Conclusion

In this chapter, we introduced MCC2$\star$, an adapted version of the MCC2 algorithm where pruning is done at each level of the beam search based on the FiRE-n method developed in section 3.3. The integration of the pruning allowed for the algorithm to be executed 20% to 25% faster than the original version on relatively small datasets. The comparability property of the FiRE scores with the FiRE-n scores allows MCC2$\star$ to prune candidates of no interest at each level of the beam search without getting rid of the insightful ones, making it possible to return slightly better solutions than the original algorithm. The integration of FiRE-n into the MicroCellClust algorithm did not affect its scalability as the improvements observed in small populations were increased in larger populations.

Chapter 5

A clustering approach: retrieving the most critical clusters

As mentioned in chapter 3, the FiRE-n approach requires the groups submitted for classification to be specified as input, thereby restricting the applicability of the method to scenarios where the groups to classify are known in advance. For instance, FiRE-n can be used as a way of filtering candidate groups and used as heuristics for MCC2 as seen in chapter 4 .
The present chapter introduces an alternative method employing a clustering approach to identify critical groups of cells, instead of the classification approaches proposed in chapter 3. This method will once again rely on the FiRE algorithm and will make use of the rareness and homogeneity concepts described in section 3.1.1.

5.1 Retriever of critical clusters

This alternative method, called Retriever of Critical Clusters (ReCC) aims at returning the k most critical groups of a specified size, given a scRNA-seq dataset.

5.1.1 Generation of groups

A beam search approach is used to generate the candidate groups. The algorithm starts by evaluating all the possible pairs of cells and retrieves the k pairs that have the best criticality score. Then, at the next level of the beam search, each of these best pairs of cells is attached to each cell of the dataset to form all the possible triplets from the best pairs while excluding groups with duplicate cells. The criticality of these triplets is then evaluated and the k triplets with the best scores are kept. This process is repeated recursively until the groups generated are of the specified size n, and the k best groups of size n are returned.

The objective behind this greedy approach is two-fold: first, circumvent the requirement of generating all possible groups of a given size which implies factorial complexity for the algorithm; second, provide the opportunity to give a list of starting groups as input of ReCC instead of starting with all the possible pairs.

5.1.2 Scoring

In the same way as for FiRE-n, ReCC assigns a criticality score to the groups at each level of group generation. This criticality score is a combination of the rareness score and the homogeneity score of each cluster. The rareness and the homogeneity of groups of cells are assessed in the following way :

- The *rareness score* of a group is set to the minimum of the FiRE scores of the cells composing this group.
- The *homogeneity score* of a group is set to the average number of times two cells of the group fell in the same bucket during the execution of the FiRE algorithm.

At the initial recursion level, the computation of the rareness and homogeneity scores for a pair of cells $\{a, b\}$ is calculated as follows:

$$Rareness(\{a, b\}) = \min(FiRE_a, FiRE_b) \tag{5.1}$$

$$Homogeneity(\{a, b\}) = \frac{sim(a, b)}{\frac{2\times(2-1)}{2}} = sim(a, b) \tag{5.2}$$

Where $sim(i, j)$ represents the number of times two cells i and j fell in the same bucket during the execution of the FiRE algorithm.
Then, at each level of the beam search, the rareness and homogeneity scores of all the k best groups are stored so that the scores of a group G of size i to which a cell c is added can be updated recursively in linear time as follows :

$$Rareness(G \cup \{c\}) = \min(Rareness(G), FiRE_c) \tag{5.3}$$

$$Homogeneity(G \cup \{c\}) = \frac{\frac{i\times(i-1)}{2} \times Homogeneity(G) + \sum_{g\in G} sim(g, c)}{\frac{(i+1)\times i}{2}} \tag{5.4}$$

The group rareness score is simply updated by checking it against the FiRE score of the added cell and taking the smallest value of the two. The group homogeneity score is increased by the homogeneity scores of all the cells making up the

group with the newly added cell. Since the group homogeneity score is defined as the average number of times pairs of cells composing the group were placed in a common bucket, this averaging operation is reversed before adding the additional homogeneity scores and re-applying the average over the updated number of pairs.

Then, the homogeneity and rareness scores of all the clusters are normalised using min-max normalisation which scales the values of a feature to a range between 0 and 1. That normalisation technique was employed because the range of values of the two scores varied widely, therefore, such a technique allows us to compare both scores on the same scale. Once the scores of all the clusters have been standardised, the criticality score of a cluster K is computed as :

$$\textit{Criticality score}(K) = \beta \times \textit{Rareness}(K) + (1 - \beta) \times \textit{Homogeneity}(K) \quad (5.5)$$

The β parameter allows the change of the weight given to each of the scores to get the criticality score that returns the best results for a specific application.

5.1.3 Parameters

The beam search of ReCC was implemented recursively. Each recursion level l of the ReCC algorithm takes several parameters in input :

- The FiRE scores of all the individual cells.
- The similarity matrix of all the pairs of cells of the dataset, containing at each position ij the number of times cell i and cell j fell in the same bucket during the execution of the FiRE algorithm.
- The size n of the desired groups.
- The number k of groups to return.
- The list of the k *base groups* that will be used to construct the groups of the recursion level l.
- The rareness scores of each of the *base groups*.
- The homogeneity scores of each of the *base groups*.
- β, the weight assigned to the rareness score to compute the criticality score.

After its execution, the ReCC algorithm returns the k groups of size n that exhibit the highest criticality score among the groups generated, given the weight assigned to the rareness and the homogeneity scores, along with the rareness and homogeneity scores of each of these k groups.

5.1.4 Optimisation of the criticality formula

The criticality score is thus the weighted average of the group rareness score and the group homogeneity score. The parameter β controls the weight assigned to each sub-component. The value of β can be optimised by analysing the sensitivity of the classification performance of ReCC for a variation of β. This was done in a dataset of 500 cells, 95% of which of type 293T and 5% Jurkat [1]. Figure 5.1 displays the criticality precision among the 100 groups returned by ReCC for clusters of size 2, 3, 4 and 5.

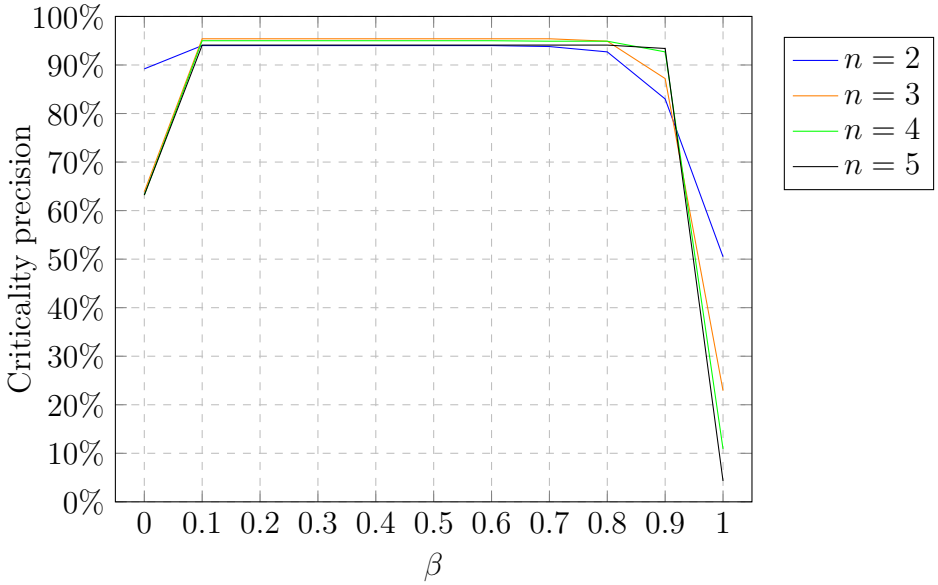

Figure 5.1: Percentage of criticality among groups retrieved by ReCC according to the β parameter

Figure 5.1 shows robust classification performance for $\beta \in]0, 0.9[$. Therefore, a value of 0.5 was chosen for the remaining of the experiments on ReCC.

5.1.5 Complexity

As hinted at previously, ReCC still faces complexity issues. The complexity of the method is linear regarding the number of cells for all but two steps: the computation of the similarity matrix and the generation of all possible pairs. Both of these operations require quadratic time and executing them would thus take away one of FiRE's main advantages: its efficiency.

[1] In this test, the experimental dataset was reduced from 1000 cells to 500 cells to keep enough unused cells to ensure independence with the datasets used in subsequent tests.

To circumvent this complexity issue, heuristics were introduced to reduce the search space of the beam search, in a similar manner to MCC2. The reduction of the dataset does not modify the level of complexity but allows for a timely execution of the algorithm. Since the objective of the method is to identify groups of homogeneous rare cells, the initial dataset of ReCC is reduced to the population of cells labelled as rare by FiRE. Filtering out the population of abundant cells with low FiRE scores has no impact on the output of ReCC since it builds homogeneous groups that maximise the minimum FiRE score of the group.

5.1.6 Pseudocode

Algorithm 4 ReCC Algorithm

1: **Input**: $\mathcal{C}$, the set of input cells
2: **Input**: **n**, the size of the desired groups
3: **Input**: **k**, the number of best groups to select
4: **Output**: The **k** groups of size **n** with the best criticality score
5: $FiRE = FiRE(L, M, H, \mathcal{C})$
6: $RareCells \leftarrow \textsc{Index}(Scores > FiRE\_threshold)$
7: $RareScores \leftarrow FiRE.scores[RareCells]$
8: $RareFiRE = FiRE(L, M, H, RareCells)$ ▷ *Re-Run FiRE to compute the homogeneity matrix on fewer cells (only rare ones)*
9: $SimMatrix \leftarrow RareFiRE.similarityMatrix$
10: **return** ReCC-Aux(RareScores, SimMatrix, **n**, **k**, [], [], [])

Algorithm 5 Pseudo Code for the recursive function of the ReCC method

Input: $\mathcal{C}$, the set of input cells
Input: **FiREScores** $\in \mathbb{R}^{|\mathcal{C}|}$, the FiRE scores of the cells
Input: **SimMatrix** $\in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$, the similarity matrix
Input: **n**, the size of the desired groups
Input: **k**, the number of best groups to select
Input: **BaseGroups** $\in \mathbb{R}^{k\times b}$, the k groups of length b returned at the previous search level
Input: **HomogeneityScores** $\in \mathbb{R}^{k}$, the homogeneity scores of each group $\in$ **BaseGroups**
Input: **RarenessScores** $\in \mathbb{R}^{k}$, the homogeneity scores of each group $\in$ **BaseGroups**
$CandidateGroups \leftarrow \emptyset$
$Homogeneity \leftarrow \emptyset$
$Rareness \leftarrow \emptyset$
if BaseGroups $== \emptyset$ **then**
    $CandidateGroups \leftarrow$ GenerateAllPairs($\mathcal{C}$)
    for all $group \in CandidateGroups$ **do**
        $Rareness[group] \leftarrow \min($**FiREScores**$[group[0]],$ **FiREScores**$[group[1]])$
        $Homogeneity[group] \leftarrow$ **SimMatrix**$[group[0]][group[1]]$
else
    for all $group \in$ **BaseGroups do**
        for all $i \in \mathcal{C}$ **do**
            if $i \notin group$ **then**
                $Candidate \leftarrow group \cup \{i\}$
                $Rareness[Candidate] \leftarrow$ Rareness($Candidate$) ▷ *See* (5.3)
                $Homogeneity[Candidate] \leftarrow$ Homogeneity($Candidate$) ▷ *See* (5.4)
                $CandidateGroups \leftarrow CandidateGroups \cup Candidate$
$Rareness \leftarrow$ MinMaxNormalization($Rareness$)
$Homogeneity \leftarrow$ MinMaxNormalization($Homogeneity$)
$Criticality \leftarrow \beta \times Rareness + (1-\beta) \times Homogeneity$
$ElectedCandidates \leftarrow$ Sort($Criticality$)$[0 : k-1]$
if $|ElectedCandidates[0]| ==$ **n then**
    return $\{ElectedCandidates, Homogeneity[ElectedCandidates], Rareness[ElectedCandidates]\}$
else
    return ReCC-Aux($\mathcal{C}$, **FiREScores**, **SimMatrix**, **n**, **k**, $ElectedCandidates$, $Homogeneity[ElectedCandidates], Rareness[ElectedCandidates]$)

5.2 Results

Throughout the experiments, ReCC was executed to return the 1000 most critical groups (parameter $k = 1000$) with a criticality score computed as the arithmetic average of the rareness and homogeneity scores of the groups (parameter $\beta = 0.5$). The FiRE algorithm was run for 100 iterations (parameter $L = 100$) and on 50 genes for the sketching (parameter $M = 50$).
The results displayed in Figure 5.2 are the average of 100 independent executions of the ReCC algorithm. The dataset used is the same as for the experiments described in chapter 3, that is, a scRNA-seq dataset of 1000 cells composed of 5% of rare Jurkat cells and 95% of abundant 293T cells.

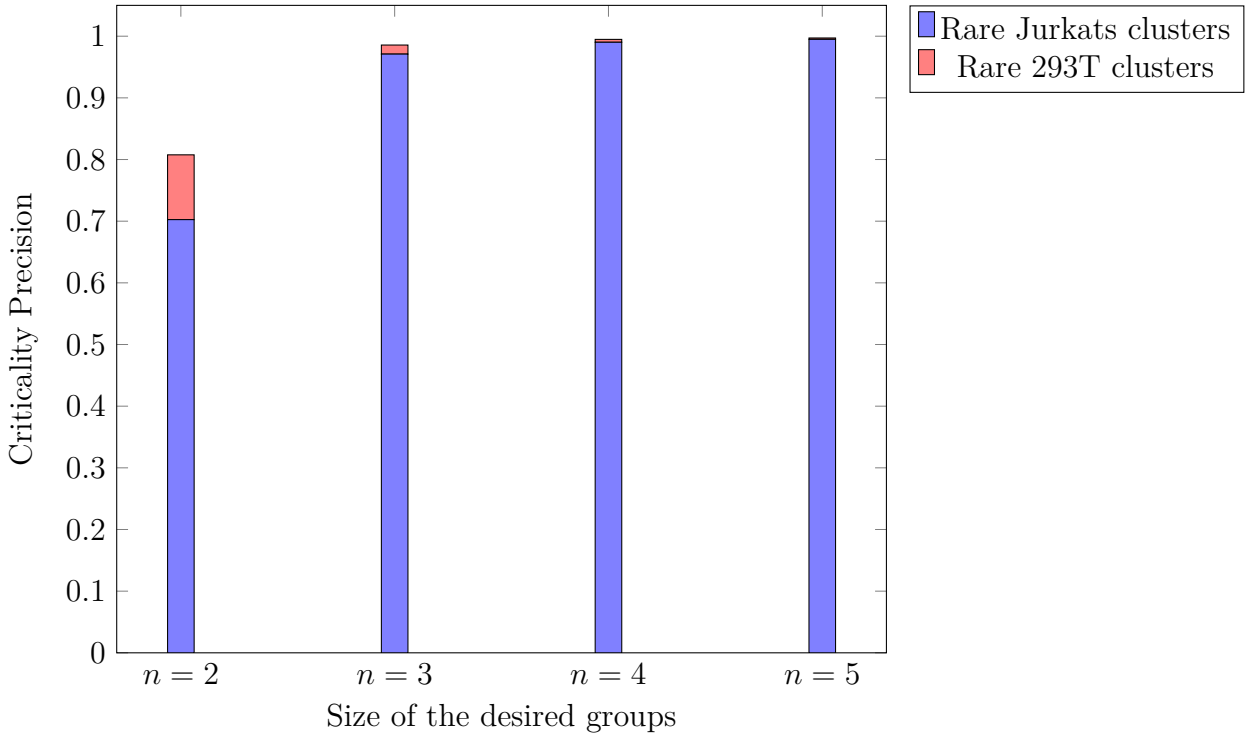

Figure 5.2: Bar Plot of the precision of criticality among groups returned by ReCC according to the size of the groups.

Figure 5.2 presents the precision of the criticality of the k groups returned by ReCC (i.e. the percentage of groups that are both rare and homogeneous). First, it can be observed that for larger group sizes (parameter n), the proportion of critical groups among the clusters returned rises. 99.7% of the groups of size 5 returned were both homogeneous and rare according to the criteria described in section 3.1.1.

Furthermore, Figure 5.2 also shows that the amount of critical 293T clusters tends to decrease when n increases. That can be explained by different factors, first, among the cells labelled as rare by FiRE, 64.29% are of type Jurkat and 35.71% are of type 293T on average. Therefore, when the size of the groups desired gets bigger, the number of possible combinations of 293T groups will also decrease. Secondly, clusters that contain Jurkat cells may have a higher rareness score because they are made of cells coming from the minority population, therefore, their criticality score will also tend to be higher and thus, they will be favoured compared to groups made of 293T. Additionally, as the k groups returned at each level of recursion serve as the foundational groups for the subsequent recursion level, if fewer 293T clusters are returned at recursion level i, less 293T candidate clusters will be constructed and assessed at level $i + 1$, precipitating a snowball effect that nearly eradicates the number of 293T clusters returned.

5.3 Conclusion

This chapter introduced ReCC, an alternative method developed to build clusters of rare cells in contrast to the scoring method developed in chapter 3. Similar heuristics to those provided by MCC2 were integrated into the method, improving the scalability of ReCC. Experiments have shown the effectiveness of the method to build and return critical groups of cells of a given size, within the experimental set-up.

Chapter 6

Conclusion

The goal of this thesis was to develop a method generalising the FiRE algorithm and to observe its impact after being plugged in the beam search of the MCC2 algorithm.

We proposed FiRE-n, an algorithm extending the FiRE algorithm, that assigns rareness scores to groups of n cells from scRNA-seq data. The objective is to attribute a high score to groups formed by rare cells that have a similar gene expression. The results of the evaluation of FiRE-n show that the scores assigned by the proposed method allow the effective classification of rare clusters. Thanks to the penalty applied to the score of groups not deemed homogeneous, FiRE-n manages to filter out most groups made of rare cells coming from different types. However, the FiRE-n method exhibits one major drawback as it requires the user to specify the groups they want to obtain a score for, limiting its use to specific applications.

Following the encouraging results of FiRE-n, we introduced MCC2$\star$, a method that integrates FiRE-n into the MCC2 algorithm as a heuristic to reduce the search space. MCC2$\star$ provides an improvement in both objective value and runtime in comparison to MCC2. The pruning implemented in MCC2$\star$ beam search allows the discard of unfavourable candidates that were selected by MCC2, allowing for an improvement in objective value that appears to be greater for larger populations of cells. MCC2$\star$ also provides a 20% to 25% speed up of the original algorithm.

In addition to FiRE-n and MCC2$\star$, the ReCC algorithm was developed to circumvent FiRe-n's main drawback (i.e. the need to specify the groups to score). ReCC is a beam search based method whose goal is to return the k most critical cell groups of size s. After evaluation, ReCC was shown to be effective at retrieving rare subpopulations of cells of a given size.

Furthermore, additional leads that fall outside the scope of the thesis have been identified.
First, metagenes could be exploited to improve FiRE-n's measure of homogeneity of cells. Metagenes are collections of genes that share common biological functions, chromosomal locations, or regulations [24]. An adaptation of the sketching method of the FiRE algorithm making use of these metagenes could be implemented. A more precise homogeneity metric could then be derived from these sketches taking the functionality of genes into account.
Secondly, in order to decrease ReCC's complexity, a heuristic could be introduced to avoid generating all pairs during the first beam search level. The pairs could be generated similarly to the pairs in MCC2, that is by linking each cell to the n cells that have the closest rareness score. That would allow ReCC to be more efficient on larger datasets.
Thirdly, another lead that could be explored is to have ReCC replace the beam search of MCC2. Instead of keeping the groups with the best objective value for the MicroCellClust problem, the groups with the best criticality score as defined in ReCC could be retained as candidates for the next beam search level. Pruning candidates using the criticality score appears to be promising with MCC2$\star$, therefore adapting the whole beam search could have interesting results in terms of objective value.
Lastly, ReCC could be adapted to be a hierarchical clustering method. Hierarchical clustering allows the clustering of the whole population of cells without any overlay between the different clusters [25]. Indeed, thus far, the same cell could be part of several groups returned by ReCC. Implementing a hierarchical clustering could allow for a complete clustering of the rare cells and for the identification of the type of cells each clusters are made of.

In conclusion, this thesis has shown the potential of refining the FiRE and MCC2 approaches by proposing two new methods, namely FiRE-n and MCC2$\star$. A third method called ReCC was proposed to tackle the problem at hand through clustering rather than classification. The performance of all methods was cautiously evaluated and the results were presented and discussed in detail. Additional workstreams could further improve those results.

Bibliography

[1] D. Jovic, X. Liang, H. Zeng, L. Lin, F. Xu, and Y. Luo, "Single-cell rna sequencing technologies and applications: A brief overview," *Clinical and Translational Medicine*, vol. 12, no. 3, p. e694, 2022.

[2] Y. Xin, J. Kim, H. Okamoto, M. Ni, Y. Wei, C. Adler, A. Murphy, G. Yancopoulos, C. Lin, and J. Gromada, "Rna sequencing of single human islet cells reveals type 2 diabetes genes," *Cell Metabolism*, vol. 24, no. 4, pp. 608–615, 2016.

[3] H. M. Levitin, J. Yuan, and P. A. Sims, "Single-cell transcriptomic analysis of tumor heterogeneity," *Trends in cancer*, vol. 4, no. 4, pp. 264–268, 2018.

[4] M. Guo, Y. Peng, A. Gao, C. Du, and J. G. Herman, "Epigenetic heterogeneity in cancer," *Biomarker research*, vol. 7, no. 1, pp. 1–19, 2019.

[5] G. Plitas, C. Konopacki, K. Wu, P. D. Bos, M. Morrow, E. V. Putintseva, D. M. Chudakov, and A. Y. Rudensky, "Regulatory t cells exhibit distinct features in human breast cancer," *Immunity*, vol. 45, no. 5, pp. 1122–1134, 2016.

[6] V. Y. Kiselev, T. S. Andrews, and M. Hemberg, "Challenges in unsupervised clustering of single-cell rna-seq data," *Nature Reviews Genetics*, vol. 20, no. 5, pp. 273–282, 2019.

[7] A. Jindal, P. Gupta, Jayadeva, and D. Sengupta, "Discovery of rare cells from voluminous single cell expression data," *Nature communications*, vol. 9, no. 1, p. 4719, 2018.

[8] A. Gerniers *et al.*, "Microcellclust: mining rare and highly specific subpopulations from single-cell expression data," *Bioinformatics (Oxford, England)*, vol. 37, no. 19, pp. 3220–3227, 2021.

[9] A. Brazma and J. Vilo, "Gene expression data analysis," *FEBS Letters*, vol. 480, no. 1, pp. 17–24, 2000. Functional Genomics.

[10] R. Lowe, N. Shirley, M. Bleackley, S. Dolan, and T. Shafee, "Transcriptomics technologies," *PLoS computational biology*, vol. 13, no. 5, p. e1005457, 2017.

[11] B. Hwang, J. H. Lee, and D. Bang, "Single-cell rna sequencing technologies and bioinformatics pipelines," *Experimental & molecular medicine*, vol. 50, no. 8, pp. 1–14, 2018.

[12] A. M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner, "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells," *Cell*, vol. 161, no. 5, pp. 1187–1201, 2015.

[13] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, *et al.*, "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets," *Cell*, vol. 161, no. 5, pp. 1202–1214, 2015.

[14] J. Cao, J. S. Packer, V. Ramani, D. A. Cusanovich, C. Huynh, R. Daza, X. Qiu, C. Lee, S. N. Furlan, F. J. Steemers, *et al.*, "Comprehensive single-cell transcriptional profiling of a multicellular organism," *Science*, vol. 357, no. 6352, pp. 661–667, 2017.

[15] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[16] D. Grün, A. Lyubimova, L. Kester, K. Wiebrands, O. Basak, N. Sasaki, H. Clevers, and A. Van Oudenaarden, "Single-cell messenger rna sequencing reveals rare intestinal cell types," *Nature*, vol. 525, no. 7568, pp. 251–255, 2015.

[17] L. Jiang, H. Chen, L. Pinello, and G.-C. Yuan, "Giniclust: detecting rare cell types from single-cell gene expression data with gini index," *Genome biology*, vol. 17, pp. 1–13, 2016.

[18] Z. Wang, W. Dong, W. Josephson, Q. Lv, M. Charikar, and K. Li, "Sizing sketches: A rank-based analysis for similarity search," *SIGMETRICS Perform. Eval. Rev.*, vol. 35, p. 157–168, jun 2007.

[19] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[20] V. Branders, P. Schaus, and P. Dupont, "Mining a sub-matrix of maximal sum," *arXiv preprint arXiv:1709.08461*, 2017.

[21] "10.8. Beam Search — Dive into Deep Learning 1.0.0-beta0 documentation,"

[22] A. Gerniers and P. Dupont, "MicroCellClust 2: a hybrid approach for multivariate rare cell mining in large-scale single-cell data," 12 2022.

[23] P. J. Van Laarhoven, E. H. Aarts, P. J. van Laarhoven, and E. H. Aarts, *Simulated annealing*. Springer, 1987.

[24] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, *et al.*, "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15545–15550, 2005.

[25] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14863–14868, 1998.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**