DTU Bioinformatics and Systems Biology
Department of Health and Technology

Developing a marker-scaled clustering algorithm for improved population identification in single cell cytometry data

Master Thesis

Developing a marker-scaled clustering algorithm for improved population identification in single cell cytometry data

Master Thesis
February, 2024

By
Dimitrios Loukas

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Health and Technology, Brovej, Building 118, 2800 Kgs. Lyngby Denmark

Approval

This thesis has been prepared over six months at the Section for Bioinformatics, Department of Health and Technology, at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Science in Engineering, MSc Eng.

It is assumed that the reader has a basic knowledge in the areas of statistics.

Dimitrios Loukas - s220514

.....

Dimitrios Loukas

.....

11/02/2024

Contents

1 Introduction

The immune system, a complex and dynamic defense mechanism, is fundamental to human health. It consists of two primary parts: the innate and adaptive immunity. Innate immunity acts as the first line of defense, responding rapidly to a wide range of pathogens through physical barriers, and a variety of immune cells including neutrophils, macrophages, natural killer (NK) cells, dendritic cells, and mast cells, along with various soluble factors. In contrast, adaptive immunity involves a more specific and long-lasting response, characterized by the generation of memory cells that remember specific pathogens. This adaptive response is mediated by T and B lymphocytes, including specialized subsets like helper T cells (CD4+), cytotoxic T cells (CD8+), regulatory T cells, memory T and B cells, and plasma cells, which undergo a complex development process involving differentiation and maturation. During their development, immune cells express different markers that indicate their stage of differentiation and function. These markers, known as lineage markers, play an important role in identifying and describing the various types of immune cells. For example, T cells, which are key to adaptive immunity, are identified by the CD3 lineage marker, while B cells are known for markers like CD19 and CD20. The dynamic expression of these lineage markers during cell maturation and differentiation play a major role in the formation of distinct immune cell phenotypes, which helps us understand the different roles immune cells play and how the immune system works as a whole.

Accurate identification of immune cell subsets with lineage markers are important in immunology research. These markers help to track how various immune cells develop, function, and react, improving understanding of immune responses and aiding in diagnosing and treating immune-related disorders. Single-cell cytometry has become an important technique for detailed analysis of individual immune cells, as it offers insights into how cells behave in response to infections or in diseases like cancer. Advances in flow cytometry and mass cytometry (CyTOF) have greatly helped in studying immune cells. These methods allow detailed study of many markers on each cell, providing a full picture of each cell's traits and actions. However, cytometry includes various technical limitations, such as the restricted number of markers that can be simultaneously analyzed, which presents a challenge in obtaining a complete cellular profile. There's a need for sophisticated methods to make sense of the patterns in this data, which includes information from thousands of cells and their various markers.

Traditionally, manual gating has been used to sort cytometry data. This method involves picking out cell groups based on their marker patterns, offering a basic way to categorize cell types. But, manual gating has its limits because it relies on what is already known about markers and can be subjective. It's also less useful for high-dimensional data, especially when trying to spot rare cell types or small differences in cells. These limitations point out the need for better, more objective methods for analyzing cytometry data that can handle its complexity and draw out important findings. This need has driven the creation of automated analysis techniques, which offer a better understanding of complex data from cytometry. Automated analysis methods often use algorithmic clustering, aiming to sort cells based on marker expression profiles more objectively than manual gating. These methods, typically described as unsupervised or semi-supervised, can reveal new cell types or states that might be overlooked in manual analysis. However, a detailed look shows that these methods still rely heavily on human input. Choosing parameters, making sense of results, and the need to repeat the analysis process involve expert judg-

ment. Therefore, while these methods are an improvement over manual gating, they don't completely remove the need for expert knowledge. Combining computational algorithms with expert knowledge in immunology is key for making sense of cytometry data in a way that is meaningful biologically. This joint approach ensures that the results are not only statistically significant but also biologically accurate and relevant [1].

The primary objective of this research is to explore the development of a marker-scaled clustering method designed for single-cell cytometry data analysis. This method aims to address some limitations of both manual gating and existing automated analysis techniques by incorporating biological information. It specifically uses groups of markers defined by immunologists, thereby integrating biological knowledge directly into the data analysis process. This approach attempts to bridge a gap found in current methods, where the reliance on purely computational or algorithmic clustering often overlooks the biological understanding that lies beneath, which is essential for accurate cell population identification. As mentioned before, many automated methods, while advanced, still require expert input for parameter selection and result interpretation. By integrating manually defined marker groups into the clustering process, this method attempts to identify clusters not only statistically significant but also meaningful from an immunological perspective. While the effectiveness of this method in significantly improving cell population identification remains to be validated, its development represents an important step towards more informed and biologically grounded cytometry data analysis.

In this study, the analysis is carried out on two variations of cytometry data: the raw mass cytometry data and two differently marker-scaled datasets. The first marker-scaled dataset incorporates biological insights through expert-defined marker groups, while the second explores an alternative approach using log fold change for defining groups. PhenoGraph and Seurat are performed on these datasets, using k-nearest neighbors (kNN) clustering initially, followed by shared nearest neighbor (SNN) clustering with algorithms like Louvain, Leiden, or SLM to further refine the cluster identification. In the marker-scaled data, Principal Component Analysis (PCA) is used for two main purposes. It first helps in adjusting the weight of each group, ensuring a balanced representation in the analysis. Secondly, PCA, along with Uniform Manifold Approximation and Projection (UMAP), is used for a better visual interpretation of the complex cytometry data. To ensure the most effective clustering, the study uses the Adjusted Rand Index (ARI) to assess the clustering parameters of all three datasets, as well as compare the parameters of both marker-scaled datasets. The latter step helps decide which marker scaling method works best for analyzing cell groups more accurately.

2 Theory

2.1 Immune System

The field of immunology has been shaped by a series of pivotal experiments conducted by scientists over time. These studies collectively contribute to our understanding of the immune system's role in protecting the body from invaders like microbes, viruses, cancer cells, and toxins. The immune system is a complex network of cells, chemicals, and processes responsible for defending various regions of the body, including the skin, respiratory passages, and the intestinal tract. Broadly, the immune system relies on two lines of defense: innate immunity and adaptive immunity.

2.1.1 Innate immunity

Innate or natural immunity includes the inherent defense mechanisms that operate independently of prior exposure to specific antigens. This type of immunity is present from birth and serves to protect the host against injury or infection, regardless of previous encounters with the pathogenic agent. It consists of elements such as physical barriers like the skin and mucous membranes, as well as various defenses that hinder infections, including lysozymes in tears and the highly acidic environment of the stomach. Innate immunity is also characterized by the involvement of specialized cells such as phagocytes and natural killer cells, along with essential proteins like complement and cytokines, all of which play vital roles in preserving the individual's well-being. Table 2.1 summarizes the relevant cell types, highlighting their functions, structures, and membrane markers for comparison [2].

Table 2.1: Cells of innate immunity [2]

Empty Cell	**Function**	**Structure characteristics**	**Membrane marker**
Neutrophils	Phagocytosis, intracellular killing, inflammation and tissue damage	Characteristic nucleus and cytoplasm	CD67
Macrophages	Phagocytosis, intracellular and extracellular killing, tissue repair, antigen presentation for specific immune response	Characteristic nucleus	CD14
Natural killer (NK) cells	Kill infected cells and malignant cells; LAK cells kill transformed cells and malignant cells	Also known as large granular lymphocytes (LGL); activated by IL-2 and IFN to become LAK cells	CD56, CD16
K cells	Recognize antibody-coated targets	Morphologically undefined	Could be NK cells (IgG), macrophages (IgG), eosinophils (IgE) or other cells (IgG)

2.1.2 Adaptive immunity

Adaptive immunity is a specific set of secondary defense responses that occur several days to weeks after exposure to microbial antigens, following the initial innate immune response at barrier systems like the skin or mucous membranes. Unlike innate immune responses, adaptive responses are highly tailored to the antigens of the specific type or species of microbe that triggered them. Furthermore, the immune system 'remembers' these responses. Adaptive immune responses are geared toward eliminating microbes and the toxins or enzymes they produce. Therefore, these responses target molecules that are foreign to the organism and not the organism's own structural or functional molecules. Through the Major Histocompatibility Complex (MHC) system, the adaptive immune system can distinguish foreign molecules from the organism's own. The defenses and mechanisms of adaptive immunity encompass cell-mediated immunity, involving T lymphocytes that combat intracellular pathogens, and humoral immunity, where B lymphocytes combat extracellular pathogens and toxins [3]. In summary, T cells and B cells are the two primary components of adaptive immunity.

Table 2.2: Cells of adaptive immunity [4]

Immune Cell	**Markers**	**Function**
T Cells	CD3, CD4, CD8, CD69, CD25, PD-1, TIM-3, LAG3	Recognize antigens and regulate immune responses.
B Cells	CD19, CD20, CD21, CD22, CD23, CD40, IgM, IgD	Produce antibodies and activate T cells.
Dendritic Cells (DCs)	Siglec-H, CD317, CD11c, HLA-DR, CD1c, CD141, XCR1, CLEC9A	Initiate adaptive immunity by presenting antigens to T cells.
Myeloid-Derived Suppressor Cells (MDSCs)	CD11b, HLA-DR (lack), CD14 (monocytic), CD15 (polymorphonuclear)	Inhibit immune responses by suppressing T cell activation and creating an immunosuppressive environment.

The following sections will further explore the different types of cells that make up the immune defenses.

2.1.3 Immune Cells

The immune system, as discussed before, is a complex network of specialized cells and molecules, playing a pivotal role in defending the body against infections and diseases. At the heart of this system lies the hematopoietic process, responsible for producing the diverse array of immune cells that safeguard our health. Hematopoietic stem cells (HSCs), found primarily in the bone marrow, are incredibly versatile. There are several types of HSCs, each with distinct properties:

- Multipotent Hematopoietic Stem Cells (HSCs): These cells have the ability to transform into many different types of blood cells, making them the primary source of immune cells.
- Multipotent Progenitors: These progenitors are one step further down the specialization path. They have a broader range of options than multipotent HSCs but are not as versatile as their predecessors.
- Unipotent Progenitors: These are the specialists of the immune cell production process. Once they've committed to a specific lineage, such as lymphoid-primed

multipotent progenitors (LMPPs), they are dedicated to producing a particular type of immune cell.

The hematopoietic process is a complex passage that involves a hierarchy of stem and progenitor cells, with multipotent hematopoietic stem cells (HSCs) playing a central role. These cells, among others, contribute to the generation of various immune cells. As this differentiation progression unfolds, HSCs give rise to multipotent progenitors, which offer a broader range of options but are not as versatile as their predecessors. Before reaching their final specialized forms, these cells undergo maturation processes. The hematopoietic process branches into two main paths: lymphoid and myeloid lineages, as shown in Figure 2.1. Lymphoid progenitors specialize in producing a variety of immune cells, including natural killer (NK) cells, B cells, and T cells, each with its unique role in the immune response. Myeloid progenitors, on the other hand, give rise to a diverse array of cells critical to the innate immune response. These include dendritic cells, neutrophils, eosinophils, basophils, monocytes, and macrophages. Each of these cell types plays a crucial role in the body's defense against infections and other threats.

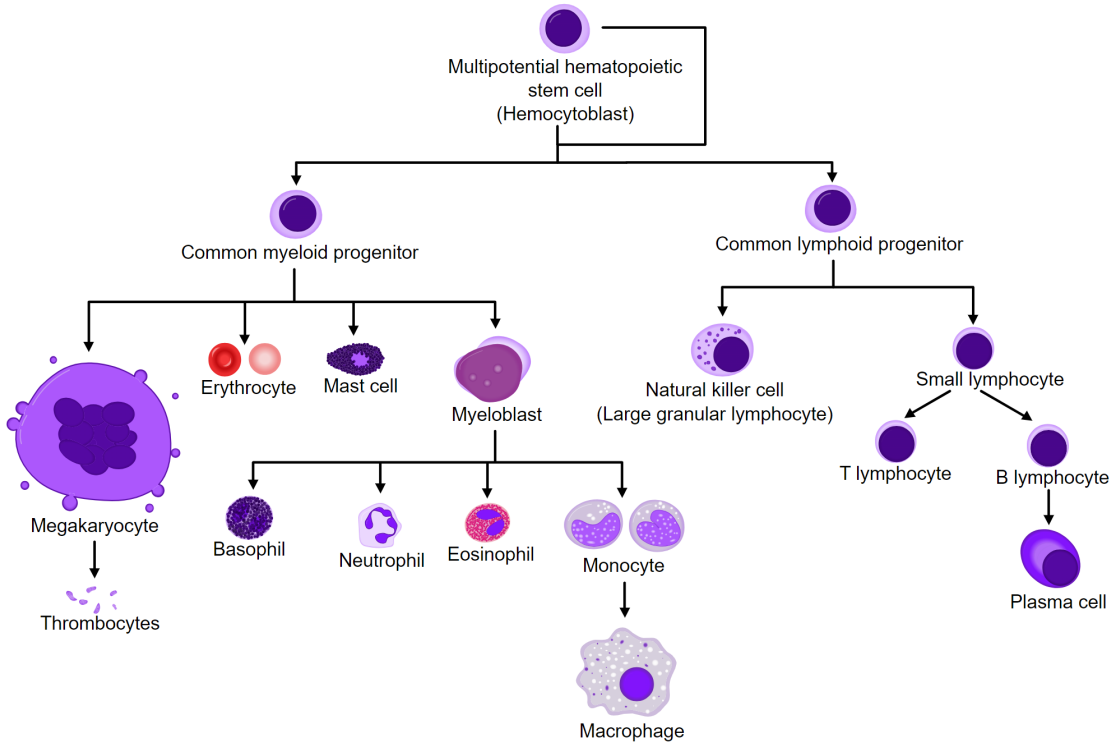


Figure 2.1: Overview of human haematopoiesis [5].

2.1.4 Lineage markers and activation markers

In immunophenotyping, immune cells are identified and categorized using specific markers (Cluster of Differentiation), including both lineage markers and those indicating functional states. Lineage markers help categorize cells into distinct populations, such as CD3, CD4, and CD8 for T cells, CD19 and CD20 for B cells, CD14 and CD11b for monocytes, and CD56 and CD161 for NK cells. Beyond lineage identification, other markers are used to characterize each cell population's activation and physiological state. For instance, CD69, CD25, and CD62L indicate activation, CD45RO and CD27 signal memory status and CCR7, CCR5, CXCR4, and CCR6 are chemokine receptors. Intracellular markers like FoxP3 for Treg cells, cytokines (e.g., IFN- , TNF- , IL-2 for TH1 cells), proliferation markers (e.g., Ki67, CFSE), and antigen-specific markers like MHC Tetramers also play a

crucial role in understanding the functional dynamics of immune cells.

In immunophenotyping, it is essential to understand that immune cells display a variety of CD markers. The specific combination of these markers, rather than a single marker, defines the cell's identity and function. Some CD markers are more prominently expressed, serving as primary identifiers for certain cell types. Others, while less pronounced, are still important for more detailed classification. For example, T cells are generally identified by CD3, but the additional presence of CD4 or CD8 markers classifies them into helper or cytotoxic T cells, respectively. This concept of a combined marker profile is central in immunophenotyping, allowing for a more precise identification and understanding of the roles of different immune cells.

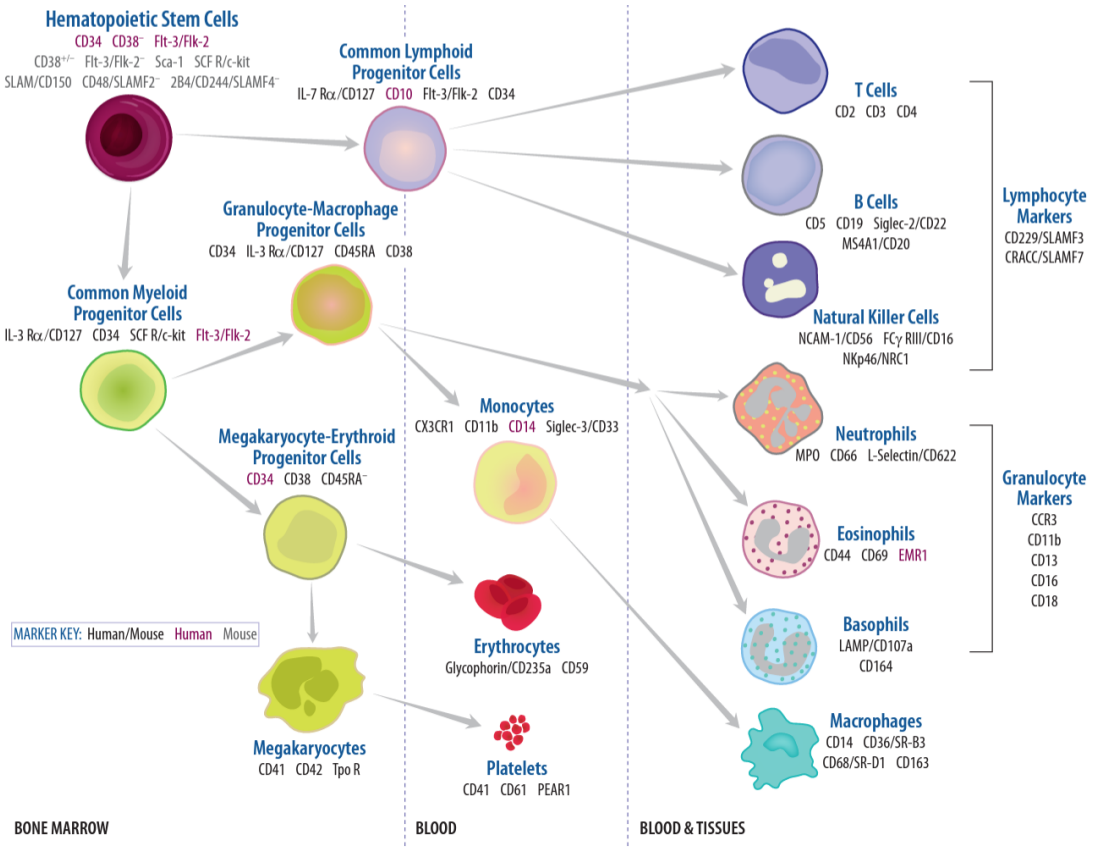


Figure 2.2: Hematopoietic Stem Cell Lineage-specific Markers. The illustration depicts a model of the hierarchy of hematopoietic stem cell differentiation and presents early lineage-committed and cell-specific markers that can be used to differentiate between immune cell types. [6].

2.2 Cytometry

Cytometry is a scientific method used to study cells and their components. It allows for the analysis of the physical and chemical characteristics of cells, such as size, structure, and protein content. It is also essential for studying immune cell differentiation by identifying lineage markers accurately.

Flow cytometry and mass cytometry are the main forms of cytometry. Flow cytometry analyzes physical and chemical attributes of cells as they move through a fluid stream under light exposure, typically from a laser. This method can examine thousands of particles per second to provide information about each cell's size, texture, and the presence of specific biomarkers via fluorescence-based detection.

Mass cytometry is an advance from traditional cytometric methods. It uses metal isotope-labeled antibodies instead of fluorescent labeling, allowing for simultaneous measurement across multiple parameters without the constraints of spectral overlap often seen in fluorescence-based techniques. Both approaches are being thoroughly examined in the following parts of this section

2.2.1 Flow Cytometry

Flow cytometry is a high-speed technology for analyzing individual cells or particles in a liquid solution as they pass by lasers. It measures both visible light scatter and fluorescence, providing valuable information about cell size and complexity. This technology can analyze mixed cell populations from various sources, including blood, bone marrow, lymph nodes, spleen, and tumors. Moreover, it goes beyond analysis, it can also sort cells for more in-depth study. Flow cytometry has widespread applications in fields like immunology, virology, molecular biology, cancer research, and infectious disease monitoring. It's particularly useful for studying the immune system's response to diseases like cancer and infections.

In traditional flow cytometry, three main systems - fluidics, optics, and electronics - work together to analyze cells. The fluidics system focuses cells in a sheath fluid into a laser beam, where they emit light scatter and fluorescence. The optical system, consisting of lasers and detectors like photomultiplier tubes (PMTs), collects this light. Dichroic and bandpass filters differentiate the wavelengths, allowing specific fluorochromes to be measured. Forward Scatter (FSC) and Side Scatter (SSC) provide data on cell size and internal complexity, while fluorescence measurements identify specific cellular characteristics, like fluorescently labeled proteins [7].

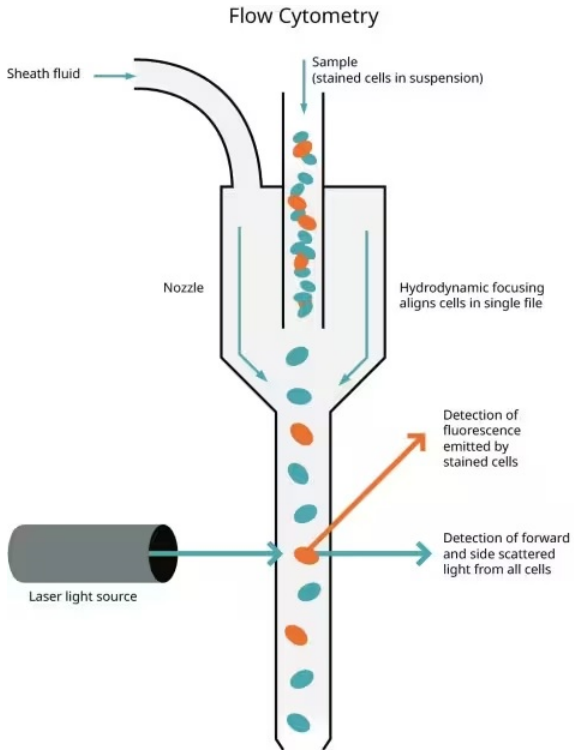


Figure 2.3: Flow cytometry diagram giving an overview of the flow cytometer. Sheath fluid focuses the cell suspension, causing cells to pass through a laser beam one at a time. Forward and side scattered light is detected, as well as fluorescence emitted from stained cells [8].

Fluoresense markers

Flow cytometry uses fluorescent markers to label and analyze specific components within samples. These markers are highly effective at binding to a range of targets, including lineage markers, proteins, antigens, and genetic material like DNA/RNA. However, the number of detectable fluorochromes at one time is limited by instrument capacity and spectral overlap challenges. This often requires focusing on one cell type at a time or using different color sets for multi-cell analysis.

2.2.2 Mass Cytometry

Mass cytometry is a technique that combines flow cytometry and elemental mass spectrometry. This combination, known as Cytometry by Time-Of-Flight (CyTOF), enhances the ability to measure multiple cellular characteristics simultaneously. While traditional flow cytometry uses fluorophores (light-emitting molecules) as markers to gauge molecular expression, their overlapping emission spectra can complicate the analysis, especially in experiments measuring numerous parameters. The workflow of mass cytometry is shown in Figure 2.4.

The key advantage of mass cytometry lies in its use of elemental mass spectrometry. This method distinguishes isotopes based on their atomic weights with high precision, allowing for the simultaneous analysis of many more cellular features compared to fluorescence-based methods. In mass cytometry, probes such as antibodies are linked to unique, stable heavy-metal isotopes instead of fluorophores. The presence and quantity of these metal isotopes in a specific mass channel serve as indicators of molecular expression, minimizing signal overlap and enhancing the accuracy of the analysis.

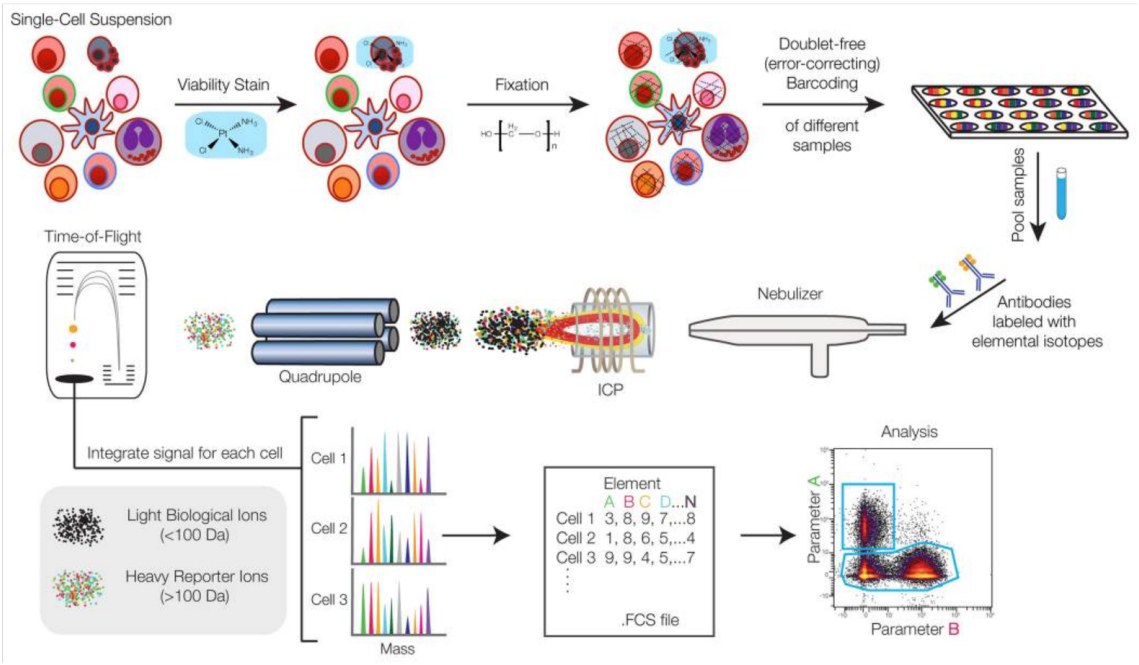


Figure 2.4: Workflow of a typical mass cytometry experiment [9].

2.3 Manual Gating

Manual gating is an important technique in cytometry for identifying and categorizing specific cell populations based on their distinct characteristics. This method was widely used before the development of automated clustering algorithms and remains significant in both flow and mass cytometry for precise cell population analysis. The process involves

interpreting data represented in scatter plots and histograms. Researchers establish gates or boundaries within these plots to isolate cells that exhibit specific features from the broader cell population, enabling targeted analysis [7].

Flow cytometry categorizes cells based on light scatter and fluorescence intensity. FSC correlates with cell size, while SSC indicates internal complexity. Similarly, in mass cytometry (CyTOF), manual gating also categorizes cells, but instead uses metal isotopes as markers, allowing for more parameters to be analyzed simultaneously due to the absence of spectral overlap.

A researcher uses scatter plots or histograms from cytometry data to set boundaries called 'gates' in order to isolate cells with specific characteristics. This involves selecting areas on the plot that represent the desired cell population, based on their light scatter and fluorescence in flow cytometry, or marker expression in mass cytometry. The gated cells are then further analyzed or sorted for subsequent experiments.

Manual gating has the advantage of precise and adaptable cell selection. However, it introduces subjectivity and can lead to variability in results because each operator may interpret and use the method differently.To address these issues, newer methods like automated gating and machine learning are employed. Automated gating uses predefined algorithms for cell population analysis, reducing human error. Machine learning, as applied in Seurat and PhenoGraph, uses data-driven algorithms for advanced cell classification and clustering.

2.4 Principal Component Analysis

Principal Component Analysis is a widely used multivariate statistical technique in the field of data analysis. It is a dimensionality reduction method, where the goal is to find a lower dimensional representation of a high dimensional data set, while preserving as much information as possible in the process. PCA builds on tools from linear algebra, and most importantly the Singular Value Decomposition. In the upcoming section, the focus will shift to an in-depth exploration of PCA. However, before diving into the complexities of this method, the discussion will begin with an introduction to fundamental concepts in linear algebra, including projections and subspaces.

2.4.1 Projections and subspaces

In the context of an M-dimensional vector space, it is defined as the collection of M-dimensional vectors represented by the equation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix}, \tag{2.1}$$

denoted as $\mathbf{x} \in \mathbb{R}^M$. It is important to note that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^M$ and real numbers a, b, the vector $\mathbf{z} = a\mathbf{x} + b\mathbf{y}$ also resides in $\mathbb{R}^M$. This property signifies that $\mathbb{R}^M$ is closed under linear transformations. Additionally, the transpose of a vector $\mathbf{x}$, denoted as $\mathbf{x}^T$, involves flipping the vector along its diagonal:

$$\mathbf{x}^\intercal = \begin{bmatrix} x_1 & x_2 & \cdots & x_M \end{bmatrix}$$

Subspaces

A subspace V in a vector space $\mathbb{R}^M$ can be a line, plane, or their higher-dimensional equivalents. The crucial characteristic of a subspace is its closure under linear transformations. Therefore, V is a collection of M-dimensional vectors in $\mathbb{R}^M$ where if $x, y \in V$, then

$$ax + by \in V,$$

for any values of a and b.

The span of a set of vectors $x_1, \ldots, x_n \in \mathbb{R}^M$ is defined as

$$V = \text{span}(x_1, \ldots, x_n) = \{z \mid z = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n\},$$

where $a_1, \ldots, a_n$ are arbitrary.

For instance, considering the span of a single vector:

$$V_1 = \text{span}\left(\begin{bmatrix}1\\ \frac{1}{2}\end{bmatrix}\right),$$

corresponds to all vectors that can be written as $\begin{bmatrix}x\\ y\end{bmatrix} = a_1 \begin{bmatrix}1\\ \frac{1}{2}\end{bmatrix}$ (for arbitrary a_1), forming a line.

A more elaborate example is a plane $V_2 = \text{span}(w_1, w_2)$ where

$$w_1 = \begin{bmatrix}1\\ -1\\ 0\end{bmatrix} \quad \text{and} \quad w_2 = \begin{bmatrix}1\\ 0\\ 0.3\end{bmatrix}.$$

The length of a vector x is defined as

$$\|x\| = \sqrt{x^T x} = \sqrt{x_1^2 + x_2^2 + \ldots + x_M^2},$$

and two vectors x, y are orthogonal if $x^T y = 0$.

A set of vectors $x_1, \ldots, x_n$ is linearly independent if

$$0 = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \implies a_1 = a_2 = \ldots = a_n = 0.$$

The basis of a subspace V is a set of vectors $v_1, \ldots, v_n$ such that

$$\text{span}(v_1, \ldots, v_n) = V,$$

and $v_1, \ldots, v_n$ are linearly independent. The dimension of the subspace is denoted by n.

This is often practical when the basis vectors have a unit length and are mutually perpendicular. An orthonormal basis satisfies $v_i^T v_j = 0$ for $i \neq j$, and it's always possible to discover an orthonormal basis for a subspace.

Projection onto a Subspace

Consider any vector x in a subspace V with an orthonormal basis $v_1, \ldots, v_n$. In the context of a subspace, x can be expressed as a linear combination of the basis vectors:

$$x = a_1 v_1 + a_2 v_2 + \ldots + a_n v_n,$$

where $a_1, a_2, \ldots, a_n$ are appropriate coefficients.

The importance of an orthonormal basis lies in its ability to simplify the computation of these coefficients. To calculate a_i, for instance, one can multiply both sides of the equation by v_i^T:

$$a_i = x^T v_i.$$

If x is not initially in V, the numbers $b_1 = x^T v_1, b_2 = x^T v_2, \ldots, b_n = x^T v_n$ can still be computed. Then, a new vector x_0 can be constructed, lying in V:

$$x_0 = b_1 v_1 + \ldots + b_n v_n.$$

x_0 represents the point in V closest to x, and it's referred to as the projection of x onto V. The coordinates of x in the subspace V are denoted by the n-dimensional vector:

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

This vector b differs from the original M-dimensional space that x belongs to.

Matrix Representation

If we gather the basis vectors $v_1, v_2, \ldots, v_n$ in a matrix V:

$$V = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix},$$

the coordinates of x in the space V can be expressed as:

$$b^T = x^T V.$$

Principal Component Analysis

In a situation where there are N observations, represented as $x_1, x_2, \ldots, x_N \in \mathbb{R}^M$, each with M features or dimensions. In the context of Principal Component Analysis, which is a technique for reducing dimensionality, the goal is to choose an integer n and find a new n-dimensional representation $b_1, b_2, \ldots, b_N \in \mathbb{R}^n$. Here, n is selected such that $n \leq M$, and each b_i aims to capture the essence of its corresponding observation x_i.

To accomplish this transformation from an M-dimensional space to an n-dimensional space, a basic approach involves choosing an orthonormal basis $v_1, \ldots, v_n$ for an n-dimensional subspace V. Then, each b_i is determined as the projection of x_i onto V. To maintain consistency in this projection even with the addition of a constant, the mean is subtracted from each x_i. The general PCA algorithm is as follows:

- Compute the mean $m = \frac{1}{N} \sum_{i=1}^{N} x_i$.
- Subtract the mean from x_i: $x_i' = x_i - m$ (and collect all x_i' into an $N \times M$ matrix X').
- Project onto V: $b_i^T = (x_i')^T V$, where $V = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix}$.

The crucial factor is the choice of the projection matrix V. When n = 1 and projecting onto a line, the objective is to maximize the variance of the projected data. This involves selecting v_1 as the vector that maximizes the dispersion of the projected data. So, we seek for:

$$v_1 = \arg\max_{\|v\|=1} W,$$

where $W = N \times \text{Variance}[b_1, \ldots, b_N]$. The variance is computed as $\text{Variance}[b_1, \ldots, b_N] = \frac{1}{N} \sum_{i=1}^{N} (b_i - \bar{b})^2$, with $\bar{b} = \frac{1}{N} \sum_{i=1}^{N} b_i$.

In the following section, we will address the maximization problem and extend our consideration to the general case where $n > 1$.

Maximizing the variance with respect to the first principal component

First, we observe that the mean of the projected data, denoted as $\bar{b}$, is zero since we have subtracted the mean from $\boldsymbol{X}$:

$$\bar{\boldsymbol{b}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{b}_i = \frac{1}{N}\sum_{i=1}^{N} \hat{\boldsymbol{x}}_i^T \boldsymbol{v}_1 = \frac{1}{N}\left(\sum_{i=1}^{N} \hat{\boldsymbol{x}}_i\right)^T \boldsymbol{v}_1 = \frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{x}_i - \boldsymbol{m}\right)^T \boldsymbol{v}_1 = 0. \tag{2.2}$$

If we define the matrix $\boldsymbol{S} = \boldsymbol{X}^T\boldsymbol{X}$, we can rewrite the variance $\boldsymbol{W}$ as:

$$\boldsymbol{W} = \sum_{i=1}^{N} \boldsymbol{b}_i^T \boldsymbol{b}_i = \sum_{i=1}^{N} (\hat{\boldsymbol{x}}_i^T \boldsymbol{v}_1)^T \hat{\boldsymbol{x}}_i^T \boldsymbol{v}_1 = \sum_{i=1}^{N} \boldsymbol{v}_1^T \hat{\boldsymbol{x}}_i \hat{\boldsymbol{x}}_i^T \boldsymbol{v}_1 = \boldsymbol{v}_1^T \hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} \boldsymbol{v}_1 = \boldsymbol{v}_1^T \boldsymbol{S} \boldsymbol{v}_1. \tag{2.3}$$

For $\boldsymbol{v_1}$ to be an orthonormal basis, it has to have a norm of 1, i.e., $\|\boldsymbol{v_1}\|_2 = \boldsymbol{v_1}^T\boldsymbol{v_1} = 1$. To achieve the maximization of the equation above under this constraint, we can introduce the Lagrangian multiplier λ and maximize the Lagrangian:

$$\mathcal{L} = \boldsymbol{W} + \lambda(1 - \|\boldsymbol{v_1}\|^2) = \boldsymbol{v_1}^T(\boldsymbol{S} - \lambda \boldsymbol{I})\boldsymbol{v_1} + \lambda. \tag{2.4}$$

Taking derivatives with respect to λ and $\boldsymbol{v_1}$, we obtain:

$$\frac{\partial}{\partial \lambda}\mathcal{L} = 1 - \boldsymbol{v_1}^T\boldsymbol{v_1} = 0, \tag{2.5}$$

$$\nabla_{\boldsymbol{v_1}}\mathcal{L} = (\boldsymbol{S} - \lambda \boldsymbol{I})\boldsymbol{v_1} = 0. \tag{2.6}$$

From the first equation, we observe that $\boldsymbol{v_1}$ should be normalized. The second equation can be rewritten as:

$$\boldsymbol{S}\boldsymbol{v_1} = \lambda \boldsymbol{v_1}. \tag{2.7}$$

This implies that v_1 ought to be an eigenvector of S with eigenvalue λ. We select v_1 as the eigenvector of S associated with the largest eigenvalue, as it maximizes the extent of dispersion.

This solves the case $n = 1$. For $n \geq 2$, the procedure is similar but requires more effort. First, we collect the b_i's into an $N \times n$ matrix

$$\boldsymbol{B}^T = [\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_N}]^T.$$

The projection can then be written as

$$\boldsymbol{B} = \hat{\boldsymbol{X}}\boldsymbol{V},$$

$$\text{where} \boldsymbol{V} = [\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_n}].$$

Define the Frobenius norm $\boldsymbol{W} = \|\boldsymbol{B}\|_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{n} \boldsymbol{B}_{ij}^2$. To maximize $\boldsymbol{W}$, ensure $\boldsymbol{v_i}^T\boldsymbol{v_j} = 0$ for $i \neq j$, as we are looking for an orthonormal basis. Formally:

$$\boldsymbol{v_1}, \ldots, \boldsymbol{v_n} = \text{The } n \text{ orthonormal vectors that maximize } W.$$

Singular Value Decomposition and PCA

The Singular Value Decomposition offers a convenient approach to calculate the n eigenvectors associated with the n greatest eigenvalues. When given any $N \times M$ matrix X, the SVD calculates three matrices:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_M \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}, \quad \boldsymbol{U} = \left[\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_N\right], \quad \boldsymbol{V} = \left[\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_M\right],$$

such that

$$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = X$$

and $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_M$, known as the singular values of X. Note that these conditions imply $\boldsymbol{v}_i^T\boldsymbol{v}_j = 0$ if $i \neq j$ and 1 otherwise; in other words, the columns of $\boldsymbol{V}$ are orthonormal.

These matrices fulfill the decomposition $X = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where the singular values are sorted in decreasing order. The first n principal components are represented as $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$, and the projections of observation i onto this subspace are denoted by $\boldsymbol{b}_i = \boldsymbol{x}_i\boldsymbol{V}_n$.

The PCA Algorithm

The PCA algorithm, applied to a matrix X, involves the following steps:

1. Subtract the mean: $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{m}$, where $\boldsymbol{m} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$.
2. Divide by standard deviation (Optional): $\boldsymbol{x}'_{ij} = \frac{\boldsymbol{x}_{ij}}{s_k}$, where $s_k = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\hat{\boldsymbol{x}}_{ik})^2}$.
3. Compute the SVD: $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{X}'$.
4. The n first principal components are $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$, and coordinates of observation i when projected onto the subspace spanned by the first n principal components are given by $\boldsymbol{b}_i^T = \hat{\boldsymbol{x}}_i^T\boldsymbol{V}_n$ or alternatively $\boldsymbol{B} = \hat{\boldsymbol{X}}\boldsymbol{V}_n$, where $\boldsymbol{V}_n = \left[\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\right]$.

The optional step of centering and normalizing each attribute is included to account for different scales among attributes.

Variance Explained by PCA

The vectors $\boldsymbol{b}_i$ in the matrix $\boldsymbol{B}$ correspond to the coordinates of vector $\boldsymbol{x}_i$ upon projection onto the n-dimensional subspace. To retrieve vector $\boldsymbol{x}_i$ in the original space, we can utilize the following formula:

$$\boldsymbol{x}'_i = \boldsymbol{V}_n\boldsymbol{b}_i = b_{i1}\boldsymbol{v}_1 + \ldots + b_{in}\boldsymbol{v}_n.$$

The variance explained by the n first principal components is computed using the Frobenius norm[10]:

$$\text{Variance Explained} = \frac{\|\boldsymbol{X}'\|_F^2}{\|\hat{\boldsymbol{X}}\|_F^2} = \frac{\sum_{i=1}^{n}\sigma_i^2}{\sum_{i=1}^{M}\sigma_i^2} \tag{2.8}$$

2.5 Uniform Manifold Approximation and Projection - UMAP

UMAP is a method that involves manifold learning and dimension reduction, incorporating concepts from topological data analysis. It offers a broad approach to manifold learning and dimension reduction while also offering specific implementations.

To start understanding UMAP, it is necessary to have some mathematical knowledge from algebraic topology and topological data analysis. This will establish a fundamental algorithm that performs well in theory but has limitations in practical application. The next stage involves incorporating basic Riemannian geometry to align real-world data with the assumptions of the topological data analysis algorithm. However, this introduces additional complexities that can be addressed through advanced mathematical techniques and fuzzy logic. By integrating these components with a novel approach for identifying a low-dimensional representation more aligned with the new data structures, we arrive at the core UMAP algorithm.

Topological Data Analysis and Simplicial Complexes

Simplicial complexes are a method for creating topological spaces using basic combinatorial elements. This simplifies handling the continuous geometry of topological spaces by reducing it to relatively straightforward combinatorics and counting. This approach is essential for the overall strategy in topological data analysis, especially dimension reduction.

The initial step involves using simple building blocks called simplices. Geometrically, a simplex provides a very basic way to create an n-dimensional object. A n-dimensional simplex is referred to as an n-simplex, and it is generated by taking the convex hull of (n+1) independent points. Therefore, a 0-simplex represents a point, while a 1-simplex forms a line segment between two zero simplices; further iterations include forming triangles or tetrahedrons with increasing dimensions through this uncomplicated process, as shown in Figure 2.5.

Figure 2.5: Low dimensional simplices.

The underlying structure is relatively straightforward, where a *-simplex can be seen as an arbitrary set of *+1 objects with faces determined by appropriately sized subsets. These simplices are the basic building blocks that, when glued together in what is called a ”simplicial complex,” can form various topological spaces. A simplicial complex consists of these simplices such that any face of any simplex in the complex is also part of the complex and the intersection of any two simplices in it shares a common face.

To apply these theoretical tools from topology to finite sets of data points, we'll look at constructing a Čech complex from an open cover of a topological space. An open

cover essentially represents the entire space through its family of sets whose union covers it completely. Utilizing this information allows for converting this into a combinatorial representation known as 0-simplices for each set in the cover; creating 1-simplices between two such sets if they intersect non-trivially; generating 2-simplices between three such sets if their triple intersection exists; and so on. Even though this process seems simple—by just examining intersections—the background theory provides assurances about how well this method captures meaningful representations corresponding to the actual topology itself. The quality and precision offered by finer covers play essential roles, emphasizing that despite its simplicity, this approach presents much about topology accurately.

A crucial next step is to comprehend how to implement this procedure with a limited set of data points. If we posit that the data points are derived from an underlying topological space, then in order to infer the topology of that space, it becomes necessary to create an open cover for it. In cases where our data actually exists within a metric space (allowing us to measure distances between points), one method for approximating an open cover involves creating spheres around each data point using a fixed radius. Given that we only have finite samples and not direct access to the topological space itself, there's no certainty whether this truly constitutes an open cover; however, considering these challenges as constraints on our approximation might be reasonable. An added benefit is that associated Čech complex will include a 0-simplex for each data point.

The decision to adopt an abstract, topological method for creating a neighborhood graph and embedding it in lower-dimensional space is driven by two main factors. Firstly, this approach offers a strong theoretical basis for our methodology, ensuring an accurate representation of the underlying topological structure. While creating a neighborhood graph and arranging it in lower-dimensional space may seem intuitive and feasible from a computational standpoint, it lacks the inherent motivation derived from capturing the data's intrinsic topological properties. Secondly, this abstract topological framework allows for a more general approach, addressing challenges associated with other algorithms. Although the final computational process is relatively simple, understanding the significance of various manipulations is essential for gaining a comprehensive understanding of the algorithm beyond mere execution.

Adapting to Real World Data

The outlined theoretical framework mentioned above, thoroughly justifies the effectiveness of using a neighborhood graph-based approach to capture manifold structures during dimension reduction. Nevertheless, translating this theory into practical implementation presents challenges. One significant challenge, involves the precise selection of an appropriate radius for the open cover's constituent balls. This decision is crucial because opting for a radius that is too small leads to disconnected components within the simplicial complex, while selecting a radius that is too large results in high-dimensional simplices that do not accurately represent the manifold structure. Addressing this difficulty necessitates careful consideration in striking a balance.

Part of this dilemma can be attributed to the Nerve theorem, which supports our approach by asserting homotopy equivalence between the simplicial complex and the union of the cover. In practical terms, when dealing with finite data at certain radii, these covers fail to encompass all parts of the envisioned underlying data manifold - resulting in disconnected components . On other occasions where there are dense clustering regions among data points , overly comprehensive covers lead unnecessarily high-dimensional shapes.In an ideal scenario , determining an appropriate raduis would assume uniformity across distribution but real world datasets hardly conform; thus posing challanges on how to set up reasonable

boundaries.

To tackle these complications,best alternative focuses on presuming uniform distribution across manifolds then delve deeper into understanding their implications.Cases whereby unequal distributions occur,distances notions vary adding complexity.Placing such assumptions enables localization- computing distance against Riemannian geometry localy.A rudimentary result:- choosing ball raius one regarding point locality.Such strategies comply with traditional handeled algorithms basedon k-neighbors graphs serving instead fixed-radius callinear connectivity.Connexitivity hinges upon 'k' value affecting estimation accuracy: larger values enhancing broad precision counterpart smaller scale targets fine details capturing.

The development of a Riemannian metric based on local assumptions offers additional benefits, including the capacity to assign weights to graph edges according to the local metric distances between points. This introduces a fuzzy topology where open sets are not strictly binary, but rather fuzzy values ranging from zero to one. Despite its conceptual appeal, fully grasping this concept requires formalization using mathematical tools from algebraic topology and adapting it for use in metric spaces and fuzzy simplicial sets. Formalizing these concepts mathematically transforms the theoretical framework into a well-defined methodology.

However, applying this approach to real data presents challenges, especially in higher dimensions. Many data points become isolated, which is unexpected for a non-pathological manifold. The absence of this property is identified as local connectivity, meaning that every point on the manifold should have a sufficiently small connected neighborhood. In the context of finite approximations, this implies that no point should be completely isolated; each should connect with at least one other point. Thus, introducing "local" connectivity concept extends (fuzzy confidence) roughly assigned probabilities related events to an individual's closest neighbor. Still, there remains an issue - namely, incompatible metrics locally defined respectively around distinct pairs.

All these challenges prompt us to investigate solutions for addressing such differences. We integrate our previously established mathematical representations, forging a family of weighted complex relationships yielding a single united representation. This approach leverages collective wisdom upon coinciding weight decisions, overcoming any uncertainties caused by existent incompatibilities.

In summary, the path thus marked leads to an additional phase – developing the transition laid out in the pathway, of courses coming up with solutions built over original thoughts leads the way forward, helping guide strategies. One part within the surge is integrating mechanic mathematics. Another side to be counted is generating visuals explaining composite features, interconnections, and emerging low-dimensional perspective. This takes the analytical process further along the journey.

Finding a Low Dimensional Representation

Ideally, our aim is to have the low-dimensional representation maintain a fuzzy topological arrangement that closely resembles the original structure. The first task revolves around establishing how to identify the fuzzy topological low dimension representation, while the second task regards the way we find the best one.

The initial query is essentially resolved by repeating the same process used to identify the fuzzy topological structure of our data. However, this time we will have a low-dimensional representation lying on a very specific manifold instead. This manifold is simply the low-dimensional Euclidean space that we are attempting to embed into. As a result, all

the previous efforts exerted to adjust the concept of distance across the manifold will be irrelevant when dealing with the low-dimensional representation. We explicitly require that the distance on the manifold conforms to standard Euclidean distance relative to the global coordinate system and does not involve varying metrics. This simplifies matters considerably. In order to achieve a good low-dimensional representation, we must optimize the hyper-parameter *min_dist* in the algorithm. This is because the distance to the nearest neighbor is a property that we would like to be globally true across the manifold as we optimize toward a good low dimensional representation.

The second inquiry, is about the matter of how can we discover an effective lower-dimensional representation. This relies on our ability to assess the degree of similarity in terms of fuzzy topological structures. If we have a measure of closeness, we can convert it into an optimization problem to find the low-dimensional representation with the closest fuzzy topological structure. However, if our measure of closeness has various properties, the nature of the optimization techniques we can apply will differ.

In the process of reconciling conflicting weights associated with simplices, we construed these weights as probabilities denoting the likelihood of the existence of each simplex. In comparing the two topological structures, since they both have the same 0-simplices, it is plausible that the comparison involves two vectors of probabilities delineated by the 1-simplices. Since these probabilities are modeled as Bernoulli variables, reflecting the presence or absence of a simplex with parameters following a Bernoulli distribution, employing cross entropy in this context would be fitting.

If the set of all possible 1-simplices is denoted as E, and weight functions are defined such that $w_h(e)$ represents the weight of the 1-simplex e in the high-dimensional scenario, while $w_l(e)$ denotes the weight of e in the low-dimensional situation, then we can calculate the cross entropy as

$$\sum_{e \in E} w_h(e) \log\left(\frac{w_h(e)}{w_l(e)}\right) + (1 - w_h(e)) \log\left(\frac{1 - w_h(e)}{1 - w_l(e)}\right) \tag{2.9}$$

The first term, $w_h(e) \log(\frac{w_h(e)}{w_l(e)}$, creates a strong attraction among the data points and is particularly important in scenarios with many dimensions. This is because this factor will be minimized as $w_l(e)$ becomes as large as possible, which occurs when the distance between the points is minimized.

The second term, $(1-w_h(e)) \log(\frac{1-w_h(e)}{1-w_l(e)})$, creates a repulsive interaction between the ends of e when $w_h(e)$ is small. This is due to the minimization of this term by making $w_l(e)$ as small as possible.

On the whole, this method of pull and push, controlled by the edge weights in the topological layout of the high-dimensional data, will enable the low-dimensional representation to stabilize into a state that fairly accurately reflects the general topology of the original data.

Bringing together these components allows us to develop the UMAP algorithm. The initial stage involves creating a fuzzy topological model, following the described methodology. Subsequently, the focus is on refining the low-dimensional representation to closely match the original fuzzy topological model through cross-entropy measurement [11].

2.6 K-Nearest Neighbors

K-Nearest Neighbor (KNN) is a versatile learning algorithm used for classification and regression tasks. However, it relies on the concepts of proximity and similarity, which have inherent limitations. KNN's key principle is its simplicity and effectiveness in making predictions based on the similarity of new data points to existing data. However, the choice of the hyperparameter 'k' becomes crucial as the dataset size increases: a smaller 'k' captures fine-grained details but may be sensitive to noise, while a larger 'k' provides a smoother decision boundary but might not capture local complexities [12]. KNN can assist in identifying cell populations within single cell cytometry data. By finding the 'k-nearest neighbors' for each cell, based on their marker profiles, you can group cells with similar marker expressions into distinct populations.

KNN algorithm

After defining a distance metric, such as *Euclidean*, *Manhattan*, or *Cosine*, the KNN algorithm computes distances between data points. This process identifies the 'k-neighbors,' which are the nearest data points from the dataset for any new data point. For classification, KNN uses the most frequently occurring class labels among these k-neighbors. In regression tasks, it calculates the average of the values associated with the k-neighbors to predict numerical values [12].

Enhanced Clustering through Graph-Based Integration

Graph-based clustering approaches, like Shared Nearest Neighbor (SNN) methods, introduce a new dimension to cell population identification. They construct a graph representation of the data, where nodes correspond to individual cells, and edges represent similarities or shared nearest neighbors between cells. This graph captures intricate marker profile interactions more comprehensively. Combining KNN with SNN techniques can significantly enhance clustering in single cell cytometry data. Initially, KNN identifies the nearest neighbors for each cell based on marker profile similarity. Then SNN techniques are applied to create a graph representing the interconnectivity of cells within the dataset.

2.7 Shared Nearest Neighbor

Shared Nearest Neighbor (SNN) clustering is based on the principle that the similarity between two points is enhanced by their mutual proximity to a set of other points. This concept was introduced by Jarvis and Patrick [13]. For example, if two points, A and B, are both near a third set of points, C, their similarity is considered stronger because it's 'confirmed' by their shared closeness to C.

Building the SNN Graph

The construction of the SNN graph starts with a proximity matrix. In the Jarvis-Patrick scheme, a link is created between two points, p and q, if they are each within the k closest neighbors of each other, a process known as k-nearest neighbor sparsification. The weight of the links in the SNN graph can be calculated as the count of shared neighbors or by a weighted formula considering the rank order of these neighbors in each point's list. The weight of a link between two points i and j is calculated using the formula:

$$\text{str}(i, j) = \sum (k + 1 - m) \times (k + 1 - n)$$

where m and n represent the positions of a shared neighbor in the neighbor lists of i and j.

Clustering Algorithm

After building the SNN graph, the algorithm prunes it by removing all edges with weights below a defined threshold. This step isolates significant shared connections and discards

weaker ones. The final clusters are identified as the connected components in the pruned SNN graph, representing groups of points with a shared neighborhood, providing a context-rich grouping.

Comparison with kNN

SNN clustering differs from kNN as it assesses the shared neighborhood structure rather than just relying on the proximity of a point to its immediate neighbors. This approach is useful in high-dimensional spaces where direct distance measurements may be less reliable due to the curse of dimensionality [14].

2.8 PhenoGraph clustering

This section covers the key principles and methods of PhenoGraph, explaining how it identifies distinct cellular states based on phenotypic traits.

The Nearest-Neighbor Graph Approach

Cell differentiation is a complex process that involves detecting density patterns within high-dimensional spaces. The Nearest-Neighbor Graph approach, inspired by prior work [15], represents each cell as a node interconnected with neighboring cells exhibiting similar phenotypic traits. This graph-based representation captures important phenotypic relationships in a compact and informative manner, reducing the complexities associated with standard geometries.

Density Detection via Community Detection

Following the construction of the Nearest-Neighbor Graph, density detection is addressed through community detection techniques. These methods, adapted from social network analysis, are crucial for partitioning large networks into distinct communities. In the context of cell differentiation, these communities represent accumulations of similar cells, indicating biologically meaningful phenotypic stability and revealing distinct cellular states. Community detection algorithms remain agnostic to the size, number or shape of subpopulations which is essential when dealing with complex cellular populations. Efficient implementations enable partitioning large graphs with minimal computational overhead [16].

Constructing Phenotypic Graphs

The PhenoGraph methodology transforms single-cell data into a graph to show phenotypic relationships among cells. This step prevents rare subpopulations from being overshadowed by more common ones. To address measurement noise, a two-iteration approach is used. The Jaccard similarity coefficient recalibrates cell similarity based on shared neighbors in the second iteration, effectively addressing local density variations and reinforcing well-supported connections. The resulting graph structure reveals the modular nature of the cellular population, highlighting interconnected modules that differentiate rare cells from noise.

2.9 Seurat clustering

In this section, we discuss Seurat, a tool used in single-cell RNA sequencing (scRNA-seq) with potential applications in analyzing lineage markers. Unlike Phenograph, which relies on k-nearest neighbors and community detection algorithms, Seurat creates a Shared Nearest Neighbor (SNN) graph from the k-nearest neighbors graph. It then uses algorithms like Louvain and Leiden for community detection. Seurat's use of SNN, combined with Principal Component Analysis (PCA), simplifies complex data while highlighting

important features, including potentially lineage markers. While both Seurat and Phenograph are effective for clustering, Seurat's approach of using both PCA and SNN might have a unique benefit in identifying cell lineages in scRNA-seq studies

Dimension Reduction

The process begins with dimension reduction, converting high-dimensional protein marker counts into a lower-dimensional space. Seurat adapts a variant of Canonical Correlation Analysis (CCA) for this purpose. Specifically, this adapted CCA identifies correlated patterns of protein expression across the dataset, effectively reducing noise and focusing on the most informative features for lineage marker analysis.

Identification of Cell Populations

Here, Seurat employs tailored versions of K-nearest neighbors (KNN) and shared nearest neighbor (SNN) algorithms for clustering CyTOF data. The KNN algorithm in this context is used to find the closest cells in terms of protein expression profile, forming a network of cells based on similarity. The SNN algorithm further refines these groups by identifying cells that share mutual neighbors, thereby enhancing the accuracy of clustering cells into distinct populations based on lineage markers.

Feature Selection and Scoring

In this crucial step, Seurat utilizes statistical methods to select and score protein markers. Techniques such as variance analysis are applied to determine which markers show significant differences across cell populations. This not only highlights the most relevant markers for lineage identification but also helps in distinguishing subtle differences between cell types.

Data Normalization and Correction

Normalization techniques are applied to transform raw protein counts into a standardized format, making it easier to compare different samples. Seurat implements algorithms for batch effect correction, such as regression-based models, to adjust for variability that might arise from differences in experimental conditions, ensuring that the clustering is reflective of true biological differences.

2.10 Evaluating Clustering Performance

In clustering analysis, finding patterns in data depends on selecting appropriate parameters and algorithms. These choices can strongly influence the resulting clusters, highlighting the need to explore different combinations. To assess clustering performance, we employ metrics that gauge the similarity or dissimilarity between various clustering outcomes. The Adjusted Rand Index is one such metric that enables us to objectively evaluate clustering quality. In the following section, a detailed examination of the Adjusted Rand Index (ARI) is presented.

The ARI is a metric used to quantify the similarity between two clustering results. It is an improvement over the Rand Index (RI) and addresses the issue of RI's variable expected value. ARI is calculated based on a contingency table that represents the relationships between clusters in two partitions.

Let's define some terms:

U : Two random partitions with multiple clusters inside.

n_{ij} : Number of objects that belong to both cluster u_i and cluster v_j.

n_i : Number of objects in cluster u_i.

n_j : Number of objects in cluster v_j.

The contingency table is constructed as follows:

	v_1	v_2	v_3	$\cdots$	v_k	Total
u_1	n_{11}	n_{12}	n_{13}	$\cdots$	n_{1k}	$n_{1\cdot}$
u_2	n_{21}	n_{22}	n_{23}	$\cdots$	n_{2k}	$n_{2\cdot}$
u_3	n_{31}	n_{32}	n_{33}	$\cdots$	n_{3k}	$n_{3\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_h	n_{h1}	n_{h2}	n_{h3}	$\cdots$	n_{hk}	$n_{h\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 2}$	$n_{\cdot 3}$	$\cdots$	$n_{\cdot k}$	n

Here:

n_{ij} : Number of elements that belong to both cluster u_i and cluster v_j.

$n_{i\cdot}$: Sum of the values in row i, representing the total elements in cluster u_i.

$n_{\cdot j}$: Sum of the values in column j, representing the total elements in cluster v_j.

n : Total number of elements.

With the contingency table in place, the ARI is calculated using the following formula:

$$ARI = \frac{\sum_{ij}\left(\binom{n_{ij}}{2} - \frac{\binom{n_{i\cdot}}{2}\cdot\binom{n_{\cdot j}}{2}}{\binom{n}{2}}\right)}{\frac{1}{2}\left(\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right) - \frac{\binom{n}{2}}{2}}$$

In this formula, $\binom{n}{2}$ represents the binomial coefficient, and the terms involving n_{ij}, $n_{i\cdot}$, $n_{\cdot j}$, and n are calculated from the contingency table.

ARI produces a value between -1 and 1, where -1 indicates perfect disagreement between clusterings, 0 indicates random agreement, and 1 indicates perfect agreement. It is a valuable metric for assessing clustering similarity while accounting for chance as well [17].

2.11 The Kolmogorov-Smirnov (KS) Test

The Kolmogorov-Smirnov (KS) test is a statistical method used to compare two sample distributions directly. It assesses the degree of agreement between two sets of observed data, which can be continuous or discrete [18].

Test Statistic (d)

The KS test statistic, denoted as d, represents the largest vertical deviation between the cumulative step functions of the two datasets being compared. In mathematical terms:

$$d = \max\left(|F_1(X) - F_2(X)|\right)$$

Where:

- d is the maximum deviation (Kolmogorov statistic).
- $F_1(X)$ is the cumulative distribution of the first dataset.
- $F_2(X)$ is the cumulative distribution of the second dataset.

Procedure

To compare two datasets using the KS test:

1. **Calculate Empirical Cumulative Frequencies:** Compute the empirical cumulative relative frequencies for each of the two datasets.
2. **Calculate KS Test Statistic (d):** Find the maximum absolute difference between the cumulative frequencies of the two datasets, resulting in the KS test statistic (d).
3. **Hypothesis Testing:** Use the KS test statistic or its p-value to assess the similarity or difference between the two datasets' distributions.

3 Implementation

This section provides a detailed description of the step-by-step implementation and evaluation of a marker-scaled clustering algorithm for population identification in single-cell cytometry data. It should be emphasized that these implementation steps were conducted with the aim of enhancing population identification.

Marker scale Hypothesis

The research project's hypothesis was driven by the difficulties associated with conventional clustering techniques used in cytometry data. This can be summarized as follows:

Clustering cytometry data poses a challenge because researchers have constraints on the number of markers they can simultaneously measure in a single experiment, thus requiring careful marker selection due to the limited panel size. More specifically, in flow cytometry, spectral overlap restricts the number of fluorescent markers that can be simultaneously analyzed, while in mass cytometry, the limitation arises from a narrower range of available metal isotope-labeled antibodies compared to fluorescent ones. While some cell types are identified by only a few lineage markers, others depend on a more comprehensive set of markers. This difference in marker representation can create challenges for distance-based clustering algorithms, as they may face difficulty in appropriately weighting the markers, especially when markers are associated with multiple cell types. This complicates the accurate separation of cell populations and undermines the effectiveness of unsupervised clustering methods.

To address these challenges, this research emphasizes the significance of a user-informed clustering approach. This method allows researchers to incorporate their insights into the selection of markers for analysis based on their knowledge about biology and experimental design, thus improving the algorithm's ability to achieve precise separation of cell populations by leveraging researchers' expertise. In this research, we leverage groups defined by immunologist experts to guide the selection of markers and enhance the clustering process, as shown in Table 3.1.

In response to the marker weighting challenge, Principal Component Analysis scaling can be used to tackle this issue. PCA scaling is performed to standardize marker sets for each anticipated cell type, ensuring that markers are fairly weighted in the clustering process even when they are shared among multiple cell types. The hypothesis is that this improves the algorithm's ability to effectively assess markers, as well as enables more accurate separation of cell populations.

3.1 Preprocessing the data

Dataset

This research, as mentioned before, is centered around the analysis of mass cytometry data in the form of counts. More specifically the initial dataset consists of 29 columns and 1215100 rows, where the rows represent the number of cells and the columns are described below:

- 1-26: Lineage Markers
- 27: Population Labels, assigned by researchers using manual gating

- 28: Subpopulation Labels, assigned by researchers using manual gating
- 29: Sample number

The dataset obtained by excluding the *Population*, *Subpopulation*, and *Sample number* columns represents the *unscaled* mass cytometry data used in this research.

Subset

At first a subset of 10,000 cells is randomly selected for analysis. The selection process aims to ensure a representative sample of the cytometry data.

Manually Defined Population Groups

In this section, the manually defined population groups, composed by expert immunologists, are introduced, as shown in Table 3.1. These groups play a key role in the marker scaling approach. Each group contains a different combination of markers that are significant. In a way, considering combinations of markers, rather than individual ones in isolation, is beneficial as it captures the complex relationships between markers that collectively define a cell population. Unique combinations of markers working together can often identify cell types. Focusing on these combinations provides the algorithm with a more comprehensive view of cell populations and can potentially enable a more precise separation.

Table 3.1: Manually Defined Population Groups

Population Group	**Markers**			
CD4pos Naive	CD4	CCR7	CD45RA	
CD8pos Eff	CD8	CD25		
CD4pos CM	CD4	CCR7	CD45RA	
CD4pos Eff	CD4	CD25		
CD4pos Treg	CD4	CD25	CD127	
NKT	CD56	CD14	CD19	
CD8pos Naive	CD8	CCR7	CD45RA	
B cells Naive	CD27	CD38		
Monocytes CD16pos	CD16			
Monocytes CD16neg	CD16			
CD8pos EM	CD8	CCR7	CD45RA	
CD4pos EM	CD4	CCR7	CD45RA	
TCRgd	TCRgd			
B cells Transitional	CD27	CD38		
pDCs	CD16	HLADR	CD123	
mDCs	CD16	HLADR	CD11c	
NK cells CD56bright	HLADR	CD16	CD56	
CD8pos CM	CD8	CCR7	CD45RA	
B cells Memory	CD27	CD38		
Basophils	CD123	CD38	HLADR	
Plasmablasts	CD27	CD38		
major pop markers	CD3	CD19	CD20	CD14

Alternate Defined Groups using Log-Fold change

In an effort to improve the identification of populations in this research, a different method for defining population groups was investigated. Unlike the manually defined groups men-

tioned earlier, this alternative approach utilizes log fold change to distinguish significant markers and differentiate among cells, thereby effectively identifying distinct populations.

More specifically, each unique population within the data set is examined. For every population, the mean counts of cell markers are calculated, both within the specific population and in comparison to other populations. This calculation is essential for determining the relative abundance of each marker across different groups. Then, the log fold change for each marker is computed. This involves taking the logarithm (base 2) of the ratio of the mean counts within the population to those outside it. Markers with a positive log fold change are then identified as significant to the population. This significance comes from the understanding that a positive log fold change indicates a greater prevalence of a marker in a particular population compared to others. These important markers are then gathered, placing them into their corresponding populations. This method not only has the potential to highlight markers that are specific to each population but can also provide a clear picture of the distinct cellular profiles

Marker Weighting Standardization

To address the challenge of marker weighting, a standardization technique involving the standard deviation squared is used. Standard deviation squared serves as an effective means to normalize marker sets within each group, ensuring that markers are fairly weighted in the analysis. By dividing each group by the square of its standard deviation, the algorithm can mitigate the impact of varying marker scales and better account for marker importance variations.

More specifically, in mass cytometry data, such as in this case, markers may have varying scales, which can lead to differences in their contributions to the clustering process. Some markers might have larger numerical values, making them appear more influential in the analysis, even if they are not necessarily more biologically relevant. On the other hand, markers with smaller numerical values might be overshadowed, despite their importance in defining cell populations.

To mitigate these scale-related issues, each group is divided by the square of its standard deviation. This operation normalizes the marker sets within each group, making the scale of all markers more uniform. The square of the standard deviation acts as a scaling factor, essentially 'flattening' the impact of large numerical values while 'amplifying' the importance of smaller ones.

This transformation ensures that markers are fairly weighted in the analysis, irrespective of their original scales. It reduces the influence of outliers and extremes, allowing the algorithm to focus on the relative importance of markers within each group. To summarize, by using this technique, the algorithm can more accurately evaluate markers and their roles in separating cell populations, ultimately enhancing the precision of the analysis.

Marker scaled dataset

The dataset referred to as the *marker-scaled* dataset is obtained after applying the standardization method that, was mentioned before, based on the manually defined groups. This dataset comprises 55 columns and 1215100 rows. Notably, the columns correspond to the groups listed in Table 3.1, and it is possible for the same marker to appear in multiple groups.

An additional *marker-scaled* dataset is defined using the alternative groups defined using the log-fold change. This dataset is undergoes the same procedure of standardization mentioned above.

Visualization and Exploration of Marker-Scaled Data

To gain the initial insights of the hypothesis in action, a Principal Component Analysis (PCA) visualization is needed. PCA was performed on a subset of 10,000 cells from the mass cytometry data. This subset was subjected to PCA both in its original unscaled form and after applying marker scaling. The primary objective was to visualize the distribution of cells in the first two principal components (PC1 vs. PC2) to assess whether any indications of improved separation between cell populations could be observed.

3.2 Clustering with RPhenograph

Herein the PCA results of both non scaled and marker-scaled data discussed in the previous sections are fed into the RPhenograph algortihm. This clustering process involves leveraging the Louvain clustering method provided by Rphenograph to identify distinct cell populations within the cytometry data.

3.3 Analysis using Seurat

While Phenograph primarily uses a k-nearest neighbors approach for clustering and then identify communities using Louvain method, Seurat expands on this with multiple clustering algorithms, such as Leiden and Smart Local Moving(SLM) [19]. These algorithms excel at identifying subtle patterns in complex, high-dimensional data. This section elaborates on the Seurat analysis pipeline applied in this research, covering its application to both unscaled and marker-scaled datasets.

Seurat Object

The process begins by creating three seurat object, one for the unscaled and one for each of the marker-scaled data. A Seurat object is a structured data container used in the Seurat software package. It stores various types of data and analysis results associated with single-cell studies.

- Raw Data: This includes the count matrix, where rows typically represent genes, or markers in this case, and columns represent individual cells.
- Processed Data: After initial processing steps such as normalization and scaling, the processed data is also stored within the Seurat object.
- Analytical Results: Results from various analyses like dimensionality reduction (e.g., PCA), clustering outcomes, and results from differential expression analysis are held within the object.
- Metadata: The Seurat object can also contain additional information about the cells and genes/markers, such as quality control metrics and cell-type annotations.

In order to create these Seurat objects, the data needs to converted in a specific format called *dgcMatrix*, which essentially is a sparse matrix in compressed, column-oriented form. The Seurat object is then initialized using *CreateSeuratObject*, incorporating the count matrix and the transformed matrix for unscaled and marker-scaled respectively.

Normalizing the Data

Next follows the Normalization, which is performed to adjust the scale of data. This adjustment is done to ensure that variations in the expression levels of different markers are properly scaled for accurate comparison and analysis. Seurat's default normalization method, *LogNormalize*, is used here. Essentially, the normalization step converts raw counts into normalized expression values, which are then suitable for scaling.

Scale the data
After normalization follows the scaling of the data. This is done to center and scale the expression values of each marker to have zero mean and unit variance.

Feature Selection
This step is about identifying a subset of important markers, as it highlights features significant for later analysis stages. However, in cases where each marker's contribution is critical, such as this case, this step can be adapted. Instead of selective feature identification, all markers can be included as significant. This is done by specifying the total number of markers in the command, thereby ensuring the analysis incorporates the complete marker set.

Linear Dimensional Reduction
PCA is performed on scaled data to reduce dimensionality.

Clustering the Cells
Seurat employs a graph-based clustering approach. Cells are embedded in a KNN graph based on their PCA scores, and clustering is done using three different algorithms Louvain, Leiden and SLM algorithm. The FindClusters function optimizes modularity to identify cell communities.

Non-linear Dimensional Reduction
Finally, techniques like UMAP are applied for advanced visualization. These methods help in exploring the data in low-dimensional space, revealing the underlying structure and relationships between cells.

3.4 Parameter tuning
In cytometry data analysis, effective clustering relies on carefully selecting and optimizing parameters. This is essential for simplifying complex data into meaningful insights. The systematic exploration of these parameters aims to optimize the clustering process and ensure the best possible results from the analysis.

This part explores the importance and justification of each parameter for clustering the data. It highlights the significance of specific parameters in defining the clustering process and justifies their selection based on the need for accurate data analysis.

- **Number of Principal Components:** Using too few PCs may leave out important information, affecting cell differentiation. On the other hand, using too many PCs can add noise and lead to overfitting. The optimal number aims to capture significant data features while minimizing noise, improving the clustering process in cytometry data analysis.
- **Resolution:** The resolution parameter, which is a part of the clustering algorithm (Louvain, Leiden, or SLM), determines the level of detail in clustering. A higher resolution results in a finer subdivision of clusters, allowing for the identification of subtle cell populations, whereas a lower resolution might merge distinct groups, affecting the granularity of the analysis.
- **K number (K Neighbors):** Integral to the k-nearest neighbors (kNN) step, it defines the size of the cell neighborhood, impacting the construction of the similarity graph used for clustering. An optimal k number accurately reflects natural groupings in the data, while a poorly chosen k can lead to misleading cluster formations.
- **Clustering Algorithms (Louvain, Leiden, SLM):** Applied after PCA and kNN steps, each algorithm uniquely influences cluster formation. Louvain is efficient for

large datasets and tends to form larger clusters. Leiden offers higher resolution, effectively identifying smaller communities. SLM reveals nuanced structures, emphasizing local data characteristics. The choice of algorithm significantly impacts the interpretation of the data's community structure.

Evaluating Clustering Performance

In previous sections, we emphasized the importance of choosing the right metric for clustering evaluation. The Adjusted Rand Index (ARI) stands out as a suitable choice for this research due to its capacity to handle overlapping clusters, as well as for chance agreement between them. More specifically it means that it can differentiate small differences in marker expression profiles even when populations overlap. In other words, ARI can identify and measure overlap. It recognizes that certain cells may be part of multiple populations and assigns a higher similarity score when the overlap is meaningful.

Furthermore ARI accounts for the possibility of random chance creating apparent similarities between clusters, which is important in tasks involving high-dimensional data like mass cytometry. It corrects the similarity score by comparing observed agreement to what's expected by chance. This factor helps to avoid overly optimistic evaluations, particularly when dealing with noisy or complex data. It removes random similarities and leads to a more dependable assessment of clustering performance.

Implementing ARI in Parameter Optimization

The process of parameter optimization was conducted for unscaled and both marker-scaled data and the best parameters found are shown in Table 3.2. In this step, a comparison was conducted among different marker-scaled approaches, selecting the one with the highest Adjusted Rand Index (ARI) for representation in the rest of the analysis. The following section outlines how ARI is implemented in the parameter optimization process.

1. **Parameter Exploration Framework:** The parameter exploration process involves systematically testing different combinations of parameters such as the number of Principal Components (PCs), the value of k for KNN, the resolution for clustering, and the choice of clustering algorithms (Louvain, Leiden, SLM). Each parameter combination results in a specific clustering solution.

2. **Clustering Execution:** For each parameter combination, the clustering algorithm is executed on the preprocessed single-cell cytometry data. This generates clusters based on the chosen parameters.

3. **ARI Calculation:** After obtaining clustering results for each parameter combination, ARI is calculated to quantitatively evaluate the quality of these clusters. ARI assesses the agreement between the obtained clusters and a ground truth, which in your case may represent the true population labels or expert-defined clusters.

4. **Comparing Parameter Combinations:** ARI assigns a similarity score to each clustering solution. Higher ARI scores indicate better agreement with the ground truth, suggesting that the parameter combination has led to more accurate clustering. This score allows for a direct comparison between different parameter settings.

5. **Optimal Parameter Selection:** The parameter combination that yields the highest ARI score is considered the optimal choice for clustering the single-cell cytometry data. This selection is based on the assumption that higher ARI values reflect a clustering solution that best captures the underlying population structure.

Table 3.2: Optimized parameters that result in the highest Adjusted Rand Index (ARI) scores. The parameters include the number of principal components (PCs), the value of k for KNN, the resolution, and the selected clustering algorithm. The *marker-scaled* data presented in this table, which exhibits the highest Adjusted Rand Index (ARI) among the two approaches, is represented by the manually defined groups method.

Data type	PCs	k	Resolution	Clustering algorithm
unscaled	10	20	0.1	SLM
marker-scaled	15	15	0.1	Louvain

3.5 Implementing in a Large Dataset

This section discusses in detail the difficulties of adapting parameters from a small dataset to a larger one, as well as the necessary modifications and potential complications that may arise when such adjustments cannot be made due to limited computational resources. The key factors that demand thorough investigation for effective parameter adjustment include cluster density, noise levels and data distribution. Importantly, in this case, the smaller dataset represents the larger one accurately, maintaining identical proportional characteristics. This subset is thus a valuable reference point for our analysis. The key factors outlined above are explored in relation to this particular scenario in the following part.

- **Cluster Density:** Cluster density, in this case, is not a significant concern as it seems, as the representative subset matches the larger dataset's cluster proportions, ensuring consistent cluster density. This minimizes the need for density-related adjustments.
- **Noise:** Noise levels are expected to be less problematic since the representative subset maintains identical proportions. A t-test was performed to compare the Normalized Coefficient of Variation, which is calculated as the standard deviation divided by the mean and multiplied by 100, and represents noise levels. Normalizing this metric by the mean ensures consistent, size-independent noise assessment across the two subsets of different sizes. The results indicated similar noise levels across markers, as shown in the Figure 3.1, where the distribution of normalized Coefficient of Variation (CV%) values is presented.

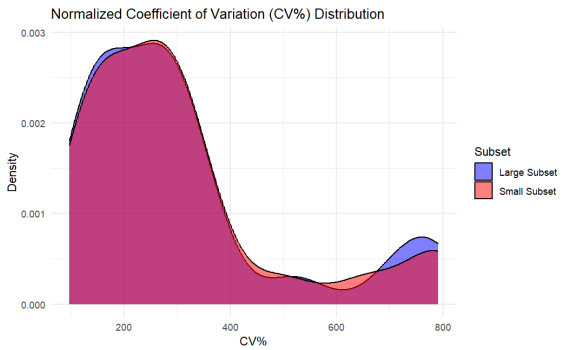


Figure 3.1: Comparison of Normalized Coefficient of Variation (CV%) Distributions between the Small and Large Subsets. The plot shows the distribution of CV% values, representing noise levels as a percentage of the mean, for both subsets, with blue representing the Smaller Subset and red representing the Larger Subset.

- **Data Distribution:** The Kolmogorov-Smirnov (KS) test was applied to analyze the distributional similarity of CD markers, showing significant consistency across all markers. In Figure 3.2, the marker with the most distinct distribution between the subsets is shown, as determined by the KS test.

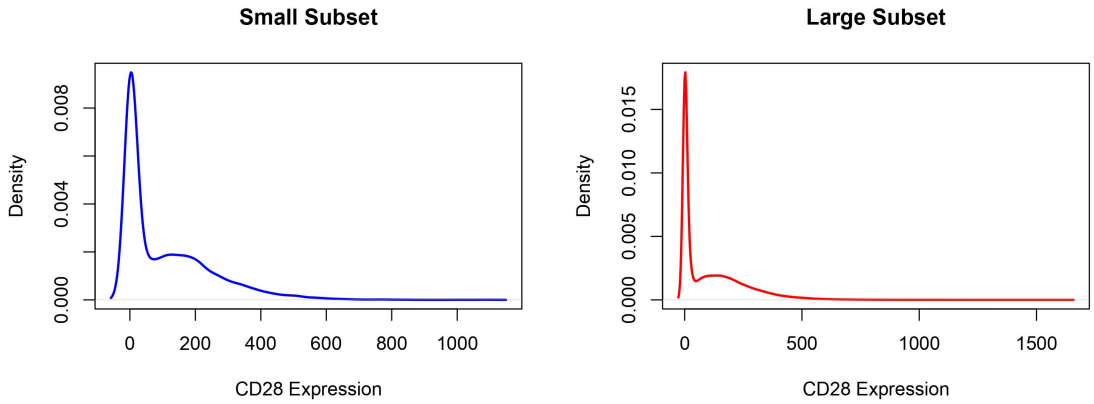


Figure 3.2: Side-by-side comparison of CD28 expression density distributions between the small subset (left, 10,000 cells) and large subset (right, 500,000 cells).

After analyzing the factors mentioned above, we can conclude that it is safe to assume, that the optimized parameters derived from the representative subset of 10,000 immune cells are suitable for the larger subset of 500,000 with adequate confidence. However, caution is advised, especially regarding other factors, such as cluster overlapping and algorithm sensitivity, as testing for them can be more challenging.

Perform Clustering on the Large dataset

In summary, two distinct subsets, each containing 500,000 cells, have been established at this stage of the analysis. The first subset consists of non scaled data with raw counts, while the second subset includes data that has been scaled based on marker expression, derived from manually defined groups. Clustering analyses, employing both PhenoGraph and Seurat algorithms, have been applied to these subsets. The optimal parameters used for these analyses are shown in Table 3.2.

4 Results

4.1 PCA

PCA helps understand the dataset initially and explore cell populations. We aim to find clearer clusters in a simpler data representation, expecting better separation in the scaled version than the non-scaled one.

We performed PCA on two subsets of 10,000 and 500,000 cells each for non-scaled and scaled data. In the following analysis, we focus on results from these subsets. We examine PC1 vs PC2 plots to observe the distribution of cells and gain insights into different cell types. These plots are shown in Figure 4.1 below:

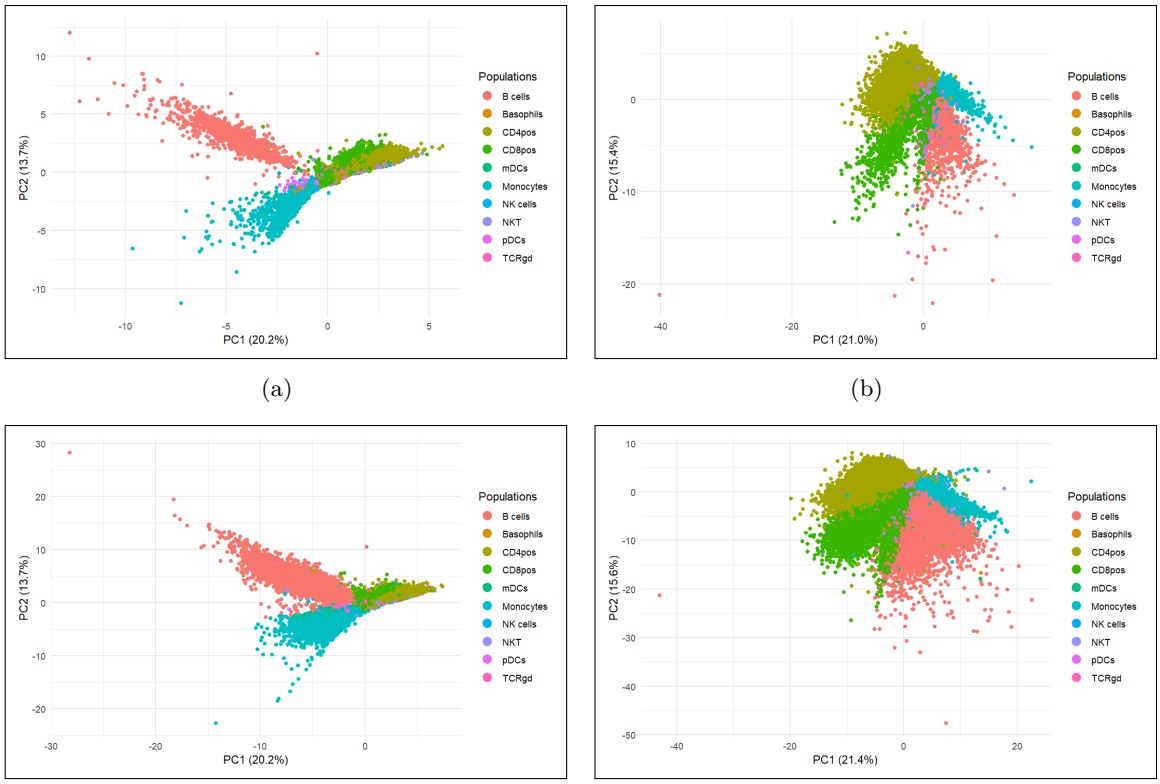
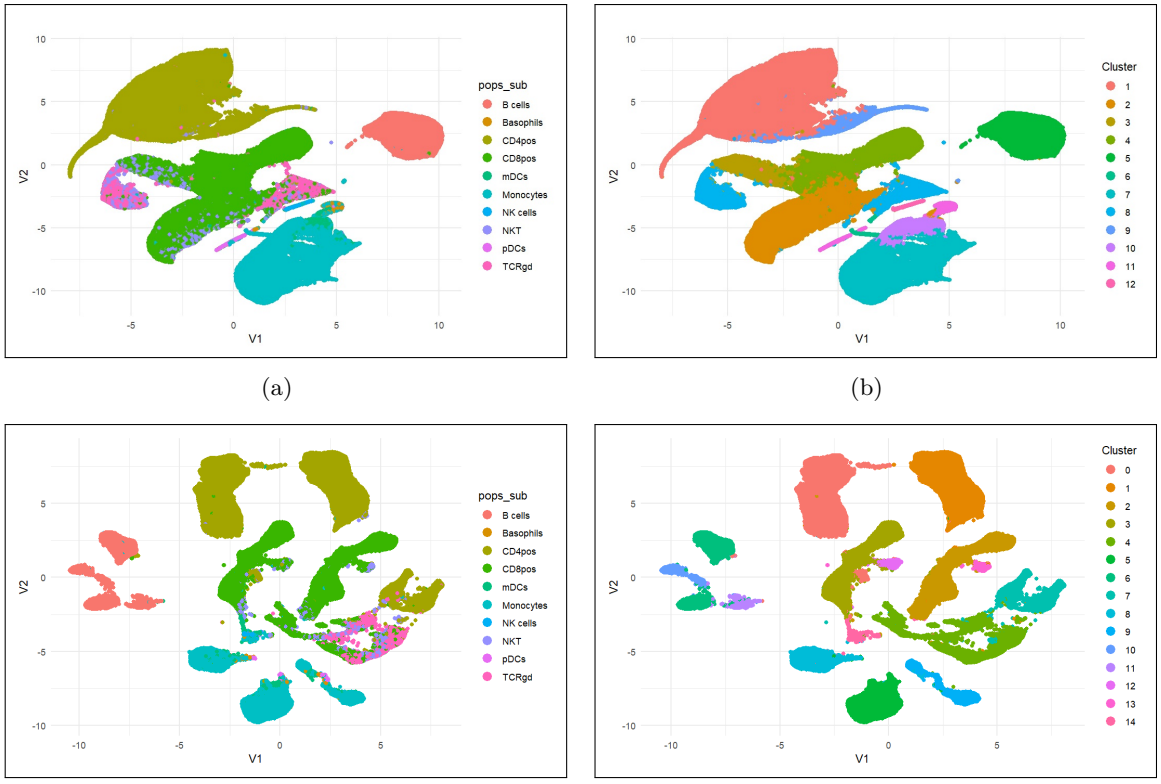


Figure 4.1: PCA visualization of two subsets from a dataset comprising 1,210,000 cells, with PC1 on the x-axis and PC2 on the y-axis. Plot (a) represents the non-scaled subset of 10,000 cells, while plot (b) shows the scaled subset of the same size. Similarly, plot (c) shows the non-scaled subset of 500,000 cells, and plot (d) shows the scaled subset of the same size.

The above PCA plots of the cytometry data show a clear difference in how cell groups are arranged between marker-scaled and non-scaled data. In both sets of 10,000 and 500,000 cells, marker-scaled data (Plots b and d) had more defined and separated clusters compared to non-scaled data (Plots a and c). This difference is especially clear in the larger set of 500,000 cells, where marker scaling significantly improved the identification of distinct cell groups. These findings support the idea that using marker scaling can help distinguish between cell types in single-cell cytometry data.

4.2 Clustering Evaluation

This section presents the results of clustering evaluation performed on a subset of 500,000 cells from a dataset of 1,210,000 cells. We use the Phenograph and Seurat algorithms on non-scaled data to assess their ability to replicate the cell populations defined by manual gating. In Figure 4.2, we visually compare the clusters assigned by Phenograph and Seurat with the populations identified through manual gating. Additionally, supplementary information is provided in the appendix. This includes additional evaluations conducted on a smaller subset of 10,000 cells, offering a deeper understanding of the clustering methodology and its application across different dataset sizes.



(a) (b)

(c) (d)

Figure 4.2: Results of Phenograph and Seurat clustering on a subset of 500,000 cells from a dataset of 1,210,000 cells using non-scaled data. Parameters (*k-neighbors = 20*, *number of principal components = 15*, *Resolution = 0.1*, *Clustering algorithm = SLM*) were optimized using a subset of 10,000 cells. The plots show a UMAP representation of PCA for the 500,000-cell subset, with the x-axis representing the first UMAP dimension (V1) and the y-axis representing the second UMAP dimension (V2). Subfigures (a) and (b) show clustering labeled with populations assigned by scientists using manual gating included in the data and clustering based on Phenograph-assigned clusters, respectively. Subfigures (c) and (d) represent clustering results using the same populations assigned by scientists and Seurat-assigned clusters, respectively.

Figure 4.2 shows different aspects of cell population identification using manual gating by scientists and computational clustering algorithms like Phenograph and Seurat. In plot (a), clear and distinct populations assigned by scientists are visible. Plot (b) displays clusters assigned by the Phenograph algorithm, which effectively identifies separate clusters, although minor differences compared to manual gating are seen, indicating possible over- or under-segmentation. Moving on to Seurat's results, plots (c) and (d) show populations assigned by scientists and clusters assigned by Seurat, respectively. Seurat's clustering

demonstrates effectiveness, yet similar to Phenograph, slight differences from manual gating are noted. Overall, both algorithms reasonably identify separate cell populations, though with some differences compared to manual gating. This suggests a validation of the clustering process to a certain extent.

4.3 Comparison of Marker Scaling vs. Non-Scaled Data with Seurat and Phenograph Clustering Algorithms

In this section, the effectiveness of marker scaling versus non-scaled data in clustering analysis using Phenograph and Seurat algorithms is examined. Marker scaling involves adjusting marker expression levels based on predefined groups defined by expert immunologists. This method was chosen over the alternative approach of defining groups using log fold change, and the choice was based on its performance evaluated by the Adjusted Rand Index (ARI) metric, as discussed in previous sections. The comparison between scaled and non-scaled data shows the impact of this preprocessing step on clustering outcomes and the identification of distinct cell populations.

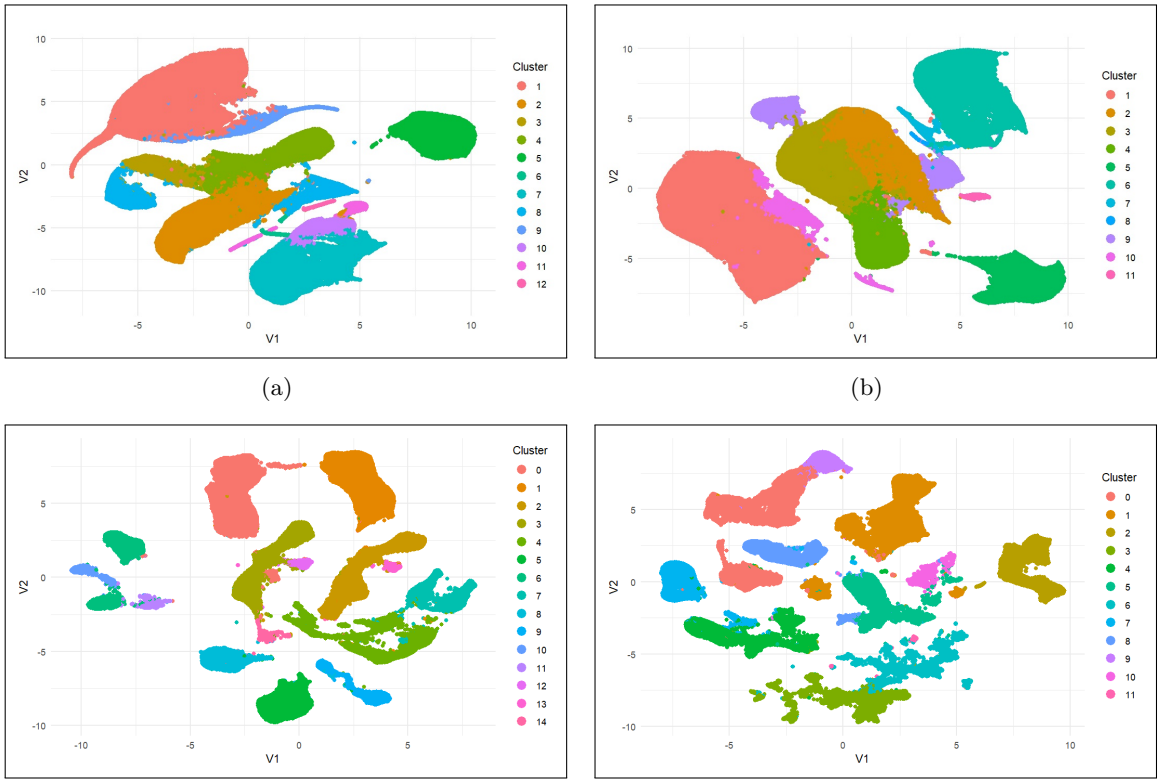


Figure 4.3: Comparison of marker scaled versus non-scaled data in clustering analysis using Phenograph and Seurat algorithms. Marker scaling adjusts marker expression levels based on predefined groups by expert immunologists, chosen over log fold change-based grouping, judged by the ARI metric as mentioned in the previous sections. The plots show a UMAP representation of PCA for the 500,000-cell subset, with the x-axis representing the first UMAP dimension (V1) and the y-axis representing the second UMAP dimension (V2). Clustering results are shown for both unscaled (a) and (c) and scaled (b) and (d) data subsets of 500,000 cells from a total dataset of 1,210,000 cells. Optimal clustering parameters were determined using a subset of 10,000 cells for both scaled and unscaled data. Specifically, for unscaled data: *number of principal components = 10*, *k-neighbors = 20*, *Resolution = 0.1*, *Clustering algorithm = SLM*; and for Marker-scaled data: *number of principal components = 15*, *k-neighbors = 15*, *Resolution = 0.1*, *Clustering algorithm = Louvain*.

Clustering Differentiation

The differences between plots (a) and (b) are small, but they have some notable distinctions. In plot (a), the clusters look more tightly packed, with dots spread out inside them, leading to a less distinct boundaries between the clusters. This could mean there's some variation within the clusters, maybe because of cells overlapping or random noise in the data. In contrast, plot (b) shows the scaled data and has clusters that aren't as tightly packed and the boundaries between the clusters seem more clear. In contrast, plots (c) and (d) exhibit different characteristics. In plot (c), the clusters appear more compact, with smoother and clearer boundaries. On the other hand, plot (d) shows clusters that are more spread out, with smaller parts of larger clusters visible nearby. This suggests that there might be more differences between cell groups in plot (d), possibly because some cells share characteristics with multiple groups or due to background noise in the data.

Number of Clusters

Marker-scaled data, as shown in plots (b) and (d), results in fewer clusters than non-scaled data, as shown in plots (a) and (c). This could suggest an improved identification of cell populations. However, it's important to note that both methods identified more clusters than the 10 distinct populations identified by manual gating. This difference might mean that the clustering process is too complex. This fact raises questions about their precision. For example, in plot (a), there are several small clusters in the center that are not present in plot (b). This could indicate that the non-scaled method is identifying noise as distinct clusters, or it could suggest that the scaled method is failing to identify subpopulations. A similar pattern is observed when comparing plots (c) and (d).

Scaled vs Non-Scaled Data

While the scaled data offers in some cases fewer, more defined clusters, it's important to consider whether this reduction reflects real differences in biology. For instance, in plot (b), the clusters appear more compact and well-defined than in plot (a). However, this could also mean that some biological variability is being lost in the scaling process, which is the opposite of what is expected, as the purpose of the predefined groups is to incorporate biological information into the clustering process. In conclusion, while there are observable differences between marker scaled and non-scaled data in clustering, definitive conclusions on the superiority of one over the other require further evaluation.

5 Conclusions

In summary, our study found some small differences between using scaled and non-scaled data for clustering. We first looked at the dataset's structure and cell populations using PCA. The PCA analysis suggested that scaled data might help separate clusters better. While scaling markers does improve the differentiation between cell groups only slightly in some cases, like in Figure 4.3a and Figure 4.3b, it seems to complicate it in others like in Figure 4.2d and Figure 4.3d. However, whether this enhancement leads to accurate identification of cell types is still uncertain. Furthermore, scaling markers also appears to decrease the number of clusters in comparison to the non scaled data, which could help define cell populations more clearly and could suggest an improved identification of cell populations, but it could also suggest that the scaled method is failing to identify subpopulations. Nevertheless, even with scaling, we still observe additional clusters compared to the populations defined by manual gating, indicating the complexity of clustering.

To finalize, our findings suggest that the impact of marker scaling on accurately differentiating cell populations remains uncertain. The results appear inconclusive, indicating the need for further investigation. Several factors may contribute to this uncertainty, including the effectiveness of the method itself, the choice of clustering algorithms, and the definition of predefined groups. Therefore, additional studies are needed. Future research could involve testing the method on new datasets to assess its applicability across different contexts. Furthermore, exploring alternative clustering algorithms such as FlowSom[20] could provide more insights on project, as FlowSom provides a viewpoint by organizing cells into a topological map based on their expression profiles. This organization allows for a visualization of how cells relate to each other in a spatial manner, which can potentially show patterns and structures within the data that might not be shown with other methods, such as Phenograph or Seurat. Lastly, a further reassessment of the groups by immunologists and a more careful selection of markers could potentially lead to better results. Overall, further investigation is essential to gain a better understanding of the effectiveness of marker scaling in single-cell cytometry data analysis.

Bibliography

[1] Christina Bligaard Pedersen and Lars Rønn Olsen. "Algorithmic clustering of singlecell cytometry data—how unsupervised are these analyses really?" In: *Cytometry Part A* 97.3 (Nov. 2019). [Online; accessed 2024-01-24], pp. 219–221. DOI: 10.1002/cyto.a.23917.

[2] Julius M. Cruse, Robert Edwin Lewis, and Huan Wang. *Immunology guidebook*. [Online; accessed 2023-09-13]. Academic Press, 2004, pp. 1–15.

[3] James F. Zachary. *Pathologic basis of veterinary disease*. [Online; accessed 2023-09-14]. Mosby, 2017, pp. 242–285.

[4] *Human immune cell marker guide*. [Online; accessed 2024-01-11]. URL: https://www.cellsignal.com/pathways/immune-cell-markers-human.

[5] Contributors to Wikimedia projects. *Hematopoietic stem cell*. [Online; accessed 2023-09-18]. Sept. 2023. URL: https://en.wikipedia.org/wiki/Hematopoietic_stem_cell.

[6] RnDsystems. *Hematopoietic Stem Cell & Lineage-specific Markers*. 1. URL: http://www.woongbee.com/0NewHome/RnD/RnD_Literature/Research/27_HSCMarker_12.pdf.

[7] adminNC. *Flow cytometry gating*. [Online; accessed 2024-01-15]. Feb. 2020. URL: https://nanocellect.com/blog/flow-cytometry-gating-everything-you-need-to-know/.

[8] *Flow cytometry introduction*. [Online; accessed 2023-09-20]. URL: https://www.abcam.com/protocols/introduction-to-flow-cytometry.

[9] Matthew H Spitzer and Garry P Nolan. "Mass cytometry: Single cells, many features". In: *Cell* 165.4 (May 2016), pp. 780–91. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4860251/.

[10] Morten Mørup Tue Herlau Mikkel N. Schmidt. *Introduction to Machine Learning and Data Mining*. 1.2. Copenhagen, Denmark: DTU, 2021.

[11] *How UMAP Works — umap 0.5 documentation*. [Online; accessed 2023-12-29]. URL: https://umap-learn.readthedocs.io/en/latest/how_umap_works.html.

[12] Diego Lopez Yse. *K-Nearest Neighbor (KNN) explained*. [Online; accessed 2024-01-10]. URL: https://www.pinecone.io/learn/k-nearest-neighbor/.

[13] R.A. Jarvis and E.A. Patrick. "Clustering Using a Similarity Measure Based on Shared Near Neighbors". In: *IEEE Transactions on Computers* C-22.11 (1973), pp. 1025–1034. DOI: 10.1109/T-C.1973.223640.

[14] Levent Ertöz, Michael Steinbach, and Vipin Kumar. *A new shared nearest neighbor clustering algorithm and its applications*. Jan. 2002. URL: https://www.researchgate.net/publication/228751217_A_new_shared_nearest_neighbor_clustering_algorithm_and_its_applications.

[15] Sean C Bendall et al. "Single-Cell Trajectory Detection Uncovers Progression and Regulatory Coordination in Human B Cell Development". In: *Cell* (2014).

[16] Vincent D. Blondel et al. *Fast unfolding of communities in large networks*. Apr. 2008. URL: https://www.researchgate.net/publication/1913681%5Ctextunderscore%7B%7DFast%5Ctextunderscore%7B%7DUnfolding%5Ctextunderscore%7B%7Dof%5Ctextunderscore%7B%7DCommunities%5Ctextunderscore%7B%7Din%5Ctextunderscore%7B%7DLarge%5Ctextunderscore%7B%7DNetworks.

[17] *M.K Hasan*. [Online; accessed 2024-01-16]. Apr. 2020. URL: https://mk-hasan.github.io/posts/2020/04/blog-post-4/.

[18] *Kolmogorov-Smirnov test- Principles*. [Online; accessed 2024-01-22]. URL: https://influentialpoints.com/Training/kolmogorov-smirnov_test-principles-properties-assumptions.htm#:~:text=This%20is%20also%20known%20as,including%20location%2C%20dispersion%20and%20shape..

[19] *Seurat - Guided clustering tutorial*. Feb. 2024. URL: %5Curl%7Bhttps://satijalab.org/seurat/articles/pbmc3k_tutorial.html#cluster-the-cells%7D%20[Online;%20accessed%202024-02-09].

[20] Sofie Van Gassen et al. "FlowSOM: Using selforganizing maps for visualization and interpretation of cytometry data". In: *Cytometry Part A* 87.7 (Jan. 2015). [Online; accessed 2024-02-10], pp. 636–645. DOI: 10.1002/cyto.a.22625.

Technical
University of
Denmark

Brovej, Building 118
2800 Kgs. Lyngby
Tlf. 4525 1700

www.byg.dtu.dk