Towards good representations of single-cell protein expression data

Christoffer Lingjærde

Thesis for Master of Science Degree at the
University of Bergen, Norway

2024

Year:	2024
Title:	Towards good representations of single-cell protein expression data
Author:	Christoffer Lingjærde

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Inge Jonassen. I am grateful that you have opened up an exciting new world for me with the highly interesting topics of this thesis. Your enthusiasm for my many (and usually bad) ideas has given me confidence and motivation. Your wisdom and feedback have been immensely valuable and your optimism is admirable. You have helped me find a direction for my future, and I will be forever grateful to you.

I want to thank my family for their love and support. Thank you to my father for providing inspiration and support when I needed it the most. This would not have been possible without you. Thank you to my mother for always looking out for me. And thank you to my sister for being her usual kind of self. And I want to give a special thanks to my girlfriend for making me a happy man every single day.

Lastly, I want to thank my friends and fellow students for the good times we have had and the memories we share.

Bergen, 2024
Christoffer Lingjærde

Abstract

Many diseases including infections and cancer can evoke an immune response that is detectable as changes in the immune cell composition in blood. In cancers originating in the immune system, the immune cell composition can also change due to uncontrolled growth of the malignant cells, shifting the normal balance between immune cell types. One example is acute myeloid leukemia (AML), which affects a precursor of several immune cell types found in blood including monocytes and neutrophils. Single-cell protein expression measurements obtained with CyTOF offer a powerful means of studying immune cell composition in blood, and this thesis concerns the analysis of such data. We specifically consider the problem of converting such data - which describe which proteins are expressed on each analyzed cell - to a representation that reveals the immune cell composition. We also study how to obtain representations that are well suited as inputs to algorithms for prediction of treatment outcome and survival. The focus will be on unsupervised clustering, and we propose a novel semi-supervised clustering algorithm and compare its performance with other methods.

Contents

Chapter 1

Introduction

One of the first symptoms of an approaching nervous breakdown is the belief that one's work is terribly important.

-Bertrand Russel

Outline of chapter

This chapter provides some context and motivation for the topic of the thesis and presents the project's main aims. A brief summary of the structure of the thesis is also provided.

1.1 Introduction

Cells constitute the building blocks of all forms of life. Cells come in many shapes and sizes and with very different properties. The size range is astonishing, with the smallest being only about 5 μm long (sperm cells) and the largest being more than 1 m long (nerve cells). The shape is highly variable, with some cells being spherical (e.g., an egg cell), others having elongated shapes (e.g., a nerve cell), and still others having strange irregular shapes (e.g., dendritic cells, which have a star-shaped appearance with long branches called dendrites). Cells also have diverse functions. Red blood cells carry oxygen from the lungs to the tissues, white blood cells fight disease agents (pathogens), and nerve cells form an electrical signaling network. Some cells are found in solid tissues, and others are found in the blood.

This thesis will focus on the cells circulating in the blood and, more specifically, on the white blood cells. These cells form part of our immune system and come in many variants, such as B cells, T cells, and NK cells. We know quite a bit about the function of individual white blood cell types today, but the complex interplay between the cells is not fully understood. One way to obtain insight into the interplay is to consider the relative abundance of the different immune cells in the blood. A change in the composition may suggest a change in what the immune system is doing (or trying to do).

One example is an infection; certain immune cells then tend to become more abundant and this is directly related to their role in fighting the pathogen. Another example is cancer; in that case, a person has been ”infected” by rogue cells that do not come from the outside but rather have their origin in healthy cells in the same individual. These rogue cells may trigger an immune response that changes the immune cell composition in the blood (Delves et al., 2017). Treating the cancer can also change the immune cell composition, and the change may be related to how well the treatment works. The treatment induces stress not only to the tumor cells but also to healthy cells in the body, and determining the exact mechanism behind a change in immune cell composition may be difficult. Nevertheless, we may try to learn the association between immune cell composition in blood and response to treatment based on data from many patients.

This thesis was motivated by the desire to learn such associations, using data from a study performed at Haukeland Sykehus on patients with acute myeloid leukemia (AML). At our disposal, we had single-cell protein expression data derived from blood samples taken before and after treatment and treatment response and survival data. For many analyses, such as the estimation of immune cell fractions and survival prediction, the data must first be condensed into a more useful format. In short, we need to find useful representations of the data for downstream analyses. In this thesis, we investigate how clustering can be used to obtain such representations.

1.2 Aims

The overall aims of this thesis were:

- To gain insight into how clustering is best used to derive representations of single-cell protein expression data that reveal the individual cell types present.

- To gain insight into how clustering is best used to derive representations of single-cell protein expression data that are useful as input variables to predictors of treatment response and survival.

1.3 Structure of thesis

The thesis consists of six chapters and an appendix. Chapter 2 introduces the reader to the biological concepts used later in the thesis. This includes describing what we are looking for in the single-cell expression data, how such data are obtained, and the background for the data used in this thesis. Chapter 3 presents the methods used in the thesis. This includes several clustering algorithms, cluster performance metrics, and visualization techniques for high-dimensional data. Methods for assessing the shape of a distribution (entropy) and changes to the shape (Kullback-Leibler divergence) are also discussed. Finally, the chapter briefly summarizes some methods for the analysis of survival data. Chapter 4 presents a novel clustering algorithm that, in addition to the data, takes one or several subsets of features as input to guide the clustering. This semi-supervised clustering algorithm directs attention to specific features in the input that are known a priori to be important. Known in machine learning as zero-shot classification, this approach offers great flexibility and potentially more relevant clusters for downstream applications. Chapter 5 presents results for a real dataset representing single-cell protein expression in blood samples from AML patients and for simulated data. Chapter 6 discusses the results and suggest some topics for further work. The appendix contains some additional results.

Chapter 2

Biological background

Outline of chapter

This chapter provides biological background information for the remaining part of the thesis. We first look at blood and its role in the immune defense and will see that different types of immune cells can be distinguished from each other by examining their surface proteins. We will examine a technique for protein quantification in single cells called mass cytometry. Finally, a type of blood cancer called acute myeloid leukemia (AML) that will be central to the investigation in this thesis will be discussed.

2.1 The composition of blood

Blood acts as a transport medium within an animal. Oxygen and nutrients are transported to tissues; carbon dioxide and waste are transported from tissues to excretory organs for disposal; signals are transmitted by hormones; and blood acts as a defense system against pathogens (Hine, 2015). Two types of cells are abundant in blood: red blood cells (erythrocytes) and white blood cells (leukocytes). Red blood cells are responsible for oxygen transport and are by far the most common cell type in blood. White blood cells are involved in immune defense and comprise several cell types that play different roles in the fight against disease agents (pathogens). All cells in the blood have developed from hematopoietic stem cells, a type of cells found in the bone marrow (the word hematopoiesis is derived from the greek words *hema* which means ”blood” and *poiesis* which means ”to create”). These stem cells can develop (or differentiate) into two distinct types of cells: myeloid progenitors and lymphoid progenitors. Cells derived from the former are called myeloid cells, while cells derived from the latter are called lymphoid cells (see Figure 2.1). In this thesis, we will focus on five different immune cell types: B cells, T cells, NK cells, monocytes, and neutrophils. For the T cells, we will distinguish between the two subtypes T killer cells and T helper cells.

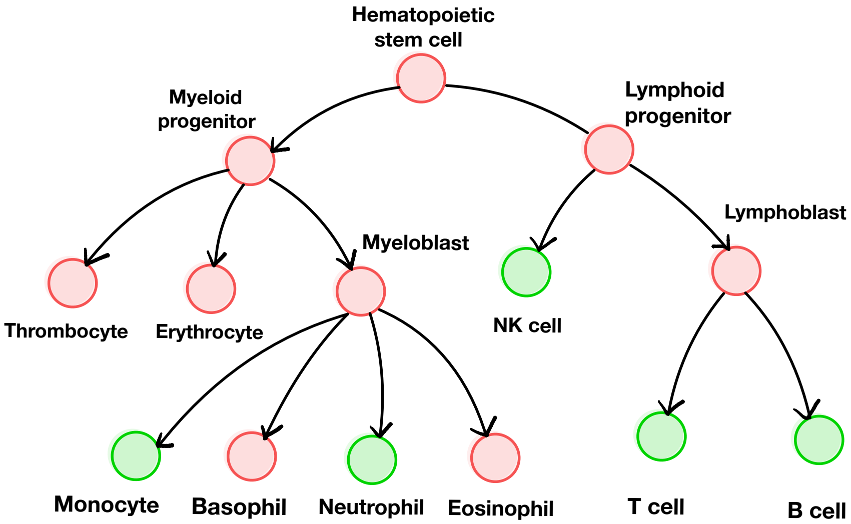


Figure 2.1: Immune cells. All cells in the blood originate from hematopoietic stem cells that are found in the bone marrow. Through multiple stages of development, they can differentiate into any of the cell types shown as leaves in the tree. Apart from thrombocytes (which produce blood platelets) and erythrocytes (red blood cells), all leaves represent types of immune cells. The cell types considered in this thesis are shown in green. The two immune cell types that we will leave out of later discussions are basophils and eosinophils. These are particularly active in fighting parasites and generally have low abundance.

Different immune cell types can, to some extent, be distinguished from each other in a microscope by visual inspection. A more precise classification requires molecular analyses. Different immune cell types differ in what proteins are expressed on their surface, and with modern techniques, these proteins can be identified for individual cells (see Section 2.2). Of particular relevance are the CD (cluster of differentiation) proteins. Table 2.1 summarizes how some of these proteins (also called markers) are expressed in some major immune cell types. There are several hundred CD markers, and more extensive overviews of CD marker expression in immune cells can be found for example in Kalina et al. (2019).

Cell Type	CD3	CD4	CD8a	CD14	CD16	CD20	CD34	CD45	CD64
B cell						●		●	
Monocyte				●				●	●
Neutrophil				●	●			●	
NK cell					●			●	
Progenitor cell							●	●	
T helper cell	●	●						●	
T killer cell	●		●					●	

Table 2.1: Protein markers for immune cell identification. Shown are some of the proteins that can be expressed on the surface of immune cells. These (and others) can be used to distinguish between different types of immune cells. Some proteins can also be used to distinguish between immune cells and other cells, such as CD45. Red dots indicate proteins normally expressed on the cell's surface. The table was constructed on the basis of information in Delves et al. (2017).

2.2 Measuring protein expression on single cells

In the previous section, we described how an immune cell type can be determined by observing which proteins are expressed on its surface. To perform such observations requires sophisticated technology, and a popular method is cytometry by time of flight (CyTOF). The description of CyTOF provided below is derived from various sources, including (Nowicka et al., 2017; Bendall et al., 2014). A key feature of CyTOF is that it enables the measurement of the expression levels of more than 40 proteins inside and on the surface of a single cell. We can identify and analyze almost all cell types by adapting which proteins we want to include in the analysis.

The idea behind CyTOF is simple but clever: rather than detecting the presence of a particular protein directly (which is hard), we translate the problem into detecting the presence of a particular metal ion (which is easy with mass spectrometry). The translation part consists of attaching a particular metal ion to each protein. To do this, one must (a) find a molecule that will attach to that specific protein (and no others) and (b) attach a metal ion to that molecule. Fortunately, a group of naturally occurring proteins, the antibodies, fit the first bill. Antibodies are an important part of our immune system, and they come in many different variants that can bind to different proteins. Before use in CyTOF, these antibodies are modified by attaching metal isotopes to them. The whole process can be summarized as follows:

CyTOF analysis

1. **Input:** A suspension where each cell is separated from the others.
2. **Tag the cells:** Mix the single-cell suspension with metal-tagged antibodies that bind to their target proteins on each cell. Now, the metals bound to a cell identify what cell type it is.
3. **Vaporize the cells:** After washing away unbound antibodies, the cells are put into small water droplets to form a mist and then sent through a hot gas to reduce each cell to many tiny particles.
4. **Mass spectrometry:** The particles are electrically charged and sent through a mass spectrometer that separates metals from each other by their mass-to-charge ratio (each metal has a unique ratio).
5. **Putting it all together:** The continuous stream of measurements is segmented into protein expression for individual cells by aggregating detected metal ions observed within a very short time window.
6. **Output**: A data file where each row represents a cell and each column represents the expression of a specific protein inside or on that cell.

2.3 Measuring cell composition

CyTOF analysis of a blood sample may involve measuring the proteins in one million cells or more. These cells obviously represent only a tiny fraction of the total cell population in the donor's blood and one may ask how representative the measurement is for the total population. Since the blood is continuously circulating through the body at high speed, we may assume that the composition is more or less the same everywhere in the circulation system. However, the composition may change over time. For example, the immune system responds dynamically to infections, and a person with an active infection may experience a sharp increase in the number of neutrophils and changes in other immune cell types as well. Cancer can also change the composition of immune cells in the blood, and later in this thesis we will investigate this for one particular type of cancer called AML. Other factors may also potentially affect the immune cell composition, such as stress.

We have so far focused on the cell surface proteins which reveal what type of cell we are dealing with. The internal proteins are useful for a different reason. These proteins can reveal important information about the actions taken by the cell. In combination, the external and internal proteins thus tell a story about what cell types we have and what each cell type is doing. The proportion of cells of each cell type is also highly informative.

2.4 Acute Myeloid Leukemia (AML)

Acute Myelogenous Leukemia (AML) is the most prevalent type of acute leukemia in adults (Appelbaum et al., 2006). In AML there is an increase in myeloid cells in the bone marrow and blood (Lowenberg et al., 1999). In healthy individuals, these cells, known as myeloblasts, typically mature into white blood cells such as granulocytes and monocytes. Only a small percentage of the cells in blood and bone marrow (about 2%) are myeloblasts in healthy individuals. In patients with AML, the myeloid blast cells fail to develop into healthy blood cells, and the percentage of such cells in the blood is much higher than in healthy individuals (De Kouchkovsky & Abdul-Hay, 2016). As a result, the blast cells will crowd out the healthy blood cells, such as white blood cells, red blood cells, and platelets.

AML is a rapidly progressing form of blood cancer with lethal outcomes in only a few months if not treated. With appropriate treatment (involving intensive chemotherapy and other treatment modalities), the 4-year survival ranges from approx. 29% to 94% depending on the disease subtype (Tangen et al., 2024). AML is also characterized by substantial intratumor cellular heterogeneity (Tislevoll et al., 2023). In other words, even within a single patient, the tumor cells can display significant variations in genetic makeup.

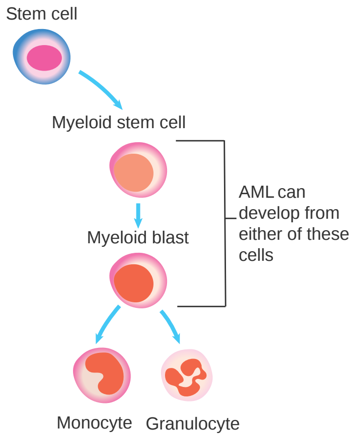


Figure 2.2: AML origin. The diagram shows the cell differentiation steps from a stem cell to mature monocytes and granulocytes (Commons, 2024). In AML, the precursors of monocytes and granulocytes become malignant through genomic alterations. This increases the number of myeloid blast cells, and the rogue cells fail to develop into monocytes or granulocytes.

The degree of intratumor heterogeneity has important clinical implications: the more genetically diverse the tumor, the more likely a given therapy fails to eradicate all disease subclones. Thus, high genetic diversity may be an indication for combination therapies involving multiple drugs (although at the risk of inducing more severe side effects).

Chapter 3

Methods

Outline of chapter

In this chapter, we present the theory behind the methods used in the thesis. We first describe the clustering algoriths K-means and hierarchical clustering. We we also describe Metaclust, which is a hybrid of K-means and hierarchical clustering. To assess the effectiveness of these methods, we employ performance metrics such as the Silhouette Score and Adjusted Rand Index, which provide insights into the compactness and agreement of clusters. Additionally, we introduce Shannon Entropy and Kullback-Leibler Divergence for feature engineering, aiding in the prediction of treatment response and survival. Lastly, we discuss fundamental survival analysis tools, including Kaplan-Meier curves, Log-Rank tests, and the Cox Proportional Hazards Model, essential for evaluating patient outcomes.

3.1 K-Means Clustering

The theory summarized in this chapter is based on James et al. (2013). K-means clustering is a popular and straightforward algorithm for partitioning a dataset into clusters based on the similarity (or, equivalently, the distance) between pairs of data points. It is based on the principle that a good data partition has minimal within-cluster variation. Suppose a cluster C contains n observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in R^p$. Then, we define the within-cluster variation of C as

$$V(C) = \frac{1}{n}\sum_{s=1}^{n}\sum_{t=1}^{n}\sum_{j=1}^{p}(x_{sj} - x_{tj})^2 \tag{3.1}$$

Defining the cluster centroid $\mathbf{c} = (1/n)\sum_{s=1}^{n}\mathbf{x}_i$ we may rewrite the equation above as follows:

$$\begin{aligned} V(C) &= \frac{1}{n}\sum_{s,t,j}((x_{sj} - c_j) + (c_j - x_{tj}))^2 \\ &= \frac{1}{n}\sum_{s,t,j}(x_{sj} - c_j)^2 + \frac{1}{n}\sum_{s,t,j}(c_j - x_{tj})^2 + \frac{2}{n}\sum_{s,t,j}(x_{sj} - c_j)(c_j - x_{tj})^2 \end{aligned}$$

The first two terms on the right hand side are easily seen to be identical, and the third term is zero since we average over differences to the mean value. After some simplification we get

$$V(C) = 2\sum_{s=1}^{n}\sum_{j=1}^{p}(x_{sj} - c_j)^2 \tag{3.2}$$

which shows that the within-cluster variation can be calculated very efficiently by simply calculating the distance from each point in the cluster to its cluster center. To minimize the sum of the within-cluster variations across all the K clusters, we seek a partition $C_1, \ldots, C_K$ of the observations that minimizes the loss function

$$L = \sum_{k=1}^{K} V(C_k) \tag{3.3}$$

At first sight, this appears to be a computationally hard problem since there are a total of K^n ways to divide n observations into K groups (or slightly less if we require all groups to contain at least one sample). For example, with $K = 10$ clusters and $n = 100$ data points, the number of possible assignments will exceed the number of atoms in the universe (estimated to be approximately equal to 10^{82} (Baker, 2021), making an exhaustive search algorithm useless.

Fortunately, good approximate solutions to the problem can be found with much less effort. The k-means algorithm is one example of this. Initially, it selects tentative cluster centers at random. These initial cluster centers are not expected to reflect the true cluster centers, and the algorithm iteratively moves them to gradually represent the true centers more accurately. The algorithm alternates between (a) assigning observations to the nearest tentative center and (b) relocating the centers to the average location of the observations assigned to them. Ultimately, the cluster assignments will stabilize, and we then say that the algorithm has converged. See Figure 3.1 for an example. Importantly, the solution found by k-means clustering is not guaranteed to be the global optimum in the sense of minimizing the criterion in Equation 3.3. It will be a local minimum of the cost function, and hopefully, this local minimum will have a loss value close to the global minimum. To ensure we have found a good local minimum, we may perform the algorithm several times for different choices of randomly selected starting points for the initial cluster centers. Next, we pick the result with the lowest value of the cost function in Equation 3.3. For more details, see Algorithm 1.

Algorithm 1: k-means clustering

Input: Observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ and the desired number of clusters k.
Output: A set of k cluster centers $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$

```
1. Randomly select the positions of the cluster centers.
2. Until assignments stop changing:
    (a) Assign every data point to the closest cluster center.
    (b) For each cluster, change the cluster center to the average
        of all the data points assigned to the cluster.
```

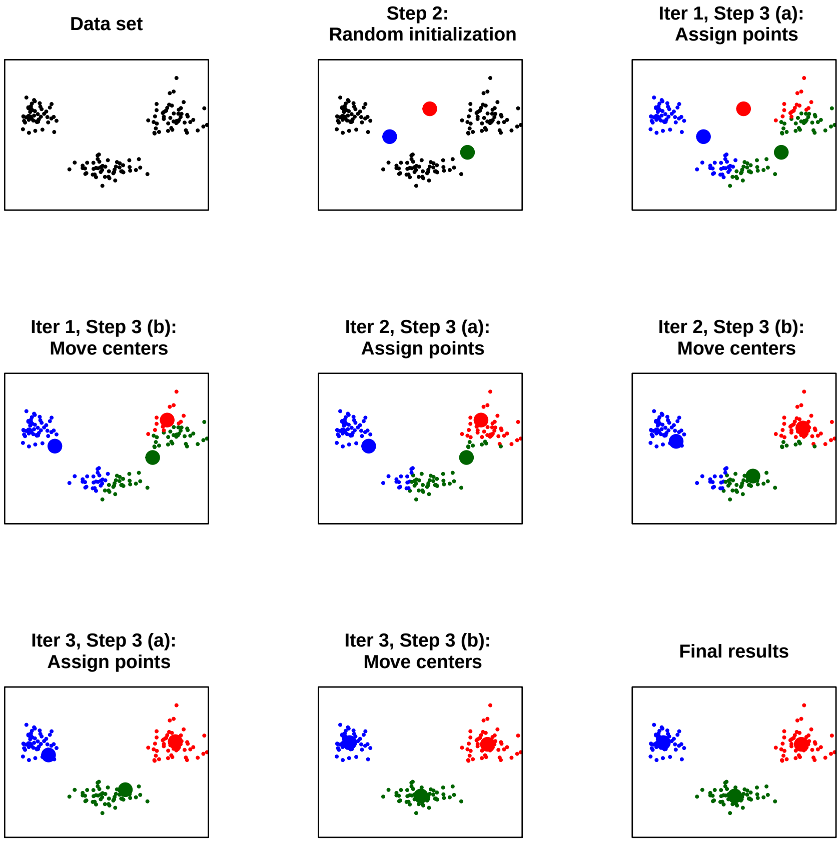

Data set
Step 2: Random initialization
Iter 1, Step 3 (a): Assign points
Iter 1, Step 3 (b): Move centers
Iter 2, Step 3 (a): Assign points
Iter 2, Step 3 (b): Move centers
Iter 3, Step 3 (a): Assign points
Iter 3, Step 3 (b): Move centers
Final results

Figure 3.1: Illustration of performing K-Means for K=3. Top left: the observations in the data set are shown. Top center: initial cluster centers are randomly selected. Top right: observations are assigned to the clusters based on the shortest Euclidean distance. Middle left: cluster centers are relocated based on the average position of assigned observations. The remaining illustrations repeat the process of assigning points and relocating clusters until assignments no longer change.

Repeating the clustering process many times with different initial centers can be time-consuming. Alternative strategies such as k-means++ have been developed to answer this challenge. K-means++ only differs from regular k-means (3.3) by the choice of initial cluster centers. Rather than choosing all centers at random, it begins by selecting *one* initial cluster center from the data points uniformly at random. Subsequent centers are chosen from the remaining observations, with the probability of selection for each point being proportional to the square of its distance from the nearest existing cluster center. This method aims to spread out the initial centers, which can potentially lead to improved clustering results and speed of convergence.

K-means clustering is widely used due to its simplicity and computational efficiency, making it suitable for a wide range of applications. However, it does have certain limitations, such as sensitivity to the initial choice of cluster centers, difficulty in clustering data of varying sizes and densities, and the assumption that clusters are spherical and evenly sized (see Figure 3.2).

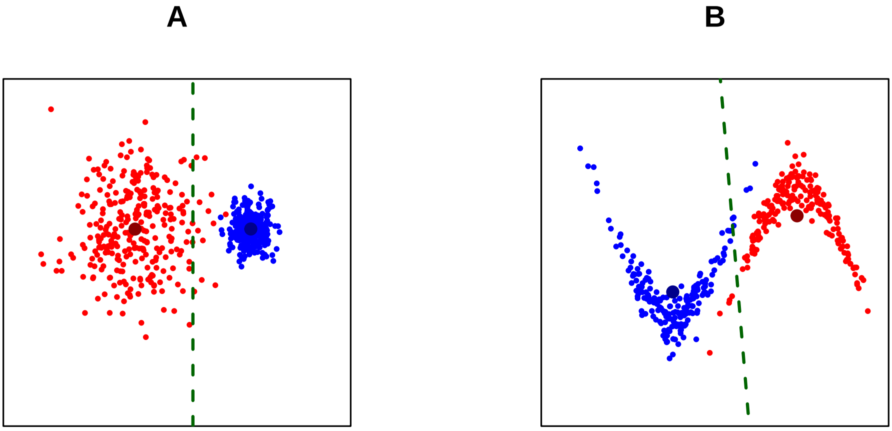


Figure 3.2: Problems with k-means. The larger points show the position of the cluster centers, and the dashed green line shows the decision boundary. Panel A illustrates the problem with k-means when there are clusters of different sizes. We see that several red points are located to the right of the boundary and thus would be assigned to the incorrect cluster. Panel B shows an example of k-means failing to capture non-spherical clusters.

3.2 Hierarchical Clustering

An alternative approach for clustering is hierarchical clustering. The starting point is a specification of a distance measure $dist(\mathbf{x}_1, \mathbf{x}_2)$ between pairs of observations and a specification of how the distance between two clusters is to be calculated. There are several choices for both distance measures.

Distance between two observations

A common choice for the distance between observations is squared Euclidean distance (or the square root of that):

$$d(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} (x_i - y_i)^2$$

Another common choice is Manhattan distance, also called the ”taxicab” distance, defined as:

$$d(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} |x_i - y_i|$$

which can be interpreted as the distance one would travel between points if one has to follow a north-south direction or an east-west direction only. Yet another distance measure is based on the cosine similarity between the observation vectors:

$$d(\mathbf{x},\mathbf{y}) = 1 - \cos\alpha = 1 - \frac{\mathbf{x}\cdot\mathbf{y}}{||\mathbf{x}||\cdot||\mathbf{y}||}$$

where α denotes the angle between the vectors $\mathbf{x}$ and $\mathbf{y}$ and $||\cdot||$ is the Euclidean (L^2) norm.

Distance between two clusters

Measures of distance between clusters are called linkage methods. In single linkage, the distance between two clusters C_1 and C_2 is defined as

$$dist(C_1,C_2) = \min\{d(\mathbf{x},\mathbf{y}) \;\text{ s.t. }\; \mathbf{x}\in C_1,\, \mathbf{y}\in C_2\}$$

According to this definition, two clusters are close to each other if at least one member of C_1 is close to at least one member of C_2. In other words, the clusters are merged based on their nearest points (see Figure 3.3). This ability allows the clustering process to trace intricate structures while merging clusters, making it particularly suited for identifying clusters with irregular, non-spherical shapes.

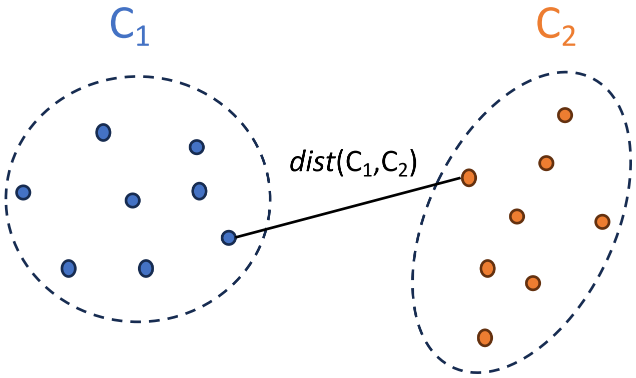


Figure 3.3: Single linkage. Illustration of the distance between clusters using single linkage.

Single linkage clustering can be sensitive to outliers due to its dependence on minimum pairwise distances. When an outlier is near another dense cluster, it can distort the actual cluster configuration by serving as a bridge to other clusters. This phenomenon, known as ”chaining,” can cause clusters to merge based on a series of points nearby rather than structural cohesion. Such a scenario can obscure meaningful relationships and result in less interpretable clusters. Figure 3.4 illustrates this. Furthermore, extreme outliler might

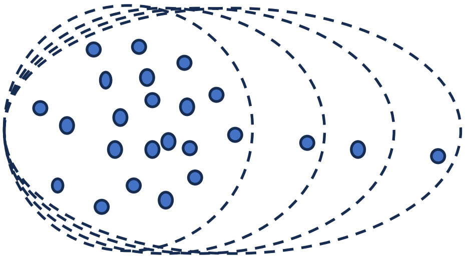


Figure 3.4: Chaining phenomenon. The outliers form a chain that distorts the true cluster configuration.

Complete linkage offers yet another solution. The distance between two clusters C_1 and C_2 is then defined as

$$dist(C_1, C_2) = \max\{d(\mathbf{x}, \mathbf{y}) \ \text{ s.t. } \ \mathbf{x} \in C_1, \ \mathbf{y} \in C_2\}$$

Here, the distance between two clusters is defined as the longest distance between any two observations where one is a member of C_1 and the other is a member of C_2. In other words, the clusters are merged based on the maximal distance between any pair of points in the two clusters (see Figure 3.5).

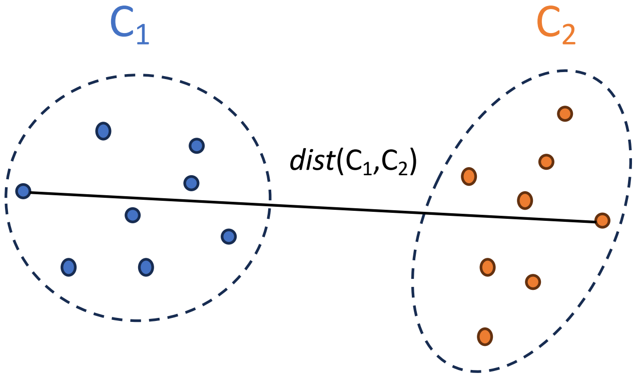


Figure 3.5: Complete linkage. Illustration of the distance between clusters using complete linkage.

Using complete linkage tends to make the clusters more spherical in shape. This happens because we always try to minimize the cluster's diameter when merging, which promotes spherical clusters' growth.

In average linkage, the distance between two clusters C_1 and C_2 is defined as the average distance between all pairs of observations, one from each cluster:

$$dist(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{\mathbf{x} \in C_1} \sum_{\mathbf{y} \in C_2} d(\mathbf{x}, \mathbf{y})$$

Here, $|C_1|$ and $|C_2|$ represent the number of observations in clusters C_1 and C_2, respectively. This method balances the influence of all members of the clusters, providing a compromise between single linkage and complete linkage. It tends to produce more compact clusters and is less prone to the chaining effect observed in single linkage. However, it can still suffer from elongated cluster shapes.

The last distance measure between clusters that we will consider in this thesis is the Ward linkage, proposed by Jh Jr (1963). The Ward linkage aims to minimize the total within-cluster variance. The distance between two clusters C_1 and C_2 is defined as the increase in the sum of squared deviations from the mean (i.e., variance) when the clusters are merged

$$dist(C_1,C_2) = \sum_{\mathbf{x}\in C_1\cup C_2} ||\mathbf{x}-\mu_{C_1\cup C_2}||^2 - \sum_{\mathbf{x}\in C_1} ||\mathbf{x}-\mu_{C_1}||^2 - \sum_{\mathbf{x}\in C_2} ||\mathbf{x}-\mu_{C_2}||^2$$

where μ_{C_1}, μ_{C_2} and $\mu_{C_1\cup C_2}$ are the centroids (means) of clusters C_1, C_2, and their union $C_1 \cup C_2$, respectively.

Ward linkage generally performs well in terms of reducing within-cluster variance and maintaining compact clusters. This makes it a preferred method for many practical applications where cluster compactness is desired. However, it may not perform as well in scenarios where the clusters have irregular shapes or significantly different sizes.

The algorithm

Hierarchical clustering starts by assigning each data point to a separate cluster. At each step of the algorithm, the two clusters that are closest to each other are identified and merged to form a single cluster. This merging process is repeated iteratively, reducing the number of clusters by one in each round until all data points are merged into a single cluster. See Algorithm 2 for details. It is common to summarize the algorithm's results in a dendrogram, a tree-like diagram that records the sequence of merges (see Figure 3.6). The dendrogram shows the composition of each cluster by drawing a ”bridge” between two clusters that have been merged. The bridge consists of two vertical lines attached to a horizontal line where the height of the latter indicates the distance between the two clusters that merge.

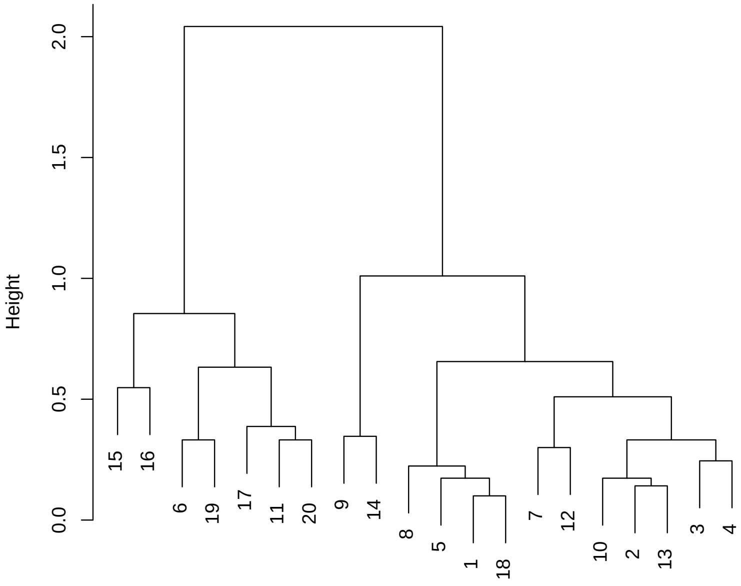


Figure 3.6: Hierarchical clustering dendrogram. The dendrogram is generated using the complete linkage method, with the y-axis representing the height or distance at which clusters are merged. The x-axis shows the indices of observations in a hypothetical dataset. This visualization illustrates how individual observations are grouped into clusters, providing insight into the hierarchical structure of the data.

A significant advantage of hierarchical clustering is the method's ability to reveal underlying structures of the data without needing to specify the number of clusters beforehand. This contrasts with methods such as K-Means, where the number of clusters has to be prespecified. The dendrogram produced by hierarchical clustering provides - as the name suggests - a visual representation of the hierarchical structure of the clustered entities. By examining the dendrogram, we aim to identify significant clusters at various levels of similarity, allowing for a better understanding of the structure than a fixed number of clusters. This flexibility makes hierarchical clustering particularly useful for exploratory data analysis, where the optimal number of clusters is not known, even to an approximation, in advance.

Algorithm 2: Hierarchical clustering

```
Input: Observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$
Output: Dendrogram (cluster tree) $\mathscr{T}$

  1. Set the leaves of $\mathscr{T}$ to be $\mathbf{x}_1, \ldots, \mathbf{x}_n$.
  2. Compute an $n \times n$ distance matrix $D = (d_{ij})$ where $d_{ij}$ is the
     distance between the $i$th and the $j$th observation.
  3. Define a pool of clusters $C_1 = \{\mathbf{x}_1\}$, $C_2 = \{\mathbf{x}_2\}$,..., $C_n = \{\mathbf{x}_n\}$.
  4. Repeat until only one cluster remains in the pool:
       (a) Identify the two clusters $C$ and $C'$ in the pool with the
           smallest pairwise distance $D^*$.
       (b) Add a bridge of height $D^*$ between $C$ and $C'$ in $\mathscr{T}$
       (c) Merge $C$ and $C'$ into a new cluster $C''$.
       (d) Insert $C''$ in the pool and remove the clusters $C$ and $C'$.
```

Imposing a minimum cluster size

When performing hierarchical clustering and cutting a dendrogram, it is necessary to specify either the number of clusters desired or the height at which to make the cut. A common issue with hierarchical clustering is that extreme outliers can often form their own small clusters due to the algorithm's deterministic nature. To address this, we propose an alternative strategy that splits the dendrogram into K clusters while imposing a minimum size requirement for each cluster.

Our approach involves iteratively increasing the number of clusters from K to $K+1$, $K+2$, and so on, until at least K clusters that meet the minimum size requirement are identified. Clusters that do not satisfy this requirement are discarded. This method ensures that the final clustering solution consists of sufficiently large and meaningful clusters, reducing the impact of outliers and small, insignificant clusters.

3.3 Metaclust

Introduction

K-means and hierarchical clustering are both very popular clustering methods, but for different reasons. The main strengths of k-means is its simplicity and scalability to large datasets. As described in Algorithm 1, we only have to calculate the distances between the observations and the centroids in each iteration, a calculation of order $O(KN)$ where K is the number of centroids and N is the number of samples. The algorithm is thus fast since it usually converges in much less than N iterations. A drawback with K-means is that the algorithm is biased towards spherical and evenly sized clusters. This will be harmful if, for example, the true clusters are strongly non-spherical. Furthermore, we must specify the number of clusters to run the algorithm.

Hierarchical clustering, on the other hand, can tackle most of these limitations. The bias towards clusters of a certain shape is still present, but the user can choose between several linkage methods to fit the data at hand. For example, single linkage can produce clusters with an elongated shape, while complete linkage favors more spherical clusters. The choice of different linkage methods thus gives the user more flexibility in the search to find the underlying structures of the data. Hierarchical clustering does not favor evenly sized clusters and commonly produces clusters of very different sizes. Also, hierarchical clustering does not require the number of clusters K to be prespecified. The drawback of hierarchical clustering is that distances between all pairs of observations must be calculated and stored. As a result, hierarchical clustering is often too memory intensive and time consuming for larger datasets. In this section, we propose an algorithm that combines K-Means and hierarchical clustering to leverage the strengths of both methods. A very similar method was proposed by Peterson et al. (2018).

The algorithm

The first step is to perform k-means on a set of input observations $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$. The purpose of this step is not to find the final clustering but rather to reduce the set of observations $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$ into a set of centroids $\mathbf{c}_1, \dots, \mathbf{c}_r \in \mathbb{R}^p$ where $r << n$. We want these centroids to represent the original set of observations sufficiently, and therefore, we should choose a high number of centroids r. In the analysis performed in this thesis, a dataset of approximately 1 million cells was reduced to a set of 500-1000 centroids. These centroids are now supposed to represent our dataset. This can be considered a sophisticated method of sampling the data, while still representing the entire dataset. See Figure 3.7 for an example. The second step of the algorithm is to cluster the centroids $\mathbf{c}_1, \ldots, \mathbf{c}_r$ further, using hierarchical clustering. This allows the freedom to choose the linkage method, enabling us to detect more intricate structures in the data than the spherical clusters we would find using k-means. Furthermore, the algorithm's output will be a dendrogram; we don't need to specify the desired number of output clusters beforehand.

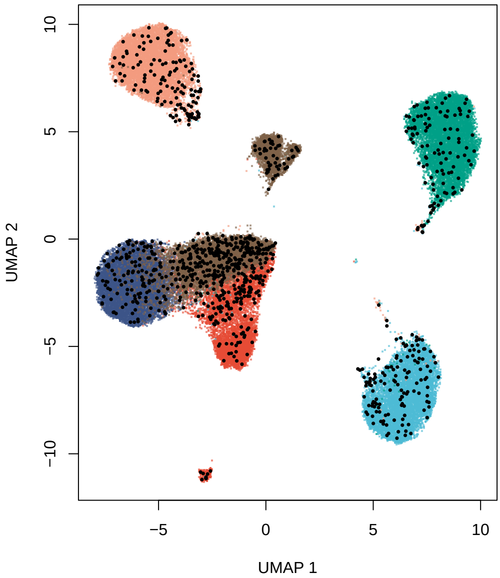


Figure 3.7: Centroid locations. UMAP showing a 2D projection of the 36-dimensional simulated dataset described in Section 3.8. The colored points show the original dataset, where the color corresponds to simulated cell types. The black points are the centroids found in step 1 of the Metaclust algorithm.

Algorithm 3: Metaclust

Input: Observations $\mathbf{x}_1,\ldots,\mathbf{x}_n \in \mathbb{R}^p$, desired number of clusters Q in the preliminary clustering, and desired number of clusters K in the final clustering.
Output: Dendrogram (cluster tree) $\mathscr{T}$

```
1. Perform k-means to find clusters $\{C_1,\ldots,C_r\}$
2. Calculate cluster centroids $\{\mathbf{c}_1,\ldots,\mathbf{c}_r\}$
3. Perform hierarchical clustering using the centroids as input
4. Cut the dendrogram to produce $K$ clusters
```

Using the k-means output as input to hierarchical clustering circumvents the time complexity limitations of hierarchical clustering while we still can assign labels to the entire set of observations $\mathbf{x}_1,\ldots,\mathbf{x}_n \in \mathbb{R}^p$. This approach is inspired by the clustering algorithm FlowSOM. This algorithm identifies multiple smaller clusters using the self-organizing map (SOM) algorithm. These clusters are then combined to form so-called ”metaclusters”. In Chapter 5, we will consider the performance of Metaclust, using Ward linkage.

3.4 Panel guided clustering

In all the clustering methods discussed so far, the samples are represented by vectors $\mathbf{x} = (x_1,\ldots,x_p)$ in some feature space X. This space could be simply $X = \mathbf{R}^p$ for some $p > 0$, or it could be something else, including a subset of Euclidean space. This feature space must be endowed with some distance measure $d(\mathbf{x},\mathbf{y})$ to allow the clustering algorithm to determine how close two samples are to each other. We have already discussed some possible distance measures when X is Euclidean space. These distance measures were symmetric in the features, i.e., all features counted equally in the distance calculation. This is reasonable if the features are all on the same scale and equally important. In some cases, however, we wish to impose some structure on the clustering by assigning weights to individual features.

For example, suppose we wish to cluster cells from a CyTOF experiment and are particularly interested in T killer cells. In that case, we may want to emphasize features distinguishing such cells from others. This could be accomplished by assigning large weights to the features CD3 and CD8, and low weights to other features. This could be implemented as a distance measure

$$d(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} w_i(x_i - y_i)^2 \tag{3.4}$$

where $w_i > 0$ for the features of interest and $w_i = 0$ for other features.

As an extension, suppose we are interested in clustering a cell population with several types of cells (e.g., T killer cells, T helper cells, and monocytes), each characterized by their own set of features. We could then define several distance measures, each tailored to distinguish cells of a certain type from other cells and perform multiple clusterings with respect to each panel of features. Finally, we could merge the results of these clusterings to form a final clustering of all cells.

We will pursue this idea further in Chapter 4 and introduce a new clustering algorithm to incorporate prior knowledge about expected combinations of protein markers for distinct cell types. By leveraging this information, we aim to enhance both the accuracy and reliability of the clustering outcomes. Our objective is to refine the algorithm's ability to organize the data into clusters, with each cluster accurately reflecting a specific, predefined cell type.

3.5 Silhouette Score

Clustering data is an unsupervised problem, meaning we cannot know the correct partition (if any) of the observations we aim to cluster. Consequently, we cannot measure the accuracy of clustering using a test set as we do in supervised learning. Moreover, determining the appropriate number of clusters is a fundamental challenge in clustering algorithms. We can inspect the dendrogram for hierarchical clustering to identify a natural division of clusters. In contrast, we must predefine the number of clusters for K-Means before executing the algorithm. While there is no universally correct method to select the optimal number of clusters, the silhouette score was proposed by Rousseeuw (1987) as a valuable metric to guide this decision.

The silhouette score evaluates how well an observation fits within its assigned cluster compared to other clusters. To quantify how well an observation i fits within its assigned cluster, we calculate the average distance from i to all other observations in the same cluster, denoted as $a(i)$. To assess how well-separated the observation is from other clusters, we define $b(i)$ as the average distance between observation i and all points in the nearest neighboring cluster. The silhouette score for observation i is then defined as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{3.5}$$

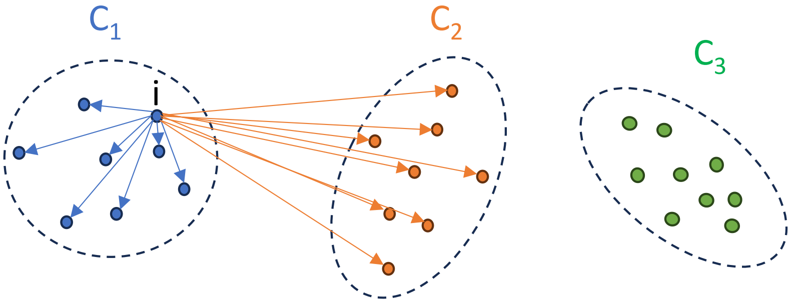


Figure 3.8: Calculation of silhouette score. Illustration of the calculation of $a(i)$ and $b(i)$ using euclidean distance. $a(i)$ is given by the average length of the blue arrows, and $b(i)$ is given by the average length of the orange arrows. Notice that no observation in C_3 affects the silhouette score of observation i.

As a result of scaling by $max(a(i), b(i))$, the silhouette score ranges from -1 to 1. A score close to 1 indicates that the observation is very similar to other points in its cluster and well-separated from other clusters. A score near 0 suggests the observation is on the boundary between two clusters, where $a(i) \approx b(i)$. A negative score implies that the observation may have been assigned to the wrong cluster. By averaging the silhouette scores of all observations in a dataset, we can obtain an overall measure of clustering quality. When applying a clustering algorithm, we can compute the average silhouette score for different numbers of clusters and select the number corresponding to the highest average score. While the silhouette score is intuitive and easy to interpret, it has a significant drawback: computing all pairwise distances between observations can be computationally intensive.

However, this issue can be somewhat mitigated by using a sampled dataset to determine the number of clusters that maximize the silhouette score and then applying this optimal number to cluster the full dataset.

3.6 Adjusted Rand Index

Assume we have two different groupings of the same set of observations, resulting in two distinct label assignments for each observation. For example, we may have performed two different clustering algorithms on the observations. It can often be informative to know how similar these two groupings are. We assume here that the group labels are arbitrary in each of the two groupings to be compared, so just comparing the labels for each observation in both groupings is not informative. One popular approach proposed by Rand (1971) introduces a metric called the Rand Index (RI) that considers all pairs of observations and quantifies how often the two groupings agree on their label assignments of the two observations. The idea is as follows. Let $U(x)$ and $V(x)$ denote the group labels for observation x in the first and second grouping, respectively. Consider a pair of distinct observations x and y. If $U(x) = U(y)$, then the observations are assigned to the same group in the first grouping. If we also have $V(x) = V(y)$, then the observations are also assigned to the same group in the second grouping, and we say that the pair of observations x and y is a true positive (TP). Similarly, if $U(x) \neq U(y)$ and $V(x) \neq V(y)$, then we say that the pair is a true negative (TN). The Rand Index (RI) is the proportion of all possible pairs of observations that are either TN or TP. Formally, we define it as

$$RI = \frac{TP + TN}{\binom{n}{2}} \tag{3.6}$$

where the denominator $\binom{n}{2} = n!/(2!(n-2)!)$ is the number of ways of picking two elements out of n elements when we ignore the order in which the elements are picked. While the Rand Index effectively compares two groupings, it has a flaw. The metric does not account for the possibility of agreement by chance. This means that even two random and independent groupings could have a high RI simply because they coincidentally agree on some pairs of observations. To address this, Hubert & Arabie (1985) proposed to correct (adjust) the Rand Index for the expected similarity between two random groupings, and also normalized this index to the range $[-1,1]$. This alternative metric is called the Adjusted Rand Index (*ARI*), which provides a measure that ranges from -1 to 1. *ARI* is defined as

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]} \tag{3.7}$$

Here, $\mathbb{E}[RI]$ is the expectation of the Rand Index for a random clustering, and $\max(RI)$ is the maximum possible value of the Rand Index. Details on the calculation of $\mathbb{E}[RI]$ and $\max(RI)$ can be found in Hubert & Arabie (1985). *ARI* has an expected value of zero when the agreement is due to chance alone and is equal to one when the groupings are identical. *ARI* can also be negative, implying that the agreement between the groupings is worse than what we would expect by chance.

3.7 Fowlkes-Mallows Index

An alternative metric that can be used to compare two different groupings of a set of observations is the Fowlkes-Mallows Index (FMI), introduced by Fowlkes & Mallows (1983). The FMI focuses on precision and recall, making it useful for evaluating how well the clustering algorithm identifies pairs of points that belong to the same cluster. FMI is particularly suitable for binary or two-class problems where precision and recall are critical metrics. Unlike the Adjusted Rand Index (ARI), which adjusts for chance, FMI directly evaluates the clustering performance based on true positive and false positive rates. That being said, the clustering performed in this thesis mainly handles multi-label data, and consequently, ARI will be the more suitable metric for measuring the similarity.

3.8 A simulation model for CyTOF data

A simulation model was developed to evaluate the performance and robustness of clustering algorithms on CyTOF data. This model was based on manually gated data from the original CyTOF dataset. Manual gating identified six significant cell populations: B cells, CD4$^+$ T cells, CD8$^+$ T cells, monocytes, NK cells, and neutrophils. These populations were used to calculate statistics for each cell type, forming the basis for the simulation model. Each cell type was separately modeled using a Gaussian Mixture Model (GMM), allowing for covariances between marker expressions to be estimated. Real CyTOF data that has not been normalized has non-negative marker expressions. Any expression value below zero was truncated to zero to ensure the non-negativity property was present in the simulation model.

A GMM is a mixture of several normal distributions. The probability density function of a GMM with C components can be written as

$$p(\mathbf{x}|\boldsymbol{\beta}) = \sum_{c=1}^{C} \pi_c \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

where the $\pi_c > 0$ determine the relative weighting of the different components and satisfies $\sum_{c=1}^{C} \pi_c = 1$, and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \Sigma_c)$ is a (multivariate) normal distribution with mean $\boldsymbol{\mu}_c$ and covariance matrix Σ_c. The parameters $\{\pi_c, \boldsymbol{\mu}_c, \Sigma_c : c = 1, \ldots, C\}$ are estimated using the Expectation-Maximization (EM) algorithm. This iterative technique finds maximum likelihood estimates of parameters in probabilistic models by alternating between the Expectation step (E-step) and the Maximization step (M-step). The E-step involves calculating the expected value of the log-likelihood function with respect to the current distribution estimate. In the M-step, the expected log-likelihood found in the E-step is maximized with respect to the model parameters. The alternation between these steps continues until the parameter changes stop or fall below some threshold. Details and proof of convergence are found in Laird et al. (1987).

The rationale for employing a simulated model with multiple components lies in the inherent complexity and diversity within the identified cell types. In most cases, the cell types identified through manual gating can be further differentiated into subtypes. For instance, manual gating may identify CD8$^+$ T cells, but within this population, there are subtypes such as CD8$^+$CD45RA$^+$ naive T cells and CD8$^+$CD45RO$^+$ memory T cells (Russo et al., 2019). Similarly, NK cells can be divided into CD56$^+$CD16$^-$ naive NK cells and CD56$^+$CD16$^+$ mature NK cells.

These examples illustrate that each manually gated cell type comprises multiple subtypes with distinct protein expression profiles. Consequently, it is reasonable to assume that each cell type's observed protein expression distribution results from a mixture of several underlying distributions. By modeling the data with up to three components, we better capture the heterogeneity within each cell type, leading to a more accurate and robust simulation model.

3.9 Uniform Manifold Approximation and Projection (UMAP)

The dataset in this thesis comprises cells with 36 measured protein expressions each. Dimensionality reduction techniques are essential to visualize such high-dimensional data. While Principal Component Analysis (PCA) is a common method, it fails to capture sufficient variance in the first few components for effective visualization in this dataset.

An alternative approach is t-SNE. This technique is used to visualize the same dataset in Tislevoll et al. (2023). tSNE focuses on preserving local structures but has a quadratic time complexity ($O(N^2)$), making it computationally expensive for large datasets. A recent advancement is Uniform Manifold Approximation and Projection (UMAP), introduced by McInnes et al. (2018), which offers a more efficient time complexity of $O(N \log N)$. This makes UMAP more scalable when large datasets are considered, and it has even been shown to preserve global data structures better (Murphy, 2022)

UMAP constructs a high-dimensional graph by connecting each observation to its nearest neighbors based on a chosen distance metric (e.g., Euclidean distance). The edges are weighted to reflect similarities between observations. The number of neighbors considered in the initial graph is the main parameter of the method, with values typically ranging from 5 to 100. Increasing the number of neighbors enhances the preservation of global structures but can dilute local details. The high-dimensional graph is then projected into lower dimensions through a series of steps, including graph construction, spectral embedding, and optimization via cross-entropy minimization. While this rather complex iterative process is detailed by McInnes et al. (2018), it essentially aims to maintain both local and global structures and does so more effectively than other methods.

The resulting low-dimensional representation can be used for visualization, cluster identification, or as a reduced feature set for further analysis.

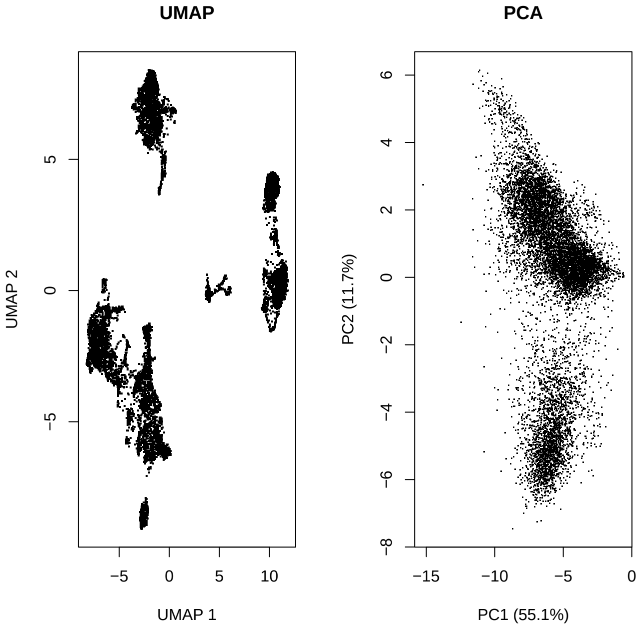


Figure 3.9: UMAP vs PCA. Dimensionality reduction with UMAP and PCA was performed on a sample of 10,000 cells from the 36-dimensional dataset considered in this thesis. UMAP was performed with the 10 nearest neighbors considered in the initial high-dimensional graph. The left plot shows the two-dimensional projection resulting from UMAP, and the right plot shows the first two principal components found by PCA. We clearly see that UMAP reveals clearer, more distinct clusters than PCA, which is more spread out, demonstrating UMAP's superiority in preserving local and global relationships in high-dimensional data.

3.10 Shannon Entropy

In information theory, Shannon entropy is a common metric for quantifying the unpredictability or diversity inherent in a discrete probability distribution. It is also commonly used to assess whether an observed frequency distribution is centered on a few of the possible outcomes or is more evenly spread out across all outcomes. Suppose $p(i)$ represents the probability (or observed relative frequency) of the ith outcome where $i = 1, 2, \ldots, n$. Then, the Shannon entropy is defined as (MacKay, 2003)

$$H = -\sum_{i=1}^{n} p(i) \log p(i) \tag{3.8}$$

where the base b of the logarithm is context-dependent. In information theory, $b = 2$ is the most common choice, as this allows a natural interpretation of the entropy as the number of bits of information. In general, if we let b equal the number of possible outcomes, the entropy will always be in the range $[0,1]$. When some of the $p(i) = 0$, we adopt the convention that $0 \times \log 0 \equiv 0$. The rationale for this choice is that $\lim_{\theta \to 0} \theta \log \theta = 0$ (MacKay, 2003).

Shannon entropy has many interesting properties. For example, we always have $H \geq 0$. To see this, note that since probabilities are restricted to the interval $[0,1]$, we must have $p(i) \geq 0$ and $\log p(i) \leq 0$. Hence, the product of the two must be negative, and the negative sum of such terms must be non-negative. $H \geq 0$ holds with equality if and only if $p(i) = 1$ for any given i, indicating a scenario where no diversity is present in the distribution.

Furthermore, the maximal value of H is achieved and equal to $\log n$ if and only if all outcomes are equally probable (i.e., $p(i) = \frac{1}{n}$ for all i) (Shannon, 1948). To see this, let (for notational convenience) $p(i) = p_i$ for $i = 1, \ldots, n$. Then, according to Gibbs' inequality (Brémaud, 2012), we have

$$-\sum_i p_i \log p_i \leq -\sum_i p_i \log q_i \tag{3.9}$$

where $q(i) = q_i$ is any other discrete probability distibution with possible outcomes $i = 1, \ldots, n$ that may or may not be identical to $p(i)$. Furthermore, the two sides are identical if and only if $p_i = q_i$ for all i. Since the left-hand side of the inequality is the entropy, we may write the above inequality as

$$H \leq -\sum_i p_i \log q_i \tag{3.10}$$

Suppose we let the second probability distribution be uniform, i.e., let $q_i = 1/n$ for all i. Then (using the logarithm with base n), we have $\log q_i = -\log n = -1$ and the inequality becomes $H \leq \sum(1/i)n = 1$. Thus, assuming we use logarithm with base equal to the number of possible outcomes n, we have $H \leq 1$, with equality if and only if the distribution p_i is uniform.

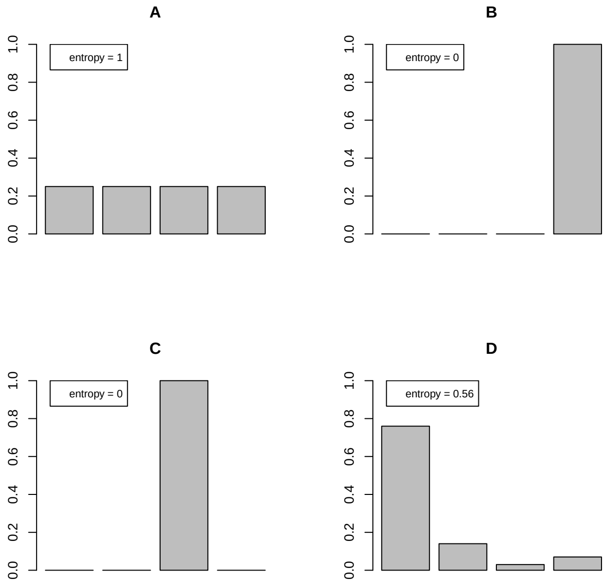


Figure 3.10: Entropy for different discrete distributions. The entropy is calculated using Equation 3.8 with the logarithm of base 4, s.t. $0 \leq H \leq 1$. In plot A we see that the uniform distribution maximizes the entropy. Plot B and C show that when $p(i) = 1$ for any i, there is no diversity, and $H = 0$.

3.11 Kullback-Leibler divergence

Entropy measures uncertainty or diversity within a single probability distribution. On the other hand, Kullback-Leibler (KL) divergence, also known as relative entropy, quantifies how one probability distribution differs from another. It is also commonly used to compare two observed frequency distributions over the same outcomes. If we have two distributions $p(i) = p_i$ and $q(i) = q_i$ for $i = 1, \dots, n$, the KL divergence is defined as

$$D(p||q) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right) \tag{3.11}$$

This measure is asymmetric, meaning that in general, $D(p||q) \neq D(q||p)$ (MacKay, 2003). KL divergence is also non-negative. To see this, we can rearrange Gibbs' inequality from Equation 3.9:

$$\begin{aligned}
-\sum_i p_i \log p_i &\leq -\sum_i p_i \log q_i \\
\sum_i p_i \log p_i &\geq \sum_i p_i \log q_i \\
\sum_i p_i \log p_i - \sum_i p_i \log q_i &\geq 0 \\
\sum_i p_i (\log p_i - \log q_i) &\geq 0 \\
\sum_i p_i \log\left(\frac{p_i}{q_i}\right) &\geq 0
\end{aligned}$$

with equality if and only if $p_i = q_i$ for all i.

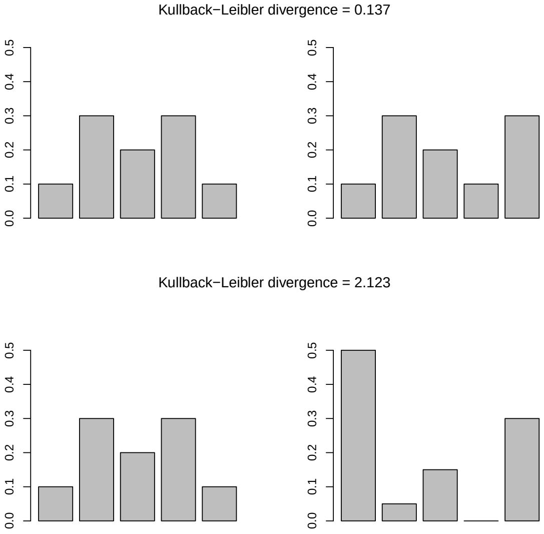


Figure 3.11: Illustration of Kullback-Leibler divergence. The top panel depicts two closely resembling distributions, resulting in a relatively low KL divergence. In contrast, the bottom panel displays significantly dissimilar distributions, leading to a higher KL divergence.

3.12 Survival data

Suppose we have a study population where we have observed the time lapsed between a predefined start point and some predefined event for each individual. For example, the study population could be all patients enrolled in a clinical study, the start point might be the time of inclusion in the study, and the event might be death from any cause. However, some patients may still be alive at the end of the study period considered, and what happens to them beyond that point remains unknown to us. For such patients, the observed time will be the last follow-up time, and we say that the survival time is right-censored. A patient may also withdraw from the study or be lost to follow-up before the end of the study period and before they experience the event. For such patients, the survival time will be the last observation, and these survival times are also referred to as right-censored. There are other types of censoring, but we will focus on right-censored data as this is the most prevalent type of censoring and is the case of the dataset considered in this thesis analysis.

In principle, we may analyze survival data using similar methods to those we use to analyze other continuous valued measurements. For example, we could, in principle, perform regression with survival times as response values using a generalized linear model (GLM) with a gamma distribution (see Figure 3.13). This, however, would not take into account that some time points correspond to events and others to time of censoring. We could circumvent this problem by simply discarding the censored data points and fitting the model to the event times only. This, however, would lead to reduced statistical power since the censored time points are informative. It would also lead to a biased estimator (ref).

Several methods have been developed for analyzing survival data. These enable more accurate and comprehensive survival probabilities and hazard rate estimates. Some of the common ones are discussed below.

3.13 Modeling survival times

It is common in survival analysis to model observed survival times as realizations of a continuous random variable T. Suppose the probability density function (or PDF) of this variable is $f(t)$ where $t \geq 0$ so that the

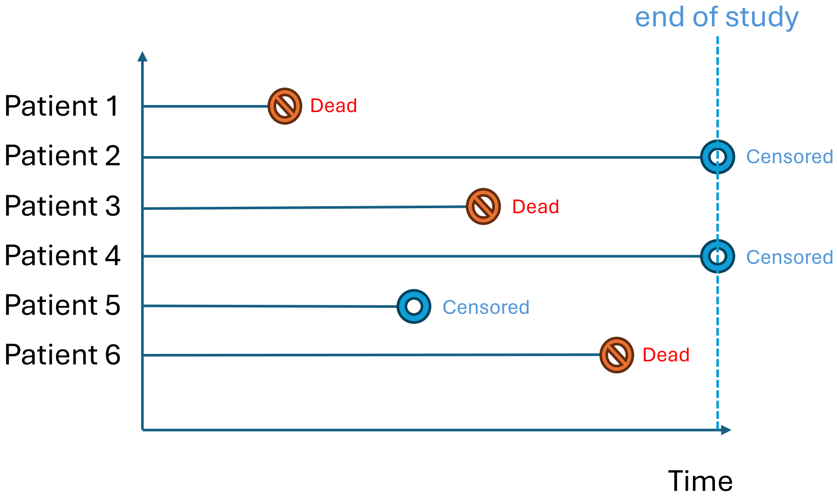


Figure 3.12: Swimmer's plot of survival data with right-censoring. Patient 1, 3 and 6 experience the event (i.e. death). Patient 2 and 4 are alive at the end of the study and therefore censored. Patient 5 has dropped out of the study before the end of the study period and before experiencing the event.

probability of T occurring in the interval $[a,b]$ is $Pr(a \leq T \leq b) = \int_a^b f(s)\,ds$. The PDF contains all the information there is about the distribution, but it is often more convenient and intuitive to consider the survival function

$$S(t) = Pr(T > t) = \int_t^\infty f(s)\,ds \tag{3.12}$$

which gives the probability that an individual survives at least until time t. For example, in a clinical study, we would often be interested in comparing the survival $S_1(t)$ in a patient population receiving a new treatment with the survival $S_2(t)$ in a patient population receiving standard treatment. If $S_1(t) > S_2(t)$ for a given time point t, we would conclude that more patients survive until time t under the new treatment than under standard treatment. We may still have $S_1(t') < S_2(t')$ for another time point t', in which case it matters which time point we focus on when determining which treatment is best. This difficulty is avoided if one assumes *proportional hazards*. This will be described in more detail later.

The survival function satisfies $S(0) = 1$ and is a non-increasing function of time with $S(t) \to 0$ as $t \to \infty$. Thus the way survival is defined, all patients will eventually experience the event. Whether we observe this event or

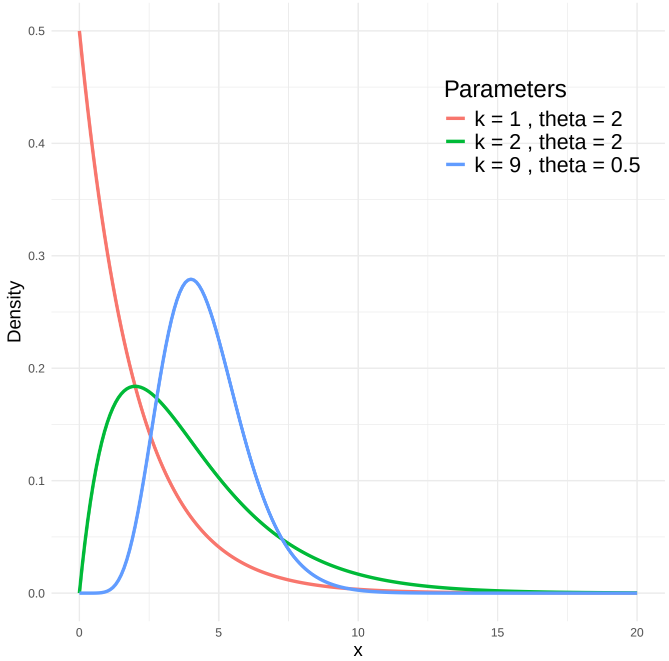


Figure 3.13: Gamma distribution for different shape (k) and scale (θ) parameters

not, however, depends on whether the patient is censored before the event happens or not. Figure 3.14 shows an example of a survival function.

Yet another way to describe a survival distribution is through the hazard function $h(t)$. This function represents the instantaneous risk of experiencing the event at time t given that they have survived until that time. It is defined as (Aalen et al., 2008)

$$h(t) = \frac{f(t)}{S(t)} \tag{3.13}$$

and can be interpreted as the instantaneous probability of the event happening at time t adjusted for the proportion of the population still at risk. The cumulative hazard

$$H(t) = \int_0^t h(s)\,ds \tag{3.14}$$

is the total accumulated hazard at time t. The relationship between the survival and the cumulative hazard is given by $S(t) = e^{-H(t)}$ (ref).

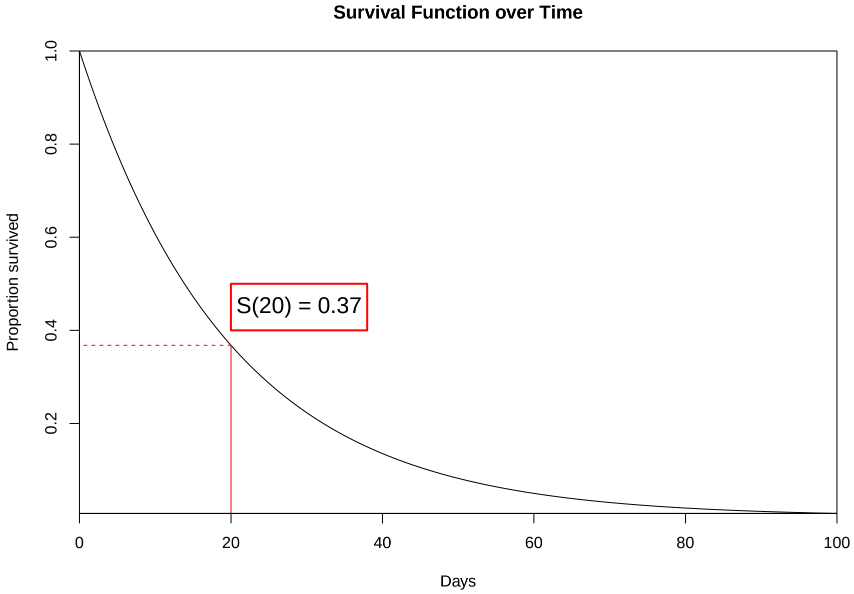


Figure 3.14: Survival function. Illustration of the survival function $S(t)$ for hypothetical survival data. We see that $S(20) = 0.37$, which means that the probability of a patient surviving at least 20 days from the start of the study is 37%.

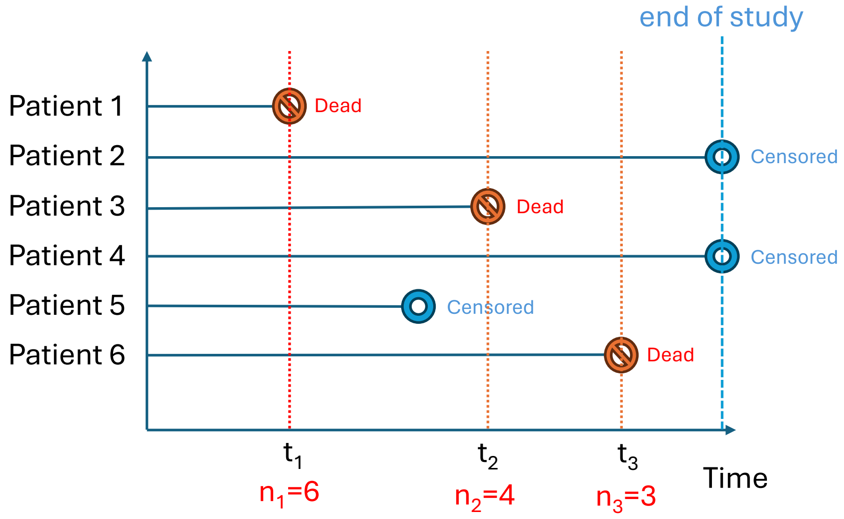


Figure 3.15: Hypothetical survival data

3.14 The Kaplan-Meier estimator

One of the first objectives in a survival analysis is often to estimate the survival curve $S(t)$. This is straightforward for uncensored data, as it is simply the proportion of subjects alive at any time. To accommodate right-censored data, we may instead use the Kaplan-Meier estimator. The starting point is a set of observed survival times, some of which represent events and some of which may represent right-censored observations. Let $t_1, t_2, \dots, t_n$ denote the unique event times and denote the number of subjects experiencing the event at time t_i as d_i and the number of subjects at risk (i.e. still not experienced the event and still not censored) at time t_i as n_i. The Kaplan-Meier estimator of the survival $S(t)$ is then defined as:

$$\hat{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \tag{3.15}$$

where $t_i \leq t$ is a shorthand notation for the set of indices i that satisfy the inequality. The formula implies that the survival estimator is piecewise constant with steps at the event times.

Example. Consider the survival data in Figure 3.15. There are six patients and three unique event times t_i, each with $d_i = 1$ $(i = 1, 2, 3)$. The number at risk at t_1 is $n_1 = 6$, so according to Eq.3.15 we have:

$$S(t_1) = 1 - \frac{d_1}{n_1} = 1 - \frac{1}{6} = \frac{5}{6}$$

The number at risk at t_2 is $n_2 = 4$ since one patient has died and one has been censored. Thus, $1 - \frac{d_2}{n_2} = \frac{3}{4}$ of the patients at risk survive past t_2. The overall probability of surviving past t_2 is therefore

$$S(t_2) = (1 - \frac{d_1}{n_1}) \cdot (1 - \frac{d_2}{n_2}) = S(t_1) \cdot (1 - \frac{d_2}{n_2}) = \left(\frac{5}{6}\right)\left(\frac{3}{4}\right) = \frac{5}{8}$$

Similarly, we find that $\frac{2}{3}$ patients at risk survive past t_3 and the overall probability of surviving past t_3 is

$$S(t_3) = S(t_2) \cdot \frac{2}{3} = \left(\frac{5}{8}\right)\left(\frac{2}{3}\right) = \frac{5}{12}$$

The resulting function is shown in Figure 3.12.

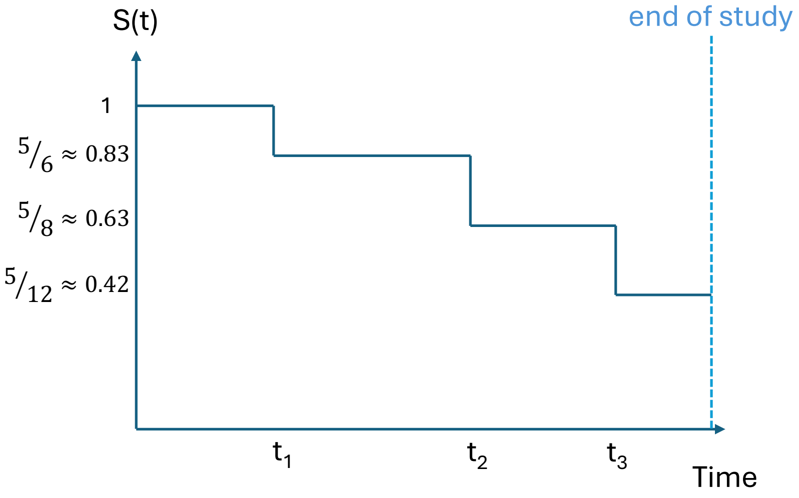


Figure 3.16: Kaplan-Meier survival curve. The survival function is estimated from a hypothetical clinical study. Each step down in the curve represents an event, and points at which the curve flattens indicate periods without events.

The Kaplan-Meier estimator is a non-parametric estimator since it does not assume anything about the probability distribution for the event times. This makes the estimator very versatile and applicable in situations where the distribution of survival times is unknown. Moreover, in section 3.15, we describe how two or more Kaplan-Meier curves can be compared using the log-rank test to determine if there are statistically significant differences between the survival functions of different groups. For instance, in clinical studies comparing two or more treatments, Kaplan-Meier curves allow researchers to visually and statistically compare the probability of survival over time among different treatment groups. This can be crucial in understanding which treatments are more effective at treating the disease.

3.15 Log-rank test

Suppose we conduct a clinical study where the patients are divided into a number of groups that receive different treatments, and for each individual, we record the survival. To assess whether the survival in all the groups is identical or not, we may apply the log-rank test. A small p-value indicates that not all the groups share the same survival. Another way of thinking of the log-rank test is that of a method for assessing the association between a categorical variable (with a finite number of levels) and survival. A small p-value is then an indication of an association being present.

The intuition behind the log-rank test is that a difference in survival in two or more groups must be linked to a difference in the proportion of events in the groups. Thus, by comparing the proportion of events in the groups, we can determine whether survival is different or not. For simplicity, we focus on the case with only two groups. The proportions of events in the groups can change over time, so we first determine the proportion experiencing the event at each individual time point. We do this by examining the number of events and the number at risk in each group at each time point t_i and calculating the ratio between the two. We next summarize all the proportions in a single score.

A more formal description of this statistical test called the log-rank test, is as follows. Define $t_1, \ldots, t_n$ as the times at which events occur in either of the two groups and let d_{1i} and d_{2i} be the number of events at time t_i in the two groups. Also, let n_{1i} and n_{2i} denote the number of subjects a risk just prior to time t_i in the two groups. Finally, let $d_i = d_{1i} + d_{2i}$ and $n_i = n_{1i} + n_{2i}$. We wish to test the null hypothesis that the two groups have identical hazard functions:

$$H_0 \ : \ h_1(t) = h_2(t) \quad \text{for all } t$$

We do this by comparing the total number of events in group 1 with the expected number of events if the null hypothesis is true. The former is given by $\sum_{i=1}^{n} d_{1i}$ and the latter by $\sum_{i=1}^{n} \frac{d_i}{n_i} n_{1i}$, so the difference between the two is

$$D = \sum_{i=1}^{n} d_{1i} - \sum_{i=1}^{n} \frac{d_i}{n_i} n_{1i}$$

Normalizing the above difference by its standard deviation s under the null hypothesis leads to the log-rank test statistic $W = D/s$, which can be shown to follow an approximate standard normal distribution (James et al., 2013). This enables us to calculate a p-value for the null hypothesis of no survival difference between the two groups. An expression for the standard deviation is (James et al., 2013)

$$s = \sqrt{\sum_{k=1}^{n} \frac{d_k (n_{1k}/n_k)(1 - n_{1k}/n_k)(n_k - d_k)}{n_k - 1}}$$

The above approach takes into account censored subjects since the number at risk includes all subjects still being observed at the given time point. This includes all subjects experiencing an event and all subjects being censored at this time point or later. Another useful feature of the log-rank test is that it makes minimal assumptions about the underlying survival distributions. It is non-parametric, making it robust and widely applicable in various contexts where the form of the survival distribution is unknown or difficult to model. However, we have to assume that the hazard functions for the compared groups are proportional over time. This is called the proportional hazards assumption and implies that the ratio of the hazard functions between the two groups is constant over time. If one group has a higher risk of the event occurring at a given time, this increased risk is assumed to be consistent across all time points. The log-rank test is most potent and appropriate when this assumption holds. If we violate the proportional hazards assumption, the log-rank test results may be unreliable, and other methods may be more suitable.

3.16 Cox's proportional hazards model

As described above, the log-rank test can be used to assess the association between a categorical variable and survival. Sometimes, the categorical variable is derived from a continuous variable through thresholding. For example, we may have a continuous variable representing a risk score and a derived categorical variable with levels ”low,” ”intermediate,” and ”high,” depending on the magnitude of the score. The log-rank test cannot be used to assess directly the association between the continuous variable and survival, but Cox's proportional hazards model - also called Cox regression - can be used for that purpose. In fact, Cox regression allows estimation of the strength of association between both continuous and categorical variables on survival. Furthermore, Cox regression allows investigation of the joint association of multiple variables with survival.

Cox regression is, in a sense, the closest analogy to linear regression for survival data. Just as linear regression models the relationship between a dependent variable and one or more independent variables by estimating their coefficients, Cox regression models the relationship between the hazard function and covariates, estimating the impact of each covariate on the risk of the event occurring over time.

In the proportional hazards model, the hazard function for an individual i with feature vector $\mathbf{x}_i$ at time t is given by (Cox, 2018)

$$\begin{aligned} h(t \mid \mathbf{x}_i) &= h_0(t)\exp(\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \qquad (3.16)\\ &= h_0(t)\exp\sum_{j=1}^{p} \beta_j x_{ij} \qquad (3.17) \end{aligned}$$

where $\beta_1, \ldots, \beta_p$ are the parameters of the model. Note that there is no intercept term β_0 in the model; this is encapsulated in the baseline hazard function $h_0(t)$ which represents the hazard function for an individual with all features equal to zero (i.e. $x_{i1} = \ldots = x_{ip} = 0$). Note that $h_0(t)$ is not dependent on the feature vector $\mathbf{x}_i$ and thus is shared across all individuals.

The proportional hazards model, as the name suggests, makes the assumption of proportional hazards, which implies that the hazard ratios between any two individuals are constant over time. Specifically, the ratio of the hazard functions for two individuals $i = 1$ and $i = 2$ is given by:

$$\frac{h_1(t)}{h_2(t)} = \frac{h_0(t)\exp(\sum_{j=1}^{p} \beta_j x_{1j})}{h_0(t)\exp(\sum_{j=1}^{p} \beta_j x_{2j})} = \frac{\exp(\sum_{j=1}^{p} \beta_j x_{1j})}{\exp(\sum_{j=1}^{p} \beta_j x_{2j})} \tag{3.18}$$

Since the baseline hazard function $h_0(t)$, which is the only time-dependent factor, is canceled out, we see that the hazard ratio does not depend on time. As a consequence, we must assume that the groups are not experiencing significantly different risks at different times.

The parameters of the proportional hazards model, $\beta_1, \beta_2, \ldots, \beta_p$, are estimated using the partial likelihood method as proposed by Cox (1972).

Unlike the full likelihood, which requires specifying the baseline hazard function $h_0(t)$, the partial likelihood focuses on the order in which events occur. This sequential approach is similar to the Kaplan-Meier estimator and the log-rank test. The partial likelihood is given by

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{\exp(\sum_{j=1}^{p} \beta_j x_{ij})}{\sum_{i^* \in R(t_i)} \exp(\sum_{j=1}^{p} \beta_j x_{i^* j})} \tag{3.19}$$

Here, n is the number of observed events, x_{ij} is the j-th feature value for the individual experiencing the event at time t_i, and $R(t_i)$ is the set of patients at risk of experiencing the event at time t_i. The partial likelihood function $L(\boldsymbol{\beta})$ is then maximized with respect to the regression parameters $\boldsymbol{\beta}$. By maximizing $L(\boldsymbol{\beta})$, we obtain the values of $\boldsymbol{\beta}$ that best explain the observed order of events, effectively capturing the relationship between the covariates and the survival times. Note that Cox's version of the partial likelihood does not allow for ties between patients in event times, and other alternative methods such as Efron (1977) have been proposed for approximating the partial likelihood in the case of tied survival times.

The estimated parameters $\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p$ can provide insight into the effect of the features on the survival of the patient. If we increase the feature x_{ij} for the ith patient by one unit, the hazard ratio is given by $\exp(\beta_j)$. If the hazard ratio is less than one, then the hazard will decrease, and thus, the survival of the patient will increase, and if the hazard ratio is greater than one, then the hazard will increase, leading to decreased survival. After estimating the parameters $\boldsymbol{\beta}$, we still need to estimate the baseline hazard function $h_0(t)$ to use the model given by Eq.3.16. The most common (Lin,

2007) method for this purpose is the Breslow estimator. This approach is discussed in detail by Breslow (1972).

Chapter 4

Clustering by attention to multiple protein subsets

Outline of chapter

This chapter presents a novel clustering algorithm called CAMPS for single-cell protein expression data. By incorporating prior knowledge about expected combinations of protein markers for distinct cell types, we aim to enhance both the accuracy and reliability of the clustering outcomes.

4.1 Introduction

Suppose we have a dataset with single-cell protein expressions for many cells in a cell population. If the cell population comprise several different cell types, each characterized by a unique protein expression profile, then we might reasonably expect unsupervised clustering to be able to identify each cell type as a separate cluster. However, the separation between different cell types becomes blurred when the data are high-dimensional and noisy. This is a consequence of the fact that the concept of localness is ill-defined in high dimensions James et al. (2013). One way to circumvent this problem is to calculate distances between points in a low-dimensional subspace. For example, if we happen to know that only a specific subset of features are relevant to the clustering, then we might use the subspace spanned by those features. We actually have several such subsets for the single-cell protein expression data, each corresponding to the protein markers identifying a particular cell type. Using these subsets one at a time to define pairwise distances, we can direct attention to relevant protein features for individual cell types and thus guide the clustering. This is the basic principle behind the CAMPS algorithm. In addition to the data, the algorithm takes one or several subsets of features as input. This assumes that the user has knowledge of which features are relevant for each cell type, and for immune cells, we have such knowledge (Delves et al., 2017). Such knowledge also exists for many other cell types. Known in machine learning as zero-shot classification, this approach can be a powerful means of deriving more relevant clusters. In the following sections, we describe the CAMPS algorithm step by step.

4.2 Protein panels

A protein panel is a subset of the proteins that can be used to identify a particular cell type. For example, a protein panel for T killer cells might consist of the two proteins CD3 and CD8. All T killer cells have high expression of both these proteins and no other cells express both CD3 and CD8. In CAMPS, we define several protein panels to cover all the cell types that we expect to encounter and that we are interested in. The panels are jointly represented as a binary matrix $S = (s_{ij})$, where $s_{ij} = 1$ if the jth protein is a member of the ith protein panel/subset and $s_{ij} = 0$ if the jth protein is not part of the ith protein panel. An example of this is shown in Figure 4.1.

$$S = \begin{array}{cl} \begin{array}{ccccccc} \text{CD3} & \text{CD4} & \text{CD8} & \text{CD14} & \text{CD16} & \text{CD20} & \text{CD64} \end{array} & \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} & \begin{array}{l} \text{T helper cells} \\ \text{T killer cells} \\ \text{B cells} \\ \text{Monocytes} \end{array} \end{array}$$

Figure 4.1: Panel matrix. Rows correspond to protein panels and columns correspond to protein markers. Values are 1 for markers that should be present and 0 for markers of no relevance.

The first row/protein panel indicates the protein markers associated with T helper cells, defined specifically by high expression levels of CD3 and CD4.

The third row defines our subset of protein markers for identifying B-cells, characterized by high levels of CD16 and CD20. The combination of protein markers that we associate with a specific cell type is available from existing literature.

4.3 Data reduction

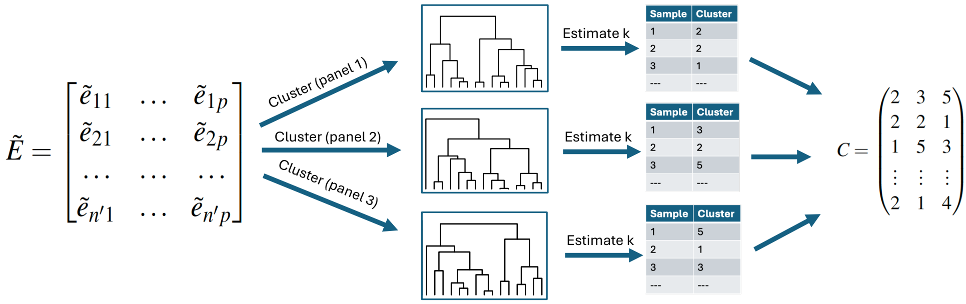
The first task in CAMPS is to reduce the number of objects to be clustered by replacing groups of similar objects with one common representative. Known as vector quantization, this operation substantially reduces the computing time and memory use in downstream processing. This is relevant since the second step of the algorithm requires computation and storage of all pairwise distances between objects. To achieve the above object reduction, we apply k-means clustering and select k large enough to keep important structural features of the original data set. The choice of k is a trade-off between achieving a compact representation (small k) and a faithful representation of the original data (large k). In this thesis, we selected $k = 1000$. The common representatives were simple averages of the object vectors in a group (i.e., centroids); if we expect a substantial number of outliers, then one might consider using the median instead.

Let $E = (e_{ij})$ denote the original $n \times p$ expression matrix, where n is the number of cells and p is the number of proteins. The data reduction transforms E into an $n' \times p$ matrix $\tilde{E} = (\tilde{e}_{ij})$ where n' is the number of common representatives (centroids). Visually:

$$E = \begin{pmatrix} e_{11} & \dots & e_{1p} \\ e_{21} & \dots & e_{2p} \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ e_{n1} & \dots & e_{np} \end{pmatrix} \xrightarrow{\text{k-means}} \tilde{E} = \begin{pmatrix} \tilde{e}_{11} & \dots & \tilde{e}_{1p} \\ \tilde{e}_{21} & \dots & \tilde{e}_{2p} \\ \dots & \dots & \dots \\ \tilde{e}_{n'1} & \dots & \tilde{e}_{n'p} \end{pmatrix}$$

4.4 Panel clustering

The next step in CAMPS is clustering the rows in the reduced data set $\tilde{E}$ with respect to each predefined protein panel. For this purpose, we used hierarchical clustering with Ward linkage, which generally seems to work well. Each clustering (hereafter called sub-clustering) utilizes one of the predefined protein panels and focuses exclusively on the expression of the proteins included in the respective protein panel. Rather than using a fixed number of clusters, we use the silhouette score to determine the optimal number of clusters in each clustering. The possible choices for the number of clusters were limited to the interval 3-10 in the analyses reported in this thesis. Note that the number of clusters can differ across the sub-clusterings. The approach just described is reminiscent of a single step of the manual gating method for clustering the data into cell types. The combined result of all the sub-clusterings is a matrix $C = (c_{ij})$, where c_{ij} denotes the cluster label assigned to the ith centroid (common representative) during the jth sub-clustering. Thus, for each centroid that we cluster, there will be as many cluster labels as there are predefined protein panels. See Figure 4.2 for an overview of the process.



$$\tilde{E} = \begin{bmatrix} \tilde{e}_{11} & \dots & \tilde{e}_{1p} \\ \tilde{e}_{21} & \dots & \tilde{e}_{2p} \\ \dots & \dots & \dots \\ \tilde{e}_{n'1} & \dots & \tilde{e}_{n'p} \end{bmatrix}$$

Sample	Cluster
1	2
2	2
3	1
---	---

Sample	Cluster
1	3
2	2
3	5
---	---

Sample	Cluster
1	5
2	1
3	3
---	---

$$C = \begin{pmatrix} 2 & 3 & 5 \\ 2 & 2 & 1 \\ 1 & 5 & 3 \\ \vdots & \vdots & \vdots \\ 2 & 1 & 4 \end{pmatrix}$$

Figure 4.2: Panel clustering. The centroids found in the first step are clustered with respect to each of the protein panels. This results in a dendrogram for each protein panel. The silhouette score method is used to determine the number of clusters for each dendrogram. The cluster labels obtained from all the panel clusterings are collected in a matrix C.

4.5 Deriving the final clustering

The final step of the CAMPS algorithm is to derive a single combined clustering of the cells from the matrix C found in the previous step. Recall that C contains multiple cluster labels for each centroid and that each centroid represents several cells. In the following, we focus first on the clustering of the centroids; when this has been accomplished, we let all the single cells that a centroid represents inherit these cluster labels.

To facilitate this final clustering, we must establish a method for measuring the distance between centroids based on their labels from the sub-clusterings. A straightforward approach might be to use the Euclidean distance; however, this method assumes that, for example, label 1 is closer to label 2 than to label 3 in the jth sub-clustering, which is misleading since the numerical value of cluster labels is assigned arbitrarily and does not reflect actual proximity. An alternative measure could involve assessing the dissimilarity between centroids, essentially counting how many sub-cluster labels differ between them and using this count as the distance measure. While this method directly reflects discrepancies in clustering outcomes, it has a significant drawback. As previously mentioned, our goal in each sub-clustering is to identify a cluster that specifically encapsulates the relevant cell types. Suppose two centroids fall into the same cluster in a given sub-clustering. In that case, it is crucial to determine whether this cluster indeed corresponds to a *significant cell type* or is merely an incidental grouping. Dissimilarity alone fails to convey this level of detail. Instead, we propose a different method to quantify the distance between centroids. This new approach will reflect the clusters' biological significance, distinguishing between meaningful and incidental groupings, thus allowing for a more nuanced final clustering.

The main idea is to determine which clusters in each sub-clustering are of interest and which are not. Suppose we are performing a sub-clustering for the protein panel that aims to identify T helper cells. In this case, we are only considering the features CD3 and CD4, and the clusters identified might look like what we see in Figure 4.3. We clearly see that the green cluster contains the T helper cells, which we are trying to separate in this particular sub-clustering. The other clusters are not of interest. Centroids that share the label are clearly both T helper cells, so this should receive a large score. In other words, we want to pay much attention to this particular sub-clustering if two centroids share the same green cluster label. If the centroids share the blue or red cluster label or do not share the label at all, then we will not pay attention to this particular sub-clustering. To accomplish this, we assign an attention weight to each cluster. We do this by finding the average expression value for each protein considered in the subset, and taking the minimum of these. In the plot, the green cluster has an average expression of 0.8 for both CD3 and CD4. The minimum of these is 0.8, so we let the attention to this sub-clustering be 0.8 if the centroids share the same label here. The red cluster shows a high expression of CD3 but a low expression of CD4. If we take the minimum of these, we find that the attention weight will be low. Similarly, the attention weight for the blue cluster will be low.

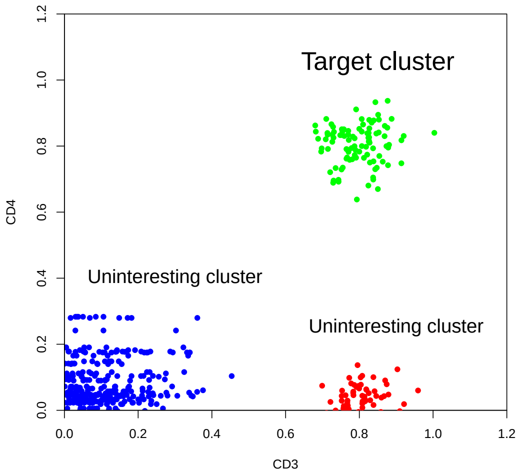


Figure 4.3: Example of sub-clustering.

Formally, the attention weights are defined as follows. Consider a protein panel k and a cluster j in that protein panel. For each protein r in that panel, we first calculate the median expression x_{kjr} across all the samples (centroids) in the jth cluster. We then define the attention weight as

$$w_{kj} = \frac{\min(x_{kj1}, \ldots, x_{kjr})}{\sum_s \min(x_{ks1}, \ldots, x_{ksr})}$$

All the proteins in the panel must have a high median expression to obtain a high attention weight. The denominator normalizes the weights within each protein panel to 1, ensuring that all the protein panels receive the same total weight. The final step of the CAMPS algorithm is to perform hierarchical clustering of the rows of C with the distance measure

$$d_{ij} = -\sum_k I(c_{ik} = c_{jk}) w_{k,c_{ik}},$$

and using Ward linkage. Here, $I(\cdot)$ is the indicator function and is 1 if the condition is true and 0 otherwise. The sum is over the protein panels k. Note that the distances as calculated above are negative or zero; to use them in hierarchical clustering, we use instead $\tilde{d_{ij}} = d_{ij} - \min(d_{ij})$, which ensures that all distances are nonnegative. Finally, the dendrogram in cut into K clusters to generate cluster labels. Cells inherit the label for the centroid representing them.

4.6 Complete algorithm

Algorithm 4: CAMPS

Input: (a) A protein expression matrix $E = (e_{ij})$ where e_{ij} is the expression of the jth protein in the ith cell; (b) A binary matrix $S = (s_{ij})$ specifying protein panels (subsets) where $s_{ij} = 1$ if the jth protein is a member of the ith subset and $s_{ij} = 0$ otherwise; (c) The desired number of clusters $K > 0$.
Output: A cluster label y_i for each cell.

1. **Data reduction:** Cluster rows in E by k-means with $k = 1000$, calculate centroid for each cluster and collect centroids in $\tilde{E}$.

2. **Panel clustering:** For each protein panel: perform hierarchical clustering with Ward linkage to the rows of $\tilde{E}$, and cut the dendrogram into q subclusters (where q is found by maximizing the silhouette score) to derive cluster labels for the rows. Combine the results from all the panel clusterings in a matrix $C = (c_{ij})$ where c_{ij} is the cluster label for the ith centroid in the jth clustering.

3. **Calculate Attention Weights:** For each protein panel k and for each cluster j in that protein panel, calculate median protein expression for each protein r in the protein panel. Let $\tilde{w}_{kj}$ be the minimum of these medians. Finally, calculate attention weights $w_{kj} = w_{kj} / \sum_s \tilde{w}_{ks}$.

4. **Final clustering:** Perform hierarchical clustering of the rows of C with the distance measure $d_{ij} = -\sum_k I(c_{ik} = c_{jk}) w_{k,c_{ik}}$ and with Ward linkage. Here, $I(c_{ik} = c_{jk})$ is the indicator function and is 1 if the class labels match and 0 otherwise. The sum is over the protein panels k. Finally, cut the dendrogram in K clusters to generate cluster labels. Cells inherit the label for the centroid representing them.

Algorithm 4 can be extended to include negative protein markers, i.e., proteins that should not be expressed in a protein panel. The only change we have to make is how attention weights are calculated. Let $x_1, \ldots, x_q$ and $y_1, \ldots, y_r$ denote median protein expressions for positive and negative protein markers, respectively. Then we let $\tilde{w}_{kj} = \min(x_1, \ldots, x_q) - \max(y_1, \ldots, y_r)$.

4.7 The protein panels considered

For further reference, Table 4.1 shows the protein panel that was used for the CAMPS algorithm in this thesis.

Cell type	**CD8a**	**CD3**	**CD56**	**CD14**	**CD4**	**CD64**	**CD16**	**CD66b**	**CD20**	…
B cells	0	-1	0	0	0	0	0	0	1	…
CD4+ T cells	-1	1	0	0	1	0	0	0	0	…
CD8+ T cells	1	1	0	0	-1	0	0	0	0	…
Monocytes	0	0	0	1	0	1	-1	0	0	…
Neutrophils	0	0	0	-1	0	0	1	1	0	…
NK cells	0	-1	1	0	0	0	1	0	0	…

Table 4.1: The protein panels used in the thesis analysis. Only the proteins that were part of at least one of the panels are shown.

Chapter 5

Results

Outline of chapter

In this chapter we use the previously described clustering methods including the novel CAMPS method to derive compact representations of real and simulated CyTOF data. After presentation of the datasets, we investigate the ability of the cluster methods to distinguish between cell types and their ability to provide useful representations for prediction of treatment response and overall survival.

5.1 The AML dataset

Clinical features

In a recent study conducted at Haukeland University Hospital in the period 2016-2023, a cohort of 32 acute myeloid leukemia (AML) patients were administered standard induction chemotherapy (Tislevoll et al., 2023). Two of the patients were discarded from the analysis reported here for reasons outside my control, resulting in a total of 30 patients available for this thesis project. Access to the data was provided through a collaboration between the University of Bergen and Haukeland University Hospital.

Treatment response was evaluated approximately 17 days after treatment, resulting in a binary classification of patients into complete responders (CRs) and others (patients with partial response, stable disease, or progressive disease). Patients were also followed for some time after the treatment to obtain overall survival (i.e., time to death by any cause). This resulted in a continuous variable indicating either time to death by any cause or the last recorded observation of the patient for those still alive at the end of the study period. The estimated overall survival (OS) is shown in Figure 5.1 and clearly demonstrates that even with the given treatment, the survival is very poor, with a median survival of less than 1.5 years.

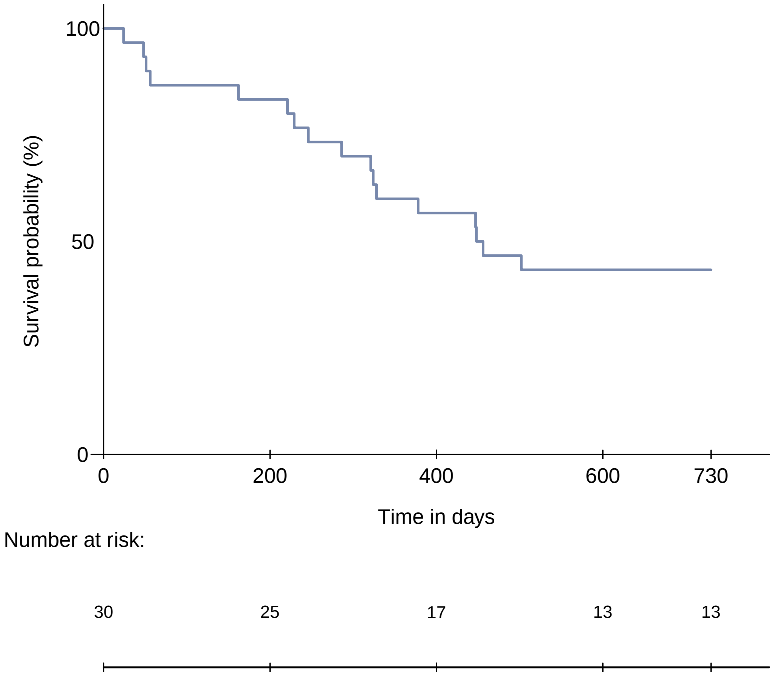


Figure 5.1: Kaplan-Meier survival curve for all patients. Here, we see the estimated overall survival probability of the AML patients over time. The plot illustrates the proportion of patients surviving over a period of 730 days (approximately 2 years). The y-axis represents the survival probability as a percentage, while the x-axis indicates the time in days. The number of patients at risk at various time points is indicated below the x-axis, with an initial cohort of 30 patients. The survival curve demonstrates a gradual decline in survival probability, with significant drops at various intervals, reflecting the mortality events over the study period.

Protein expression data

Mass cytometry (CyTOF) was used to analyze blood samples taken from the patients (see Section 2.2 for a description of CyTOF). This resulted in the quantification of 36 proteins of which 21 were proteins expressed on the surface of cells (surface markers) and 15 were proteins expressed in the interior of the cells (intracellular markers). Table 5.1 provides an overview of the proteins. Note that the panel includes the protein CD8a but not CD8. CD8a is one of the subunits of the CD8 molecule, and the expression of CD8a is, therefore, closely related to the expression of CD8. The terms CD8 and CD8a will be used interchangeably in the rest of the thesis.

Surface proteins	**Intracellular proteins**
Axl_1H12	Caspase3_cleaved
CD3	CyclinB1
CD4	pAkt
CD7	pAxl
CD8a	pCREB
CD11	pErk
CD14	pHistone3
CD16	pNFkB
CD20	pP38
CD25	pRB
CD33	pS6
CD34	pSTAT1
CD38	pSTAT3
CD45	pSTAT5
CD56	p4E_BP1
CD64	
CD66b	
CD90	
CD117	
CD123	
HLA.DR	

Table 5.1: Proteins in the CyTOF dataset. There are 36 proteins in total, of which 21 are expressed on the surface of a cell (left column) and 15 are expressed inside the cell (right column). The letter ”p” in front of some of the intracellular proteins indicates that only the phosphorylated form of the protein is measured.

Briefly, the protein expression data consisted of three measurements for each patient based on blood samples obtained at the initiation of induction therapy (0 hours), after 4 hours, and after 24 hours. The measurement at 4 hours was incomplete (3 missing values) and was excluded from the current analysis. An extension of the analysis to include these measurements would be straightforward, but it was concluded that this would not add significantly to the goal of the present analysis. Preliminary studies (Tislevoll et al., 2023) furthermore suggest that the samples taken at 4 hours are less informative for treatment response and survival than the samples taken at 24 hours.

Cells found in blood primarily consist of red blood cells (erythrocytes) and immune cells (lymphocytes and cells of myelogenic origin). Other cell types may occasionally occur, such as epithelial cells in patients with solid tumors, but these are of no concern in our analysis. Red blood cells are usually removed before CyTOF analysis by lysis (i.e. breaking up the cells) or centrifugation, and this was also done in this study. Hence, the single-cell data represent almost exclusively immune cells and the combination of surface proteins found in a cell provides valuable information about the type of immune cell.

The dataset's detailed protein expression profiles allow for in-depth analysis of cellular behaviors and responses to treatment. The surface markers play a crucial role in identifying various cell types within the blood samples, while the intracellular markers provide valuable information on the signaling pathways and functional states of these cells.

Tislevoll et al. (2023) demonstrates that early signaling changes, particularly in ERK1/2 and p38 MAPK phosphorylation, significantly predict patient survival. The dataset from Haukeland University Hospital offers a rich resource for exploring the single-cell protein expression landscape in AML patients, aiding in evaluating early responses to induction therapy.

The number of cells analyzed by CyTOF in each blood sample ranged from 16,037 to 530,283. The upper limit of this range posed serious computational problems with the available hardware, and the variability itself caused additional challenges in the comparative analyses. A subsampling approach was employed to address these issues. To this end, a fixed number of cells were randomly sampled without replacement from each blood sample. This streamlined subsequent analyses by reducing the computational load without seriously compromising the diversity of the data. The number of cells was chosen to be slightly less than the lower limit of cells per sample, or specifically N=16,000. This number was found to be sufficiently small to ensure efficient processing and analysis in later methods, while still being large enough to capture the biological variability within each sample. The original dataset totaled approximately 10 million cells across all the 60 available blood samples (two samples per patient). After subsampling, the total size of the CyTOF dataset was reduced to roughly 1 million cells, i.e., a reduction to 10% of the original size (see Figure 5.2). Some of the methods used required even fewer cells than this to run efficiently and manage memory limitations. In those cases, the subsampled data were further subsampled for performance.

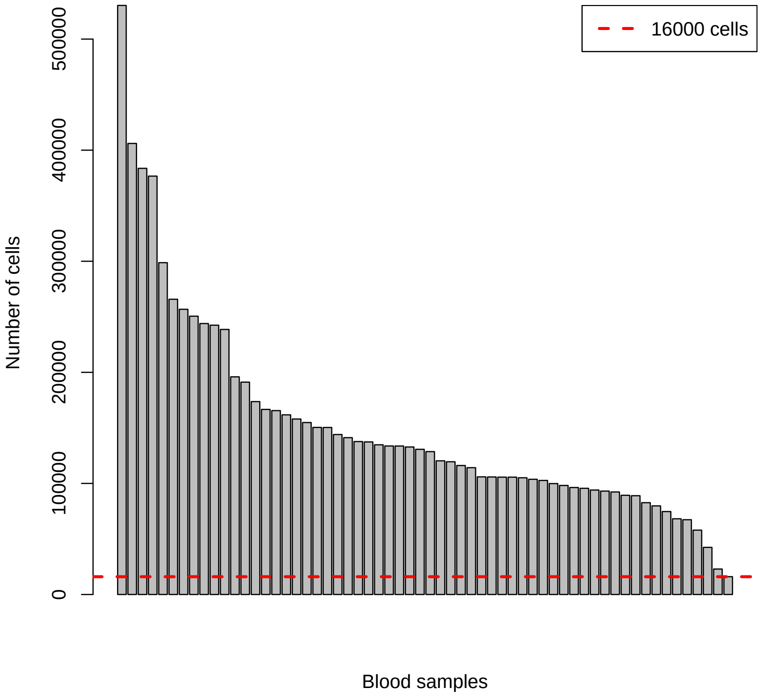


Figure 5.2: Number of cells per sample. The sizes are sorted in decreasing order. The red dotted line shows the number of cells subsampled from each blood sample.

Several visualizations were generated to verify that subsampling maintained the integrity and representativeness of the data. One such plot is shown in Figure 5.3 where CD3 and CD8 marker intensities are plotted against each other. High values of CD3 and CD8 are used to characterize T cells, and the plot demonstrates that the key cell population is still well represented in the subsampled data. Similar plots were inspected for other cell types.

Raw CyTOF data are essentially counts and typically have a skewed distribution with a long right tail. It is common to transform the data to prevent that large values dominate too much in downstream analyses. A

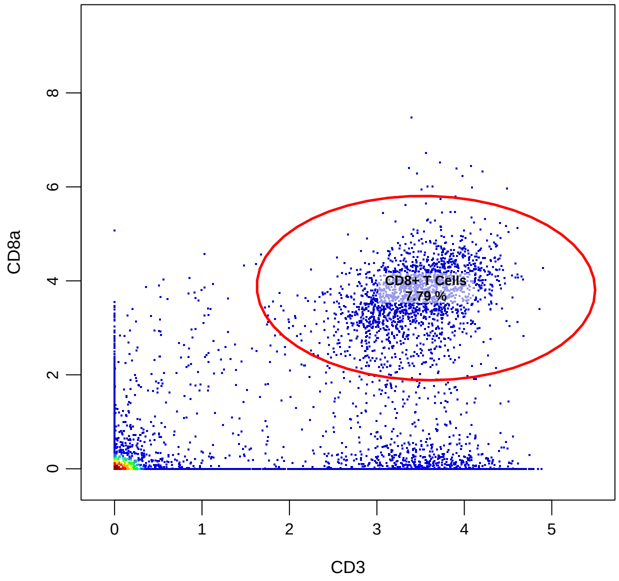


Figure 5.3: Scatter plot of CD3 vs. CD8 after subsampling. Each dot represents a single cell, and the cells within the red circle are gated as CD8+ T cells, showing that the CD8+ T cell populations are well represented in the subsampled data.

log transform is one way of achieving this, but it cannot be used directly if the data contains zeros or negative values. Such values can appear in raw CyTOF data due to background subtraction. An alternative transformation that does not have this problem is the arcsinh function

$$arcsinh(x) = \frac{\ln(x + \sqrt{x^2 + 1})}{a}$$

where $a > 0$ is referred to as the scale factor (see Figure 5.4). This transformation can handle negative input values and approximates a log transformation at higher values. In addition, it behaves linearly near zero. The scale factor a controls how far from zero the transformation behaves approximately linearly; larger values of a result in a more extensive linear region. Due to these properties, the arcsinh transformation is widely used for preparing CyTOF data for clustering. Bendall et al. (2011) recommend 5 as the most appropriate value for the scale factor a for CyTOF data, which is the value used for the transformation in this thesis. Before the downsampled dataset was used for analysis, the protein expressions were arcsinh-transformed.

Figures 5.5 and 5.6 illustrate the impact of this transformation on the protein expression distributions. The top panel in Figure 5.5 shows the raw expression values for markers CD66b, CD4, and HLA.DR, which exhibits highly skewed distributions with extreme outliers. After applying the arcsinh transformation, as seen in the bottom panel, the distributions become more normalized, and the boundaries between high and low expression intensities become apparent. In Figure 5.6, we observe that the transformed expressions now operate within similar ranges across the markers, which is beneficial for subsequent clustering analyses using Euclidean distance. This normalization prevents any single marker from disproportionately influencing the clustering results due to its initial broader range, ensuring that all markers contribute more equally to the analysis.

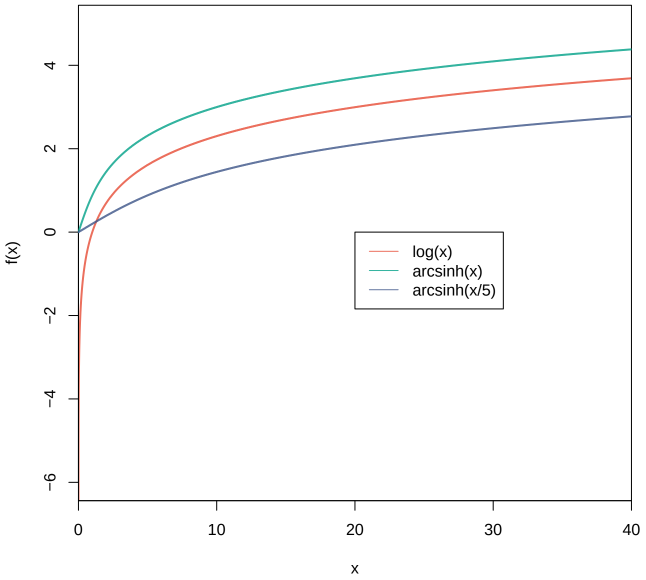


Figure 5.4: Illustration of the arcsinh transformation compared to the log transformation. The plot shows three functions: $\log(x)$ (red), $\text{arcsinh}(x)$ (green), and $\text{arcsinh}(x/5)$ (blue). The arcsinh transformation handles negative values and large ranges of CyTOF data while maintaining properties similar to the log transformation at higher values. The $\text{arcsinh}(x/5)$ transformation, with a scale factor of 5, provides a linear approximation near zero, which is particularly useful for normalizing CyTOF data.

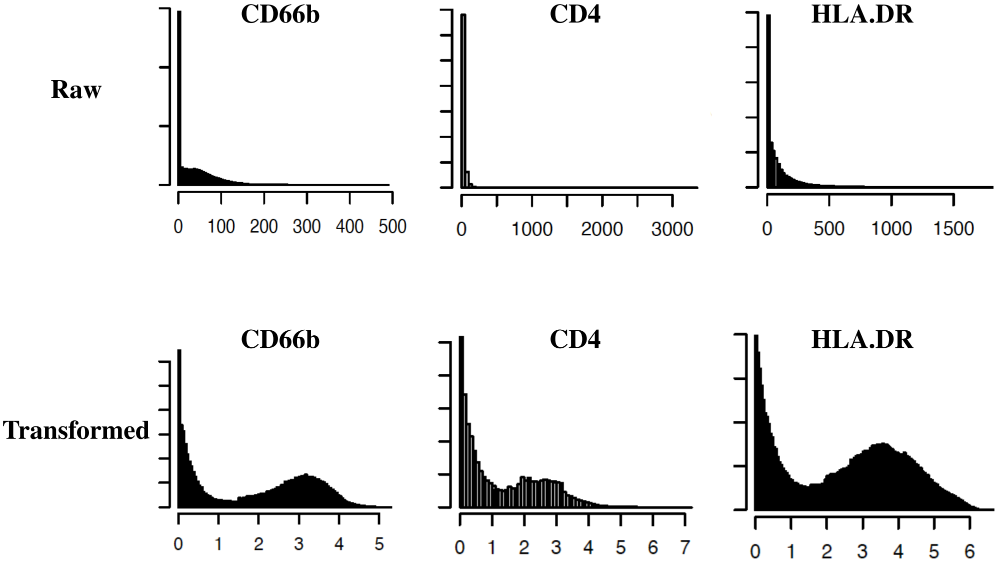


Figure 5.5: Expression values for selected markers before and after arcsinh transformation. The top panel shows the raw expression values for markers CD66b, CD4, and HLA.DR, illustrating the typical skewed distributions observed in CyTOF data. The bottom panel displays the same markers after applying the arcsinh transformation, which normalizes the distributions and makes them more suitable for subsequent analyses. The transformation helps to stabilize variance and reduce the influence of extreme values, facilitating better differentiation between high and low marker intensities.

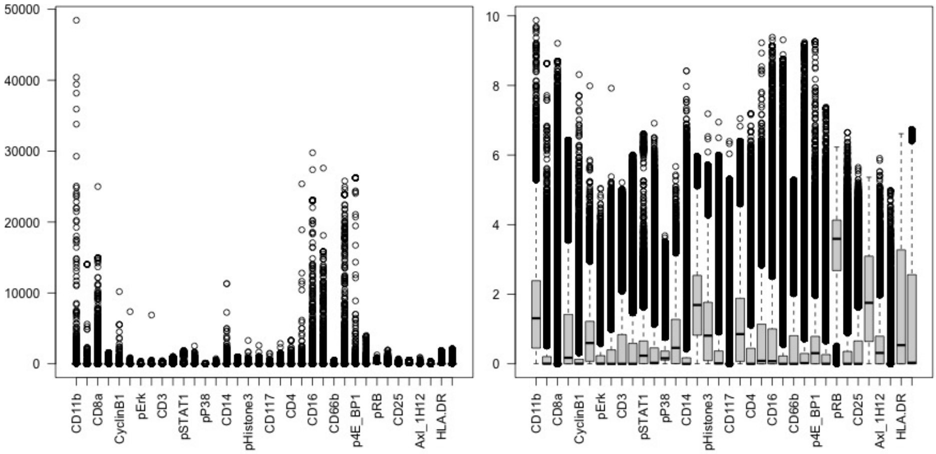


Figure 5.6: Boxplots showing the distribution of protein expressions for all markers before and after arcsinh transformation. The boxplot on the left depicts the raw expression values. The boxplot on the right presents the same markers after the arcsinh transformation.

5.2 Identifying cell types manually

Initially, the CyTOF dataset of single cells lacked cell type annotations. Manual gating was performed using the sequential gating strategy in Russo et al. (2019) and the R package CytoExploreR (Hammill, 2021) to generate cell annotations and obtain the necessary statistics and characteristics for modeling specific cell types.

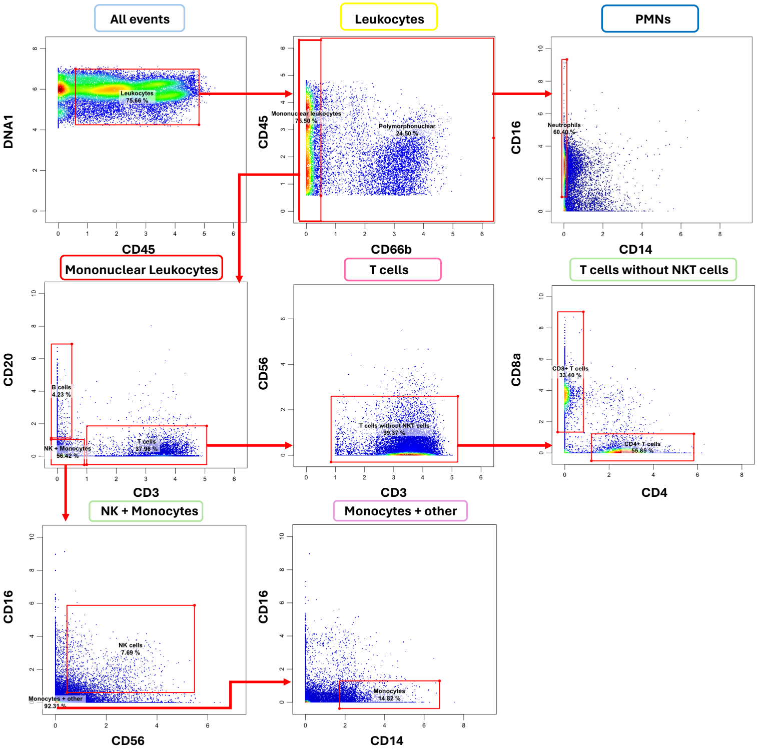


Figure 5.7: Manual gating strategy. A gating scheme showing the manual gating strategy and boundary selection used to identify immune cell populations from CyTOF data. The color of the box around the title of the plot corresponds to the color of the populations shown in the gating tree (Fig.5.8). The detailed gating strategy was as follows: First, leukocytes were identified by gating on $CD45^+$ and DNA1. Subsequently, mononuclear leukocytes ($CD66b^-$) and polymorphonuclear leukocytes ($CD66b^+$) were differentiated. From the population of polymorphonuclear leukocytes, neutrophils were identified by gating on $CD14^-CD16^+$. The mononuclear leukocyte population was further classified into T cells ($CD3^+CD20^-$) and B cells ($CD3^-CD20^+$). The remaining cells ($CD3^-CD20^-$) were divided into NK cells ($CD56^{dim}CD16^{dim}$) and monocytes ($CD14^+CD16^-$). Finally, $CD4^+$ T cells and $CD8^+$ T cells were gated from the population of T cells.

This approach allowed for the creation of a simulated dataset to evaluate the clustering performance of subsequent methods. Minor adjustments to the gating strategy in Russo et al. (2019) had to be made since some of the protein markers used in the paper were not available in the data analyzed here. The result for the AML dataset is shown in Figures 5.7 and 5.8.

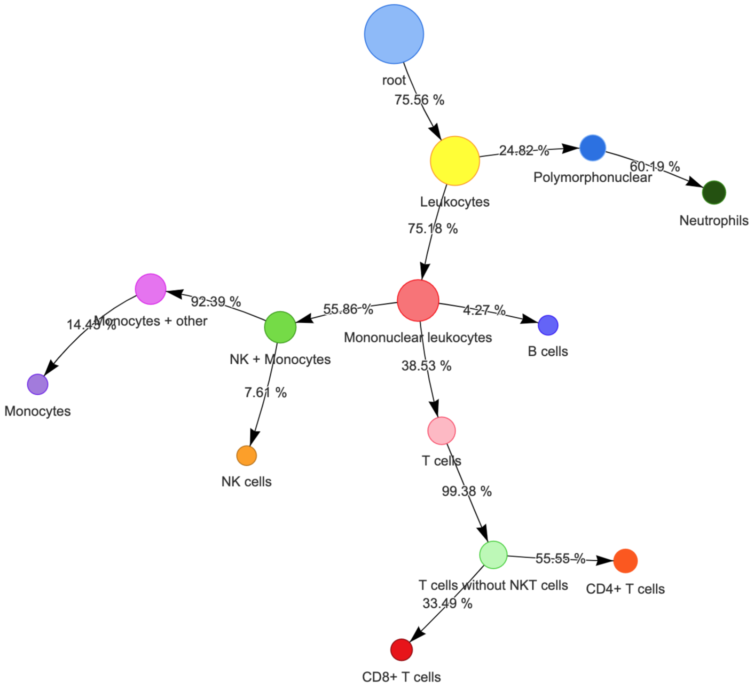


Figure 5.8: Gating tree. Nodes represent cell populations, and edges represent steps in the manual gating of the AML data. The root node at the top of the figure represents all cells, and the path from the root to a leaf node represents the steps performed to isolate the cell type represented by the leaf node. Numbers on the edges represent the percentage of cells in the parent node that are transferred to the child node. Smaller nodes correspond to smaller subpopulations of cells, and the colors are consistent with those used in Figure 5.7 and Figure 5.9.

After manual gating, the proportion of cells assigned to each cell type in each sample was calculated. The distributions of these proportions are shown in Figure 5.9 and show that the mean and variance differ quite substantially between cell types.

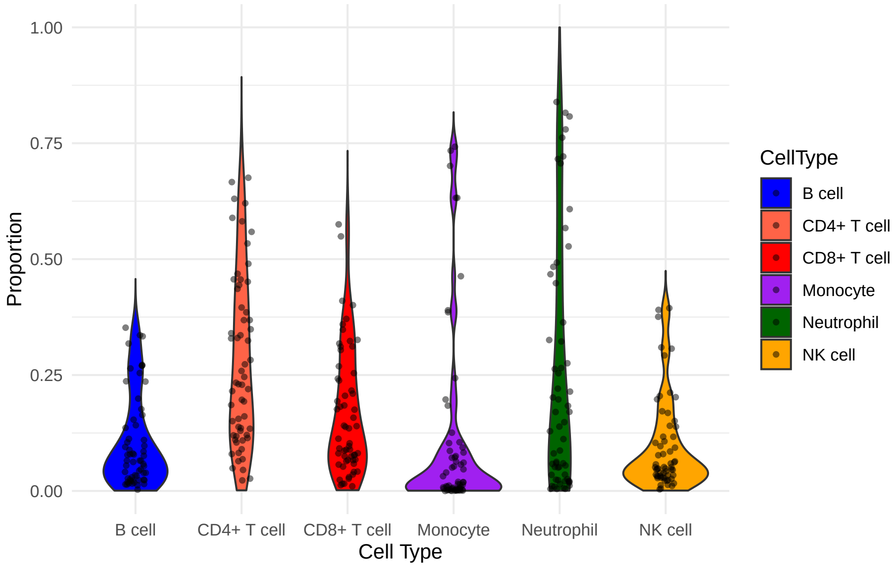


Figure 5.9: Violin plot showing the distribution of cell types across samples. The colors of the violins correspond to the color of the populations shown in the gating tree (Fig.5.8).

The mean and variance are smallest for B cells and NK cells, intermediate for T killer cells and monocytes, and highest for T helper cells and neutrophils. For comparison, the mean values above are compared to ordinary proportions for different cell types in blood in healthy individuals (see Table 5.3). Observe that the proportion of T cells is well above the reference range, while the proportion of neutrophils is well below the reference range. In addition, both monocytes and NK cells are slightly outside their reference ranges. Such shifts in distributions are expected in AML (senior consultant Marianne Brodtkorb, personal communication).

When performing manual gating on CyTOF data obtained from blood samples of patients with AML, several challenges arise due to the nature of the disease. AML can significantly affect cell surface marker expressions and the relative frequency of different cell populations, complicating the gating process. The presence of leukemic blasts and altered cell phenotypes can obscure the plots, making it difficult to establish clear gating boundaries. This complexity is compounded by the high variability in the expression of markers caused by the malignancy, leading to a high proportion of aberrant or ”corrupted” cells (Tislevoll et al., 2023) that can further confound the analysis. The variability in neutrophil proportions shown in 5.9 can possibly be

Cell type	Estimated (%)	Reference value (%) Sender et al. (2023)	Reference range (%) DIPS
B cells	6	4	1.4 - 6.8
T cells	48	21	12.8 - 30.5
Monocytes	11	8	3.0 - 10.9
Neutrophils	28	62	36.1 - 73.4
NK cells	6	4	1.4 - 5.6

Table 5.2: Cell type proportions. For each cell type the table shows the proportion of immune cells of that type in the AML dataset (found by manual gating as previously described), the reference value according to Sender et al. (2023), and the reference range used by the central laboratory at Radiumhospitalet, Oslo and obtained from DIPS by senior consultant Marianne Brodtkorb. This reference range was provided for the *i*th cell type as a range $[a_i, b_i]$ on the number of cells per unit volume, and was translated to a percentage range $[A_i, B_i]$ using the formulas $A_i = 100a_i/(a_i + \sum_{j \neq i} c_j)$ and $B_i = 100b_i/(b_i + \sum_{j \neq i} c_j)$ where $c_j = (a_j + b_j)/2$. This corresponds to the assumption that for the *i*th cell type the percentage range is calculated under the assumption that all other cell types are fixed at their mean value.

attributed to the effects of AML, which can lead to neutropenia, a condition characterized by abnormally low levels of neutrophils. Neutropenia occurs due to the reduced production of normal hematopoietic cells, including neutrophils, caused by the proliferation of leukemic blasts (Hansen et al., 2020).

Despite these challenges, the primary objective of the gating process in this study was not to achieve perfect cell-type annotations. Instead, the goal was to obtain sufficient statistics and characteristics related to various cell types. This information was essential for creating a simulated dataset to evaluate the performance of subsequent analytical methods described in this thesis. By focusing on generating these statistical parameters rather than on precise cell classification, the manual gating served its purpose effectively, even in the presence of AML-related complexities.

5.3 Simulating CyTOF data

For each identified cell type, a separate Gaussian mixed model (GMM) was fitted to the manually gated data for each of the following cell types: B cells, T helper cells, T killer cells, monocytes, NK cells, and neutrophils. The components were modeled as having variable shapes, equal sizes, and

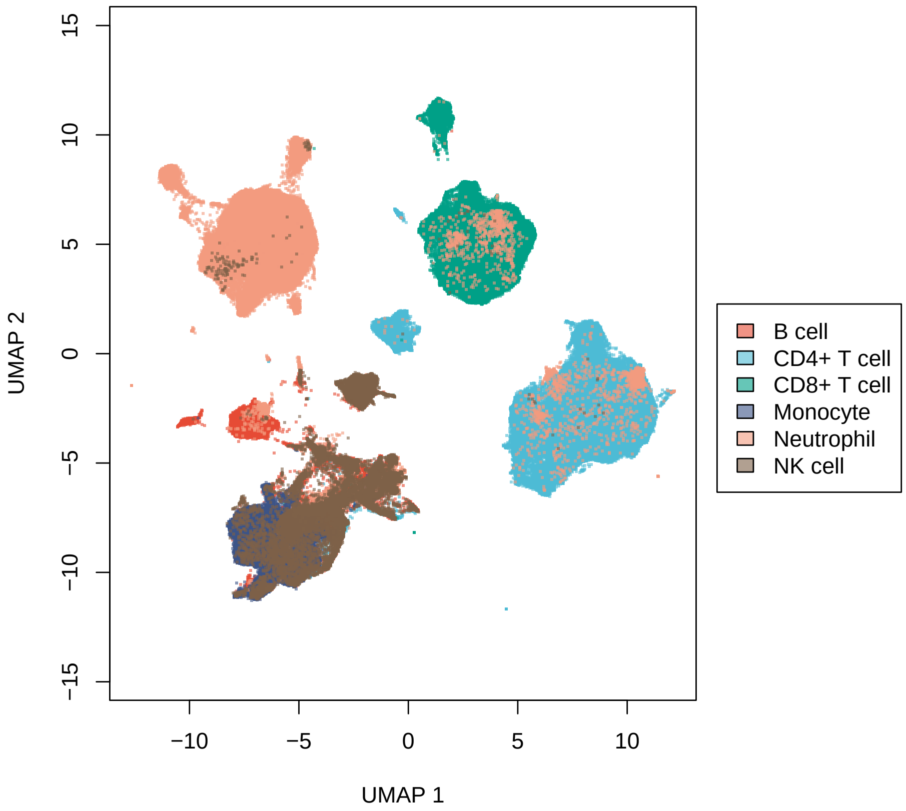


Figure 5.10: UMAP projection of manually annotated cells. The color of the datapoints corresponds to the cell type shown in the box.

equal orientations (referred to as VEE). Fitting the GMM involved determining the optimal number of components (up to three) and estimating each component's means, covariances, and mixture weights. The optimal number of components used for the model was selected according to BIC. Three components gave the best BIC for all six GMMs used to model the cell types.

Using the GMMs, a specified number of cells could be generated for each cell type and then combined into a single dataset. This approach allowed for creating a simulated dataset with known true cell types and proportions, facilitating the assessment of clustering algorithms' performance by evaluating the purity of the clusterings. Furthermore, the simulation model allowed for the creation of multiple datasets, which was used to investigate the sensitivity of the algorithms to changes in the training data. The simulation model also provided the flexibility to adjust the variance of the simulated data. This feature enabled the evaluation of clustering algorithms under varying levels of noise. By adjusting the variance, different scenarios could be simulated. This was used to benchmark the performance of different clustering algorithms and understand their robustness and limitations in various conditions. Two different simulation sets were generated:

- **Sim 1:** n=60,000 simulated cells with equal proportions of the different cell types, i.e. 10,000 cells of each cell type.
- **Sim 2:** n=60,000 simulated cells with cell type proportions matching the proportions found in the AML dataset (Table 5.2).

The number of cells per cell type for the different simulations is shown in Table 5.3.

	B cell	**T helper**	**T killer**	**Monocyte**	**Neutrophil**	**NK cell**
Real n=381,400	23,277	116,014	69,940	40,612	108,382	23,175
Sim 1 n=60,000	10,000	10,000	10,000	10,000	10,000	10,000
Sim 2 n=60,000	3,661	18,250	11,002	6,388	17,050	3,649

Table 5.3: Number of cells per cell type in the real and simulated datasets.

Figure 5.11 and Figure 5.12 show how the simulation 1 dataset compares to the original dataset. Note that the sharp edges present in the original cells result from the selected boundaries during manual gating (see Fig.5.7). The general trend of the original cells is evidently preserved in the simulation. That being said, some instances could prove problematic for further analysis. For example, the top left plot in Fig 5.11 shows the distribution of B cells. We see that a significant proportion of the simulated cells show a low expression of CD20, which is the marker used to identify B cells. Furthermore, by investigating the UMAP projection of simulation 1 (shown in Fig.5.13), we see that CD4$^+$ T cells, CD8$^+$ T cells and monocytes are well separated from the other cell types. However, the B cells, monocytes, and NK cells are more tightly packed together. Figure 5.14 shows that the UMAP projection of simulation 2 follows a similar pattern.

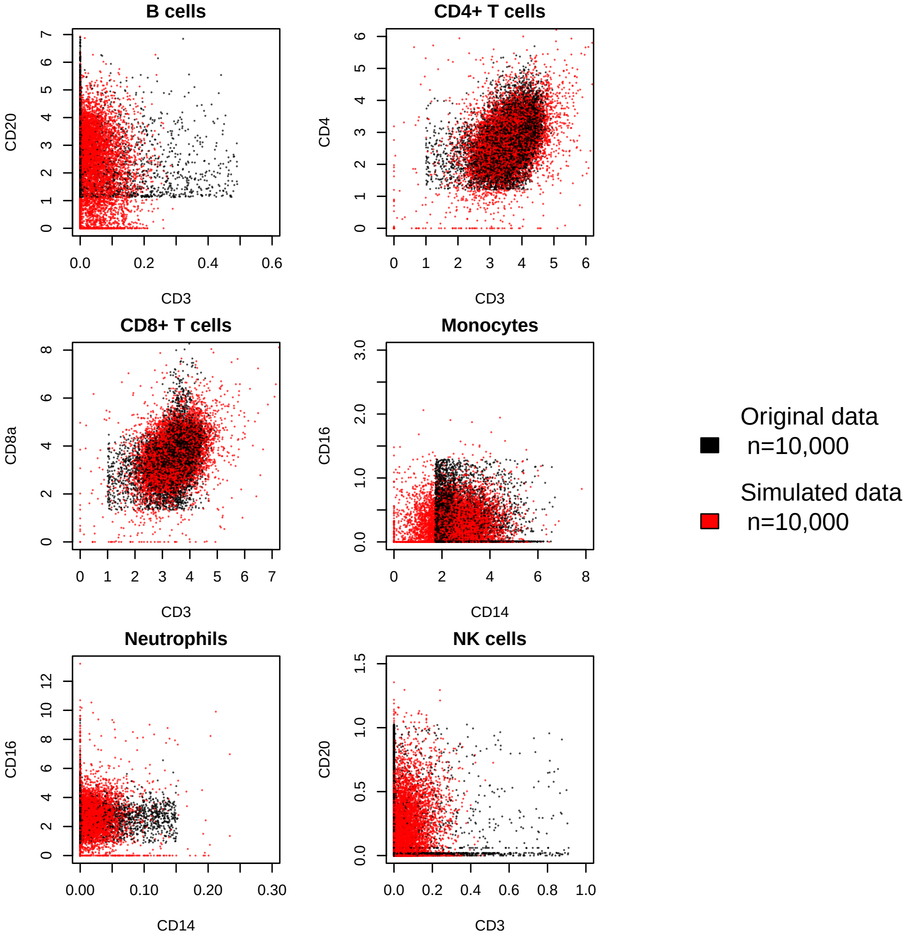


Figure 5.11: Surface proteins for the AML dataset and simulation 1. The scatter plots show the expression of selected surface markers for the AML data and for simulated cells (Sim 1). The red cells are simulated, and the black cells are from the original dataset. Each plot shows 10,000 simulated cells and 10,000 randomly sampled original cells.

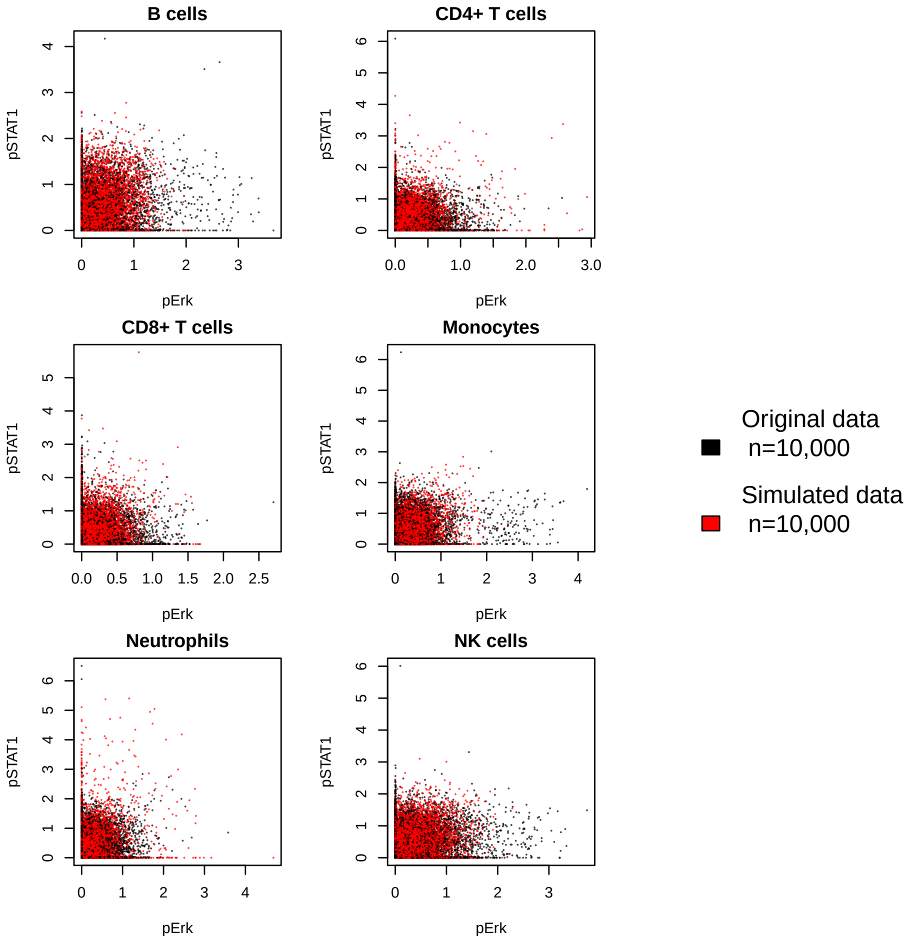


Figure 5.12: Intracellular proteins for the AML dataset and simulation 1. The scatter plots show the expression of selected intracellular proteins for the AML data and for simulated cells (Sim 1). The red cells are simulated, and the black cells are from the original dataset. Each plot shows 10,000 simulated cells and 10,000 randomly sampled original cells.

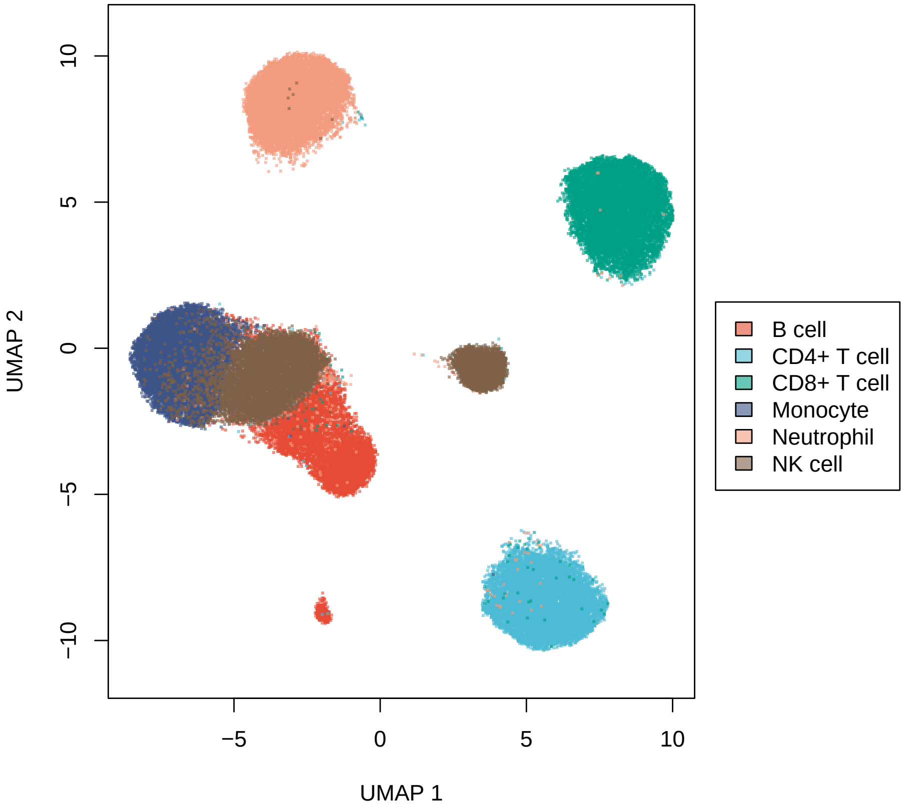


Figure 5.13: UMAP projection of cells from simulation 1. The color of the datapoints corresponds to the simualted cell type

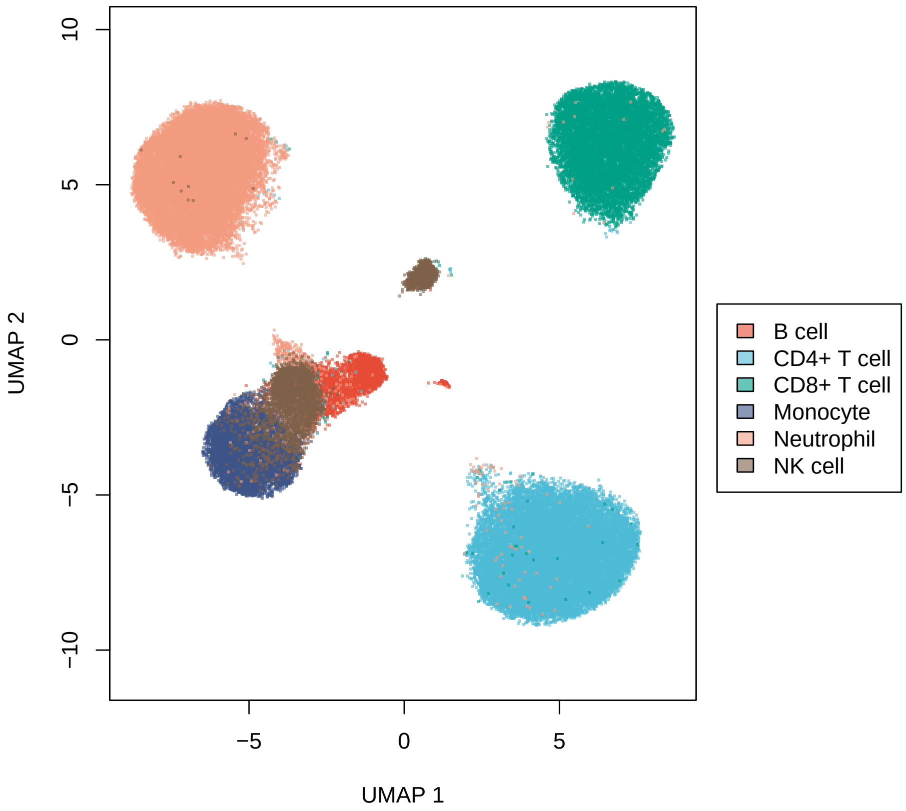


Figure 5.14: UMAP projection of cells from simulation 2. The color of the datapoints corresponds to the simualted cell type

5.4 Clustering CyTOF data

The attention will in this and the following sections be directed towards a comparison of various clustering methods on the specific task of clustering CyTOF data. In this section, we will present the clustering algorithms used, the datasets considered, and the parameters of the model. We will also inspect the degree of concordance or discordance between the clustering results of different methods; later we will investigate how well the different methods are able to recapture the true cell types (Section 5.5) and how useful the methods are for construction of input features for prediction of treatment outcome (Section 5.6). An overview of the clustering methods (and variants thereof) to be studied in the following is provided in Table 5.4.

Method	**Acronyms**
CAMPS	camps
K-means	kmeans
Metaclust	mclust
Hierarchical	hc.single, hc.average, hc.complete, hc.ward
Hierarchical w/minimum cluster size	hc.single*, hc.average*, hc.complete*

Table 5.4: Clustering methods. The table lists the various approaches considered in the following for identifying a prespecified number K of clusters from single-cell CyTOF data. Linkage methods considered in hierarchical clustering were single, average, complete, and Ward. In CAMPS, K-means and Metaclust, the desired number of clusters is an input parameter. In hierarchical clustering, the desired number of clusters was identified by a flat cut of the dendrogram into K subclusters. In hierarchical clustering with minimum cluster size, the dendrogram was iteratively split in $K, K+1, \ldots$ clusters until K clusters satisfied the cluster size requirement (clusters not satisfying the requirement were discarded). Full names and acronyms will be used interchangeably.

The methods in the table all allow (indeed require) the specification of the desired number of clusters K to produce a cluster assignment of the input samples. For hierarchical clustering, it is implicitly assumed that the cluster assignment is found by horizontally cutting the dendrogram into K subclusters that are named $C1, C2, \ldots$ from left to right. In this thesis, we both consider cluster assignment with K fixed and with K determined from the data using the silhouette score (as described in Section 3.5).

Datasets

As previously described, we considered both real data (from AML patients) and synthetic data for the empirical comparison of clustering methods.

AML data: these were either unfiltered or filtered, depending on our question. The filtering removed all cells except those that were identified as one of the six target cell types in the manual gating described in Section 5.2. Only filtered data were considered for comparing cluster assignments (this section and section 5.5). This allowed the assessment of the cluster performance under ideal circumstances where most dead cells, non-leukocytes, and immune cells with unusual expression profiles had been removed. Only unfiltered data were considered to compare prediction performance based on cluster composition (Section 5.6). Using unfiltered data ensures that this analysis stays as close as possible to future use of the methods to predict patient outcomes from un-gated CyTOF data.

Synthetic data: these were only considered when measuring the clustering method's ability to identify cell types in Section 5.5. We mainly focus on the two simulated datasets described in Section 5.3, to investigate how different proportions of the cell types will affect the results. We also utilized simulated data to measure the methods' robustness to noisy data by scaling the variance of the protein expressions of generated cells. Finally, we investigated the robustness to changes in the dataset by generating several simulated datasets with the same parameters and saw how much the results changed over the different runs.

Fixed number of clusters

For the task of identifying cell types, all cluster algorithms were applied to the filtered AML data and both of the simulated datasets with $K = 4$, $K = 6$, or $K = 8$. Since exactly six cell types were isolated during manual gating and then used to create the simulation model, we expect $K = 6$ to best capture the true grouping of the data. When the clusters were estimated to measure quality as features for prediction, we estimated K with the Silhouette method.

Estimated number of clusters

When we clustered the data for the purpose of identifying cell types, we knew that there were exactly six distinct cell types (assuming the manual gating was performed properly) present in our filtered and simulated datasets. However, this knowledge would not be available to us in a real setting where we are faced with a single-cell dataset without annotations. Therefore, in addition to applying the clustering with the fixed K-values described above, the silhouette score was used to estimate the number of clusters. When we clustered for the purpose of identifying cell types (Section 5.5), we tried K-values ranging from 3 to 8. Figure 5.15 shows the silhouette scores for the methods camps, kmeans, hc.ward and mclust. The other clustering methods are omitted for reasons we will see later. Table 5.5 shows the K-values that maximized the silhouette score. When we clustered for the purpose of making features for prediction (Section 5.6), we are faced with the more complex unfiltered dataset, and therefore a large range of K-values between 5 and 15 were considered. Figure 5.16 shows the silhouette scores for the methods camps, kmeans, hc.ward and mclust.

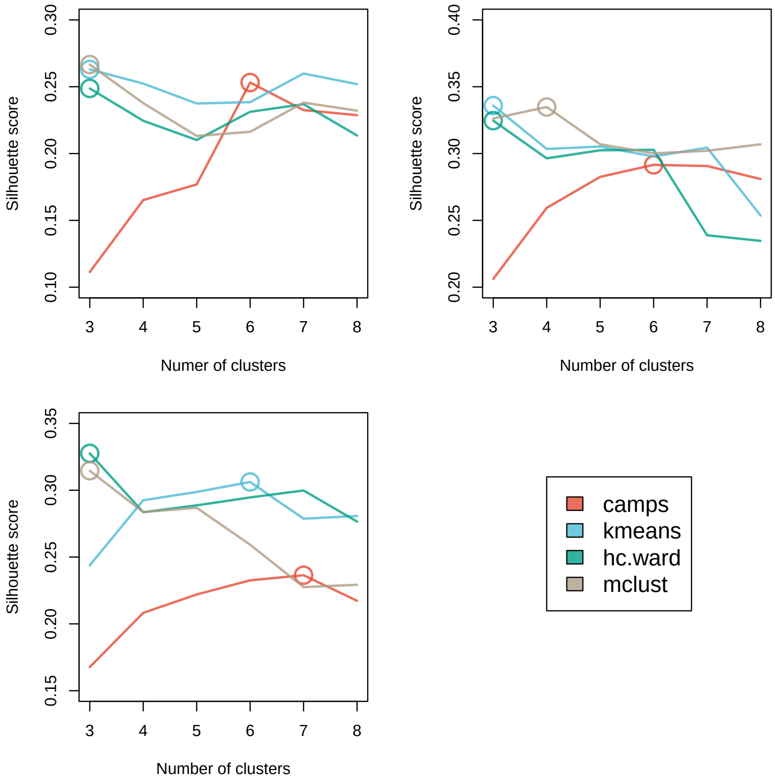


Figure 5.15: Silhouette scores for K values ranging from 3 to 8. The silhouette score was computed for CAMPS, K-Means, HClust with ward linkage, and MClust. Circles indicate the maximum score for the clustering algorithm. The top left plot displays the scores for the first simulated dataset, the top right plot shows the scores for the second simulated dataset, and the bottom left plot presents the scores for the real dataset.

..

Model parameters

Except for the number of clusters estimated, the parameters for the clustering methods remained consistent across the different datasets considered.

Dataset	camps	kmeans	hc.ward	mclust
Unfiltered AML data	6	10	11	11
Filtered AML data	7	6	3	3
Simulation 1	6	3	3	3
Simulation 2	6	3	3	4

Table 5.5: Optimal K-values estimated by the silhouette score. For unfiltered AML data we considered K-values in the range between 5 and 15. For the other data sets, we considered K-values in the range from 3 to 8.

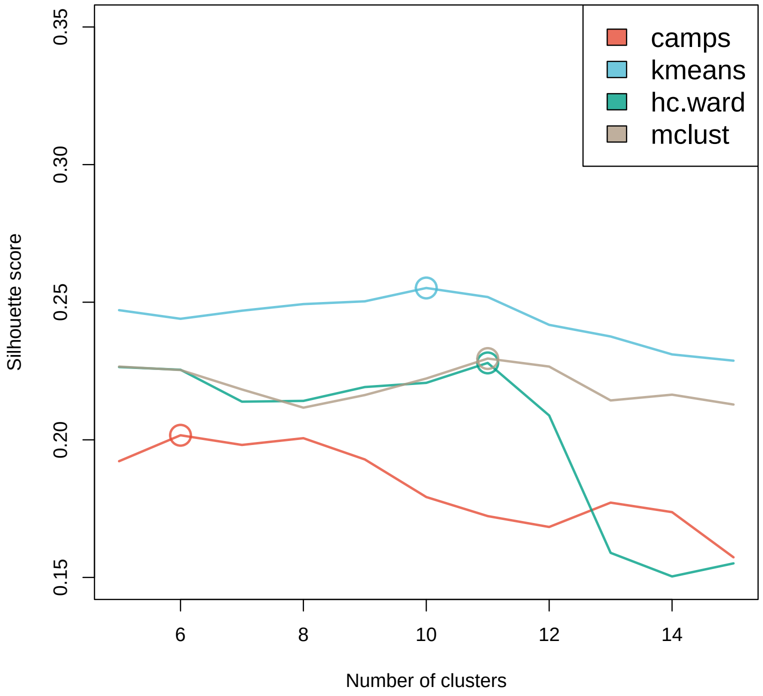


Figure 5.16: Silhouette scores for K values ranging from 5 to 15. The silhouette score was computed for CAMPS, K-Means, HClust with ward linkage, and MClust. Circles indicate the maximum score for the clustering algorithm.

Euclidean distance was utilized as the distance measure for all methods. For K-means, a maximum of 100 iterations was performed using K-means++ initialization. Step 1 of the CAMPS algorithm involved scaling the expression values by protein markers (column-wise) to the 95th percentile, followed by K-means clustering with $K = 1000$, using K-means++ initialization and a maximum of 150 iterations. Details on the input file defining the protein panels are described in Table 4.1. For the Metaclust algorithm, Step 1 was performed using K-means with $K = 500$, K-means++ initialization, and a maximum of 300 iterations. The resulting centroids were then combined using hierarchical clustering with ward linkage to produce the final clustering result.

Regardless of the linkage method, hierarchical clustering involves computing a distance matrix for all pairs of observations. This proved to be too memory-intensive for any of the datasets shown in Table 5.3. Consequently, a downsampled dataset, consisting of a random sample of 10,000 cells taken without replacement, was used to ensure manageability and efficiency in the clustering process. Additionally, hierarchical clustering with a minimum cluster size requirement was performed, ensuring at least three observations per cluster for single linkage and fifty observations for complete and average linkage. The minimum size requirement for ward linkage had no effect on the results and was therefore excluded.

When estimating the number of clusters K using the silhouette score for any of the aforementioned methods, the same downsampled dataset used in hierarchical clustering was considered. This is because computing the silhouette score of a clustering, like hierarchical clustering, involves calculating the pairwise distances between all observations, leading to memory issues with the available hardware. For K-means, CAMPS, and Metaclust, once the optimal K was estimated, a new clustering was performed on the entire dataset using the determined K.

The running times of the various clustering algorithms varied significantly. K-means was notably fast, typically converging within a few minutes. Hierarchical clustering methods were also relatively quick, but this speed is primarily due to the need to downsample the dataset to make the computations feasible. On the other hand, MClust and CAMPS were the most time-demanding methods. This increased computational time is mainly due to the initial step in both algorithms, which involves running K-means with a very high K value. This step can be computationally expensive. However, once this initial step is completed, the remainder of the clustering process for both metaclust and camps is almost instantaneous. This characteristic means that selecting different values of K or calculating the silhouette score is very fast after the initial step is finished.

Comparing clustering outcomes

As a first inspection of the clustering results, we consider the sizes of the clusters when K=6 on the filtered AML dataset (see Figure 5.17). The first thing to notice is that essentially all observations in hc.single, hc.single*, and hc.average are contained in a single cluster. This pattern was consistent when we clustered the simulated data as well, across various choices of K. Regardless of whether K was set to 4, 6, 8 or 10, these methods tended to produce a few dominant clusters containing the majority of the observations, with the remaining clusters containing very few observations. This behavior indicates that these linkage methods are not well-suited for our data, as they fail to partition the dataset into meaningful clusters adequately. This issue was also evident in hc.complete, hc.complete*, and hc.average* in most scenarios. To achieve a more balanced data partitioning, we had to increase the minimum observation criterion to such an extent that a significant number of cells were discarded, resulting in uninformative results. Therefore, we decided to focus on hierarchical clustering with Ward linkage, which did not exhibit the tendency to consolidate most observations into a few clusters. Ward linkage provided a more effective dataset partitioning, making it a more suitable choice for our hierarchical clustering analyses moving forward.

For the purpose of comparing the similarity of the clustering results, the adjusted Rand index (see Section 3.6) was calculated between all pairs of clustering assignments. Since the hierarchical clustering methods were performed on a downsampled dataset, kmeans, camps, and metaclust were also applied to this sampled dataset to ensure direct comparability of the clustering assignments. The results with $K = 6$ on the unfiltered AML dataset are shown Figure 5.18. We see that camps, kmeans, hc.ward and metaclust produce the most similar results. Specifically, the ARI between the camps assignments and kmeans assignments is 0.95, indicating a high degree of agreement between the groupings. We also notice that hc.average, hc.single, and hc.single* have ARI close to 0 when compared to the other clustering results. This is unsurprising given the disproportionate cluster sizes we see in Figure 5.17. The inspection of the similarity between clusterings performed on the other datasets and with other K-values tells similar stories.

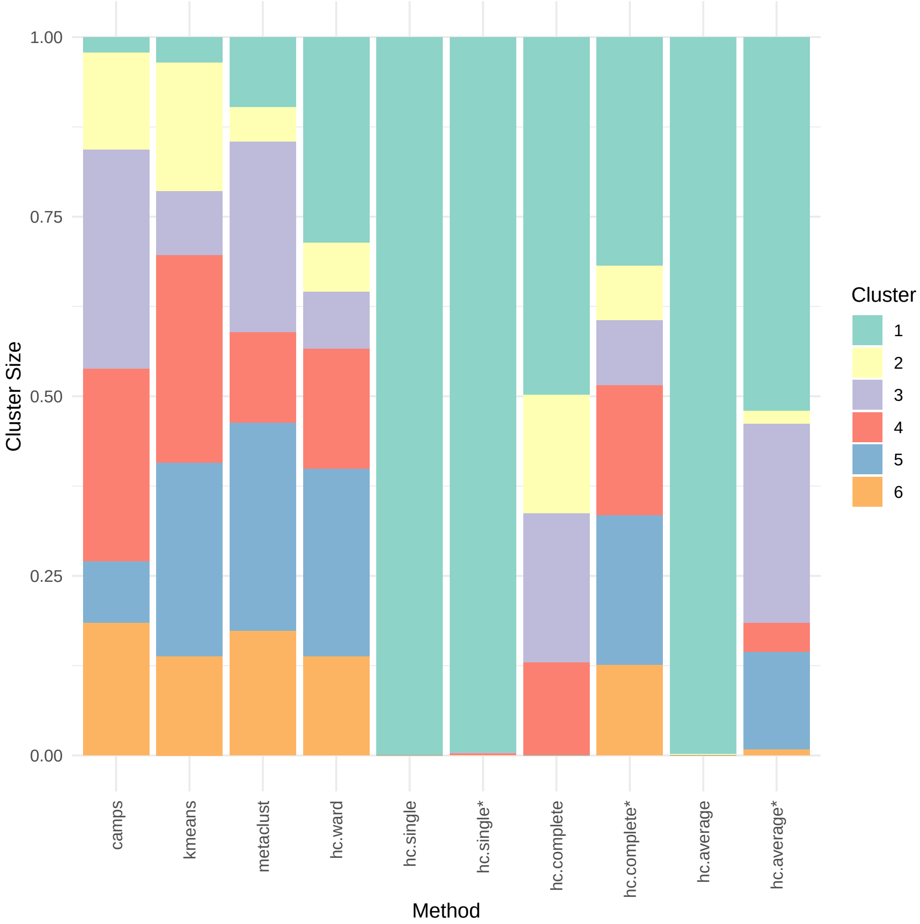


Figure 5.17: Cluster sizes. Stacked bar plot showing the relative sizes of clusters for different clustering methods used on the real CyTOF dataset with K=6. Each bar represents a clustering method, with the colors within each bar indicating the different clusters. The y-axis represents each cluster's proportion relative to the clusters' total size for that method.

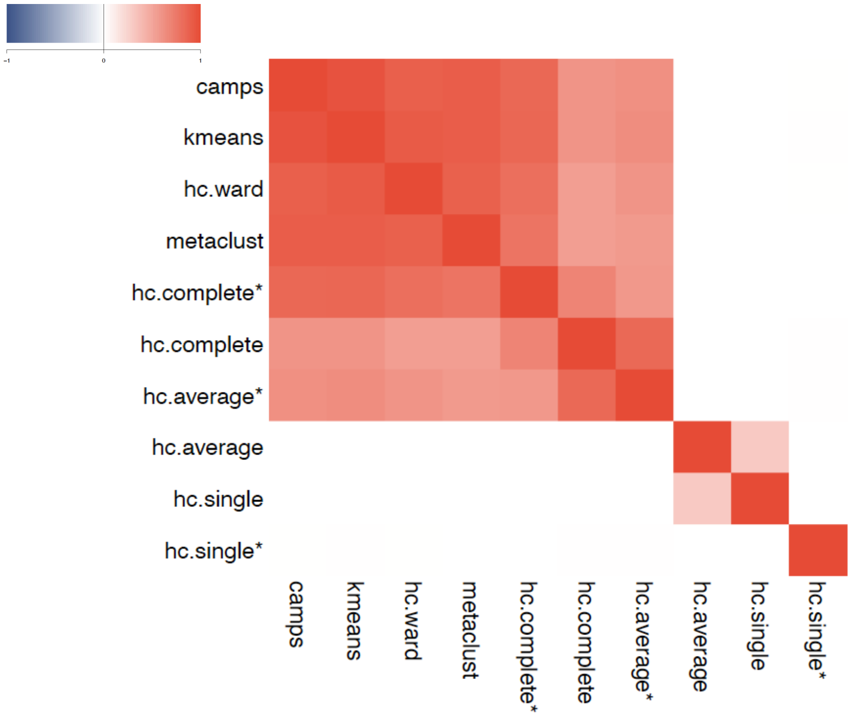


Figure 5.18: Similarity of clustering results. Heatmap displaying the adjusted Rand index (ARI) of the cluster assignments produced by various clustering methods for K=6 on a downsampled version of the filtered AML dataset. The color intensity represents the similarity of the cluster assignments, with darker shades indicating higher similarity. This visualization provides a comparative assessment of how closely the different clustering methods agree regarding cluster assignments.

5.5 Quality of cell type identification

In this section, we evaluate the effectiveness of various clustering methods in identifying individual cell types. We focus on datasets with known true labels: the filtered AML data containing manually labeled cells, and the simulated data. The primary analysis centers on the results when $K = 6$, corresponding to the actual number of cell types identified through manual gating, and the number of simulated clusters (see Sections 5.2 and 3.8).

We begin by examining the performance of hierarchical clustering methods—single linkage (hc.single), average linkage (hc.average), and complete linkage (hc.complete)—on the filtered AML data. These results are then compared to those obtained using an additional criterion for a minimum number of observations per cluster. Figure 5.19 illustrates these comparisons.

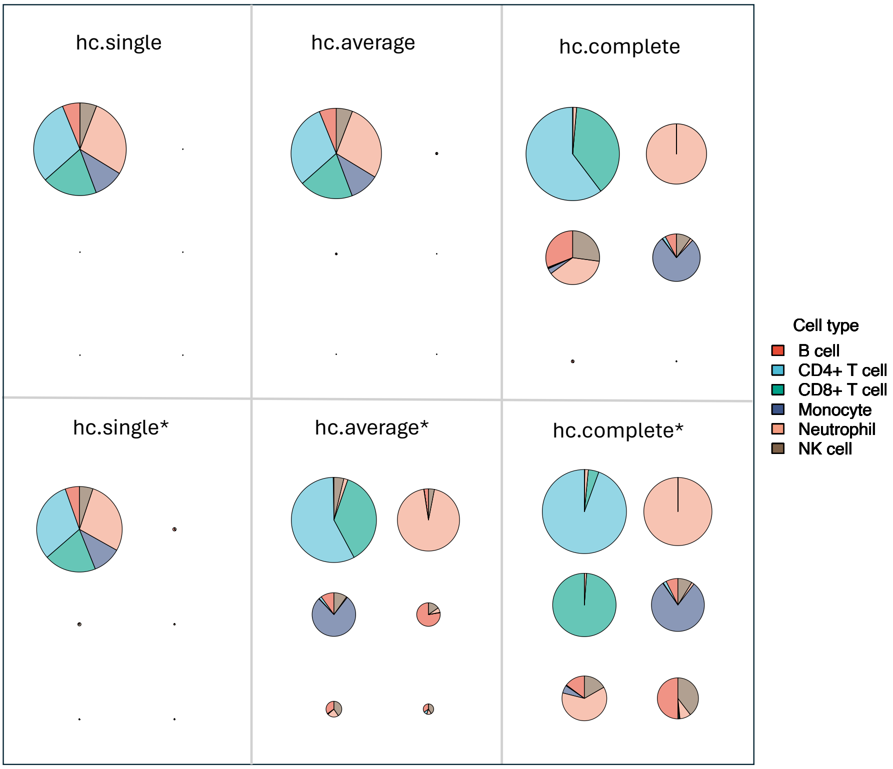


Figure 5.19: Celltype composition of clusters. Each disk represents a cluster, and the area of the disk is proportional to the size of the cluster. Colors indicate actual cell type. The top panel shows the clustering results from using hierarchical clustering using single, average, and complete linkage without the minimum observations criterion. The bottom panel shows the same linkage methods with the minimum observations criterion included. All methods use K=6 (fixed) and are applied to the filtered AML dataset described in the section above.

As shown previously (Figure 5.17), single linkage tends to assign nearly all cells to a single cluster, resulting in an uninformative clustering. Applying a minimum observations criterion (min_obs = 3) did not enhance the results. Average linkage produced similar outcomes to single linkage. However, introducing the minimum observations criterion (min_obs = 50) noticeably improved performance, yet it still failed to differentiate the cells adequately. Complete linkage demonstrated the most promising results, though two clusters remained completely empty. Enforcing the minimum cluster size requirement (min_obs = 50) for complete linkage yielded more meaningful clusters, successfully separating several cell types. Despite this improvement, we later demonstrate that Ward's method outperformed all other hierarchical clustering methods. Therefore, hierarchical clustering with Ward linkage will be our selected hierarchical clustering method for further comparisons.

Next, we investigated how well kmeans, hclust with Ward linkage, mclust and camps were able to differentiate the cell types in the filtered AML dataset (see Figure 5.20). K was fixed to 6 for all methods.

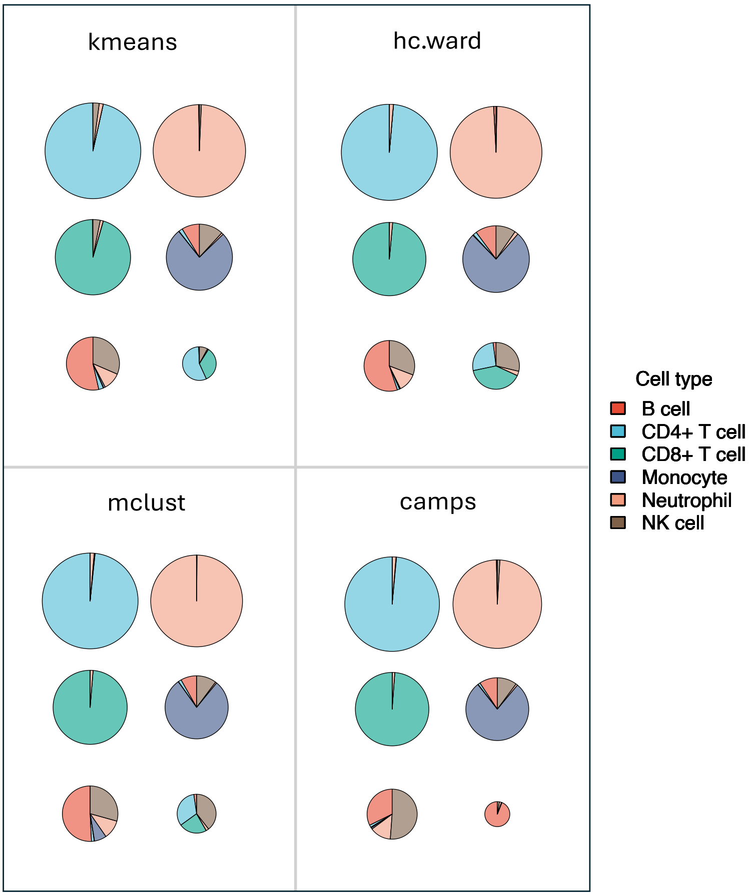


Figure 5.20: Celltype composition of clusters. Each disk represents a cluster, and the area of the disk is proportional to the size of the cluster. Colors indicate actual cell type. All clustering methods displayed use K=6 (fixed) and are applied to the filtered AML dataset described in the section above.

By visual inspection, none of the clustering methods clearly appear to outperform or underperform the others. K-means, hc.ward, and mclust all produce very similar clusters. However, camps performs slightly better in identifying a single cluster that contains all the CD8$^+$ T cells. The Adjusted Rand Index (ARI) was calculated to compare the estimated groupings with the manually identified groupings of the cells (see Section 5.2). The ARI of the clustering methods with $K = 6$ on the filtered AML dataset are displayed in Figure 5.21.

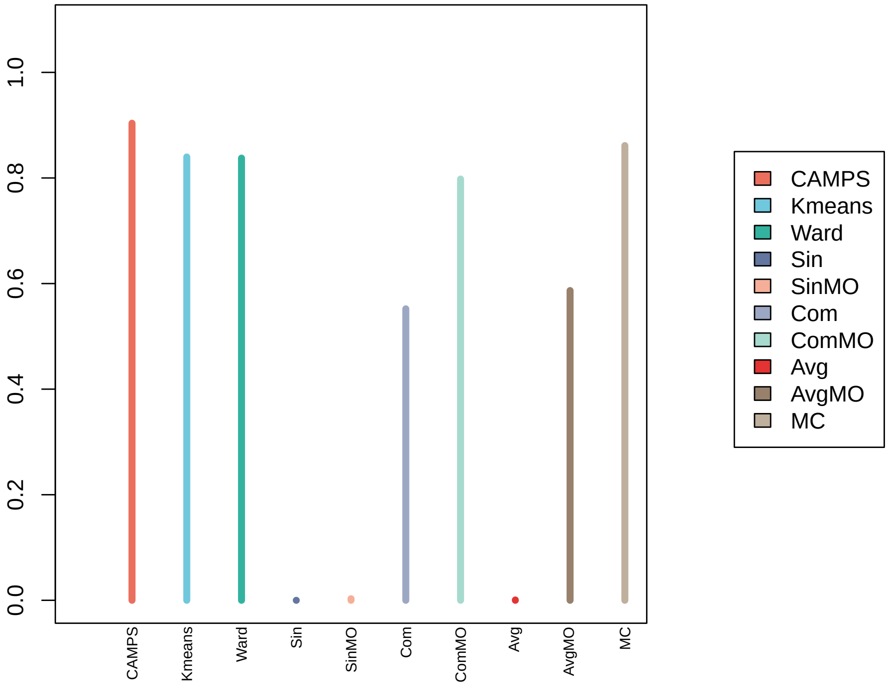


Figure 5.21: ARI values for K=6 on filtered AML dataset. The height of the bar shows the adjusted Rand index for the different clustering methods. The color of the bar represents the clustering method that was applied. ”MO” at the end of the model names indicates that the minimum observation criterion was used.

The camps algorithm achieved the highest ARI value of approximately 0.904. It outperforms all other clustering techniques, including mclust (ARI=0.861), kmeans (ARI=0.840), and hclust with ward linkage (ARI=0.838). These methods showed strong clustering performance but did not match the accuracy of the camps algorithm. In general, we see that linkage methods other than Ward perform worse.

We next investigated the effects of estimating the number of clusters instead of the fixed approach, as this would be a more realistic setting for real applications (see figure 5.22). In Table 5.5, we saw that when K is estimated by silhouette score, then $K = 7$ for camps, $K = 6$ for kmeans, $K = 3$ for hc.ward, and $K = 4$ for mclust when the filtered AML dataset was considered.

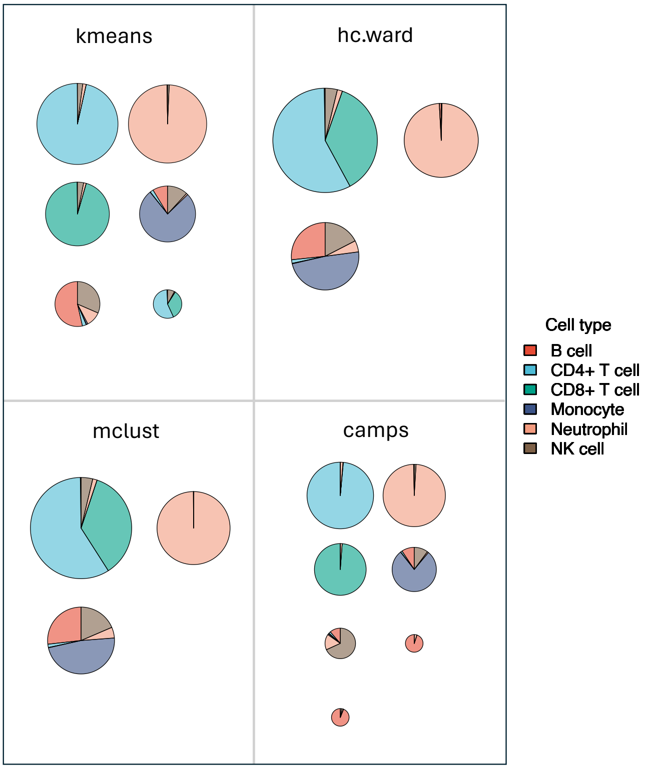


Figure 5.22: Celltype composition of clusters. Each disk represents a cluster, and the area of the disk is proportional to the size of the cluster. Colors indicate actual cell type. All clustering methods displayed use K estimated by silhouette score and are applied to the filtered AML dataset described in the section above. ARI to each clustering is displayed in 5.23.

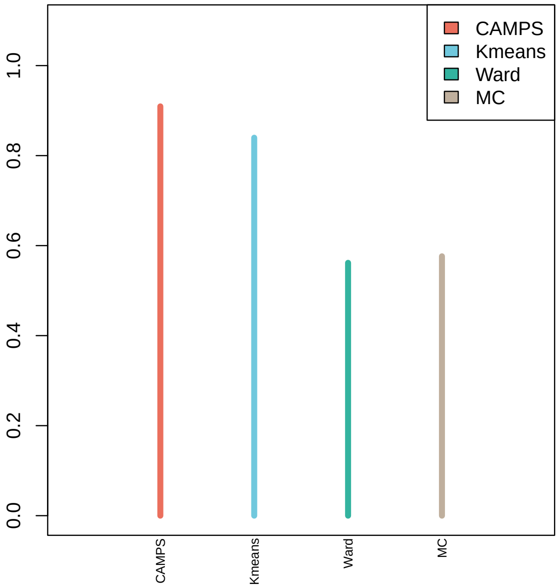


Figure 5.23: ARI values for K chosen by silhouette on filtered AML data. The height of the bar shows the adjusted Rand index for the different clustering methods. The color of the bar represents the clustering method that was applied.

Again, we see that CAMPS performs better than the other clustering methods on the filtered AML dataset. The number of clusters used by ward and mclust (K=3) is clearly the reason why they are underperforming in this instance. The adjusted Rand index is shown in Figure 5.23.

We then proceeded to measure the clustering methods' ability to identify the simulated cell types in the first simulated dataset (see table 5.3. In this dataset, 10,000 cells of each cell type were generated for a balanced dataset. We first considered $K = 6$ fixed for all methods. The results of the clusterings are shown in Figure 5.24

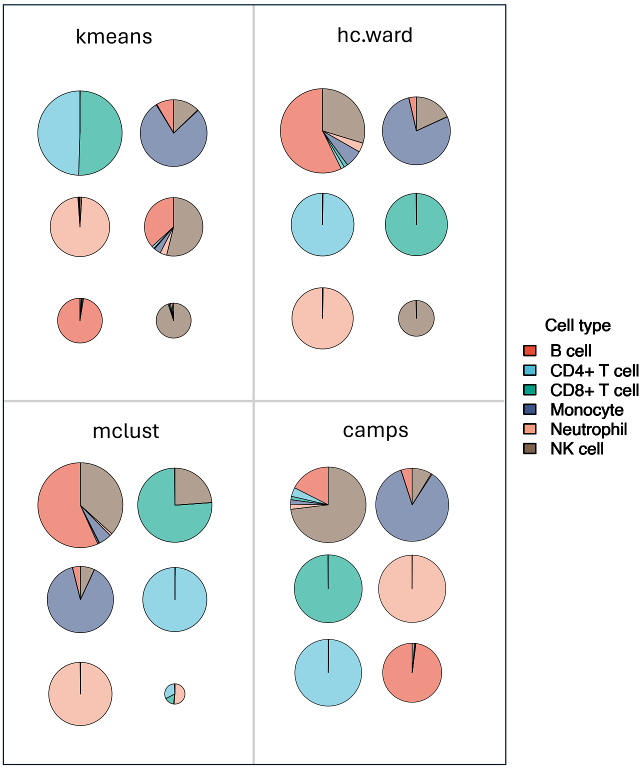


Figure 5.24: Celltype composition of clusters. Each disk represents a cluster, and the area of the disk is proportional to the size of the cluster. Colors indicate actual cell type. All clustering methods displayed use $K = 6$ (fixed) and are applied to the first simulated dataset described in the section above. ARI to the clusterings shown here (and others) is displayed in 5.25.

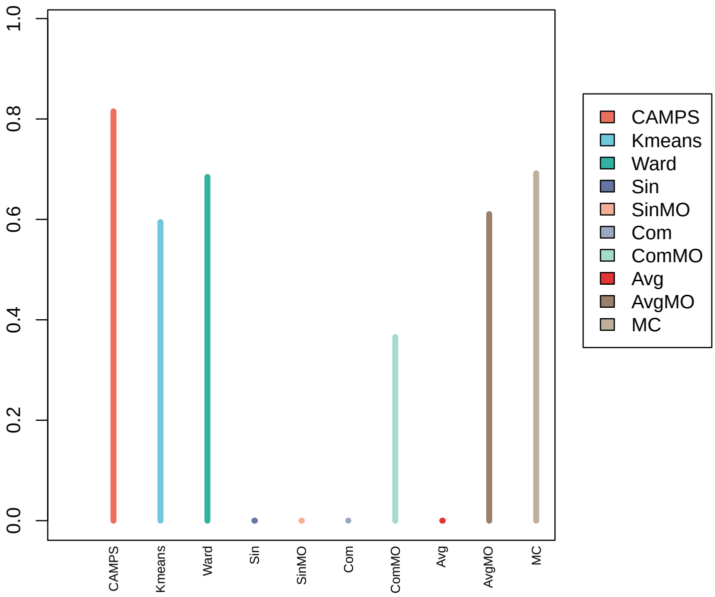


Figure 5.25: ARI values for K=6 (fixed) on the first simulated data set (equal proportions). The height of the bar shows the adjusted Rand index for the different clustering methods. The color of the bar represents the clustering method that was applied.MO” at the end of the model names indicates that the minimum observation criterion was used.

In this instance, we see some bigger differences between the results. Kmeans clearly struggles to separate the two major subtypes of T cells. hc.ward and mclust can separate these, but the cluster sizes are unbalanced, with some clusters being much smaller than others even though the simulated cells are of equal proportions. CAMPS on the other hand is able to do this much better, with very balanced cluster sizes. In particular, we see that CAMPS is able to differentiate between the B cells and the NK cells better than hc.ward and mclust. The adjusted Rand index is shown in Figure 5.25. We also investigated the results when the silhouette score estimated the optimal number of clusters. The results are shown in Figure 5.26.

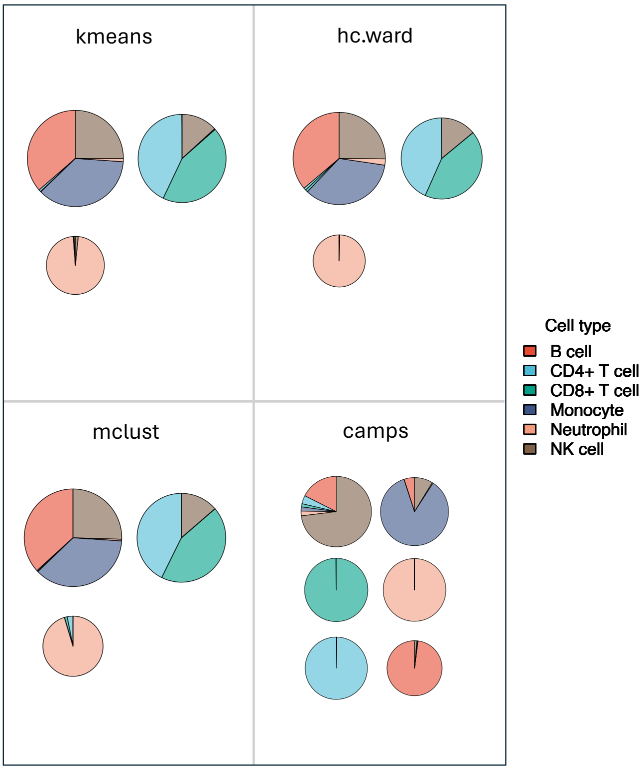


Figure 5.26: Celltype composition of clusters. Each disk represents a cluster, and the area of the disk is proportional to the size of the cluster. Colors indicate actual cell type. All clustering methods displayed use $K = 6$ (fixed) and are applied to the first simulated dataset described in the section above. ARI to the clusterings shown here (and others) is displayed in 5.27.

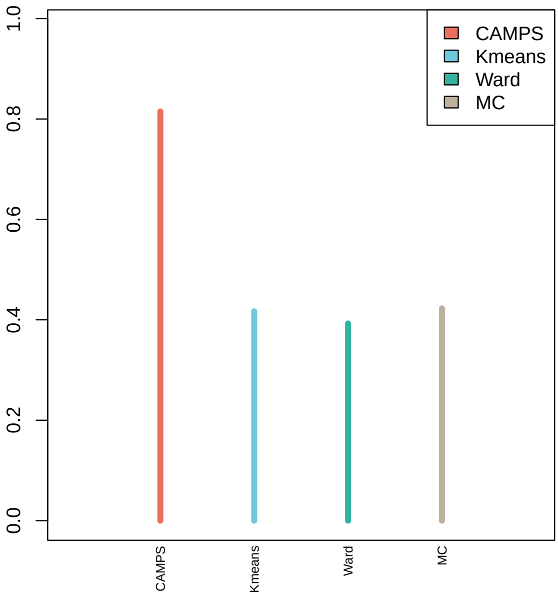


Figure 5.27: ARI values for K estimated by silhouette on the first simulated data set (equal proportions). The height of the bar shows the adjusted Rand index for the different clustering methods. The color of the bar represents the clustering method that was applied.

The silhouette score estimated 3 clusters for kmeans, hc.ward, and mclust, leading to under-clustering of the data. Camps, however, identified the ”correct” number of clusters with silhouette score, and outperformed all the other methods. The adjusted Rand index is shown in Figure 5.27.

The analysis of comparing the clustering performance of known cell types was repeated for the filtered AML dataset and both of the simulated data sets. Figure 5.28 compares all methods used on all three datasets with $K = 6$ (fixed). Furthermore, the process was repeated for different values of K. These results are shown in Figure 5.29.

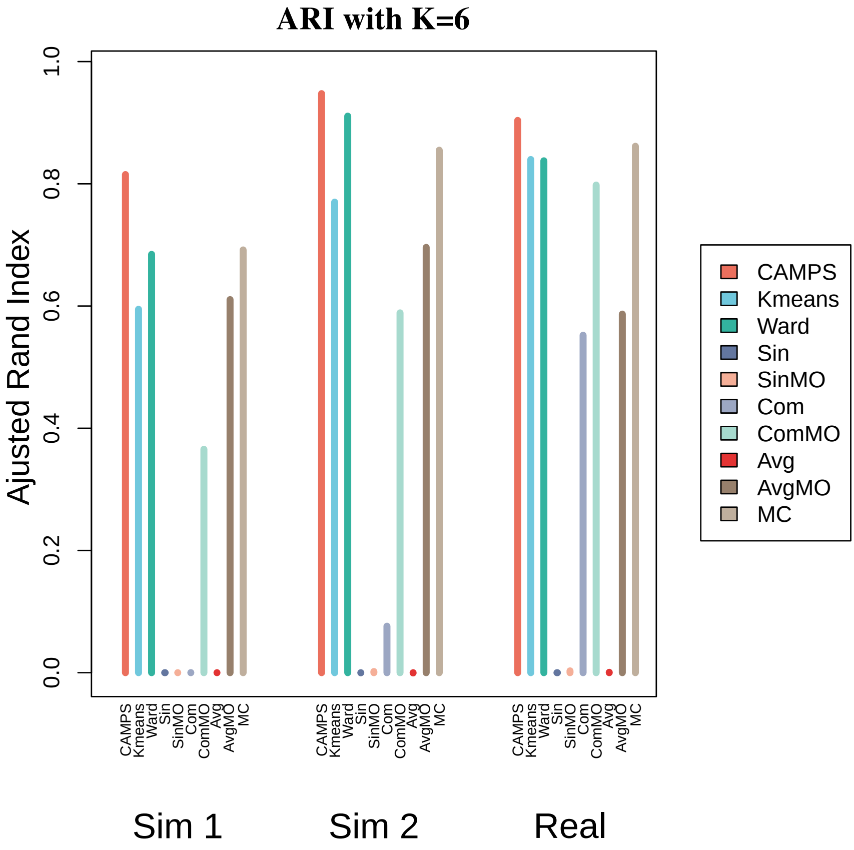


Figure 5.28: ARI on all datasets for K=6. Barplot showing the adjusted Rand index (ARI) for the clustering methods used on sim1, sim2 and the real filtered AML dataset. All clusters are estimated with K=6. SinMO, ComMO, and AvgMO refer to HClust with single, complete, and average linkage using the minimum observations criterion.

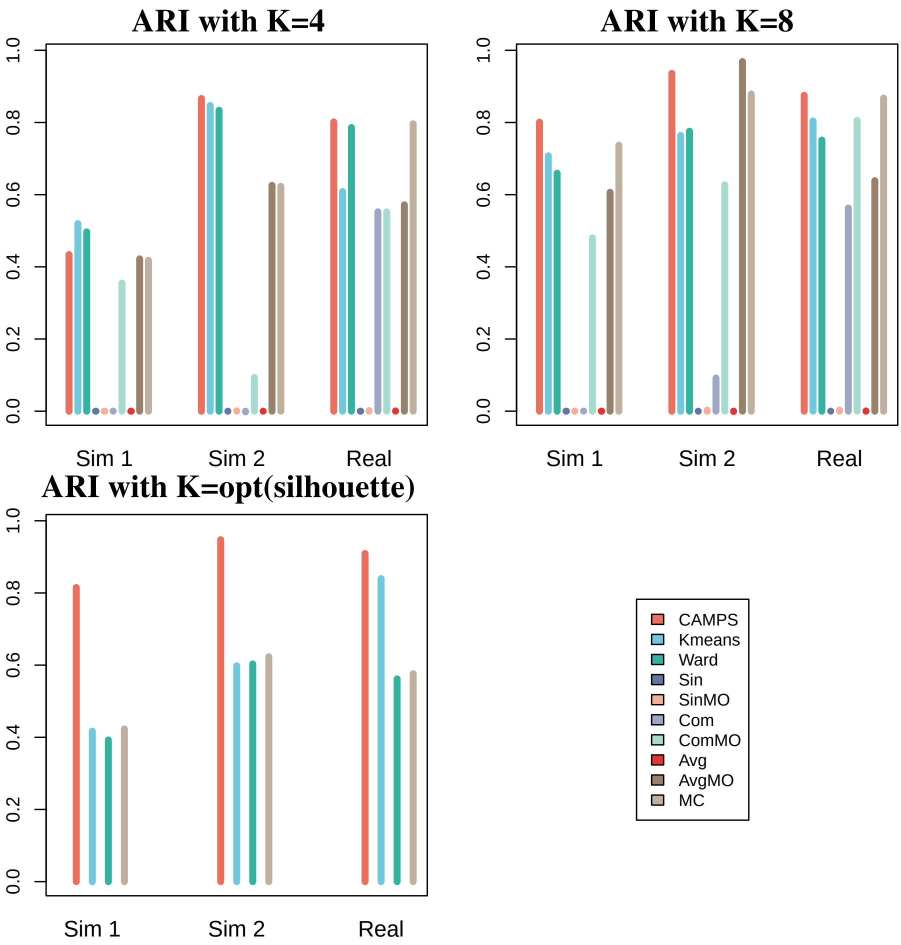


Figure 5.29: ARI on all datasets for K=4, K=8, and with K estimated by silhouette. Barplots showing the adjusted Rand index (ARI) for the clustering methods used on sim1, sim2, and the real filtered AML dataset. The top left plot shows the results from clustering with K=4, while the top right plot shows the ARI for K=8. The bottom right plot shows the ARI when the optimal number of clusters is estimated by maximizing the silhouette score. The silhouette score was only maximized for CAMPS, K-Means, HClust with ward distance, and MClust. SinMO, ComMO, and AvgMO refer to HClust with single, complete, and average linkage using the minimum observations criterion.

Robustness to noisy data

We wanted to determine how the clustering algorithms responded to increased or decreased noise in the expression values. For this, we used our simulation model (see Section 3.8) to generate cells where the covariance matrix Σ was multiplied by a scaling factor (alpha) in the range $0.25, 0.35, 0.45, \ldots, 3.95$. The clustering methods were then tasked with clustering these generated datasets, and the adjusted Rand index was computed to find how well the clusters represented the real (simulated) cell types. The results are shown in Figure 5.30.

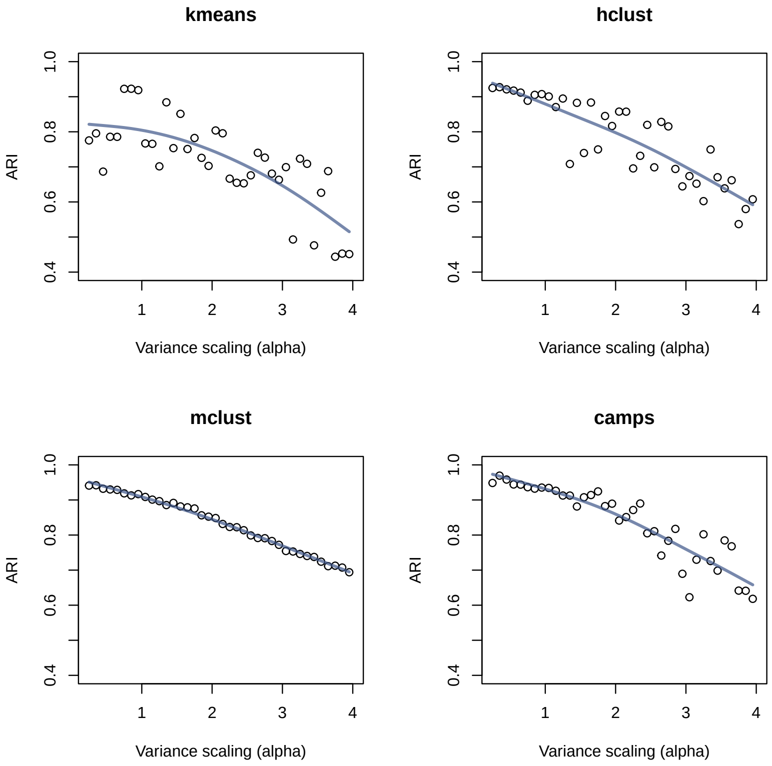


Figure 5.30: Performance of clustering methods on different degrees of noise in the simulated data. The plots show how ARI is affected for each of the methods by the scaling of alpha on Σ in the simulation model.

We see that mclust, in particular, stands out as being very robust to changes in the variance of the marker expressions in the simulated cells, with a very stable reduction in performance. Camps also show stability for smaller alpha values but exhibit more variability in the performance for larger alpha values. Figure 5.31 shows the four curves together.

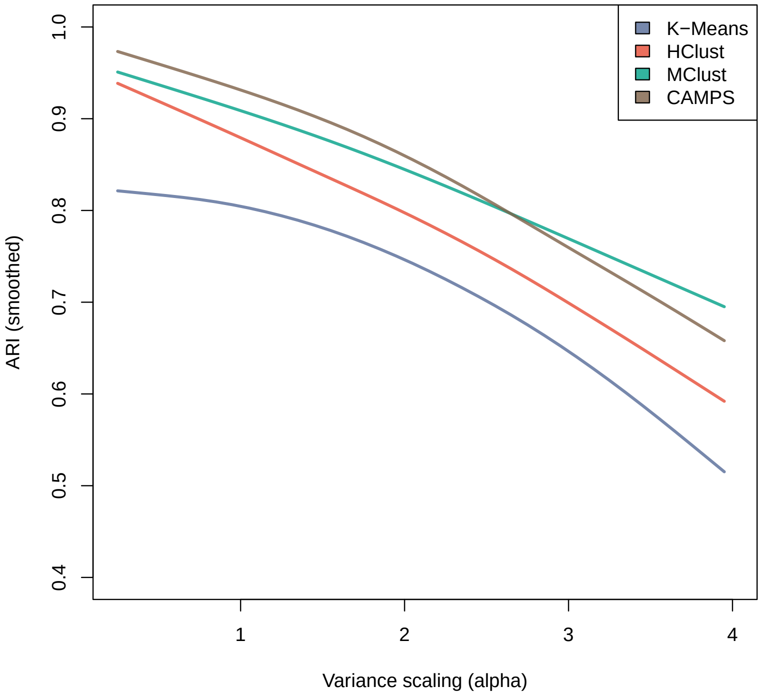


Figure 5.31: Performance of clustering methods on different degrees of noise in the simulated data. The color represents the different clustering methods. The data points are smoothed using smooth splines.

We observe that CAMPS achieve the best performance in identifying cell types for lower values of the scaling factor alpha. As the variance in the data increases, mclust eventually outperforms the other methods.

Finally, we plotted heatmaps showing the median protein expression in the clusters estimated with CAMPS with K estimated by the silhouette score and when clustering the entire unfiltered AML dataset (see Figure 5.32). Similar results are shown for the k-means algorithm in Figure 5.33.

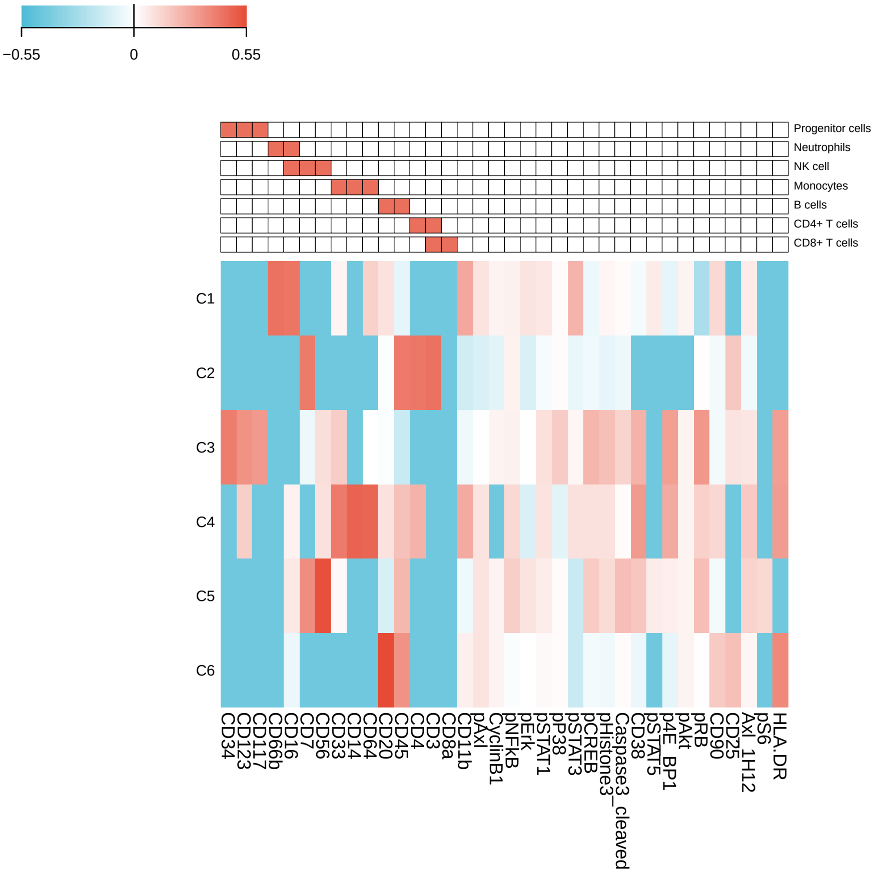


Figure 5.32: Heatmap showing the median protein expressions in the clusters estimated by CAMPS with K estimated by silhouette score (K=6) when clustering the entire unfiltered AML dataset. Each column represents a different protein marker, and each row corresponds to one of the six clusters (*C*1 to *C*6). The color intensity indicates the median protein expression. The bars above the heatmap show some of the major immune cell populations we expect to be present in the samples. The red boxes indicate that we expect the protein to be expressed in the cell type.

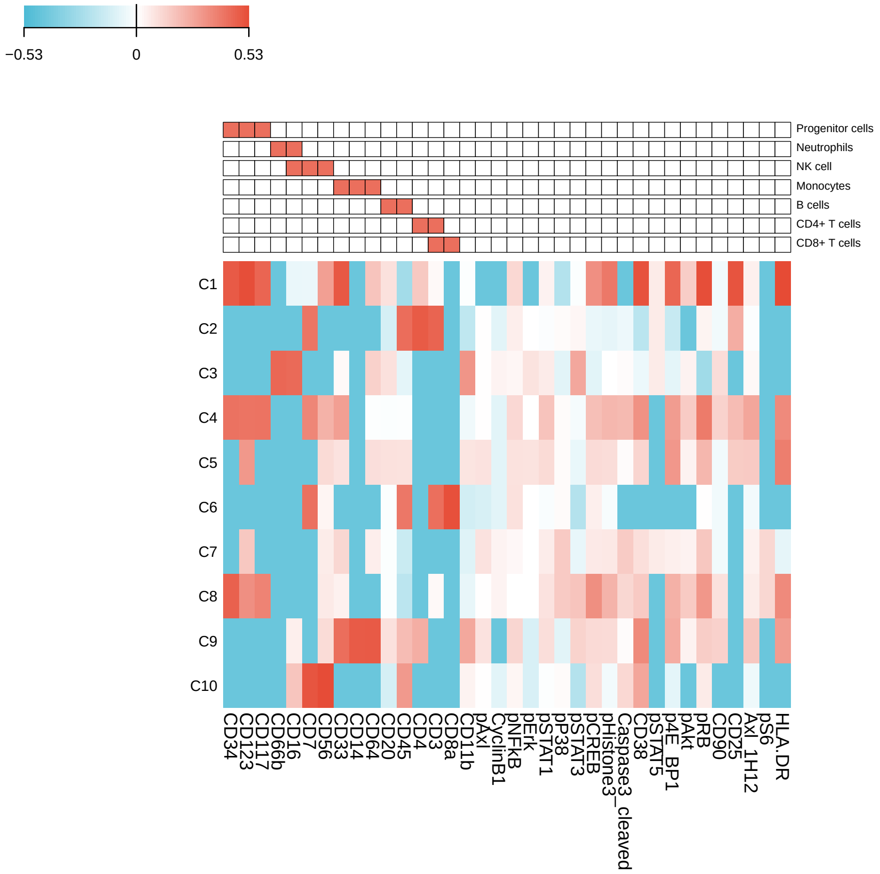


Figure 5.33: Heatmap showing the median protein expressions in the clusters estimated K-Means with K estimated by silhouette score (K=10) when clustering the entire unfiltered AML dataset. Each column represents a different protein marker, and each row corresponds to one of the six clusters ($C1$ to $C10$). The color intensity indicates the median protein expression. The bars above the heatmap shows some of the major immune cell populations we expect to be present in the samples. The red boxes indicate that we expect the protein to be expressed in the cell type.

5.6 Quality as features for prediction

The merit of the different clustering methods can be measured in ways other than their ability to identify cell types. In this section, we will consider the clusters' ability to form useful features for predicting treatment response and survival in the AML data set.

Making the features

The features used for prediction are all based on the patient's proportion of cells assigned to each of the clusters identified by K-means, metaclust, camps, and hierarchical clustering with ward distance. Each of the 30 AML patients had blood samples taken at 0 hours and 24 hours after the start of treatment. For each patient, we isolated the cells from the 0-hour sample and determined the proportion of these cells assigned to each cluster. We then repeated this process for the patient's 24-hour sample. Figure 5.34 shows how the 24-hour samples are distributed among the clusters estimated by camps with $K = 6$ (estimated by silhouette).

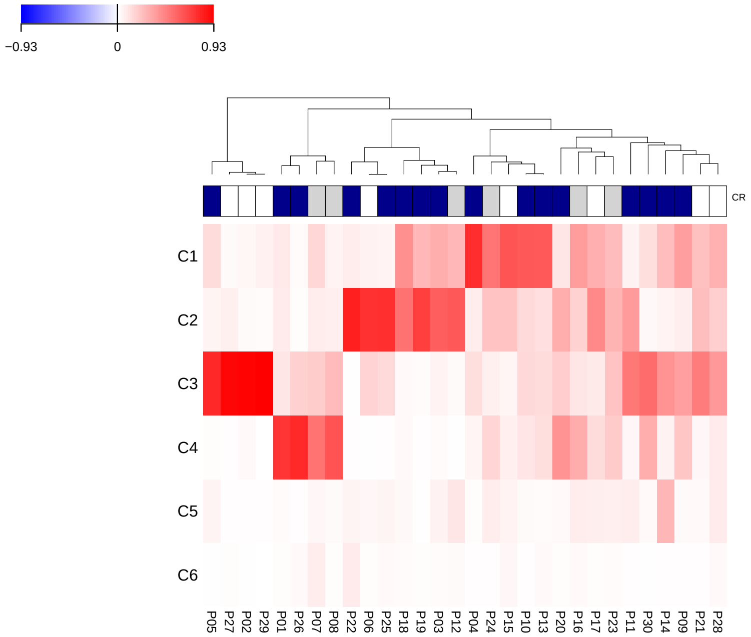


Figure 5.34: Proportion of cells in clusters. Heatmap showing the proportion of cells assigned the clusters estimated by CAMPS with K=6 (estimated by silhouette score) for each patient's 24-hour blood sample. Each column represents a different patient, and each row corresponds to one of the six clusters (C1 to C6). The color intensity indicates the proportion of cells assigned to each cluster, with darker shades of red representing higher proportions. The dendrogram at the top groups patients based on the similarity of their cluster proportions. The annotation bar at the top (CR) indicates whether the patient was cancer-free (blue box) 17 days after the start of treatment or not (white box). Grey indicates that the complete response variable is not available for the patient.

We calculated the entropy (see Section 3.10) of the distribution of cells in the clusters for each patient's 0-hour sample and 24-hour sample. These entropies served as our first two predictive features. To capture changes over time, we computed the absolute difference in entropy between the samples taken at 0 and 24 hours. Additionally, we assessed the Kullback-Leibler divergence (see Section 3.11) between the cluster proportions at 0 hours and 24 hours, offering a fourth feature. These four features, derived from the clustering methods, will be utilized to predict treatment response and survival outcomes in the AML dataset. An overview of the features used for prediction and their acronyms are found in Table 5.6

Feature	Acronyms
Entropy at 0 hours	ent0
Entropy at 24 hours	ent24
Absolute change of entropy	delta
Kullback Leibler divergence	kl

Table 5.6: Features for prediction. The table overviews the features used to predict treatment response and survival. The features were calculated for each of the 30 patients in the study, and are based on the proportions of cells assigned to the various clustering methods.

Predicting response to treatment

Next, we investigated and compared the predictive capabilities of the features formed by the different clustering methods. We employed logistic regression using the features described to predict the binary complete response (CR) variable. The distribution of the response variable is shown in Figure 5.35. Note that 6 patients were missing the response variable.

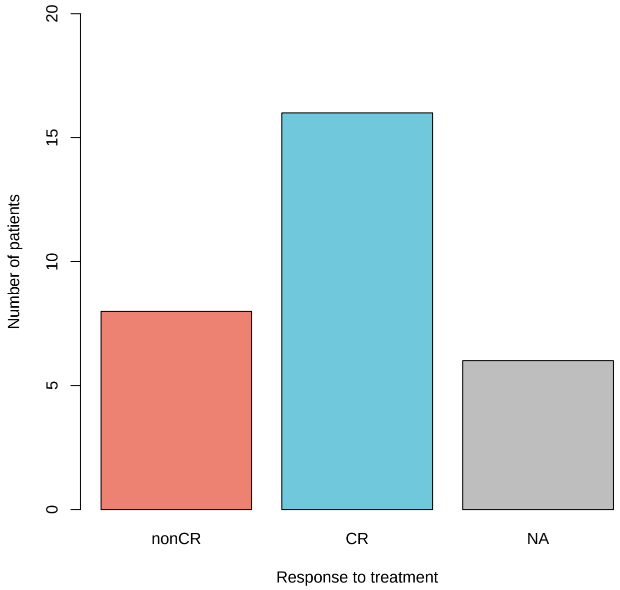


Figure 5.35: Distribution of the CR response variable. Barplot showing the number of patients with non-complete response (nonCR), complete response (CR) and missing values (NA). 8 patients had nonCR, 16 had CR and 6 were missing from the clinical data provided.

Feature	kmeans	hclust	mclust	camps
ent0	0.380	0.507	0.419	0.182
ent24	0.727	0.836	0.928	0.389
delta	0.194	0.320	0.094	0.069
kl	0.131	0.447	0.162	0.165

Table 5.7: P-values from logistic regression. The features are derived from the clustering results of kmeans, hclust, mclust, and camps with K chosen by silhouette scores. P-values are rounded to three decimal places.

Table 5.7 shows the p-values of performing logistic regression. Delta entropy is the most promising feature for predicting CR, particularly when derived from mclust and camps clustering methods. However, none of the features achieved statistical significance. Inspection of the distribution of the feature variables revealed that some of these were right-skewed (distributions are shown in Figure 5.36). Therefore, we decided to log-transform the features used in the logistic regression model to potentially improve the predictive power. The resulting p-values are shown in Table 5.8.

Feature	kmeans	hclust	mclust	camps
log(ent0)	0.339	0.383	0.316	0.101
log(ent24)	0.578	0.712	0.943	0.170
log(delta)	0.430	0.734	0.036	0.045
log(kl)	0.077	0.326	0.097	0.040

Table 5.8: P-values from logistic regression using log-transformed features. The features are derived from the clustering results of kmeans, hclust, mclust, and camps with K chosen by silhouette scores. P-values are rounded to three decimal places.

The log transformation of features enhances the predictive power of certain features, particularly delta entropy and KL divergence when derived from the mclust and camps clustering methods. These results suggest that log-transformed features might be more effective in capturing the underlying relationships in the data, improving the accuracy of the logistic regression model in predicting treatment response.

Survival prediction

The next step in our analysis was to predict the overall survival of patients using features derived from different clustering methods. For this purpose, we employed the Cox proportional hazards model (see Section 3.16). The p-values obtained from the Cox regression are presented in Table 5.9.

Feature	kmeans	hclust	mclust	camps
ent0	0.4597	0.1850	0.5894	0.8888
ent24	0.0880	0.0112	0.0515	0.4740
delta	0.0029	0.0154	0.0019	0.0005
kl	0.0642	0.0652	0.0145	0.0207

Table 5.9: P-values from Cox proportional hazards model (likelihood ratio test). The features are derived from the clustering results of kmeans, hclust, mclust, and camps with K chosen by silhouette scores. P-values are rounded to four decimal places.

None of the clustering methods show significant p-values for the entropy of the 0-hour blood sample, indicating it does not independently predict overall survival. The p-values for hclust (0.0112) and mclust (0.0515) suggest that entropy at 24 hours might be a meaningful predictor of survival, with hclust showing a significant association. Overall, delta entropy stands out as the most robust predictor of patient survival, with significant p-values across all considered clustering methods. The Kullback Leibler divergence is also a significant predictor of patient survival when derived from camps (p=0.0207) and mclust (p=0.0145).

Following the Cox regression analysis, we utilized the logrank test (see Section 3.15) to further investigate the predictive capabilities of our different clustering-derived features. To apply the logrank test, we need to stratify the population of patients. We did this by calculating the p-value using different splits of the features. Each feature was split at the the following quantiles: $0.250, 0.275, 0.300, \ldots, 0.750$. We then used the logrank test to find the split that resulted in the lowest p-value. The p-values found by using the logrank test with the optimal threshold for stratifying the patients can be found in Table 5.10. Figure 5.36 shows the distribution of the features and where the population was split.

Feature	**kmeans**	**hclust**	**mclust**	**camps**
ent0	0.0811	0.0477	0.1516	0.2185
ent24	0.0024	0.0009	0.0111	0.1187
delta	0.0005	0.0164	0.0033	0.0026
kl	0.1151	0.0896	0.0199	0.0185

Table 5.10: P-values from logrank test The population of patients was stratified at the quantile of the features that minimized their p-value. The features are derived from the clustering results of kmeans, hclust, mclust, and camps, with K chosen by silhouette scores.

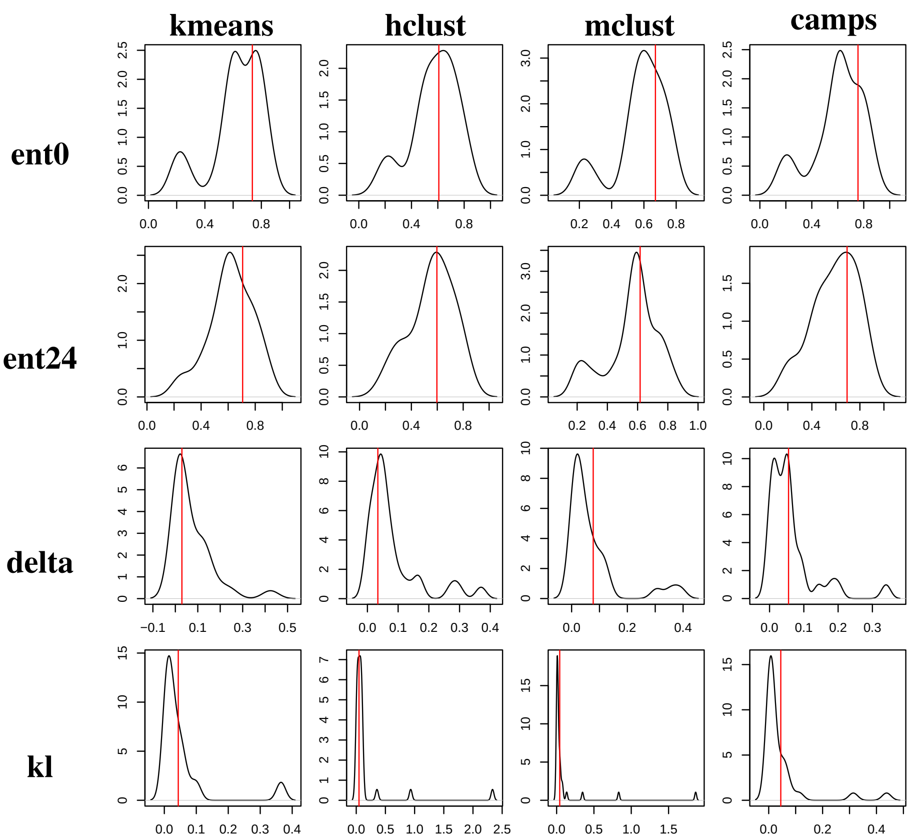


Figure 5.36: The distribution of feature variables. The plots show the distribution of the feature variables (ent0, ent24, delta and kl) for the different clustering methods. The number of clusters is estimated using the silhouette score. The red line shows the split of the variable that corresponded to the lowest p-value from the log-rank test. The exact value of the split is shown in table 5.11.

Feature	kmeans	hclust	mclust	camps
ent0	0.736	0.607	0.671	0.756
ent24	0.706	0.597	0.617	0.693
delta	0.029	0.034	0.077	0.055
kl	0.034	0.105	0.034	0.045

Table 5.11: Optimal thresholds The table shows the thresholds of the features that gave the lowest p-value for the logrank test for each clustering method.

The consistent low p-values for delta entropy across all clustering methods suggest that changes in entropy are highly predictive of patient survival.

The significance of entropy at 24 hours for multiple methods (especially hclust, p=0.0009) indicates that the state of the cellular landscape after 24 hours of treatment is also highly predictive of survival. The Kullback-Leibler divergence also shows some predictive capability, particularly with mclust and camps. No particular clustering methods stands out as superior to the others with

Kaplan Meier plots showing estimated survival curves when splitting by the optimal threshold:

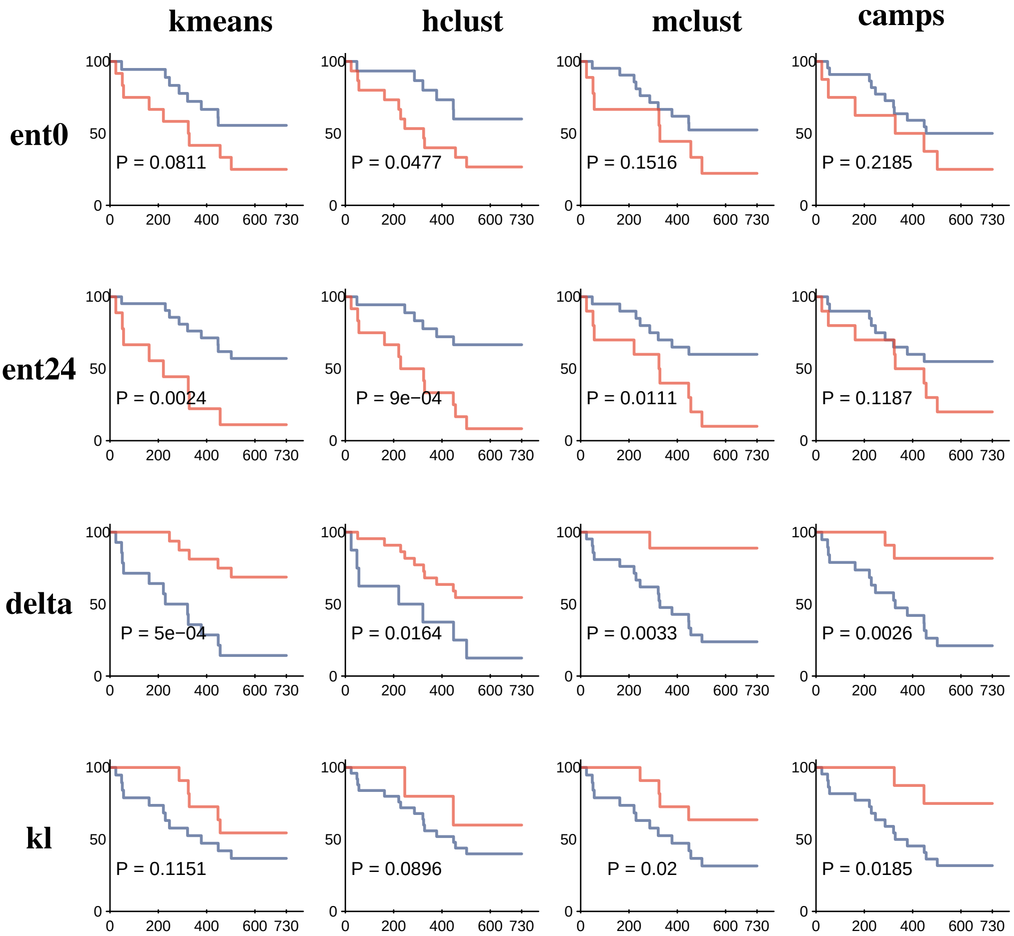


Figure 5.37: Kaplan-Meier plots for patients stratified by the optimal feature threshold. The provided Kaplan-Meier plots display the estimated survival probabilities for patients stratified by the entropy-based features (ent0, ent24, delta, and kl) and clustering methods (kmeans, hclust, mclust, and camps). The red and blue lines represent different patient groups based on the feature values that minimized their p-values in the log-rank test. Each plot includes the corresponding p-value.

In the Cox regression, the delta entropy yielded a low p-value for both K-Means (p=0.0029) and CAMPS (p=0.0005). We take a closer look at the Kaplan-Meier curves when stratifying the delta entropy for these methods in Figure 5.38 and 5.39. We note that K-Means had a more balanced split of the patients, with groups of 16 and 14. CAMPS had a slightly more unbalanced division, with groups of 11 and 19.

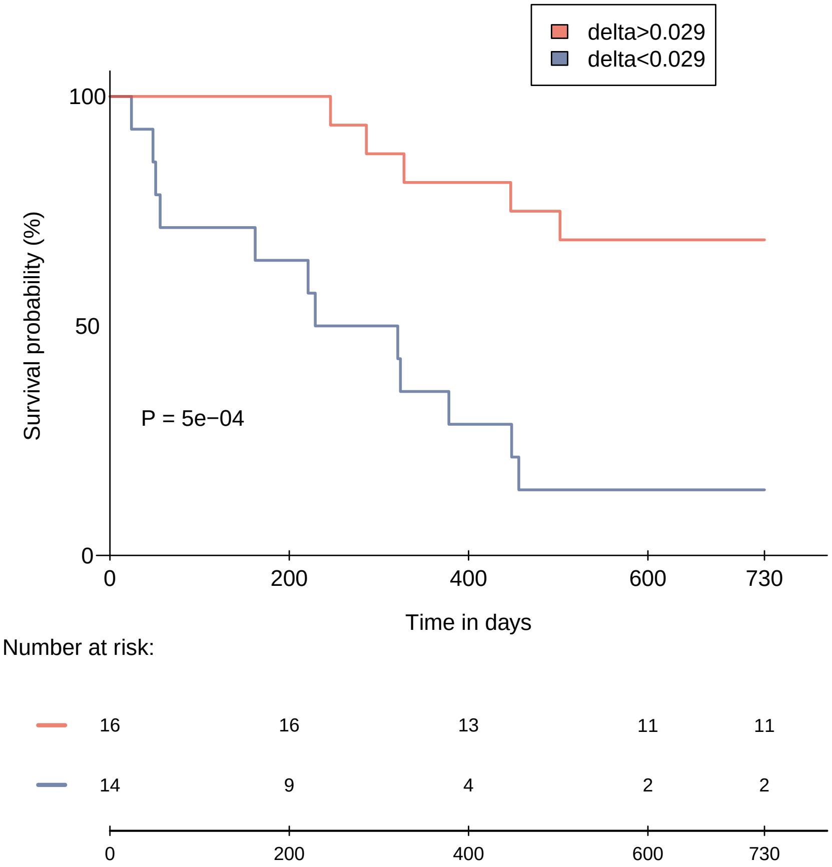


Figure 5.38: Kaplan-Meier curve for population split at optimal delta entropy derived from K-Means (K=10). The orange line represents the patients with delta entropy greater than 0.029, and the blue line represents those with delta entropy less than 0.029. The curves depict the percentage of patients surviving over time, shown on the y-axis. The x-axis shows the survival time in days. The separation between the curves indicates the difference in survival between the two groups. The p=0.0005 suggests the statistical significance of this difference. The number of patients at risk at each time point for the groups is shown below the plot.

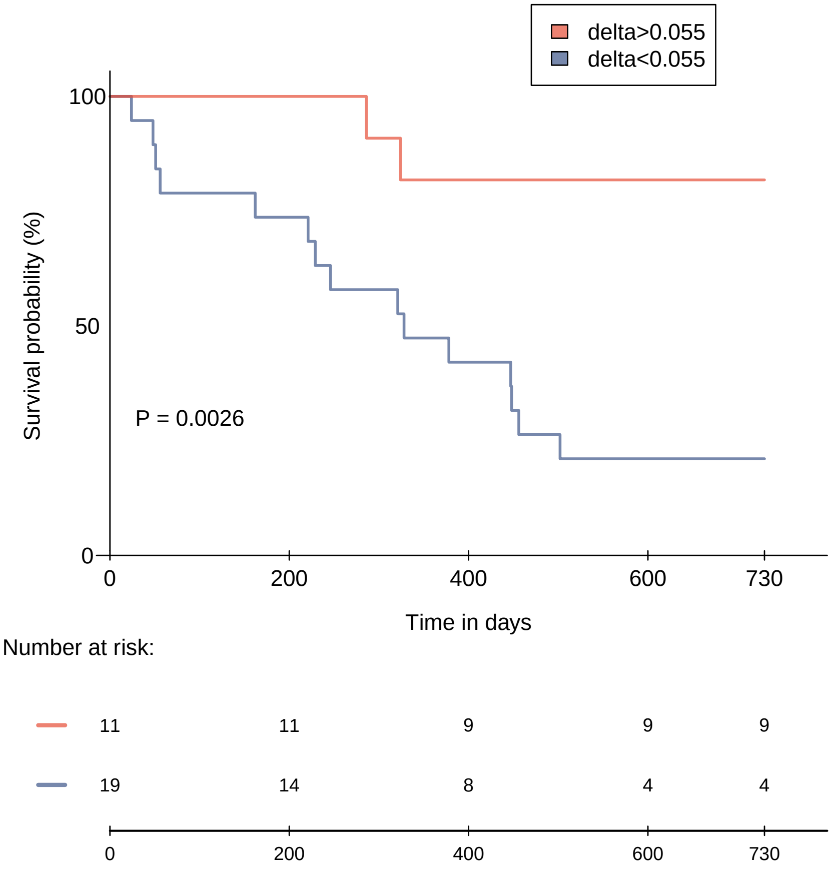


Figure 5.39: Kaplan-Meier curve for population split at optimal delta entropy derived from CAMPS (K=6). The orange line represents the patients with delta entropy greater than 0.055, and the blue line represents those with delta entropy less than 0.055. The curves depict the percentage of patients surviving over time, shown on the y-axis. The x-axis shows the survival time in days. The separation between the curves indicates the difference in survival between the two groups. The p=0.0026 suggests the statistical significance of this difference. The number of patients at risk at each time point for the groups is shown below the plot.

Chapter 6

Discussion and conclusion

Introduction

The goal of this thesis has been to find effective representations of CyTOF data, leveraging clustering algorithms to investigate cellular landscapes and predict patient outcomes. What we define as effect representations depends on what we try to achieve. To achieve this, we proposed and evaluated several clustering techniques, including the proposed CAMPS algorithm, across real and simulated datasets. Here, we reflect on our findings and discuss their implications, strengths, and potential weaknesses.

One of the primary challenges in analyzing CyTOF data is its high dimensionality and noise, which complicate the clustering of cell types based on protein expression profiles. Traditional clustering methods like K-Means and hierarchical clustering have inherent limitations when applied to such complex datasets. K-Means tends to favor spherical clusters and requires pre-specification of the number of clusters, while hierarchical clustering, though more flexible, can be computationally expensive and sensitive to noise and outliers.

Novel clustering algorithms

In this thesis, I have proposed two novel clustering methods. We first introduced Metaclust, which combines the strengths of K-Means and hierarchical clustering. By leveraging the scalability of K-Means, the flexibility and control offered by hierarchical clustering can be applied to large datasets. CAMPS also borrows strength from K-Means and allows the user to incorporate domain knowledge about the behavior of specific cell types to enhance the clustering results.

Datasets

We considered several datasets to evaluate the effectiveness of our clustering methodologies and their applications in CyTOF data analysis. The datasets span both real and simulated data, providing a comprehensive framework for testing our proposed algorithms.

A Gaussian mixture model was employed to simulate the CyTOF data. Several considerations were necessary, including maintaining the non-negative property of realistic data. This was handled by truncating values below zero. While simple, this approach can negatively impact downstream analysis. If a large portion of the protein expressions in the simulated cells are negative and then truncated to zero, an artificial compact cluster at exactly zero is created. This can mislead clustering algorithms into identifying a small, compact cluster that doesn’t truly exist. Moreover, the models are based on manually gated cell types. If this step is not performed accurately, the simulated data will reflect these shortcomings, embodying the ”garbage in, garbage out” concept.

We generated two datasets to assess the performance of our methods in different settings. The first simulated dataset contained equal proportions of cell types. Although this might not be realistic, it allowed for easier interpretation of the results. The second dataset reflected the realistic distribution of cell types. One of the main advantages of using simulated datasets for performance assessment is knowing the true labels of the data. In contrast, manually gated cell types in real data likely contain some incorrect annotations, introducing bias.

Manual gating

The manual gating strategy employed in this thesis was based on the methodology outlined in Russo et al. (2019). However, several of the protein markers used in that study were unavailable in the provided dataset, necessitating adaptations. For instance, initial steps involving the gating out of dead cells, which typically represent a very small percentage of the total cells, had to be skipped. Additionally, other adjustments were made, which could potentially compromise the integrity of the annotated cells.

Manual gating involves selecting boundaries through visual inspection, a task usually performed by experts. In this thesis, I performed the gating, which could further impact the integrity of the results. To mitigate this, rather conservative gating boundaries were chosen. The aim was not to identify the accurate cell types for as many cells as possible but to isolate distinct characteristics of specific cell types for simulation purposes. This conservative approach, while useful for defining clear cell type characteristics, may underestimate the true noise level in the data. Consequently, this could lead to the overestimation of the performance of clustering methods in later analyses.

Clustering single cell-data

Clustering was performed on the datasets, and the results were carefully inspected and compared. It quickly became evident that hierarchical clustering with single, average, and complete linkage had significant shortcomings. These methods tended to consolidate the majority of observations into a small number of clusters, leaving most clusters with only a handful of cells.

In response to these results, we developed an alternative approach for cutting the dendrogram while imposing a minimum size requirement for clusters. This modification had no effect on the single linkage approach but showed significant improvements for complete and average linkage methods. However, despite these improvements, they still did not perform as well as other methods. Overall, Ward linkage emerged as the most appropriate linkage method.

Furthermore, an inspection of the similarity of cluster assignments using the adjusted Rand index revealed that CAMPS and K-Means produced the most similar results. In contrast, single, complete, and average linkage methods were the outliers, demonstrating lower similarity in cluster assignments.

Quality of cell type identification

The effectiveness of the clustering algorithms in identifying individual cell types was carefully measured. When the number of clusters was fixed, there were small differences between the performance of K-Means, hierarchical clustering with Ward linkage, Metaclust, and CAMPS. Generally, CAMPS tended to perform slightly better, particularly in its ability to separate B cells and NK cells more effectively than the other methods. In real-world applications where the number of clusters needs to be estimated, CAMPS showed very promising results. The other methods often under-clustered the data by selecting a low K-value, leading to poor identification of cell types.

The datasets used for measuring cell type identification performance were simplified versions of realistic data, focusing only on major cell types identified by manual gating. This approach potentially overlooked rare populations. Including broader assessments with rare cell types could provide a more comprehensive understanding of the algorithms' performance and limitations.

The robustness of the clustering methods to noise and changes in the data was also evaluated by scaling the variance of protein expression in the cells and generating several datasets. Metaclust stood out for its robustness to changes in variance across datasets. CAMPS also demonstrated stability at lower levels of noise. In contrast, K-Means exhibited the most variability, likely due to the random initialization of cluster centroids. In this analysis, K-Means was only initialized once, but in general, it should be initialized multiple times to reduce the randomness inherent in the algorithm. A redeeming factor is the use of K-Means++ initialization, which reduces the randomness of the method, but these results indicate that this might not be sufficient to obtain stable results.

Quality as features for prediction

For prediction, we considered features derived from the algorithms' clustering results. Specifically, we examined the entropy of the patients' 0-hour and 24-hour samples. We also considered the absolute change in entropy and the Kullback-Leibler divergence to capture changes over time. Initially, we focused on the binary problem of predicting the response to treatment. The results were not particularly promising, likely due to the missing response variable for six patients, which significantly reduced the size of an already small dataset. Upon inspecting the distribution of the features, we found that several were right-skewed. Consequently, we performed a log transformation of these features. We observed some minor statistical significance post-transformation, particularly for the delta entropy derived from Metaclust and CAMPS. These findings suggest that log-transformed features might be more effective in capturing the underlying relationships in the data, thereby improving the accuracy of the logistic regression model in predicting treatment response.

We then considered the prediction of the survival of the patients. We found that the entropy at 0 hours was not an independent predictor of survival. The entropy at 24 hours showed some more promising results, with significance when derived from hierarchical clustering with Ward linkage. Delta entropy, however, showed consistently low p-values across all clustering methods, suggesting changes in entropy are highly predictive of patient survival, in particular when derived from CAMPS and Metaclust. If delta entropy is high, patient survival tends to increase. Assuming clusters correspond to cell types, this indicates that significant alterations in the cellular landscape suggest effective treatment. Such changes may reflect an increase in healthy blood cells (like neutrophils) or a reduction in malignant cells. As outlined in Tislevoll et al. (2023), cytometry data is a valuable tool for early response evaluation in AML patients, and the findings in this thesis underscore this potential. These results highlight the importance of dynamic changes in the cellular composition of blood in response to treatment. Monitoring these changes can provide critical insights into the effectiveness of therapeutic interventions and help tailor treatment strategies to individual patients, potentially improving outcomes in AML treatment.

Other features derived from the clustering results could also be used for prediction. For instance, the proportion of cells in each cluster could serve as separate predictive features. However, due to the small number of patients included in the study, it was decided not to pursue this approach. Utilizing such features in a small dataset might lead to overfitting and unreliable predictions.

Potential expansions of CAMPS

CAMPS, designed to incorporate domain-specific knowledge through feature subsets, demonstrated improved performance in differentiating major cell populations. By guiding the clustering process with predefined protein panels, CAMPS could effectively highlight relevant features for each cell type. This approach aligns with the concept of zero-shot classification, offering a practical way to enhance clustering accuracy by leveraging known marker behaviors. However, we have not been able to test the method's performance on a completely realistic dataset. We could only consider a subset of the full dataset containing cells from major cell populations found by manual gating. When faced with smaller populations and perhaps even unexpected cell types, the performance remains unknown. Furthermore, to better understand the relevance of CAMPS in the field, it should be compared to the current popular choice of clustering algorithm for CyTOF data; FlowSOM. Complications related to FlowSOM's requirement of a special file type often used for CyTOF data (but not available to me), meant that FlowSOM had to be excluded from the analysis of this thesis.

The current implementation of CAMPS only accommodates ternary weights for markers (1, 0, or -1), which might oversimplify the importance of certain proteins. Future work could explore extending this to include continuous weights, allowing for a more nuanced representation of marker relevance. The CAMPS algorithm is highly inspired by manual gating, where subpopulations of cells are gated out sequentially. CAMPS could, in theory, be extended to this sequential approach as well, further enhancing the method's capabilities.

Conclusion

In conclusion, we have gained insight into how clustering can be used to derive representations of CyTOF data that are useful for the determination of cell type composition. In particular, we have shown that the proposed method CAMPS is capable of outperforming several traditional clustering methods for this purpose. We have also explored how clustering combined with entropy-based features can be used to find representations of the data that are useful for the prediction of treatment outcome and survival.

Bibliography

Aalen, O., Borgan, O., & Gjessing, H. (2008). *Survival and event history analysis: a process point of view*. Springer Science & Business Media.

Appelbaum, F. R., Gundacker, H., Head, D. R., Slovak, M. L., Willman, C. L., Godwin, J. E., Anderson, J. E., & Petersdorf, S. H. (2006). Age and acute myeloid leukemia. *Blood*, *107*, 3481–3485.

Baker, H. (2021). How many atoms are in the observable universe? *https://www.livescience.com/how-many-atoms-in-universe.html*, .

Bendall, S. C., Davis, K. L., Amir, E.-a. D., Tadmor, M. D., Simonds, E. F., Chen, T. J., Shenfeld, D. K., Nolan, G. P., & Pe'er, D. (2014). Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development. *Cell*, *157*, 714–725.

Bendall, S. C., Simonds, E. F., Qiu, P., Amir, E.-a. D., Krutzik, P. O., Finck, R., Bruggner, R. V., Melamed, R., Trejo, A., Ornatsky, O. I. et al. (2011). Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*, *332*, 687–696.

Brémaud, P. (2012). *An introduction to probabilistic modeling*. Springer Science & Business Media.

Breslow, N. E. (1972). Discussion of the paper by d. r. cox. *Journal of the Royal Statistical Society: Series B (Methodological)*, *34*, 187–220.

Commons, W. (2024). File:diagram showing the cells in which aml starts cruk 297.svg — wikimedia commons, the free media repository. URL: https://commons.wikimedia.org/w/index.php?title=File:Diagram_showing_the_cells_in_which_AML_starts_CRUK_297.svg&oldid=845250657 [Online; accessed 22-April-2024].

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, *34*, 187–202.

Cox, D. R. (2018). *Analysis of survival data*. Chapman and Hall/CRC.

De Kouchkovsky, I., & Abdul-Hay, M. (2016). Acute myeloid leukemia: a comprehensive review and 2016 update. *Blood cancer journal*, *6*, e441–e441.

Delves, P. J., Martin, S. J., Burton, D. R., & Roitt, I. M. (2017). *Roitt's essential immunology*. John Wiley & Sons.

Efron, B. (1977). The efficiency of cox's likelihood function for censored data. *Journal of the American statistical Association*, *72*, 557–565.

Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, *78*, 553–569.

Hammill, D. (2021). *CytoExploreR: Interactive Analysis of Cytometry Data*. URL: https://github.com/DillonHammill/CytoExploreR r package version 1.1.0.

Hansen, B.-A., Wendelbo, Ø., Bruserud, Ø., Hemsing, A. L., Mosevoll, K. A., & Reikvam, H. (2020). Febrile neutropenia in acute leukemia. epidemiology, etiology, pathophysiology and treatment. *Mediterranean journal of hematology and infectious diseases*, *12*.

Hine, R. (2015). *A dictionary of biology*. OUP Oxford.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of classification*, *2*, 193–218.

James, G., Witten, D., Hastie, T., Tibshirani, R. et al. (2013). *An introduction to statistical learning* volume 112. Springer.

Jh Jr, W. (1963). Hierarchical grouping to optimize an objective function. *J. Amsr. Statis. Assoc.*, *58*, 236–244.

Kalina, T., Fišer, K., Pérez-Andrés, M., Kuzílková, D., Cuenca, M., Bartol, S. J., Blanco, E., Engel, P., & van Zelm, M. C. (2019). Cd maps—dynamic profiling of cd1–cd100 surface expression on human leukocyte and lymphocyte subsets. *Frontiers in Immunology*, *10*, 2434.

Laird, N., Lange, N., & Stram, D. (1987). Maximum likelihood computations with repeated measures: application of the em algorithm. *Journal of the American Statistical Association*, *82*, 97–105.

Lin, D. (2007). On the breslow estimator. *Lifetime data analysis*, *13*, 471–480.

Lowenberg, B., Downing, J. R., & Burnett, A. (1999). Acute myeloid leukemia. *New England Journal of Medicine*, *341*, 1051–1062.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, .

Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Nowicka, M., Krieg, C., Crowell, H. L., Weber, L. M., Hartmann, F. J., Guglietta, S., Becher, B., Levesque, M. P., & Robinson, M. D. (2017). Cytof workflow: differential discovery in high-throughput high-dimensional cytometry datasets. *F1000Research*, *6*.

Peterson, A. D., Ghosh, A. P., & Maitra, R. (2018). Merging k-means with hierarchical clustering for identifying general-shaped groups. *Stat*, *7*, e172.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*, 846–850.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Russo, M. A., Fiore, N. T., Van Vreden, C., Bailey, D., Santarelli, D. M., McGuire, H. M., Fazekas de St Groth, B., & Austin, P. J. (2019). Expansion and activation of distinct central memory t lymphocyte subsets in complex regional pain syndrome. *Journal of neuroinflammation*, *16*, 1–17.

Sender, R., Weiss, Y., Navon, Y., Milo, I., Azulay, N., Keren, L., Fuchs, S., Ben-Zvi, D., Noor, E., & Milo, R. (2023). The total mass, number, and distribution of immune cells in the human body. *Proceedings of the National Academy of Sciences*, *120*, e2308511120.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, *27*, 379–423.

Tislevoll, B. S., Hellesøy, M., Fagerholt, O. H. E., Gullaksen, S.-E., Srivastava, A., Birkeland, E., Kleftogiannis, D., Ayuda-Durán, P., Piechaczyk, L., Tadele, D. S. et al. (2023). Early response evaluation by single cell signaling profiling in acute myeloid leukemia. *Nature communications*, *14*, 115.