CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

A COMPARISON OF SCRNA-SEQ ANALYSIS WORKFLOWS FOR DIFFERENTIALLY EXPRESSED GENES IDENTIFICATION IN BREAST CANCER

Mateus do Amor Devino Pereira

Dissertation submitted to the Graduate Program in Computer Science of the Federal Center for Technological Education of Rio de Janeiro, CEFET/RJ, as partial fulfillment of the requirements for the degree of master.

Advisors:
Eduardo Bezerra da Silva, D.Sc.
Kele Teixeira Belloze, D.Sc.

Rio de Janeiro,

January 23, 2025

A Comparison of scRNA-seq Analysis Workflows for Differentially Expressed Genes Identification in Breast Cancer

Master's Dissertation in Computer Science, Federal Center for Technological Education of Rio de Janeiro, CEFET/RJ.

Mateus do Amor Devino Pereira

Approved by:

Prof. Eduardo Bezerra, D.Sc. (CEFET/RJ)

Prof. Kele Teixeira Belloze, D.Sc. (CEFET/RJ)

Prof. Pedro Henrique González Silva, D.Sc. (CEFET/RJ)

Prof. Fabrício Alves Barbosa da Silva, D.Sc. (Fiocruz)

Rio de Janeiro,

January 23, 2025

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

P436 Pereira, Mateus do Amor Devino
A comparison of scrna-seq analysis workflows for differentially expressed genes identification in breast cancer / Mateus do Amor Devino Pereira. — 2024.
71f. : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2024.
Bibliografia : f. 67-71
Orientador: Eduardo Bezerra da Silva
Coorientadora: Kele Teixeira Belloze

1. Análise numérica - Processamento de dados. 2. Aprendizado de máquina. 3. Mamas - Câncer. I. Silva, Eduardo Bezerra da. (Orient.). II. Belloze, Kele Teixeira.(Coord.). III. Título.

CDD 004.0151

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

ABSTRACT

A Comparison of scRNA-seq Analysis Workflows
for Differentially Expressed Genes Identification
in Breast Cancer

Mateus do Amor Devino Pereira

Advisors:
Eduardo Bezerra da Silva, D.Sc.
Kele Teixeira Belloze, D.Sc.

Abstract of dissertation submitted to the Graduate Program in Computer Science of the Federal Center for Technological Education of Rio de Janeiro, CEFET/RJ, as partial fulfillment of the requirements for the degree of master.

In recent years, *-omics* research has played a major role in understanding cellular biology. In particular, single cell RNA-sequencing data analysis has been successful in drug discovery and biomarker identification of complex diseases, such as cancer. However, when it comes to analyzing scRNA-seq data, several steps in the workflow can be done in different ways. This study provides a comparison of different combinations of techniques in a scRNA-seq workflow, evidencing how each of these steps impact the results. We perform experiments on three datasets of transcriptome data, with each dataset having different magnitudes of number of samples. Our experiments consist of applying a set of preprocessing and clustering techniques to breast cancer scRNA-seq data. We also perform an ablation study to understand how each technique influenced experiment outcomes.

Keywords:
scRNA-seq; Machine Learning; Breast Cancer

Rio de Janeiro,

January 23, 2025

Contents

I Introduction **14**

I.1 Contextualization 14

I.2 Motivation 15

I.3 Objectives 15

I.4 Methodology 15

I.5 Organization 16

II Theoretical Foundation **17**

II.1 Cancer 17

II.1.1 The Eukaryotic Cell Cycle 17

II.1.2 Oncogenes, Tumor Suppressor Genes and The Hallmarks of Cancer 18

II.2 RNA-Sequencing 19

II.3 Gene Expression Matrices 19

II.4 Quality Control 20

II.5 Data Normalization 21

II.5.1 Maximum Absolute Scaling 21

II.6 Dimensionality Reduction 21

II.6.1 Principal Component Analysis 22

II.6.2 Unified Manifold Approximation and Projection 22

II.6.3 T-Distributed Stochastic Embedding 23

II.7 Clustering 24

II.7.1 K-Means 25

II.7.2 Spectral Clustering 26

II.7.3 Louvain 27

II.7.4 DBSCAN 27

II.8 Clustering Metrics 28

II.8.1 Silhouette Coefficient 28

II.8.2 Cluster Homogeneity 28

II.8.3 Rand Index 29

II.9 Bootstrapping 29
II.10 Discussion 30

III Related Work 31
III.1 Tool Development 31
III.2 Imputation 32
III.3 Cell Classification 33
III.4 Gene Regulatory Networks 34
III.5 Clustering 34
III.6 Discussion 35

IV Methodology 36
IV.1 Problem Statement 36
IV.2 Proposed Methodology 36
IV.2.1 Quality Control 37
IV.2.2 Normalization 37
IV.2.3 Dimensionality Reduction 38
IV.2.4 Cell Clustering 38
IV.3 Experiment Design 39

V Experiments 41
V.1 Datasets 41
V.2 Exploratory Analysis 41
V.2.1 GSE75688 42
V.2.2 GSE161529 43
V.3 Pipeline Templates 45
V.4 Experiments 46
V.4.1 Homogeneity Score 47
V.4.2 Rand Index 48
V.4.3 Wall Time 49
V.4.4 Process Time 50
V.5 Differential Expression Visualizations 51
V.6 Ablation Study 51
V.6.1 P1 Ablation 52
V.6.2 P2 Ablation 53
V.6.3 P3 Ablation 53

V.6.4 P4 Ablation 53
V.6.5 P5 Ablation 54
V.6.6 P6 Ablation 54
V.6.7 P7 Ablation 54
V.6.8 P8 Ablation 55
V.6.9 P9, P10, P11 and P12 Ablation 55
V.6.10 Ablation Summary 55

VI Conclusion 64
VI.1 Retrospective Analysis 64
VI.2 Contributions, Limitations and Future Work 65

References 66

A Wall and Processing Times 72

B Clustering Results Visual Representations 74

C Expression of Known Genes 83

List of Figures

II.1 The cell cycle. 18

II.2 Visual comparison of maximum absolute scaling. 22

IV.1 Single Cell RNA-sequencing analysis pipeline high-level steps. 37

V.1 Gene expression levels distribution on GSE75688. 43

V.2 Gene expression levels distribution on GSE161529. 44

V.3 Pipeline structure and techniques. 45

V.4 Homogeneity score results. 57

V.5 Rand index results. 58

V.6 Wall time. 59

V.7 Process time. 60

V.8 Expression of genes related to breast cancer on Pipeline P1, small dataset. 61

V.9 Expression of genes related to breast cancer on Pipeline P1, medium dataset. 61

V.10 Expression of genes related to breast cancer on Pipeline P1, large dataset. 62

B.1 Toy dataset UMAP true labels. 74

B.2 Toy dataset UMAP for pipelines P1, P2, P3 and P4. 75

B.3 Toy dataset UMAP for pipelines P5, P6, P7 and P8. 76

B.4 Toy dataset UMAP for pipelines P9, P10, P11 and P12. 77

B.5 Medium dataset UMAP true labels. 78

B.6 Medium dataset UMAP for pipelines P1, P2, P3 and P4. 79

B.7 Medium dataset UMAP for pipelines P5, P6, P7 and P8. 80

B.8 Large dataset UMAP true labels. 81

B.9 Large dataset UMAP for pipelines P1, P2, P4, P5, P6 and P8. 82

C.1 Expression of genes related to breast cancer on Pipeline P2, small dataset. 83

C.2 Expression of genes related to breast cancer on Pipeline P4, small dataset. 83

C.3 Expression of genes related to breast cancer on Pipeline P5, small dataset. 84

C.4 Expression of genes related to breast cancer on Pipeline P6, small dataset. 84

C.5 Expression of genes related to breast cancer on Pipeline P8, small dataset. 85

C.6 Expression of genes related to breast cancer on Pipeline P9, small dataset. 85
C.7 Expression of genes related to breast cancer on Pipeline P10, small dataset. 86
C.8 Expression of genes related to breast cancer on Pipeline P12, small dataset. 86
C.9 Expression of genes related to breast cancer on Pipeline P2, medium dataset. 86
C.10 Expression of genes related to breast cancer on Pipeline P4, medium dataset. 87
C.11 Expression of genes related to breast cancer on Pipeline P5, medium dataset. 87
C.12 Expression of genes related to breast cancer on Pipeline P6, medium dataset. 88
C.13 Expression of genes related to breast cancer on Pipeline P8, medium dataset. 88
C.14 Expression of genes related to breast cancer on Pipeline P2, large dataset. 89
C.15 Expression of genes related to breast cancer on Pipeline P4, large dataset. 89
C.16 Expression of genes related to breast cancer on Pipeline P5, large dataset. 90
C.17 Expression of genes related to breast cancer on Pipeline P6, large dataset. 90
C.18 Expression of genes related to breast cancer on Pipeline P8, large dataset. 91

List of Tables

V.1 Accession GSE75688 cells distribution by cancer type. 42

V.2 Accession GSE75688 cells distribution by cell type. 42

V.3 Experiments executed. 47

V.4 Homogeneity Scores for Different Pipelines Across Datasets 48

V.5 Rand Index Values for Different Pipelines Across Datasets 49

V.6 Wall times for different pipelines on datasets of varying sizes. Values for the small dataset are in seconds, while values for medium and large dataset are in hours. 49

V.7 Process times for different pipelines on datasets of varying sizes. Values for the small dataset are in seconds, while values for medium and large dataset are in hours. 50

V.8 Ablation study results. 63

A.1 Ablation study wall and processing times 73

List of Abbreviations

AI	Artificial Intelligence	15
ANN	Artificial Neural Network	33, 34
ARI	Adjusted Rand Index	34
AUROC	Area Under The Receiver Operating Characteristic Curve	33
CNN	Convolutional Neural Network	33
DAWN	Denoising Autoencoder With Neuronal Approximator	35
DBSCAN	Density-Based Spatial Clustering Of Applications With Noise	27, 32, 48, 55
DEG	Differentially Expressed Genes	34
DNA	Deoxyribonucleic Acid	18
DT	Decision Tree	33
EDA	Exploratory Data Analysis	41
ELM	Extreme Learning Machine	33
EM	Expectation-maximization Clustering	34
FPKM	Fragments Per Kilobase Million	37
GBM	Gioblastoma Multiforme	34
GEO	Gene Expression Omnibus	41, 44, 64
GNB	Gaussian Naive Bayes Classifier	33
GRN	Gene Regulatory Network	31, 34, 35
GS	Mitogenic Growth Signals	19
ICA	Independent Component Analysis	35
KNN	K Nearest Neighbors	33, 65
ML	Machine Learning	15, 21, 22, 24, 31, 32, 33, 35, 37, 41
NMI	Normalized Mutual Information	34
PCA	Principal Component Analysis	22, 35, 38, 39, 48, 49, 50, 53, 54, 55, 56
RF	Random Forest	33, 34
RNA-SEQ	RNA-sequencing	15, 17, 19, 31, 33, 42
RPKM	Reads Per Kilobase Million	37
SCRNA-SEQ	Single Cell RNA-sequencing	15, 17, 19, 20, 22, 24, 27, 30, 31, 32, 34, 35, 36, 37, 38, 39, 43, 44, 55, 56, 64, 66

SIMLR	Single-cell Interpretation Via Multi-kernel LeaRning	35
SNE	Stochastic Neighbor Embedding	23, 24
SVM	Support Vector Machine	33, 34
TME	Tumor Microenvironment	31
T-SNE	T-Distributed Stochastic Neighbor Embedding	23, 24, 32, 35, 38, 46, 47, 48, 49, 50, 51, 55, 56
TAR	Tape ARchive	44
TCGA	The Cancer Genome Atlas	64
TPM	Transcripts Per Million	37
TSG	Tumor Supressor Genes	18, 30
TSV	Tab-separated Values	42, 43
UMAP	Uniform Manifold Approximation And Projection	22, 23, 38, 39, 46, 48, 49, 50, 54, 55, 56

Chapter I Introduction

I.1 Contextualization

Cancer is a set of genetic diseases characterized by an uncontrollable growth of cells. Normal cells die when replication errors trespass a certain threshold. Cancer cells seem to avoid control mechanisms and continue to replicate uncontrollably, even when replication errors accumulate [Hanahan, 2022]. Cancer is the leading cause of death worldwide, with about ten million deaths in 2020 [Sung et al., 2021]. Any part of the body can develop cancer. Breast cancer was the fifth type that most killed in 2020, with nearly 685,000 deaths [Sung et al., 2021]. It was also the type of cancer with the highest number of new cases registered in the same year.

Due to the heterogeneity of cancer, there is no one-size-fits-all treatment. Even the same type of cancer may have several subtypes that differ considerably. On top of that, each person is unique, and individual genetic variation may influence the outcome of treatments. Personalized approaches that take into account individual biological differences as well as the characteristics of the cancer being treated have been more successful than generic treatments [Ding et al., 2020].

Multidisciplinary approaches have contributed significantly to the advance of science. In particular, advances in Bioinformatics, a combination of Biology and Computer Science, have provided enormous amounts of data that can be used as input for models to better understand organisms, diseases, among others. Initiatives such as the Human Genome Project (HGP)[1], which had the goal of mapping all human genes (the genome), provide vast amounts of genetics data. This data has enabled scientists to understand the human body at a level previously never imagined.

The central dogma of molecular biology explains that DNA is transcribed into RNA, which is translated into proteins [Li and Xie, 2011]. Proteins are responsible for the functions executed by cells. Different types of data are provided in each of these stages. The analysis of the information at the DNA level is named genomics. At the RNA level, it is called transcriptomics. At the protein level, proteomics. This research focuses on analyzing data at the transcriptomic level.

[1] https://www.genome.gov/human-genome-project

I.2 Motivation

With the advances in genetic sequencing, it is possible to understand the course of diseases like never before. RNA-sequencing (RNA-seq) [Conesa et al., 2016] techniques have been used over the past decades, resulting in humongous amounts of data about the human body and many genetic diseases.

The traditional method of RNA-seq analyzes bulks of cells, resulting in each bulk's gene expression being represented by the mean of expression levels of each cell in bulk. This type of analysis is useful in many cases. However, due to cancer's heterogeneity, means can mask important information about expression levels of individual cells [Li and Wang, 2021]. As computational resources become more accessible, Single Cell RNA-sequencing (scRNA-seq) becomes possible.

In contrast with the traditional RNA-seq, scRNA-seq captures gene expression levels for each cell individually. While genetic sequencing is not a perfect process, the technology has become more accessible over the years, resulting in growing amounts of data being generated over time. As more data is generated, Machine Learning (ML) techniques have been used to advance the current understanding of cancer [Asada et al., 2021].

ML is a subarea of Artificial Intelligence (AI) in which computers are not explicitly programmed to perform a task. Instead, a model learns to perform such a task by looking at examples. The more quality data is available, the better results yielded by ML models tend to be. As more scRNA-seq data is generated, ML models can be trained to better understand cancer biomarkers [Asada et al., 2021].

I.3 Objectives

The main objective of this research is to compare different scRNA-seq data analysis pipelines. By the end of this study, it is expected to evidence how each step of a pipeline influences clustering results and, consequently, differential expression analysis. The comparison between pipelines should evidence which techniques perform better on the task of analyzing scRNA-seq data to identify potential therapeutic targets for breast cancer treatment. As specific goals, we have the following:

1. Analyze how different techniques influence the final results.
2. Analyze how each technique in a pipeline influences the final results.

I.4 Methodology

To achieve the expected results, we analyze pipelines on three datasets. One of the datasets consists of around 500 cells [Chung et al., 2017], while the other two consist of around 30,000 and

50,000 cells [Pal et al., 2021]. The pipelines we experiment with consist of four steps - (i) quality control, (ii) normalization, (iii) dimensionality reduction and (iv) cell clustering.

Once clustering results are available, we perform a differential expression analysis to highlight the differentially expressed genes in each cluster. For each cluster, we also compare the expression level of 20 genes known as biomarkers of breast cancer [Grunda et al., 2012]. An ablation study is performed in each pipeline to evaluate how each step influences the final result.

I.5 Organization

This dissertation is divided into six chapters, including this Introduction. Chapter II presents a theoretical foundation, containing all the background knowledge necessary to understand this work. Chapter III lists and compares related studies. Chapter IV describes, in detail, the methodology applied in this study. Chapter V provides results of the experiments executed. Chapter VI concludes the dissertation, providing a retrospective analysis and possible future work based on this research.

Chapter II Theoretical Foundation

This chapter describes the background necessary to understand the methodology applied in this research. It contains a Biology and Computer Science background and is divided into ten sections. Section II.1 describes cancer and its hallmarks. Section II.2 describes the RNA-seq process and how sequencing data is generated. Section II.3 describes how large, and sparse matrices can be represented computationally in an efficient way. Section II.4 describes the importance of quality control steps in scRNA-seq experiments. Section II.5 describes the data normalization process and its effects on the analyzed data. Section II.6 describes the dimensionality reduction process. Section II.7 describes the clustering task. Section II.8 presents different types of metrics to evaluate clustering results. Section II.9 presents the bootstrapping method. Section II.10 concludes this chapter, summarizing the concepts explained and how they relate to this research.

II.1 Cancer

Cancer is a set of complex diseases and, in order to understand it, a few cell biology concepts are necessary. This section describes the concepts relevant to our research, and is divided in two subsections. Section II.1.1 describes the cell cycle, which is necessary to understand how normal and cancerous cells differ from one another. Section II.1.2 presents the concepts of oncogenes and Tumor Suppressor Genes, as well as briefly describing the hallmarks of cancer.

II.1.1 The Eukaryotic Cell Cycle

Cells have different phases throughout their lifecycle. The cell cycle contains two main phases - interphase and mitosis. Interphase is focused on cell growth and is the phase in which cells spend most of their time. Mitosis is a phase in which cell division occurs. That is, the cell divides itself, generating two new cells. Figure II.1 illustrate the cell cycle and its phases.

The interphase can be divided into three states - G1, S and G2. The G1 state is focused on cell growth. Most cells spend most of their time in this state, in which organelles and proteins are produced. From the G1 state, the cell can move either to G0 or S phases. G0 is a state in which cell division does not occur anymore. When a cell goes into this state, it tends not to get out of it. In contrast, if a cell moves from G1 to S the cell cycle continues.

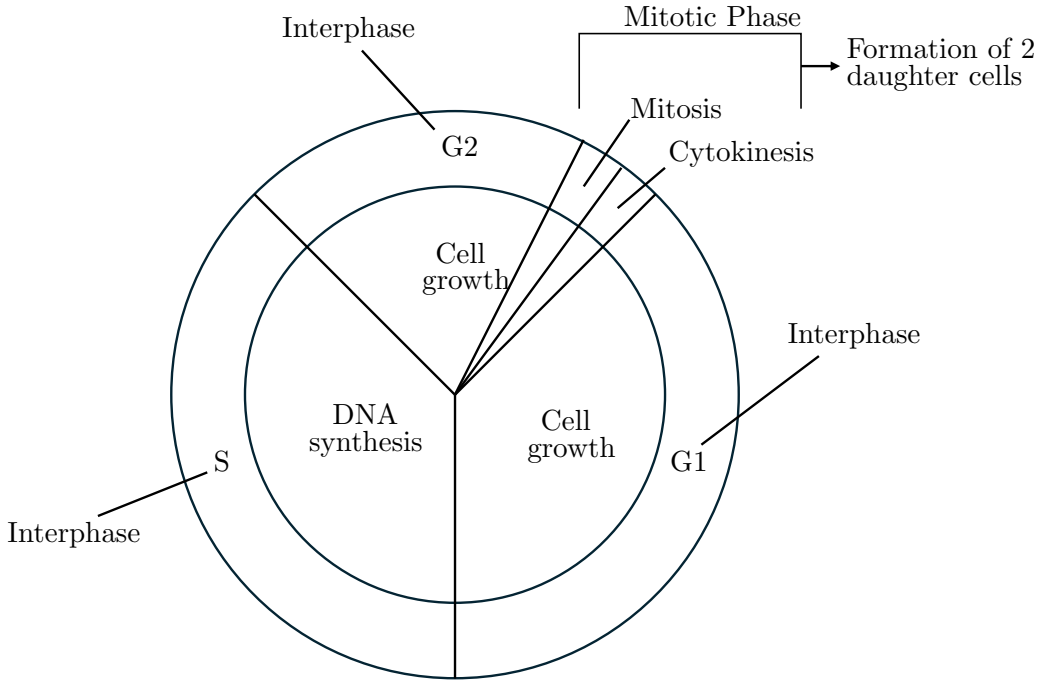


Figure II.1: The cell cycle. The cycle starts on the interphase, which is focused on cell growth, and is divided in G1, S, and S2. The second phase of the cycle is the mitotic phase, in which two daughter cells are created.

S stands for synthesis, Deoxyribonucleic Acid (DNA) synthesis. The human organism has 23 chromosomes. Most cells are diploids, except gametes, which are haploids. Diploid cells have a pair of each chromosome, while haploids only have one copy. During the S state, cells duplicate their genetic material inside the nucleus, as well as some organelles necessary for cell division.

Cells in the S state often move into the G2, which another growth state. This state revolves around preparing for the next phase, which is M. M is an abbreviation of mitosis, which is the phase in which the cell divides itself into two. After the division, the two new cells go to the G1 phase, and the cycle starts again.

There are several checkpoints throughout the cell cycle. The goal of these checkpoints is to prevent the replication of cells whose DNA has too many errors. This is done by ensuring the cell fulfills a set of requirements before proceeding through the phases of the cycle.

II.1.2 Oncogenes, Tumor Suppressor Genes and The Hallmarks of Cancer

Oncogenes and Tumor Supressor Genes (TSG) are two important concepts for understanding cancer. Oncogenes are genes whose high expression levels favor the occurrence of cancer. In contrast, TSGs favor cancer occurrence when their expression levels are low. These two types of genes are relevant to understanding the hallmarks of cancer.

There are over 200 types of cancer, which vary greatly from one another, but also share a specific set of characteristics. These shared characteristics are named the hallmarks of cancer

[Hanahan and Weinberg, 2000]. The concept was initially proposed by describing six characteristics (hallmarks) shared among cancer types. The six original hallmarks of cancer are self-sufficiency in growth signals, insensitivity to growth-inhibitory signals, evasion of programmed cell death, limitless replication potential, sustained angiogenesis, and tissue invasion and metastasis.

Normal cells need to receive growth signals to proliferate, named mitogenic growth signals (GS). Regarding normal cells, GSes of one type of cell are used to signal other types. In contrast, tumor cells seem to produce GS capable of stimulating their own proliferation. They also stimulate their neighbor cells to release GSes that contribute to their continuous growth [Hanahan and Weinberg, 2000]. These mechanisms give a name to the first hallmark - self-sufficiency in growth signals - and may explain why tumor cells grow uncontrollably.

With advances in the field, two emerging hallmarks were proposed as well as two enabling characteristics, summing up to ten hallmarks [Hanahan and Weinberg, 2011]. More recently, two emerging hallmarks and two enabling characteristics were reported [Hanahan, 2022].

II.2 RNA-Sequencing

RNA-seq is the process of analyzing a biospecimen, capturing its transcriptome expression levels [Ding et al., 2020]. The traditional form RNA-seq, also known as bulk RNA-seq, reads groups of cells. The expression level for a group is the average of the expression levels for cells in that group. This approach is useful in many scenarios, but may mask important information from highly heterogeneous groups, such as tumors.

With advancements in Biology and computational resources, a new way of RNA-seq has been developed - scRNA-seq. scRNA-seq allows reading gene expression at the cell level [Li and Wang, 2021], allowing better understanding and analysis of tumor heterogeneity.

Preprocessing steps are needed to prepare the gene expression matrix for further analysis. These steps may involve quality control, normalization, and dimensionality reduction [Andrews et al., 2021]. Once the preprocessing steps have been performed, clustering and differential expression analysis can be performed to identify potential therapeutic targets [Ding et al., 2020]. These steps will be further explained in the following sections.

II.3 Gene Expression Matrices

A gene expression matrix is a data structure that represents the expression levels of genes across a set of cells in a biological sample. In this matrix, rows correspond to individual genes, and columns correspond to individual cells. However, as most machine learning libraries expect samples (cells) to be represented in rows and features (genes) to be represented in columns, we will work with a transposed matrix. Each entry in the matrix contains a value that indicates the

expression level of a particular gene in a specific cell, typically measured as the number of detected transcripts (or counts) for that gene.

Gene expression matrices resulting from scRNA-seq experiments are large and sparse. These matrices usually consist of hundreds of thousands of lines (cells) and tens of thousands of columns (genes). These matrices cannot be fully loaded in memory, depending on the computational resources available. As many values in these matrices are zero, these matrices can be represented more efficiently by a list of tuples (i, j, v), where i denotes the row index, j the column index, and v the value of the cell M_{ij}. All cells that do not have a coordinate are assumed to have value zero.

Expression data in scRNA-seq are often non-normally distributed and heavily skewed, with many zeros and a few genes showing high expression levels. This non-normality is commonly addressed through normalization and transformation (e.g., log transformation) for better downstream analyses.

II.4 Quality Control

Due to the nature of scRNA-seq experiments, quality control measures must be applied in order to clean the data for further analysis [Andrews et al., 2021]. Cells and genes are filtered based on different criteria. The criteria used may vary depending on the type of analysis. Some of these criteria are summarized below.

Cell-Level Filtering Criteria

- Total counts per cell: Cells with unusually low or high total RNA counts may be removed, as low counts often indicate low RNA quality, and high counts can indicate doublets or multiplets.
- Number of genes detected per cell: Cells with very few or excessively high numbers of expressed genes might be excluded, with low gene counts suggesting poor quality or dying cells and high counts suggesting doublets.
- Mitochondrial gene expression percentage: Cells with a high percentage of reads from mitochondrial genes might be stressed or damaged and are often filtered out to focus the analysis on healthy cells.
- Ribosomal RNA content: High ribosomal RNA content might indicate technical biases, particularly in certain protocols, and cells with excessive ribosomal RNA counts might be excluded.

Gene-Level Filtering Criteria

- Minimum expression threshold: Genes expressed in only a small fraction of cells are often excluded, as they may be considered noise or lack sufficient statistical power.
- Variance threshold: For some analyses, only genes with sufficient variability across cells are retained, as highly invariant genes may not contribute meaningful information to clustering or differential expression.

II.5 Data Normalization

Data normalization is a preprocessing task needed to prepare raw data to be ingested by a ML model. This task is important because using features of different magnitudes may negatively affect some ML models.

II.5.1 Maximum Absolute Scaling

Maximum absolute scaling is a technique to scale features to the same magnitude. The operation performed by this normalization technique is illustrated in Equation II.1.

$$y = \frac{x}{max(x)} \tag{II.1}$$

In Equation II.1, x denotes the gene expression level for a given cell, $max(x)$ denotes the highest expression level for the same gene in all cells, and y denotes the scaled value, which is a value approximately between -1 and 1. As gene expression levels cannot be negative, our case study's values are always non-negative. As a result of the maximum absolute scaling, each gene in the expression matrix has its expression levels at the same magnitude.

Figure II.2 shows a visual representation of what happens to the data when maximum absolute scaling is applied. In this example, horizontal values vary between 0 and 1, while vertical values vary between 0 and 10 (on the left). Once maximum absolute scaling is applied, the relative distance between data points is preserved, but all features are scaled to the same magnitude. The example is applied to two dimensions for illustration purposes, but this normalization technique can be applied to any number of dimensions.

II.6 Dimensionality Reduction

Dimensionality reduction is a ML task to reduce the number of features of each sample. Using fewer features can have many advantages, such as algorithms running faster and avoiding the curse of dimensionality [Han et al., 2022]. However, dimensionality reduction must be used with caution, as it may hide important information present in the data. This section describes the dimensionality reduction techniques used in our experiments.

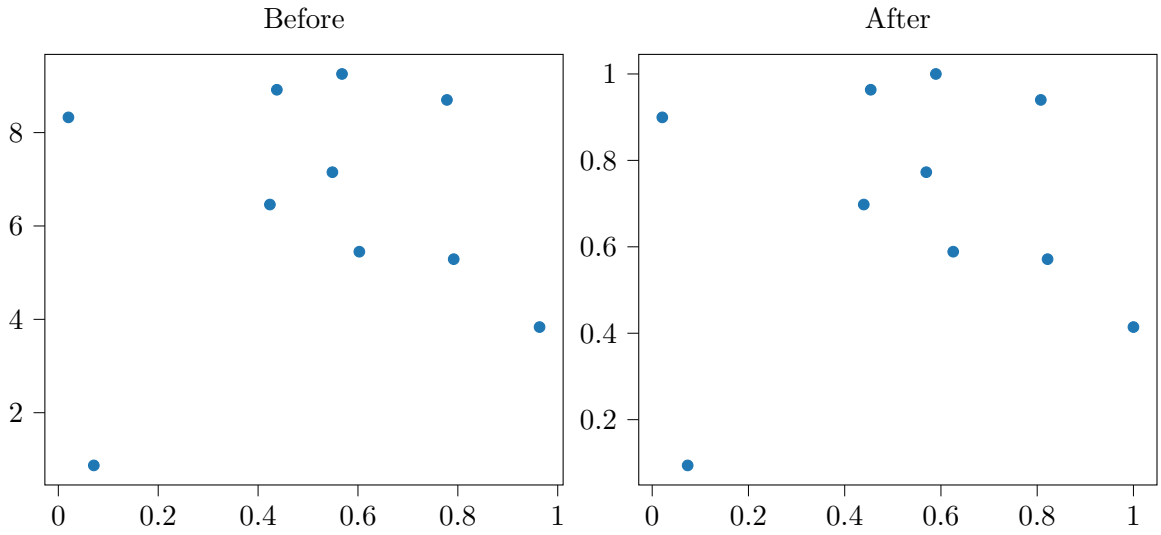


Figure II.2: Visual comparison of maximum absolute scaling. The scale of the plot on the left ranges from 0 to 1 in the X axis, and from 0 to 10 on the Y axis. On the right side, the X axis still ranges from 0 to 1, and the Y axis from 0 to 1. Comparing both plots visually, it is easy to see the relative distance between data points is preserved.

II.6.1 Principal Component Analysis

Principal Component Analysis (PCA) is a technique that identifies which features (or linear combinations of them) most explain the variance of a given dataset [Han et al., 2022]. It outputs a list of vectors, ranked from highest to lowest, by the amount of variance explained. These vectors are called principal components.

Given a dataset of n features, PCA calculates the k orthogonal vectors that most explain the variance in the dataset, with $k < n$ and k being a hyperparameter [Han et al., 2022]. The k resulting vectors can be used as features on other ML tasks, such as clustering.

A strength of PCA is its fast computing time [Halko et al., 2011]. A downside of this technique is that, because the vectors are linear combinations of the existing features, it is not trivial to define what a vector means, resulting in a low explainability of the model.

A limitation of the traditional PCA is that the whole dataset must be loaded in memory. As scRNA-seq matrices are high in dimension, they cannot always be loaded in memory in a dense format. This limitation is a challenge as the traditional PCA requires a dense matrix on the main memory to work. Given that, Incremental PCA [Artac et al., 2002] is an interesting substitute for the traditional PCA, as it allows the matrix to be loaded in chunks.

II.6.2 Unified Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP) is a KNN-based dimensionality reduction technique, mostly used to plot high dimensional data in two or three dimensions. UMAP can be defined into two steps. In the first step, a weighted graph is created. In the second step, a

similar graph with lower dimensions is built.

In more detail, in the first step, the distance between neighbors is considered to build a radius around each node on the graph. Fuzzy logic is then used to connect the nodes of the graph. The closer two nodes are, the more likely they are to be connected. Once the graph is built, a lower dimensional graph is constructed, trying to preserve the global structure (in comparison with the high dimensional graph) as much as possible.

Besides the original data, UMAP expects 4 hyperparameters - the number of neighbors to be considered, the number of dimensions of the final output, a parameter to control the graph layout, and the number of epochs to be used. The most relevant parameters are the number of neighbors and the control of the graph layout. A low number of neighbors result in better preservation of the local structure, while a higher number of neighbors result in the global structure being better preserved. As for the parameter to control the graph layout, a lower parameter choice results in points being too spread out, and a parameter too high results in points being overly connected [McInnes et al., 2018].

UMAP has three assumptions about the data [McInnes et al., 2018]. First, then data needs to be uniformly distributed on a Riemannian manifold. Second, the Riemannian metric is either constant locally or can be approximated as constant. And third, the manifold is locally connected. UMAP is a stochastic algorithm, meaning running it twice against the same input may yield different results.

II.6.3 T-Distributed Stochastic Embedding

t-Distributed Stochastic neighbor Embedding (t-SNE) is another graph-based dimensionality reduction technique, similar to UMAP. t-SNE is based on Stochastic Neighbor Embedding (SNE), with two main differences. In order to explain these differences, and why they are important, SNE will be explained.

SNE receives a matrix representing the distance between data points and returns a low-dimension representation of this matrix. In other words, having x_i and x_j representing data points in the original space, SNE returns y_i and y_j, which are points in a low-dimensional space representing x_i and x_j. The first step performed in SNE is the use of conditional probability to represent similarities [Maaten and Hinton, 2008]. Equation II.2 show the conditional probability function.

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{II.2}$$

In Equation II.2, $p_{j|i}$ represents the probability of x_i to choose x_j as its neighbor based on a Gaussian distribution centered in x_i, with σ_i denoting the variance of this Gaussian. For close points, $p_{j|i}$ is high. For points far away from each other, the value of $p_{j|i}$ is low.

Having x_i and x_j representing data points in the original space, SNE creates a map so that y_i and i_j represent x_i and x_j in a low-dimensional space. A similar conditional probability function can be used in the low-dimensional space, shown in Equation II.3.

$$q_{j|i} = \frac{exp(-||y_i - y_j||^2)}{\sum_{k \neq i} exp(-||y_i - y_k||^2)} \tag{II.3}$$

The Kullback-Leibler divergence, shown in Equation II.4, is the cost function used by SNE. The gradient descent method is used to optimize the cost function. The main hyperparameter of SNE is the perplexity, which can be interpreted as the number of neighbors to be considered [Maaten and Hinton, 2008].

$$\sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}} \tag{II.4}$$

While SNE is capable of producing good visualizations, its cost function is difficult to optimize because of the crowding problem. In a space with 10 dimensions, there can exist 11 points sharing the same distance between each other. However, this cannot be accurately represented in spaces with less than 10 dimensions. In order to represent it, SNE tends to plot these points far away from each other, resulting in other points of the dataset being crowded in the middle of the plot. t-SNE tries to tackle this problem by using a cost function that differs from SNE mostly in two ways.

The first main difference is that the cost function of SNE is not symmetric. That is, $p_{j|i}$ does not necessarily equal to $p_{i|j}$. t-SNE uses a symmetric cost function. The second difference is that t-SNE uses a Student-t distribution on the conditional probability, rather than a Gaussian. These changes alleviate both, the cost function optimization and the crowding problem [Maaten and Hinton, 2008].

t-SNE has three main limitations. First, it is not clear how well t-SNE performs when generating a map with a number of dimensions higher than 3. Second, it is sensitive to the curse of dimensionality. And third, as its cost function is non-convex, it is not guaranteed that a global optimum will be reached [Maaten and Hinton, 2008]. While random walk can be used to optimize t-SNE for large datasets, both time and memory complexities of t-SNE are $O(n^2)$ when no optimization is used.

II.7 Clustering

Clustering is an unsupervised ML task in which samples are grouped by similarity [Han et al., 2022]. Unsupervised means that the labels of each sample are not known *a priori*. In a scRNA-seq scenario, the clustering task consists of grouping similar cells based on similar gene expression

levels. This section describes the clustering techniques used in our experiments and the metric to evaluate the clustering quality.

II.7.1 K-Means

The K-Means problem consists of clustering n elements into k groups, minimizing the distance between elements in the same cluster. It is an NP-hard problem; hence finding exact solutions for large enough instances is impractical nowadays. Nonetheless, there are well-established heuristics that provide good enough results. Algorithm 1 describes the high-level steps of a well-known heuristic to solve the problem.

Algorithm 1 K-Means algorithm

Require: k, the number of clusters
Require: M, matrix containing elements to be clustered
Require: τ, a threshold used as convergence criterion
Ensure: A vector containing the cluster labels for each element
  initializeCentroids(k, M)
  initializeThreshold(τ)
  while *change* $> \tau$ **do**
    assignClustersToElements()
    recalculateCentroids()
  end while

Algorithm 1 receives three input parameters - an integer denoting the number of clusters to be used (usually denoted as k), a matrix containing the elements to be clustered (say M), and a floating point number representing the threshold to be used between iterations (denoted as τ). The implementation we use defaults τ to 0.0001, but other values can be passed as a parameter. One of M's dimensions represents the number of elements to be clustered, while the other represents the number of features of each element. Usually, rows denote elements, and columns denote features. As such, a $M_{i,j}$ denotes a matrix of i elements of j features each.

The algorithm starts by initializing k centroids, which are points representing each cluster. Then, each element in M is assigned to the cluster whose centroid is the closest. After assigning clusters to all elements in M, the centroids are recalculated and each element is clustered based on the new centroids. The process repeats until the difference between two consecutive iterations is lower than τ.

The initialization of centroids influences on the speed of convergence and the final result. A classic initialization approach is randomly selecting k elements of M as the initial centroids [Lloyd, 1982]. In 2006, a new initialization method was proposed, yielding much better results than random initialization. This newer initialization method is usually referred to as K-means++ [Arthur and Vassilvitskii, 2006].

K-means++ starts by randomly selecting an element in M as the first centroid. Then, the distance between this centroid and all remaining elements in M is calculated. After that, a new centroid is selected randomly, with probability weights based on the distance to the centroid. The farther an element is from the centroid, the higher are its chances of being selected as the next centroid. The process repeats until k centroids have been selected, then the algorithm proceeds as described in Algorithm 1. If more than one centroid has been selected, the probability weight is always based on the distance to the closest centroid.

II.7.2 Spectral Clustering

Spectral Clustering is a family of algorithms that use graph Laplacian matrices. Literature suggests that this family of algorithms performs better than classic clustering algorithms [Luxburg, 2007]. Algorithm 2 presents the steps performed in basic spectral clustering algorithms.

Algorithm 2 Spectral Clustering algorithm

Require: S, a similarity matrix between all data points
Require: k, the number of clusters to create
Ensure: A set of clusters
  constructGraph()
  computeLaplacian()
  computeEigenvectors()
  combineOriginalDatapointsAndEigenvectors()
  kmeansClustering()

As shown in Algorithm 2, spectral clustering algorithms start by creating a graph based on the similarity matrix provided as input. There are different ways to create this graph [Luxburg, 2007]. Once the graph is created, a Laplacian matrix is calculated. Then, the eigenvectors of the Laplacian matrix are calculated. The number of eigenvectors to be calculated is the number of clusters (k) to be created, which is provided as input to the algorithm. With eigenvectors calculated, a matrix with dimensions $n \times k$ is created by combining the original data points provided and the eigenvectors. Finally, K-Means is applied to this matrix and the clustering result is returned.

While these are the main steps for spectral clustering algorithms, different techniques may be applied in some of these steps, such as different approaches to creating the similarity graph. These approaches include ϵ-neighborhood, KNN graphs, and fully connected graphs. All of these graphs are weighted. In ϵ-neighborhood graphs, nodes are connected if the distance between them is lower than an ϵ threshold. In KNN graphs, nodes are connected to their k nearest neighbors. Finally, in fully connected graphs, all nodes are connected, with the similarity being used as the weight of the corresponding vertices. The choice of graph construction strategy, as well as the choice of the similarity function, influence the final result [Luxburg, 2007].

The choice of graph Laplacian to be used also influences the final results. Also, a different

algorithm other than K-Means can be used in the final clustering step. Normalization steps can be added to the steps to yield different results [Luxburg, 2007].

II.7.3 Louvain

Louvain is an algorithm for community detection in graphs. The community detection task consists of identifying clusters of highly-connected nodes in a given graph. Louvain focuses on maximizing an objective function called modularity, which is shown in Equation II.5.

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2M} \right] \delta(c_i, c_j) \tag{II.5}$$

In Equation II.5, A_{ij} denotes the weight between nodes i and j, $k_i = \sum_j A_{ij}$ denotes the sum of the weights of edges connected to i, c_i denotes the community node i is assigned, $\delta(u, v)$ is a function that returns 1 if $u = v$ or 0 otherwise, and $m = \frac{1}{2} \sum_{ij} A_{ij}$ [Blondel et al., 2008]. Algorithm 3 contains the steps executed by the Louvain algorithm.

Algorithm 3 Louvain algorithm

```
Require: G, a graph of n nodes
Ensure: A set of community clusters
  initializeCommunities()
  for n in G do
      for j neighboor if n do
          calculateSimilarity()
      end for
  end for
```

As the product of a scRNA-seq experiment is a gene expression matrix, this matrix has to be transformed into a graph in order to apply the Louvain algorithm. This transformation results in a graph where each node represents a cell, and each vertex represents the similarity between cells. This similarity is calculated by calculating the Euclidean distance between cells, based on their gene expression profiles.

II.7.4 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN), as the name suggests, is a density-based clustering algorithm. Density-based algorithms cluster data points based on the number of neighbors a data point has [Han et al., 2022]. A data point is said to be dense if it has many neighbors around it.

The DBSCAN algorithm relies on two parameters defined by the user - ϵ, which denotes the radius a data point should consider as a neighborhood; and $MinPts$, which denotes the number of neighbors a data point must have within the neighborhood to be considered dense.

Algorithm 4 DBSCAN algorithm

Require: D, a dataset containing n objects
Require: ϵ, the radius parameter
Require: $MinPts$, the neighborhood density threshold
Ensure: A set of density-based clusters
  mark all objects as unvisited
  randomly select an unvisited object p
  mark p as visited
  if the ϵ-neighborhood of p has at least $MinPts$ objects **then**
    create a new cluster C, and add P to C
    let N be the set of objects in the ϵ-neighborhood of p
    for doeach point p' in N
      if p' is unvisited **then**
        mark p' as visited
        if the ϵ-neighborhood of p' has at least $MinPnts$, add those points to N
      end if
    end for
  elsemark p as noise
  end if

II.8 Clustering Metrics

There are two types of clustering metrics - intrinsic and extrinsic [Han et al., 2022]. Extrinsic metrics need a ground truth (labels) to be calculated. In contrast, intrinsic metrics rely on calculating distances between samples, not relying on ground truth to be available. Three metrics are presented in the following sections.

II.8.1 Silhouette Coefficient

The silhouette coefficient is an intrinsic metric to evaluate clustering results [Han et al., 2022]. It produces a result between -1 and 1, where -1 denotes the worst possible result and 1 the best. Equation II.6 illustrates how this metric is calculated.

$$s(o) = \frac{b(o) - a(o)}{max\{a(o), b(o)\}} \tag{II.6}$$

In Equation II.6, o denotes a cell, $a(o)$ denotes the average distance between o and all cells in the same cluster as o, and $b(o)$ denotes the distance between o and all cells not in the same cluster as o. This metric is calculated for every cell, and the mean between all cells is considered the final score.

II.8.2 Cluster Homogeneity

Cluster homogeneity is an extrinsic metric to evaluate clustering quality [Rosenberg and Hirschberg, 2007]. It results in a value between 0 and 1 where 0 denotes the worst possible score, and 1 denotes

a perfect score. Equation II.7 illustrates the operation performed to calculate cluster homogeneity.

$$h = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases} \tag{II.7}$$

In Equation II.7, $H(C,K)$ denotes the entropy of both true and predicted labels. $H(C|K)$ denotes the entropy of true labels given the predicted labels. $H(C)$ denotes the entropy of true labels.

II.8.3 Rand Index

Rand index is another extrinsic metric for clustering evaluation [Steinley, 2004]. It also presents a value between 0 and 1 denoting worst and best possible scores, respectively. Equation II.8 presents the operation performed to calculate rand index.

$$rand = \frac{a}{a+b} \tag{II.8}$$

Given true labels and results predicted from a clustering algorithm, a denotes the count of correctly labeled samples and b denotes the number of incorrect assignments in Equation II.8.

II.9 Bootstrapping

Bootstrapping is a technique that can be used to estimate a statistical parameter for a given dataset. The technique consists of resampling a dataset with replacement multiple times [Han et al., 2022].

Given a dataset D of size s, a new dataset can be created by sampling s data points from D, with replacement. Because of the replacement, most likely some data points from D will not be selected. The sampled data points will be used as a training set, while the not sampled ones will be used a test set. Repeating this process multiple times should lead to an estimate of the statistical parameter being calculated [Han et al., 2022]. As an example, Equation II.9 shows the calculation of the accuracy for a given model.

$$Acc(M) = \frac{1}{k}\sum_{i=1}^{k}(0.632 \times Acc(M_i)_{test_set} + 0.368 \times Acc(M_i)_{train_set}) \tag{II.9}$$

In Equation II.9, $Acc(M)$ is the total accuracy of the model, $Acc(M_i)_{test_set}$ is the accuracy on the test set, and $Acc(M_i)_{train_set}$ is the accuracy on the train set, k is the number of sample sets generated, and i is the current sample set, varying from 1 to k. The caveat is that Bootstrapping tends to be too optimistic, so it works best with small datasets [Han et al., 2022].

II.10 Discussion

In this chapter, we discussed the background work necessary to understand the next chapters of this dissertation. Section II.1 introduced a few concepts about cancer, such as oncogenes and TSGs. Section II.2 introduced the scRNA-seq process, which is the process by which the data analyzed in our experiments is obtained. Section II.3 introduced gene expression matrices, and how they can be represented computationally. Section II.4 listed a few considerations for quality control when dealing with scRNA-seq data. Section II.5 introduced a few data normalization techniques, which is an important step in a scRNA-seq data analysis workflow. Section II.6 introduced a set of dimensionality reduction techniques, which are important to manage the high dimensionality of scRNA-seq data. Section II.7 introduced different clustering techniques, as clustering is also an important step in scRNA-seq data analysis. Section II.8 introduced intrinsic and extrinsic clustering metrics, which are important to assess the quality of a clustering task. Lastly, Section II.9 introduced the bootstrapping technique, which will be used to estimate statistical variables from a set of experiments we performed, which will be useful to understand the experiments in Chapter V.

Chapter III Related Work

The analysis of scRNA-seq data can be used for many purposes, such as to understand further complex diseases [Asada et al., 2021; Poirion et al., 2016; Liu et al., 2021]. This chapter presents research projects related to cancer research.

We performed a systematic search in the Scopus database to find applications of ML in scRNA-seq data analysis. We use the string *"scrna-seq" and "machine learning" and "cancer"* to search for these words in title, abstract or keywords. This search yielded 37 results as of August 2022. A brief overview of most of these papers is presented in this chapter. Also, three papers not present in the search results were added due to their relevance [Hao et al., 2021; Vieth et al., 2019; Wagner, 2020; Wolf et al., 2018].

We classify these papers accordingly to the proposed tasks and discuss each group of papers in its section. Section III.1 describes papers proposing comprehensive packages for scRNA-seq data analysis. Section III.2 presents a paper related to imputation. Section III.3 presents papers focused on cell classification. Section III.4 discusses papers focused on building Gene Regulatory Network (GRN). Section III.5 discusses papers on clustering techniques on scRNA-seq data. Section III.6 closes this chapter, summarizing the tasks presented and how our research differs from the presented papers.

III.1 Tool Development

With the ever faster-growing advancements in the scRNA-seq data analysis field, it is natural that papers toward standardization arise. While some of these papers focus on providing best practices [Nayak and Hasija, 2021; Vieth et al., 2019], other focuses on developing software packages [Mädler et al., 2021; Zeng et al., 2021; Guo et al., 2021; Hao et al., 2021; Wagner, 2020].

One of the studies we found developed a package to accelerate translational research [Mädler et al., 2021]. The package is built in Python and R, and the authors claim that its goal is to provide a balance between automation and flexibility to allow cross-study comparisons. The use cases of this package are deconvolution experiments, which combine single cell and bulk RNA-seq data.

Another tool we found is an R package for multi-omics analysis [Zeng et al., 2021]. It contains four modules, each focusing on (i) signature and tumor microenvironment (TME) estimation, (ii)

phenotype, (iii) mutations and (iv) model construction. With the increasing availability of scRNA-seq data, the authors of this tool expect it to contribute to the discovery of therapeutic targets.

A third paper we found presented an R package on scRNA-seq data specific for cancer [Guo et al., 2021]. Besides traditional scRNA-seq processing steps, developers claim the tool provides comprehensive quality control metrics, data-driven ML models for identification of tumor microenvironments for cell populations, and methods to classify cells as malignant and non-malignant cells. As shown, most packages for scRNA-seq data analysis seem to be built in R, Python, or a combination of both.

Another package developed in R is Seurat [Hao et al., 2021], which was designed for multimodal single-cell data. It provides techniques to analyze different types of data, such as transcriptomics and proteomics. Seurat can be used for quality control, normalization, clustering and many other tasks related to scRNA-seq data. The tool also provides many visualizations for each of the data types and outputs of each task. To the time of this writing, Seurat is a well maintained package that receives regular updates.

Scanpy [Wolf et al., 2018] is an open source package available in Python, and is heavily inspired in Seurat. The main functionalities available in Scanpy are data normalization, visualization, pseudotime analysis, differential expression analysis and clustering. Authors claim that the tool is scalable and easy to use, in contrast with other R-based packages. Scanpy also provides functionalities to import data from Seurat. At the time of this writing, Scanpy is still being maintained, receiving regular updates [1].

Monet [Wagner, 2020] is another open source Python package to analyse transcriptome data. It provides functionalities such as visualization (with t-SNE), clustering DBSCAN, denoising and data integration. It also provides import and export functions to transfer data from Scanpy. Monet is not currently maintained since its last commit, at the time of this writing, was in 2021 [2].

III.2 Imputation

Another task found in our search is the use of ML for scRNA-seq data imputation. Only one paper in this task was found [Yang et al., 2018]. The study applies a hybrid machine learning method to perform the imputation. In the first step, a binary classification model is trained to classify cells as tumor and non-tumor. The second step relies on finding the intersection of classification, zero-inflated and false negative model results. Lastly, a regression model is used to estimate the missing data. While no other studies on this task were found, this is an interesting task as scRNA-seq data is sparse. Part of this sparsity is caused due to low-depth sequencing,

[1] https://github.com/scverse/scanpy/commits/master/
[2] https://github.com/flo-compbio/monet/commits/master/

which fails to capture relevant biological signals.

III.3 Cell Classification

We found several papers on cell type classification. These papers focus either on immune [Jia and Hu, 2021; Torang et al., 2019] or tumor cell [Liu et al., 2022; Zhang et al., 2020; Ren et al., 2020; Kim et al., 2020] classification. One paper also focuses on reclassifying possibly mislabeled cells [Kim et al., 2019].

Another paper tries to infer tumor presence in single-cell by training classification algorithms on bulk RNA-seq data [Liu et al., 2022]. Four models were built to classify these cells - kNN, one-versus-all SVM, one-versus-one SVM, and random forest. The study concluded that it is unlikely to predict if single cells are tumors from bulk RNA-seq data.

Another paper compared different classifiers to predict cell type [Zhang et al., 2020]. Models based on several ML techniques were used, such as K Nearest Neighbors (KNN), Gaussian Naive Bayes Classifier (GNB), Decision Tree (DT), Random Forest (RF), and two types of Artificial Neural Network (ANN). The study also provided which genes most contributed to the classification in each model. The best model achieved an accuracy of 60%.

A third paper applied a Extreme Learning Machine (ELM) to classify tumor cells [Ren et al., 2020]. $L_{2,1}$ normalization is applied prior to feeding the data into the ELM to improve the robustness of the model. Five datasets were used, and the best performing model achieved an accuracy of over 97% in all of them.

The fourth paper on tumor cell classification compares different models to classify 21 types of cancer [Kim et al., 2020]. Models based on ANN, Support Vector Machine (SVM), KNN and RF were trained for the task. A dataset of 7,398 cancer and 640 normal samples was used, and models had to classify under 21 types of cancer and normal tissues. The ANNs performed considerably better than other models.

There were also papers focused on identifying immune cells. One used a 1-dimensional Convolutional Neural Network (CNN) as a classifier for different types of immune cells [Jia and Hu, 2021]. The idea behind using a CNN is that genes overlapping in the same chromosome are highly likely to share expression values. Hence, the spatial information captured by CNNs can be used to group similar genes. The model yielded an Area Under the Receiver Operating Characteristic Curve (AUROC) of 95%, and the authors say that it could be retrained to identify other types of cells.

Another immune cell classification paper proposed an Elastic-net logistic regression [Torang et al., 2019]. The code was written in R, and the model was trained to identify 15 types of immune cells. The training and testing data was obtained from 10 different studies.

One last paper on the classification proposed a model to reclassify potentially mislabeled cells in scRNA-seq experiments [Kim et al., 2019]. The model applies PCA for dimensionality reduction and AdaSampling with an ensemble of SVMs and RFs to reclassify cells. Synthetic datasets with 10 to 20% mislabeled cells were used as initial testing, with models yielding an accuracy up to 99%.

III.4 Gene Regulatory Networks

Another task we found on scRNA-seq analysis was using clustering techniques to identify Differentially Expressed Genes (DEG). These genes can be used to construct GRN, which can be further studied to understand cancer mechanisms. GRN-based studies are complex and require a robust understanding of cell biology.

One of these GRN studies focuses on Gioblastoma Multiforme (GBM), which is an aggressive type of brain cancer [Xiang et al., 2022]. scRNA-seq data is clustered to build a GRN, which is used to study potential therapeutic targets. The study predicts 12 potential targeted genes related to GBM.

Another paper analyzes scRNA-seq data of melanoma to build a GRN [Wang et al., 2021]. These networks are transformed into embeddings, and consensus clusters are identified. Cancer-related genes yielded from these clusters are further discussed in the paper.

A third paper performs a spectral clustering to identify cell subpopulations [Ye et al., 2020]. Adjusted Rand Index (ARI) is used to evaluate clustering quality (K-Means hierarchical clustering and EM are also tried). Once clusters are created, DEGs are selected to generate a Co-Expression network. This network detects interactive genes, which is useful to study conditions related to multiple genes.

III.5 Clustering

Another task we found on scRNA-seq analysis was using clustering techniques to identify DEG, which are potential therapeutic targets. In contrast with GRN-based methodologies, these studies focus on improving the clustering quality to detect DEGs.

One of these studies combines low-rank representation with K-Means to cluster cells. In contrast with similarity-based metrics, this method focuses on learning the low-rank structure of the scRNA-seq data. The study uses ARI and Normalized Mutual Information (NMI) to evaluate clustering results, which were calculated based on clustering 11 datasets. The mean of these metrics in all datasets were 0.7312 and 0.8267, respectively.

Another study proposes a hybrid approach to clustering - an ANN to compress the feature space, followed by K-Means or Expectation-maximization clustering (EM) to cluster cells [Srinivasan et al., 2020]. The ANN-based approach is named Denoising Autoencoder with Neuronal

Approximator (DAWN), which is a dimensionality reduction technique that decides the number of relevant features. The model is trained on five datasets and compared with four other methods - PCA, Independent Component Analysis (ICA), t-SNE and Single-cell Interpretation via Multi-kernel LeaRning (SIMLR). Accuracy is used to compare the results, and the proposed hybrid approach outperforms the classic methods in most scenarios.

III.6 Discussion

This chapter presented some of the many applications of ML models on scRNA-seq. We discussed tasks such as tool development, imputation, cell classification, GRN creation and clustering, and how they can support cancer research. The work described in this dissertation falls under the clustering category. We also restrict ourselves to using only breast cancer data. While some of the discussed papers in this chapter work with large datasets [Zhang et al., 2022; Guo et al., 2021], we have not identified any work in our Scopus search that uses a dataset of over 400 thousand samples for the same cancer major type. Also, our datasets consist only of human samples. The methodology we apply in our experiments is described in the next chapter.

Chapter IV Methodology

This chapter details the methodology applied in this research and is divided into three sections. Section IV.1 describes the problem being investigated. Section IV.2 describes the generic pipeline steps implemented. Section IV.3 describes our experimental design and the terminology we define to refer to our experiments.

IV.1 Problem Statement

Cancer is a complex and highly heterogeneous set of diseases. Different types of cancer change expression levels of different genes. On top of that, biological variation in individuals may also influence the likelihood of developing cancer, as well as how the disease will develop on the individual. With all this complexity involved in cancer, precision medicine yields better results than generic treatments. To apply precision medicine, it is important to identify the biomarkers of the cancer being treated.

scRNA-seq data can be used to identify these biomarkers. Clustering cells by cell type allows differential gene expression analysis to identify which genes are over-expressed in different clusters. These genes are potential therapeutic targets. Once these genes have been identified, drugs known for targeting such genes could be considered as potential treatment options.

This research aims to explore and compare different data processing and analysis pipelines to identify cancer biomarkers using scRNA-seq data. By evaluating and comparing multiple pipelines based on clustering performance metrics, we seek to identify optimal approaches for accurately clustering cells and detecting differential gene expression, with the ultimate goal of advancing precision medicine applications in cancer treatment.

IV.2 Proposed Methodology

In order to extract information out of scRNA-seq data, the raw data must be preprocessed. We create pipelines consisting of four steps: (i) quality control, (ii) normalization, (iii) dimensionality reduction, and (iv) cell clustering. Figure IV.1 illustrate these steps. Each step will be discussed in a separate section below.

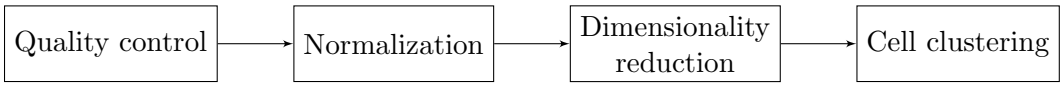


Figure IV.1: Single Cell RNA-sequencing analysis pipeline high-level steps. Each rectangle in the picture represents the steps executed in each pipeline proposed.

IV.2.1 Quality Control

Gene expression data is noisy. This noise is generated by several conditions, biological and technical. An example of biological noise is cell cycle variation. The amount of RNA available in the cell varies depending on which stage of the cycle the cell is [Andrews et al., 2021]. An example of technical variation is the batch effect, which arises from differences in sequencing conditions [Poirion et al., 2016]. Before further analysis, quality control steps are recommended (Section II.4).

The approach to handling noise depends on the type of downstream analysis to be performed. For that reason, as a first step in the pipeline, we define a filtering activity in which lowly expressed genes are discarded from the analysis. Bad quality cells can also negatively influence experiment results and should be removed. The thresholds for filtering bad quality cells are dataset specific and must be applied carefully, as important information might be removed from the process [Andrews et al., 2021].

IV.2.2 Normalization

Normalization is a set of data transformation techniques to prepare the data to be ingested by a ML model (Section II.5). Particularly for scRNA-seq data, these normalization techniques must account for variations in total counts and library size.

Noise in scRNA-seq arises from different sources. Some of these sources are technical limitations, such as low quality samples and sequencing platform-specific biases. Noise can also arise from biological sources, such as cell cycle variations. Sometimes, it is not easy to differ between noise and actual biological variations. These aspects must be taken into consideration when choosing a normalization method, since different methods are effective on different sources of noise [Lytal et al., 2020].

Many ML algorithms expect the input data to adhere to specific constraints. Normalization techniques must be chosen based on the algorithms being used in the data processing. K-Means, for instance, works better with features (gene expression levels) scaled to the same magnitude.

Besides traditional normalization techniques, there are also techniques specific to the scRNA-seq field such as Reads Per Kilobase Million (RPKM), Fragments Per Kilobase Million (FPKM) and Transcripts Per Million (TPM) [Poirion et al., 2016]. We do not explore these normalization techniques in this research. Even though these are normalization techniques, caution must be used

when comparing values across different datasets [Zhao et al., 2020].

IV.2.3 Dimensionality Reduction

In a scRNA-seq scenario, dimensionality reduction (Section II.6) is used to combine genes, reducing the number of features of each cell. This technique is fundamental in scRNA-seq analysis, as each cell has expression levels for tens of thousands of genes. Similar to normalization, different dimensionality reduction techniques should be applied depending on the data analysis objective of scRNA-seq. Dimensionality Reduction techniques are mostly used for visualization purposes, as well as a preprocessing step for clustering. There are linear dimensionality reduction methods, such as PCA, and non-linear methods, such as UMAP and t-SNE [Andrews et al., 2021].

Another reason to reduce the dimension of data is the curse of dimensionality, which are undesired effects that arise from dealing with high dimensional data. This phenomenon may result in challenges such as data sparsity, increased computing time and overfitting. Particularly in medicine, the curse of dimensionality arises from the high complexity of data, high variability and a low number of samples available [Berisha et al., 2021]. Dimensionality reduction techniques also assist in dealing with the noise created by biological and technical variations on scRNA-seq data.

IV.2.4 Cell Clustering

Clustering is an unsupervised machine learning technique that aims to group similar samples (Section II.7). There are different families of clustering techniques, such as centroid-based, hierarchical, density-based, and graph-based clustering.

An example of centroid-based clustering is K-Means, which consists of assigning each cell to a cluster based on their distance to the closest centroid. A centroid is a point in the same space as the dataset, but does not necessarily have to be a point contained in it. The main advantages of K-Means are its simplicity and fast execution time. K-Means, and centroid-based clustering techniques, in general, assume clusters are spherical and have a similar number of elements, which may not be true for scRNA-seq data.

Hierarchical clustering create a tree-like structure named dendrogram. There are two types of approaches in hierarchical clustering - agglomerative and divisive. The agglomerative approach starts with each cell belonging to a different cluster, grouping similar cells one by one until all cells belong to the same cluster. In contrast, the divisive approach starts with all cells belonging to the same cluster. Then, these cells are further broken down into different clusters based on similarity measures.

Density-based clustering techniques assume cells belonging to the same cluster form high density regions in the feature space, while cells in low density areas are not assigned to any cluster This

type of clustering technique is useful in scenarios where there is not a clear separation between cells.

In a scRNA-seq context, Graph-based clustering such as the Louvain algorithm construct a graph based on the gene expression profiles of cells. Cells with similar expression profiles are cluttered together in the graph, and are assigned to the same cluster.

Our scRNA-seq case study applies clustering techniques to group cells by cell type. Cells of the same type have similar gene expression levels, while expression levels of different cell types may vary considerably. Detecting which cell types are present in the tumor is important to define which treatment options are more likely to have a positive outcome on the patient being analyzed.

This clustering information is used for differential expression analysis. It is important to note that, while the aforementioned steps are necessary as preprocessing steps to the clustering task, the differential expression analysis is performed on raw data.

IV.3 Experiment Design

As many variables exist in our experiments, we define some terminology to explain the experiment design better. Different techniques can be applied in each step of a scRNA-seq analysis pipeline. We call *pipeline template* a set of techniques applied in scRNA-seq data. An example of a pipeline template is (i) gene filtering, (ii) maximum absolute scaling, (iii) Incremental PCA and (iv) K-Means.

Because some techniques have hyperparameters, and changing these hyperparameters influence the result, we define *pipeline instance* as a *pipeline template* associated to a set of hyperparameters. An example of pipeline instance is the aforementioned *pipeline template*, associated with (i) `mean` = 0, (ii) `none`, (iii) `batch_size` = 10000, `n_components` = 50 and (iv) `K` = 2. Some techniques have no hyperparameters (maximum absolute scaling), while others might have more than one (Incremental PCA).

Due to the stochastic nature of the methods used, we run 30 experiments for each *pipeline template*. The average and standard deviation of the metrics are collected for each *pipeline instance* to increase the statistical significance of the experiments, as well as the training time (wall and processing).

We use cluster homogeneity (Section II.8.2) and rand index (Section II.8.3) to evaluate clustering quality. We also use UMAP (Section II.6.2) to create a graphical representation of clustering results.

We, then, perform an ablation study in the *pipeline instances* with the best metrics. An ablation study removes parts of a *pipeline instance* and observes how metrics change. We define *pipeline variation* as a variation performed in a pipeline instance. An example of *pipeline variation* is removing the normalization step from a pipeline instance.

Once clustering results are available, we compare gene expression values of 20 genes known as breast cancer biomarkers [Grunda et al., 2012]. Our experiments are described in the next chapter.

Chapter V Experiments

This chapter presents experiments we performed in our research, and a discussion on results achieved. Experiments were conducted on an Ubuntu 19.10 distribution, with two Intel Xeon Gold 5120 processors, with 14 cores (28 threads) each, summing up 56 threads in total. The available memory was 187GB RAM. The code to run these experiments is available on GitHub[1].

This chapter is divided into six sections. Section V.1 describes the data used in the experiments. Section V.2 describes the Exploratory Data Analysis (EDA) performed on the datasets prior to training the ML models, as well as how labels for each cell were obtained. Section V.3 describes the *pipeline templates* used. Section V.4 describes the experiments we executed and results obtained. Section V.5 presents the expression of genes known to be related to breast cancer in some of the results obtained. Section V.6 presents the ablation study performed in each of the pipelines.

V.1 Datasets

We used two distinct datasets in the experiments, GSE75688[2] and GSE161529[3]. Both datasets were collected from Gene Expression Omnibus (GEO), a repository specialized in high-throughput gene expression data [Edgar et al., 2002]. GSE75688 consists of 549 cells collected from 11 patients. Patients have different cancer types, such as (i) ER+, (ii) ER+ and HER2+, (iii) HER2+ and (iv) TNBC [Chung et al., 2017].

GSE161529 is a larger dataset and consists of approximately 430 thousand samples (cells) collected from 69 scRNA-seq profiles performed in 52 distinct patients. This dataset includes profiles of 4 TNBCs, 4 BRCA1 TNBCs, 6 HER+, 19 ER+ and 6 lymph-node metastases of ER+ tumors. Profiles of patients without cancer are also provided, being 13 total mammary gland cells, 11 epithelial mammary gland cells, and 4 preneoplastic patients with BRCA1 mutations.

V.2 Exploratory Analysis

This section describes how each dataset is structured. It also presents general statistics about each dataset, as well as operations we performed to proceed with our experiments. Section V.2.1

[1] https://github.com/AILAB-CEFET-RJ/brca-scrna-seq
[2] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75688
[3] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE161529

describes GSE75688, and Section V.2.2 describes GSE161529.

V.2.1 GSE75688

GSE75688 consists of two files. *GSE75688_GEO_processed_Breast_Cancer_raw_TPM_matrix.txt.gz* is a compressed Tab-separated Values (TSV) file containing 57,915 rows and 566 columns. The first three columns list the Ensembl[4] id, name, and type for each gene, respectively. The following 14 columns list patients' bulk RNA-seq values (some have more than one column). The remaining 549 columns represent the gene expression levels for each cell.

The second file provided by this dataset is *GSE75688_final_sample_information.txt.gz*, which is also a compressed TSV file. It contains 528 rows and five columns. The first column lists the identifier of each cell. Cells in this dataset are identified as BCXX_YY, where XX denotes the patient number, and YY denotes the cell number. As an example, BC01_02 denotes cell 02 of patient 01. The second column denotes if the row is associated with a single cell or a bulk read. The third column labels each sample as a tumor or non-tumor. The fourth column denotes the body system the sample was extracted from. The fifth column denotes the cell type. Patients in this dataset have different molecular subtypes; only 515 out of the 549 cells are labeled. Table V.2 presents the cell distribution by cell type. The class distribution for labeled cells is illustrated in Table V.1.

Tumor type	**Cell count**
Normal	198
HER2+	130
TNBC	89
ER+	73
ER+ and HER2+	25

Table V.1: Accession GSE75688 cells distribution by cancer type.

Cell type	**Cell count**
Tumor	317
Bcell	83
Tcell	54
Myeloid	38
Stromal	23

Table V.2: Accession GSE75688 cells distribution by cell type.

We also looked at the gene expression distribution over the dataset. Figure V.1 shows the expression levels distribution for genes with highest and lowest skewness. As the figure illustrates, the expression levels do not follow a normal distribution.

[4]https://www.ensembl.org

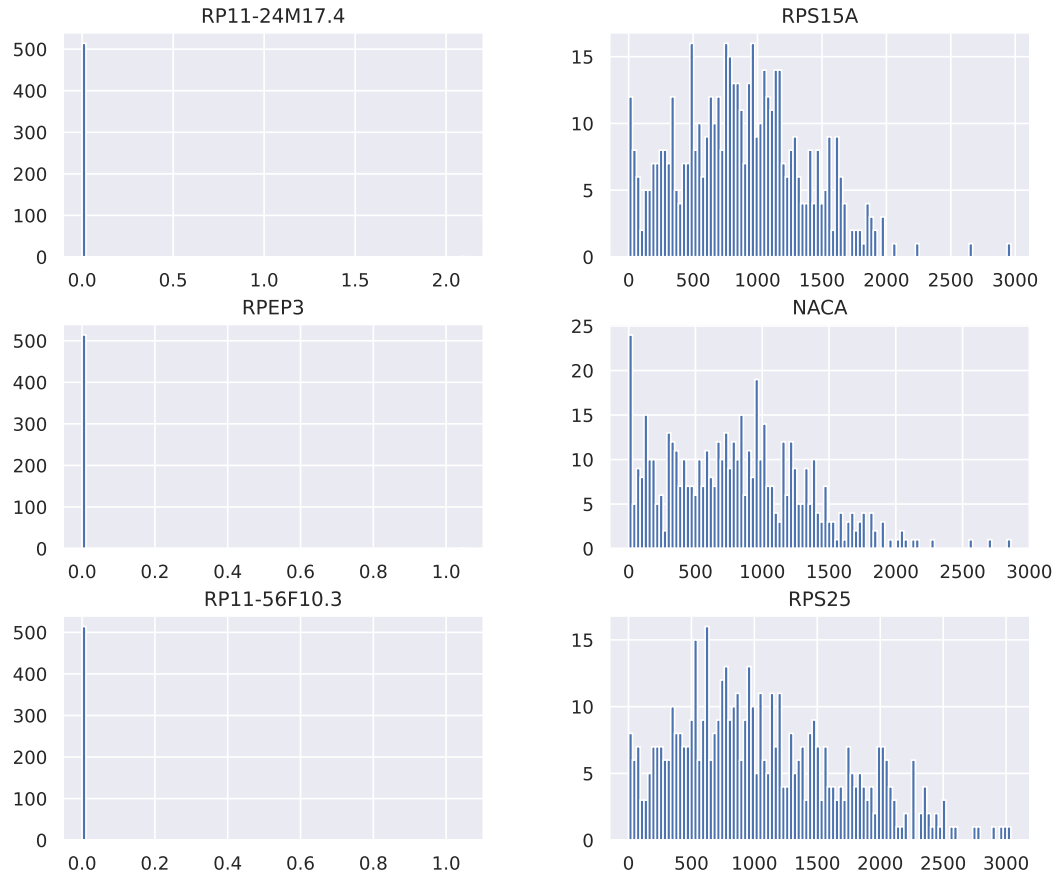


Figure V.1: Gene expression levels distribution on GSE75688. The x-axis shows the gene expression level, while the y-axis shows the frequency of cells. The left column shows the three genes whose gene expression levels have the highest skewness, while the right column shows the three with the lowest.

We used the 515 labeled cell values to proceed with our experiments, having the cell type as the label. Each cell in GSE75688 contains reads for 57915 features. 22764 of these features have zero reads in all cells, which were discarded from further analysis. We refer to GSE161529 as "small dataset" from now on.

V.2.2 GSE161529

GSE161529 is a large dataset. It consists of approximately 430 thousand samples (cells) collected from 69 scRNA-seq profiles performed in 52 patients. This dataset also contains two files structured differently than the small dataset. *GSE161529_features.tsv.gz* is a compressed TSV file with 33,538 rows and three columns and contains metadata associated with each gene. The first two columns list the Ensembl id and name of each gene. All rows of the third column have the value "Gene

Expression".

The second file in GSE161529 is *GSE161529_RAW.tar*, which is a Tape ARchive (TAR) file containing 69 pairs of files. Each pair of files is associated to a scRNA-seq profile. One file includes the gene expression matrix in the compressed mtx format, and the other contains a CSV listing the barcodes associated with each read in the matrix. All gene expression matrices in this dataset list the raw count of reads per gene.

We combined all matrices of this dataset in a single one to analyze the data. Figure V.2 shows the expression levels distribution for genes with highest and lowest skewness. As the figure illustrates, the expression levels do not follow a normal distribution.

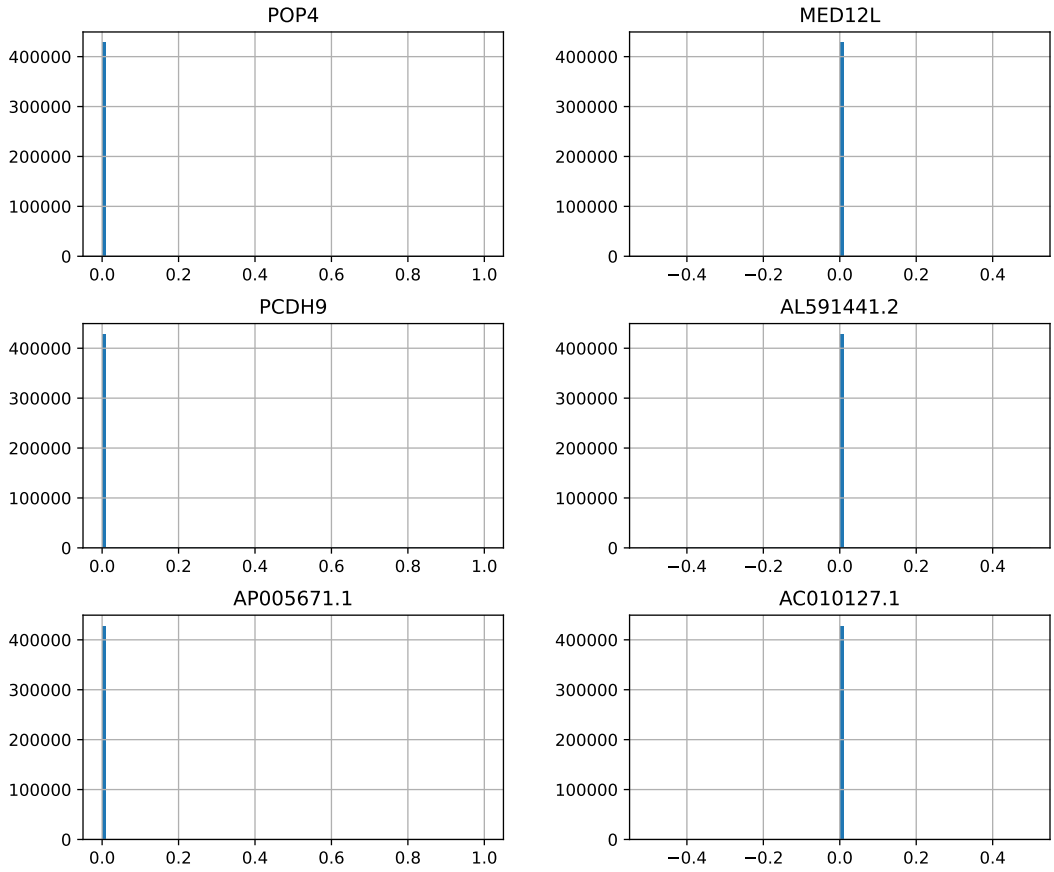


Figure V.2: Gene expression levels distribution on GSE161529. The x-axis shows the gene expression level, while the y-axis shows the frequency of cells. The left column shows the three genes whose gene expression levels have the highest skewness, while the right column shows the three with the lowest.

There is no cell type information available in GEO for the samples of this dataset. Due to time constraints and the fact that there is no scRNA-seq specialist in our team, we opted for using the

labels provided by a recent paper published by Chen et al. [Chen et al., 2022] in a high impact journal. Their work divides the dataset into 16 different subsets. Each of these subsets are used as input for a clustering algorithm, resulting in labels for each cell.

Because cell labels are a result of the clustering algorithm applied, the same subsets must be used in order to use the labels provided. We use two of the 16 groups aforementioned in our experiments. We name "large dataset" the sample group containing the integration of 13 ER+ tumors, which is the largest sample group available in the dataset. We use 57,962 of the available 63,555 cells in this group, which have been clustered into 11 distinct cell types. We name "medium dataset" the sample group containing the integration of six HER2+ tumors, which contains 31,918 cells clustered into 10 distinct cell types.

V.3 Pipeline Templates

This section describes the pipeline templates used in our experiments. Figure V.3 illustrates the pipeline structure, as well as the techniques used in our experiments.

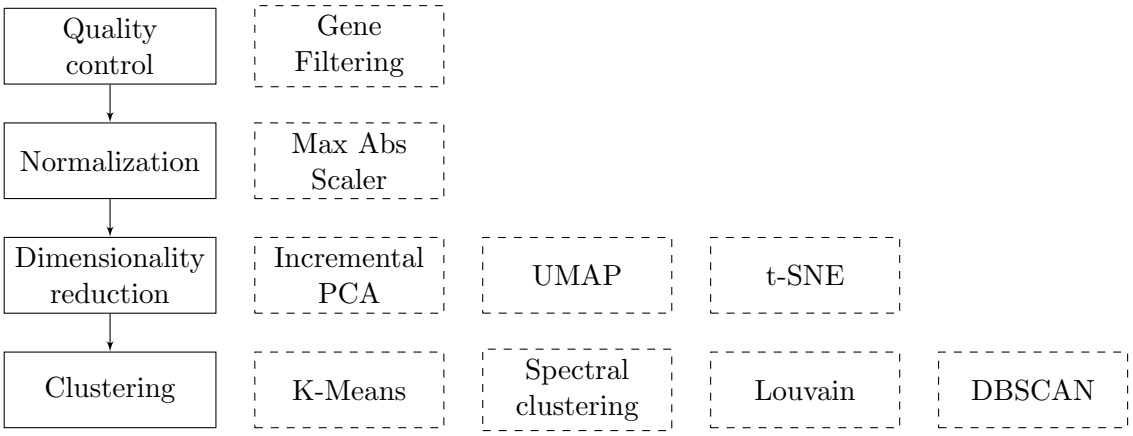


Figure V.3: Pipeline structure and techniques. Rectangles with solid borders on the left side denote the pipeline steps, and rectangles with dashed borders denote the techniques applied in our experiments. Every possible combination of techniques is used, resulting in 12 different combinations.

In Figure V.3, the rectangles with solid borders denote a step in the pipeline. The arrows close to these rectangles denote the order in which steps are executed. Hence, all pipeline templates we used follow these steps, in this order: (i) quality control, (ii) normalization, (iii) dimensionality reduction, and (iv) clustering.

Different techniques were applied in each pipeline step, which are denoted by the rectangles with dashed borders. One technique was applied in the quality control and normalization steps, three different dimensionality reduction techniques, and four clustering techniques. The combination of these techniques resulted in 12 different pipelines, which are listed below.

- P1: incremental PCA + K-Means

- P2: incremental PCA + Spectral clustering
- P3: incremental PCA + Louvain
- P4: incremental PCA + DBSCAN
- P5: UMAP + K-Means
- P6: UMAP + Spectral clustering
- P7: UMAP + Louvain
- P8: UMAP + DBSCAN
- P9: t-SNE + K-Means
- P10: t-SNE + Spectral clustering
- P11: t-SNE + Louvain
- P12: t-SNE + DBSCAN

All pipelines use gene filtering on the quality control step, following by the max abs scaling normalization method. Pipelines P1 through P4 use incremental PCA technique in the dimensionality reduction step, followed by K-Means, Spectral clustering, Louvain, and DBSCAN clustering techniques, respectively. Pipelines P5 through P8 use the same respective clustering techniques, with UMAP in the dimensionality reduction step. Finally, pipelines P9 through P12 use t-SNE in the dimensionality reduction step, followed by the previously mentioned clustering techniques.

V.4 Experiments

This section presents the experiments we ran and the results achieved. Each experiment consists of training a model with all the steps mentioned in Chapter IV, followed by an ablation study. The combinations used for each ablation study are as follows:

1. Remove only the normalization step.
2. Remove only the dimensionality reduction step.
3. Remove normalization and dimensionality reduction steps.

Three experiments were planned for each pipeline, one with each dataset used. Table V.3 shows the experiments planned and ran. Experiments with an X denote experiments ran successfully, while empty cells denote experiments that did not finish, either due to resource limitations or

time constraints. In this table, experiments with pipelines p9, p10, p11 and p12 did not finish with medium and large datasets because the t-SNE implementation we used has quadratic time complexity when the number of output dimensions is higher than three. Experiments with pipelines p3 and p7 on the large dataset did not finish because the memory available in the machine was not sufficient to load the whole dataset in RAM.

	Small	**Medium**	**Large**
p1	X	X	X
p2	X	X	X
p3	X	X	
p4	X	X	X
p5	X	X	X
p6	X	X	X
p7	X	X	
p8	X	X	X
p9	X		
p10	X		
p11	X		
p12	X		

Table V.3: Experiments executed. It was planned to execute all pipelines on all three datasets. However, some of these experiments were not completed due to time and/or computing resource restrictions. Each cell with an "X" denotes a successfully completed experiment, e.g., the top-left "X" shows that pipeline **p1** was successfully executed on the small dataset.

Due to the stochastic nature of the techniques used, each experiment was executed 30 times to increase statistical robustness of results. For each experiment, we collected the Homogeneity Score, Rand Index, Wall Time and Process time. Visualizations of the clustering results can be found on Appendix B. We discuss each metric collected in the following sections.

V.4.1 Homogeneity Score

Table V.4 presents the homogeneity scores for all performed experiments. In cells marked with "N/A", there are no results available, either due to memory limitations or time constraints. Figure V.4 provides a visual representation of the obtained values for the homogeneity score. In this figure, homogeneity score is presented in the X axis, and pipelines are presented in the Y axis. Blue, green and orange bars represent results on small, medium and large datasets, respectively. Black lines at the tip of each colored bar denote a confidence interval of 95%.

By comparing all pipelines, we see that p3 and p7, pipelines that used Louvain as the clustering algorithm, achieved the highest homogeneity scores. Pipeline p3 had a slightly better result on the small dataset, but both pipelines consistently achieved an homogeneity score close to 1 in the medium dataset. Unfortunately, there are no results available for these pipelines on the large dataset. It would be interesting to see if both pipelines would perform as consistently as they did

Table V.4: Homogeneity Scores for Different Pipelines Across Datasets

Pipeline	Small Dataset	Medium Dataset	Large Dataset
p1	0.43 ± 0.016	0.20 ± 0.006	0.10 ± 0.002
p2	0.01 ± 0.000	0.00 ± 0.000	0.00 ± 0.000
p3	1.00 ± 0.000	1.00 ± 0.000	N/A
p4	0.00 ± 0.000	0.14 ± 0.000	0.04 ± 0.000
p5	0.29 ± 0.007	0.10 ± 0.001	0.13 ± 0.001
p6	0.27 ± 0.004	0.10 ± 0.000	0.13 ± 0.002
p7	0.98 ± 0.045	1.00 ± 0.000	N/A
p8	0.23 ± 0.014	0.00 ± 0.000	0.00 ± 0.000
p9	0.02 ± 0.003	N/A	N/A
p10	0.00 ± 0.001	N/A	N/A
p11	0.05 ± 0.022	N/A	N/A
p12	0.00 ± 0.000	N/A	N/A

on the medium dataset.

In contrast, pipelines p2 had one of the worst homogeneity scores. Homogeneity scores were close to 0 in all datasets. All other pipelines had homogeneity scores lower than 0.2 on medium and large datasets. In terms of homogeneity score, Louvain pipelines achieved the best results, while K-Means, Spectral clustering and DBSCAN-based pipelines achieved far worse homogeneity scores.

V.4.2 Rand Index

Table V.5 presents the rand index scores for all performed experiments. In cells marked with "N/A", there are no results available, due to memory limitations in the machine used for performing the experiments. Figure V.5 provides a visual representation of such results. The rand index is presented in the X axis, and pipelines are presented in the Y axis. Blue, green and orange bars represent results on small, medium and large datasets, respectively. Black lines at the tip of each colored bar denote a confidence interval of 95%.

Pipeline p1 achieved the highest rand index on the small dataset, followed by p3, p5, p7, p6 and p8. Pipelines p2, p4, and t-SNE-based pipelines performed slightly worse than the previously mentioned pipelines.

While there was a huge discrepancy in the homogeneity score results, rand index results seem more balanced. Pipelines p3 and p7 achieved the best results on the medium dataset, followed by p5, p6, p4 and p1. Pipelines p2 and p8 had the worst rand indexes on the medium dataset.

On the large dataset, pipelines p3 and p7 did not execute successfully due to memory limitations, and pipelines p9, p10, 11 and p12 did not finish due to time constraints. Pipeline p6 had better results than p2 on all datasets, suggesting that UMAP was a better choice for dimensionality reduction than PCA, for pipelines that use Spectral Clustering.

Table V.5: Rand Index Values for Different Pipelines Across Datasets

Pipeline	Small Dataset	Medium Dataset	Large Dataset
p1	0.63 ± 0.011	0.55 ± 0.004	0.51 ± 0.002
p2	0.43 ± 0.000	0.28 ± 0.000	0.37 ± 0.000
p3	0.58 ± 0.000	0.72 ± 0.000	N/A
p4	0.42 ± 0.002	0.59 ± 0.000	0.47 ± 0.000
p5	0.57 ± 0.002	0.68 ± 0.000	0.61 ± 0.001
p6	0.55 ± 0.005	0.68 ± 0.000	0.60 ± 0.000
p7	0.58 ± 0.000	0.72 ± 0.000	N/A
p8	0.56 ± 0.005	0.28 ± 0.000	0.38 ± 0.000
p9	0.44 ± 0.004	N/A	N/A
p10	0.42 ± 0.001	N/A	N/A
p11	0.50 ± 0.025	N/A	N/A
p12	0.42 ± 0.000	N/A	N/A

V.4.3 Wall Time

Table V.6 presents the wall time for all performed experiments. In cells marked with "N/A", there are no results available, due to memory limitations in the machine used for performing the experiments. Figure V.6 provides a visual representation of the results, where wall time is presented in the X axis and pipelines are presented in the Y axis. Blue, green and orange bars represent results on small, medium and large datasets, respectively. Results on the small dataset are presented in seconds, while results for medium and large datasets are presented in hours. Black lines at the tip of each colored bar denote a confidence interval of 95%.

Table V.6: Wall times for different pipelines on datasets of varying sizes. Values for the small dataset are in seconds, while values for medium and large dataset are in hours.

Pipeline	Small Dataset	Medium Dataset	Large Dataset
p1	6.47 ± 0.077	0.82 ± 0.067	1.79 ± 0.012
p2	60.78 ± 1.899	2.00 ± 0.127	5.70 ± 0.198
p3	7.31 ± 0.090	1.14 ± 0.034	N/A
p4	7.09 ± 0.095	1.06 ± 0.077	3.36 ± 0.168
p5	7.65 ± 0.637	0.15 ± 0.022	0.30 ± 0.027
p6	6.71 ± 0.039	0.17 ± 0.003	0.52 ± 0.011
p7	5.63 ± 0.039	0.21 ± 0.003	N/A
p8	5.63 ± 0.025	0.12 ± 0.002	0.28 ± 0.005
p9	54.20 ± 1.091	N/A	N/A
p10	78.67 ± 8.017	N/A	N/A
p11	58.30 ± 1.019	N/A	N/A
p12	59.37 ± 2.348	N/A	N/A

Analyzing wall times on the small dataset, we see that pipelines that executed t-SNE needed much more time to finish their execution in comparison to pipelines that execute PCA and UMAP on the dimensionality reduction step. The only exception to this is p2, which uses PCA and Spectral

Clustering, and needed as much time as t-SNE-based pipelines to finish. Interestingly, p6 also uses Spectral Clustering but did not take much time to finish in comparison to p2.

On the medium dataset, PCA-based pipelines took longer to finish in comparison to pipelines that used UMAP. t-SNE-based pipelines did not finish on time due to time constraints. In fact, one successful execution of p7 took almost two weeks to finish. Since we needed at least 30 executions for each pipeline, it was not feasible to pursue the execution of t-SNE-based pipelines.

On the large dataset, p2 needed much more time to executed than all other pipelines, followed by p4, which needed about half of the time p2 did. Pipeline p1 needed about half of the time p4, and p3 did not execute successfully due to memory limitations. UMAP-based pipelines needed much less time to finish their execution, and p7 did not finish successfully for the same reason as p3.

Overall, UMAP pipelines needed far less execution time, followed by PCA and t-SNE. This is significant in terms of execution time because dimensionality reduction is the task that most accounted for execution times.

V.4.4 Process Time

Table V.7 presents the process time for all performed experiments. In cells marked with "N/A", there are no results available, due to memory limitations in the machine used for performing the experiments. Figure V.7 presents a visualization of the data shown in Table V.7, where time is presented in the X axis and pipelines are presented in the Y axis. Blue, green and orange bars represent results on small, medium and large datasets, respectively. Results on the small dataset are presented in seconds, while results for medium and large datasets are presented in hours. Black lines at the tip of each colored bar denote a confidence interval of 95%.

Table V.7: Process times for different pipelines on datasets of varying sizes. Values for the small dataset are in seconds, while values for medium and large dataset are in hours.

Pipeline	**Small Dataset**	**Medium Dataset**	**Large Dataset**
p1	298.46 ± 4.086	44.08 ± 3.733	97.29 ± 6.695
p2	3285.97 ± 102.257	62.86 ± 6.695	101.26 ± 10.222
p3	326.26 ± 4.997	55.99 ± 1.866	N/A
p4	318.24 ± 5.157	57.59 ± 4.275	172.79 ± 8.732
p5	96.37 ± 1.6	1.89 ± 0.230	4.25 ± 0.340
p6	67.32 ± 1.028	2.64 ± 0.066	10.01 ± 0.390
p7	19.74 ± 1.411	1.48 ± 0.015	N/A
p8	27.44 ± 0.981	1.74 ± 0.015	4.64 ± 0.084
p9	846.10 ± 14.133	N/A	N/A
p10	2006.22 ± 435.497	N/A	N/A
p11	842.36 ± 13.347	N/A	N/A
p12	861.53 ± 30.485	N/A	N/A

It is important to note that process times are higher than wall times because, in techniques in which parallel processing is available, the process time is multiplied by the number of threads. Pipelines using the t-SNE technique did not finish successfully due to time constraints. In fact, one successful run for pipeline p11 took over two weeks to finish, which proved experiments with t-SNE pipelines impractical.

On small and medium datasets, p2 was the pipeline with the longest process time. On the large dataset, p4 had the longest process time, which was almost twice as long as the ones for pipelines p1 and p2.

V.5 Differential Expression Visualizations

We generated visualizations of expression levels for 20 genes known to be related to breast cancer. These visualizations were generated for the seeds with highest clustering metrics for all pipelines that finished successfully, except for Louvain-based pipelines due to the number of clusters. Figure V.8 presents expression levels for pipeline p1 on the small dataset.

In Figure V.8, five clusters were generated. Cluster 0 does not seem to express any of the genes differently than other clusters. Cluster 1 highly expresses genes KRT19, MUC1 and HSPB1. Cluster 2 also does not seem to highly express any genes. Cluster 3 highly expresses genes ERBB2, GATA3, KRT19 and MUC1. Cluster 4 highly expresses genes ESR1, IL6ST, KRT19, MUC1 and SERPINA3.

A similar illustration is shown in Figure V.9, which illustrates the results for pipeline p1 on the medium dataset. Ten clusters were generated.

Clusters 0, 4, 6, 8 and 9 do not seem to express any of the 20 genes in a high level. Gene ERBB2 is highly expressed in clusters 1, 2 and 3 in comparison to other clusters. Similarly, gene KRT19 is highly expressed in clusters 1, 2, 3, 5 and 7.

Figure V.10 shows results for pipeline p1 on the large dataset, which yielded 11 clusters. Clusters 0, 7, 8 and 9 do not have high expression levels for any of the 12 genes. Gene GATA3 is highly expressed on cluster 6. Gene KRT19 is highly expressed in clusters 1, 2, 4, 5, 6 and 10. Gene HSPB1 is highly expressed in cluster 1.

This section presented visualizations for the results of pipeline p1 on small, medium and large datasets. Visualizations for other pipelines are available on Appendix C.

V.6 Ablation Study

In computational experiments, an ablation study is a systematic investigation where different components, features, or steps of a model or pipeline are selectively removed (or "ablated") to assess their impact on the overall performance. The goal is to understand the contributions and

importance of each part within a larger system, allowing researchers to isolate which aspects are most essential for achieving desired results.

This section presents the ablation study results, as well as a discussion on results achieved. Each pipeline was tested in four variations. Variation v1 was executed without scaling, v2 without dimensionality reduction, v3 without both scaling and dimensionality reduction, and v4 was executed with all steps. Each variation was tested in all three datasets. However, not all variations finished successfully. Table V.8 presents the results of the ablation study.

In Table V.8, columns HS, RI and PT denote the homogeneity score, rand index and process time, respectively. Cells with "N/A" denote the experiment did not finish successfully. Values in bold denote the best result obtained when compared to all variations of a same pipeline on the same dataset (highest for homogeneity score and rand index, and lowest for process times).

Process time is calculated by thread. As an example, two threads running for four minutes would result in a process time of eight minutes. As the machine used in our experiments has 56 threads, the process times reported are approximately 56 larger than the wall time. While mostly process time will be used for the ablation discussion, a list of process and wall times can be found at Appendix A.

V.6.1 P1 Ablation

All variations of pipeline p1 finished successfully on the small dataset. Comparing results of p1 on the small dataset, we see that the complete variation achieved the highest homogeneity score (0.52), followed by the variation without dimensionality reduction (0.40). The variations without scaling and without both, dimensionality reduction and scaling, performed similarly with a homogeneity score of 0.22. On the medium dataset we see that variations v1, v3 and v4 achieved about the same (0.26, 0.26 and 0.25, respectively), while v2 performed slightly worse (0.18). On the large dataset, the homogeneity score was about the same for all variations (between 0.10 and 0.12).

However, homogeneity score was not the only metric used. The best variation in the small dataset was v2, with a rand index of almost 0.78; followed by v4, with 0.67. Variations v1 and v3, without scaling and without both, scaling and dimensionality reduction, respectively, achieved a rand index of 0.54.

All variations of pipeline p1 finished successfully on the medium and large datasets. On medium and large datasets, the complete variation of the pipeline achieved a higher rand index, but not much different from the other variations. What draws attention in this result is that the variation without dimensionality reduction performed better even than the complete pipeline in the small dataset, even though this was not true for medium and large datasets.

Even though variations of pipeline p1 achieved similar rand index values, they had different process times. The process time for variations that did not perform dimensionality reduction (v2 and v3) was less than an hour on the medium dataset, in comparison to 48 and 29 hours on variations that performed this step (v1 and v4). This shows that the dimensionality reduction step, PCA, required the most process time.

V.6.2 P2 Ablation

All four variations of pipeline p2 executed successfully on the small dataset. On the medium and large datasets, only variations v2, v3 and v4 finished successfully. Variation v1 did not finish successfully due to memory limitations.

All variations of pipeline p2 had homogeneity scores equal or close to zero on all datasets. Rand indexes varied from 0.28 to 0.48. These metrics evidence that combining PCA with Spectral Clustering did not work well in any of the datasets. Once again, PCA was where most of the process time was spent, as variations that did not execute it finish dozens of times faster.

V.6.3 P3 Ablation

All four variations of pipeline p3 executed successfully on the small dataset. On the medium dataset, all variations executed successfully and achieved the same metrics. However, none of the four variations of the pipeline p3 executed successfully on the large dataset due to the memory available.

As all variations that finished successfully had an homogeneity score of 1, we initially thought combining PCA with Louvain was the best strategy. However, further analyzing results later showed that the homogeneity score was 1 because the pipeline had assigned one cluster to each sample. We did not have time to investigate what caused this result.

V.6.4 P4 Ablation

All four variations of pipeline p4 executed successfully on all datasets. The homogeneity score was 0 in all variations, in all datasets, except on the full pipeline on medium and large datasets, in which they were 0.14 and 0.04, respectively. Rand index values also remained the same in almost all variations in all datasets. Except on the small dataset, the complete variation of the pipeline performed slightly better than other variations, but at a much higher process time for using PCA as the dimensionality reduction technique.

V.6.5 P5 Ablation

Pipeline p5 used UMAP on the dimensionality reduction step and K-Means on the clustering step. On the small dataset, variation v1 had the highest homogeneity score (0.47), while v2 had the highest rand index (0.74). On medium and large datasets, variation v1 had the highest homogeneity score.

In comparison to pipeline p1, the only difference in p5 is the dimensionality reduction step. Pipeline p1 utilizes PCA, while p5 uses UMAP. Variations v2 and v3 of pipeline p5 had much lower process times when compared to the same variations of p1, showing that UMAP runs much faster than PCA. Also, variations v1 of pipeline p5 had the best results on the medium and large datasets in comparison to all other variations of p5.

V.6.6 P6 Ablation

Pipeline p6 uses UMAP as the dimensionality reduction step, and Spectral Clustering as the clustering step. The variation in which the normalization step is "ablated" from the pipeline had the best homogeneity scores in all three datasets when compared to other variations of the same pipeline. This variation also had the best rand index in the small and medium datasets, but the full pipeline had the best rand index in the large dataset.

Comparing all variations of pipeline p6, variations v2 and v3 had the lowest homogeneity scores and rand indexes. Neither of these variations executed the dimensionality reduction step, suggesting that UMAP was an important step for pipeline p6 to achieve better results. Comparing pipelines p2 and p6, the only different between them is that p2 executes PCA in the dimensionality reduction step, while p6 executes UMAP. While p6 achieved better results with variation v1 and v4, neither of these pipelines achieved good results, suggesting that spectral clustering was not a good choice as a clustering technique. At least not in the datasets used.

V.6.7 P7 Ablation

Pipeline p7 uses UMAP for dimensionality reduction and Louvain for clustering. While v1 achieved a homogeneity score of 0.22, all other variations achieved a homogeneity score of 1.00 on the small dataset. All variations achieved a rand index of 0.58. On the medium dataset, all variations achieved a homogeneity score of 1.00 and a rand index of 0.72. No variations finished successfully on the large dataset due to memory limitations. Similar to p3, p7 also achieved high homogeneity scores because the number of clusters created was equal to the number of cells.

In comparison to pipeline p3, the only difference is that p7 uses UMAP as a dimensionality reduction step, while p3 uses PCA. Except for variation v1 on the small dataset, both pipelines

achieved the same homogeneity scores and rand indexes on small and medium datasets. The main difference is in the process time, as UMAP finished much faster than PCA.

V.6.8 P8 Ablation

Pipeline p8 uses UMAP on the dimensionality reduction step and DBSCAN on the clustering step. On all datasets, variations v2 and v3 had a homogeneity score of 0. This was also true for the full pipeline, variation v4, on medium and large datasets. On all datasets, the variation without normalization, v1, achieved the highest metrics, both homogeneity score and rand index. This suggests that maximum absolute scaling was not a good combination with UMAP and DBSCAN.

By looking at the process time of all variations of p8, we see that variations without dimensionality reduction (v2 and v3) had a faster process time than the variations that executed this step. This is true for all three datasets. The homogeneity score results suggest that DBSCAN was not a good choice for the clustering task, which can also be observed by results in pipeline p4, which uses the same clustering technique.

V.6.9 P9, P10, P11 and P12 Ablation

All variations of pipelines p9, p10, p11 and p12 finished successfully on the small dataset. No variations of pipelines p9, p10, p11 and p12 finished successfully on medium and large datasets due to time constraints. All of these pipelines use t-SNE in the dimensionality reduction step. This technique has a $O(n^2)$ time complexity when the number of output dimensions is larger than 3, which made it impractical to run on the datasets used, given than computational resources available.

Pipelines that executed t-SNE on the small dataset are the ones with the longest process time. While most pipelines would finish under a couple of minutes on the small dataset, pipelines using t-SNE had a process time of over 10 minutes. Particularly, p10, which uses t-SNE and spectral clustering, had a process time of over 40 minutes on the small dataset. This shows that t-SNE, or at least the implementation in the Scikit-Learn library, is not suited for the dimensionality reduction step in scRNA-seq data pipelines unless it is used for visualization.

V.6.10 Ablation Summary

This section summarizes the findings of the ablation study. The pipelines that used the Louvain algorithm in the clustering step, except p11, achieved the highest homogeneity scores. While this seemed like excellent results, it was later found this was caused because each cell was assigned to a different cluster. In contrast, pipelines that used spectral clustering and DBSCAN had the worst homogeneity scores, which were close to 0 in most of the experiments.

PCA was the most expensive algorithm to run, resulting in pipelines that used it having the longest process times. Process times indicate that UMAP was much faster to run than PCA. However, this difference is not as large when comparing wall times (see Appendix A). This is because the PCA implementation as higher degree of parallelism than UMAP, even though pipelines that use UMAP had wall times faster than the ones that use PCA. The t-SNE implementation we used was not scalable for our use case, leading to experiments on medium and large datasets not finishing successfully.

Overall, results showed that none of the pipelines tested yielded good clustering metrics, except Louvain. This suggests that either the algorithms tested are not suited for scRNA-seq, or that further preprocessing is necessary on the datasets. Hyperparameter tuning might also improve results.

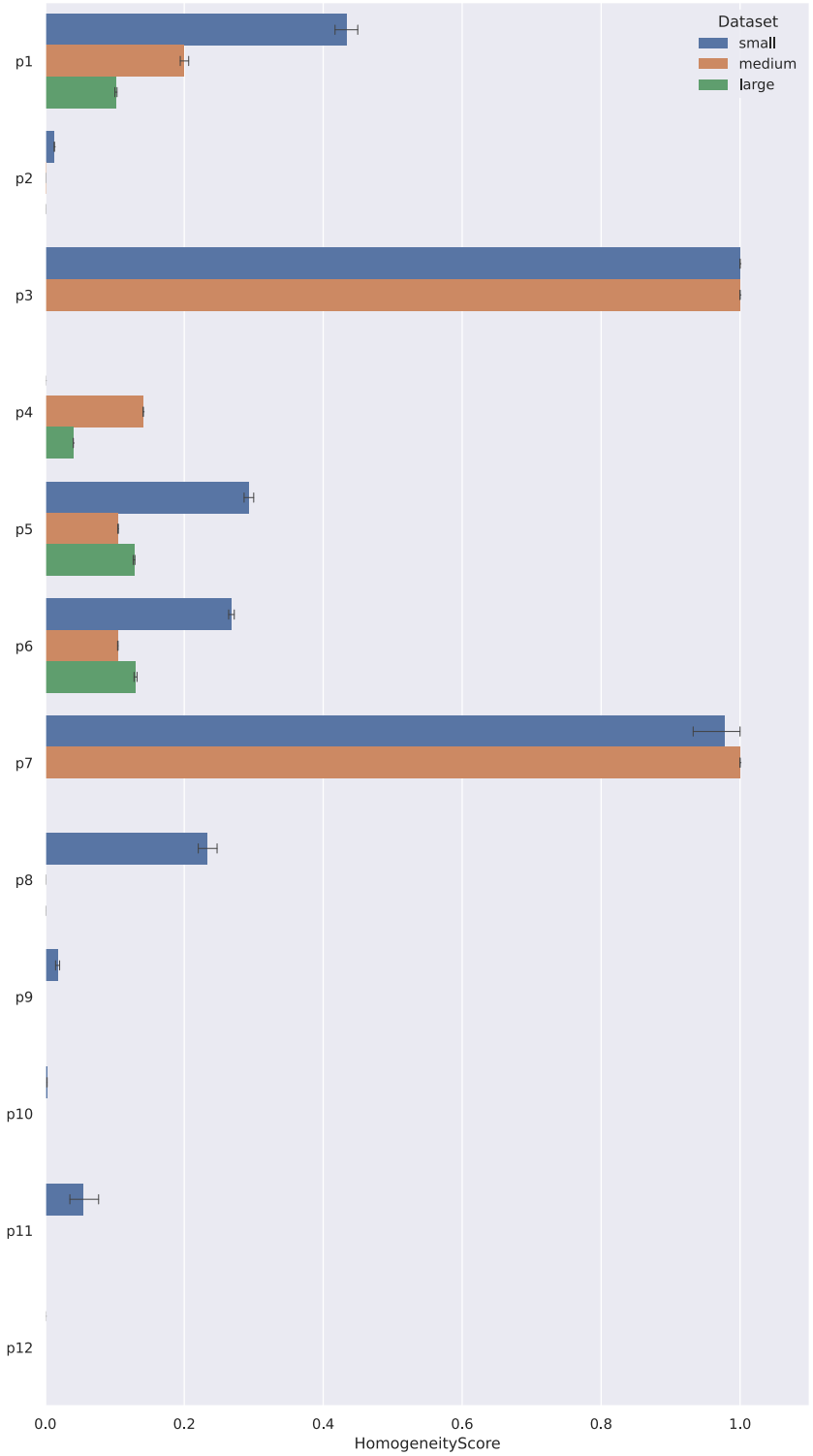


Figure V.4: Homogeneity score results. Homogeneity score is presented in the X axis, and pipelines are presented in the Y axis. Blue, green and orange bars represent results on small, medium and large datasets, respectively. Black lines at the tip of each colored bar denotes a confidence interval of 95%.

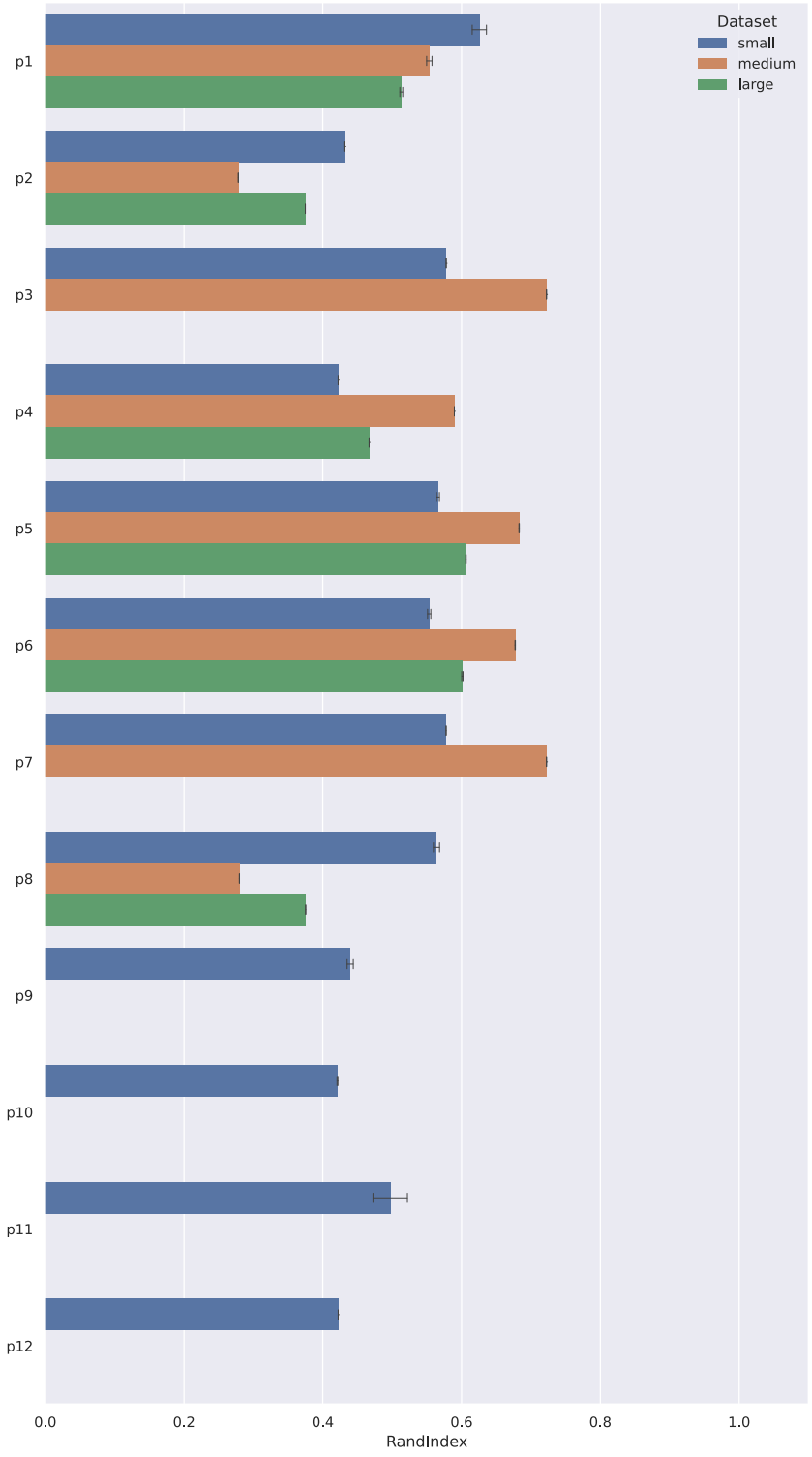


Figure V.5: Rand index results. Rand index is presented in the X axis, and pipelines are presented in the Y axis. Blue, green and orange bars represent results on small, medium and large datasets, respectively. Black lines at the tip of each colored bar denotes a confidence interval of 95%.

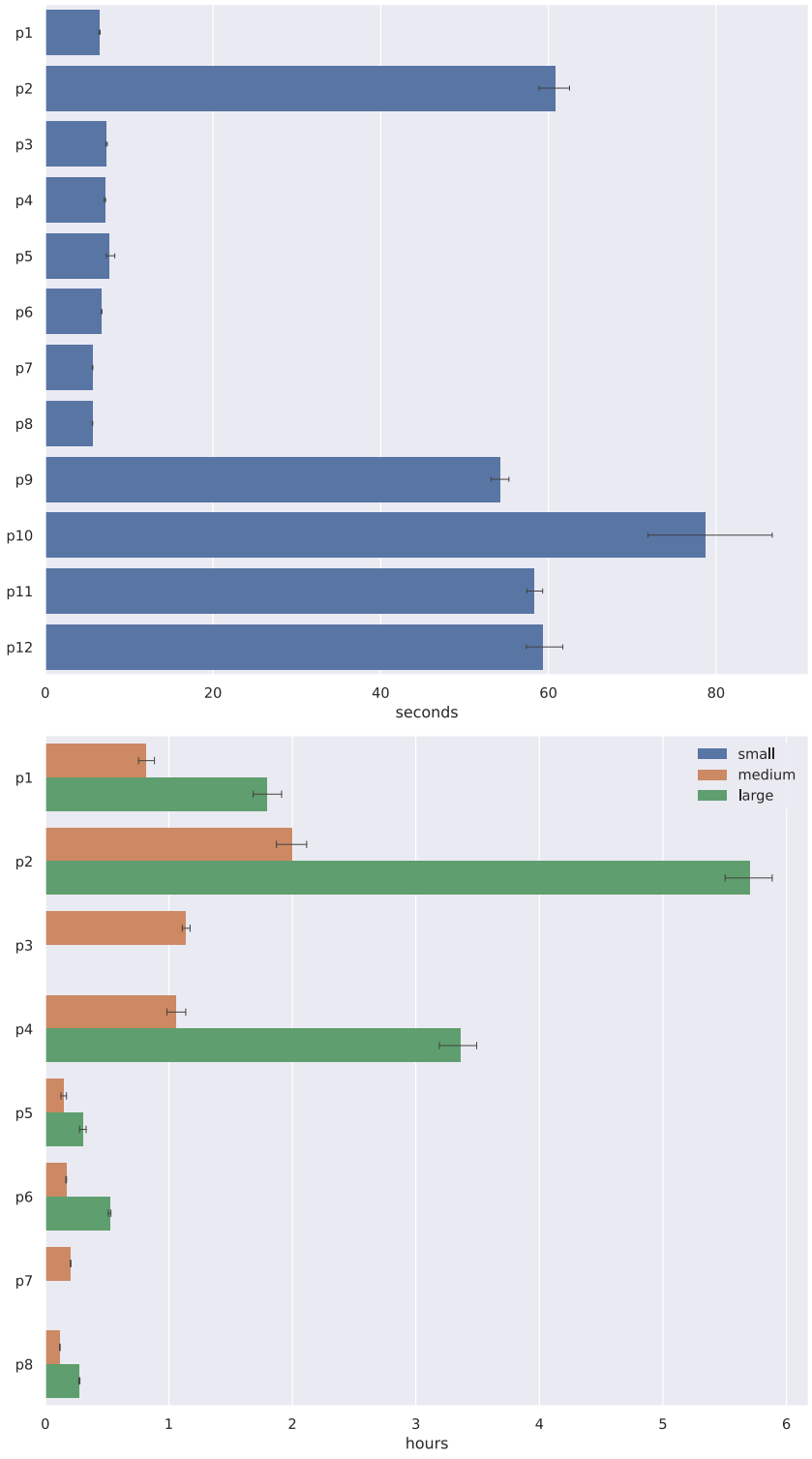


Figure V.6: Wall time. Wall time is presented in the X axis, and pipelines are presented in the Y axis. Blue, green and orange bars represent results on small, medium and large datasets, respectively. Results on the small dataset are presented in seconds, while results for medium and large datasets are presented in hours. Black lines at the tip of each colored bar denote a confidence interval of 95%.

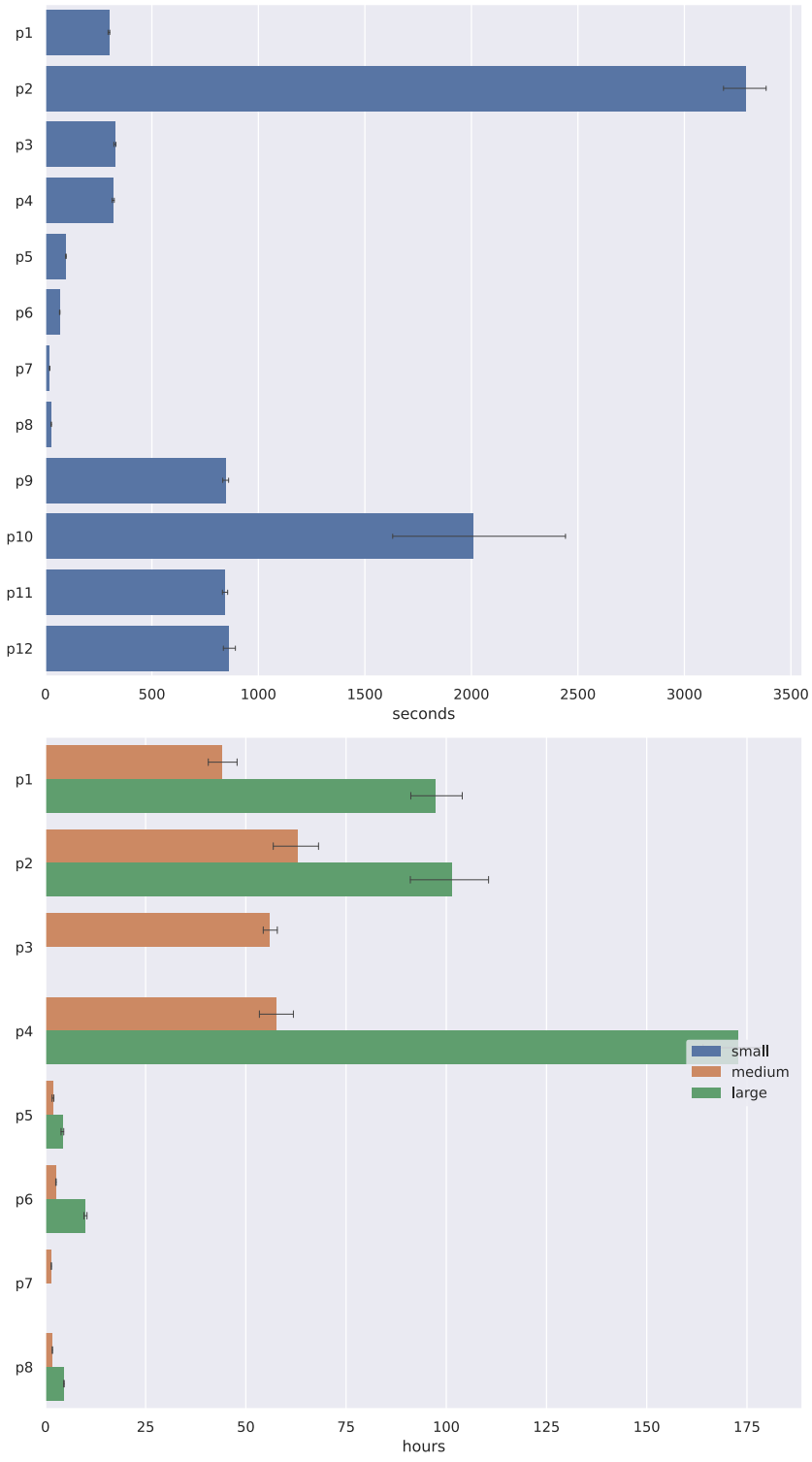


Figure V.7: Process time. Process time is presented in the X axis, and pipelines are presented in the Y axis. Blue, green and orange bars represent results on small, medium and large datasets, respectively. Results on the small dataset are presented in seconds, while results for medium and large datasets are presented in hours. Black lines at the tip of each colored bar denote a confidence interval of 95%.

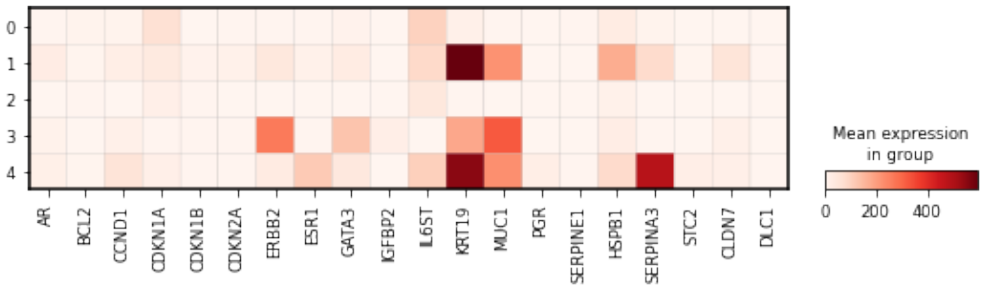


Figure V.8: Expression of genes related to breast cancer on Pipeline P1, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

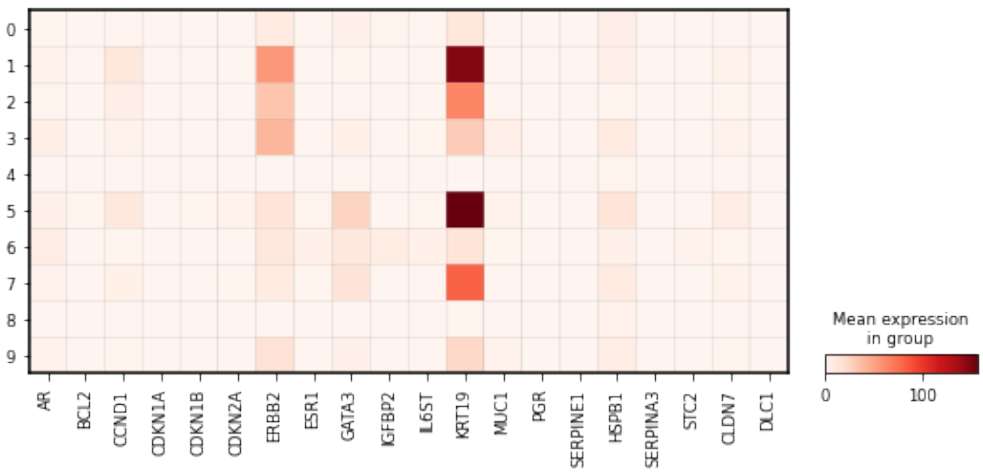


Figure V.9: Expression of genes related to breast cancer on Pipeline P1, medium dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

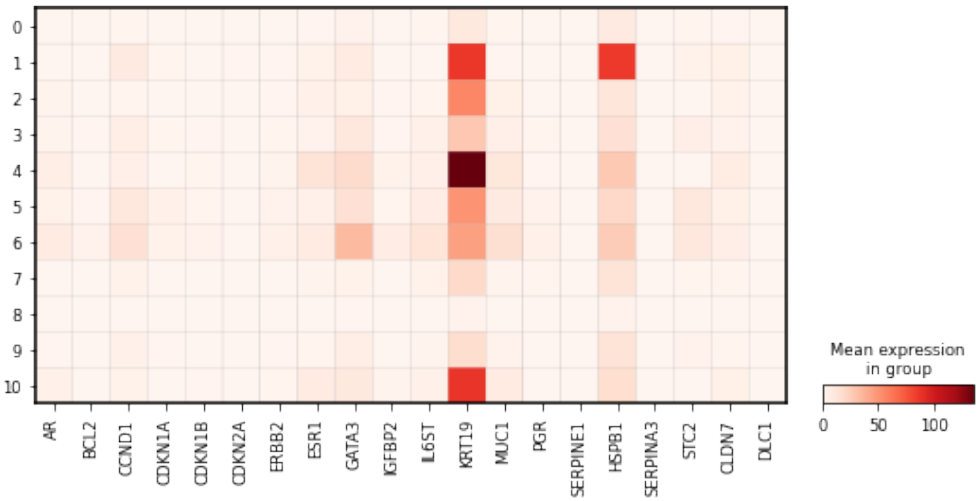


Figure V.10: Expression of genes related to breast cancer on Pipeline P1, large dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

		small			medium			large		
		HS	RI	PT	HS	RI	PT	HS	RI	PT
p1	v1	0.22	0.54	00:05:01	**0.26**	0.57	48:28:15	**0.12**	0.50	122:08:19
	v2	0.40	**0.78**	**00:00:43**	0.18	0.54	00:53:33	0.10	0.51	**02:08:03**
	v3	0.22	0.54	00:00:45	**0.26**	0.57	**00:43:22**	**0.12**	0.50	02:52:09
	v4	**0.52**	0.67	00:04:40	0.25	**0.58**	29:30:04	0.11	**0.52**	127:35:21
p2	v1	**0.02**	**0.48**	00:04:38	N/A	N/A	N/A	N/A	N/A	N/A
	v2	0.01	0.43	**00:00:40**	**0.00**	0.28	01:21:44	**0.00**	0.37	05:19:30
	v3	**0.02**	**0.48**	00:00:42	**0.00**	**0.37**	**01:11:24**	**0.00**	**0.41**	**04:56:07**
	v4	0.01	0.43	00:48:33	**0.00**	0.28	54:23:33	**0.00**	0.37	86:01:34
p3	v1	**1.00**	**0.58**	00:03:55	**1.00**	**0.72**	53:10:15	N/A	N/A	N/A
	v2	**1.00**	**0.58**	00:00:03	**1.00**	**0.72**	**03:22:32**	N/A	N/A	N/A
	v3	**1.00**	**0.58**	**00:00:03**	**1.00**	**0.72**	03:22:33	N/A	N/A	N/A
	v4	**1.00**	**0.58**	00:03:33	**1.00**	**0.72**	36:09:18	N/A	N/A	N/A
p4	v1	**0.00**	**0.42**	00:03:38	0.00	0.28	27:07:54	0.00	0.37	65:51:53
	v2	**0.00**	**0.42**	00:00:03	0.00	0.28	00:26:20	0.00	0.37	**01:11:50**
	v3	**0.00**	**0.42**	**00:00:02**	0.00	0.28	**00:26:10**	0.00	0.37	01:13:12
	v4	**0.00**	**0.42**	00:03:38	**0.14**	**0.59**	24:37:11	**0.04**	**0.47**	56:38:24
p5	v1	**0.47**	0.68	00:01:32	**0.54**	**0.74**	**00:37:09**	**0.48**	**0.65**	**01:08:12**
	v2	0.44	**0.74**	**00:00:43**	0.24	0.57	00:47:39	0.09	0.50	01:48:36
	v3	0.22	0.54	00:00:46	0.26	0.57	00:39:44	0.12	0.48	02:02:45
	v4	0.33	0.58	00:01:27	0.11	0.68	01:23:46	0.14	0.60	02:58:58
p6	v1	**0.36**	**0.62**	00:01:09	**0.46**	**0.68**	01:46:40	**0.28**	0.57	06:34:14
	v2	0.01	0.43	00:00:40	0.00	0.28	01:37:33	0.00	0.37	06:31:04
	v3	0.01	0.49	**00:00:39**	0.00	0.43	**01:11:27**	0.00	0.40	**05:54:52**
	v4	0.29	0.56	00:01:06	0.11	**0.68**	02:27:05	0.14	**0.60**	08:09:01
p7	v1	0.22	0.58	00:00:26	**1.00**	**0.72**	**00:41:54**	N/A	N/A	N/A
	v2	**1.00**	**0.58**	00:00:03	**1.00**	**0.72**	03:23:52	N/A	N/A	N/A
	v3	**1.00**	**0.58**	**00:00:03**	**1.00**	**0.72**	03:22:38	N/A	N/A	N/A
	v4	**1.00**	**0.58**	00:00:19	**1.00**	**0.72**	01:28:53	N/A	N/A	N/A
p8	v1	**0.47**	**0.61**	00:00:20	**0.39**	**0.63**	00:52:43	**0.22**	**0.50**	02:23:48
	v2	0.00	0.42	00:00:08	0.00	0.28	**00:26:13**	0.00	0.37	01:11:12
	v3	0.00	0.42	**00:00:02**	0.00	0.28	**00:26:13**	0.00	0.37	**01:10:39**
	v4	0.31	0.57	00:00:18	0.00	0.28	01:41:58	0.00	0.38	03:59:35
p9	v1	0.01	0.43	00:12:36	N/A	N/A	N/A	N/A	N/A	N/A
	v2	**0.37**	**0.71**	00:00:44	N/A	N/A	N/A	N/A	N/A	N/A
	v3	0.22	0.54	**00:00:44**	N/A	N/A	N/A	N/A	N/A	N/A
	v4	0.04	0.45	00:13:02	N/A	N/A	N/A	N/A	N/A	N/A
p10	v1	0.00	0.42	00:12:55	N/A	N/A	N/A	N/A	N/A	N/A
	v2	**0.01**	0.43	**00:00:38**	N/A	N/A	N/A	N/A	N/A	N/A
	v3	**0.01**	**0.45**	00:00:39	N/A	N/A	N/A	N/A	N/A	N/A
	v4	**0.01**	0.43	00:42:39	N/A	N/A	N/A	N/A	N/A	N/A
p11	v1	0.00	0.42	00:11:21	N/A	N/A	N/A	N/A	N/A	N/A
	v2	**0.66**	**0.63**	00:00:02	N/A	N/A	N/A	N/A	N/A	N/A
	v3	0.45	0.62	**00:00:01**	N/A	N/A	N/A	N/A	N/A	N/A
	v4	0.17	0.56	00:12:13	N/A	N/A	N/A	N/A	N/A	N/A
p12	v1	**0.00**	**0.42**	00:11:44	N/A	N/A	N/A	N/A	N/A	N/A
	v2	**0.00**	**0.42**	00:00:07	N/A	N/A	N/A	N/A	N/A	N/A
	v3	**0.00**	**0.42**	**00:00:02**	N/A	N/A	N/A	N/A	N/A	N/A
	v4	**0.00**	**0.42**	00:11:07	N/A	N/A	N/A	N/A	N/A	N/A

Table V.8: Ablation study results. Four variations of each pipeline were executed. In variation v1, the scaling step was removed. In v2, the dimensionality reduction step. In v3, both scaling and dimensionality reduction steps were removed. Lastly, in v4, all steps in the pipeline were executed. Columns HS, RI and PT denote the homogeneity score, rand index and process time, respectively. Cells with "N/A" denote the experiment did not finish successfully.

Chapter VI Conclusion

This chapter concludes the dissertation and is divided into two sections. Section VI.1 provides a retrospective analysis, describing all the approaches tried before the final version. Section VI.2 describes this research's limitations and suggests future work to expand this research.

VI.1 Retrospective Analysis

When this research project started, the original idea was to look into The Cancer Genome Atlas (TCGA) and search for breast cancer data to further understand how we could contribute to the area. After talking to a specialist in the area, we decided, due to time constraints and a cost/benefit analysis, to look into scRNA-seq data and use different techniques in a pipeline to investigate what the data would entail. However, TCGA does not have scRNA-seq data, so we had to look for another data source to proceed with our research.

After some investigation, we found out about GEO, from which we collected the two datasets used in our research (Section V.1). With both datasets in hand, we created the first end-to-end pipeline to run differential expression analysis. We opted for using Scikit-Learn because of the prior knowledge and experience of the team with the tool, besides being well-known and supported by the machine learning community.

After some trial and error, we managed to run the first pipeline, with dimensionality reduction and clustering. We used extrinsic metrics (Section II.8) to evaluate clustering results, using the cancer type of each cell as the label. These results led to a paper being published on the 15th Brazilian Symposium on Bioinformatics [Devino et al., 2022].

As scRNA-seq label information is usually not known a priori, we also tried using an intrinsic metric, Silhouette Coefficient II.8.1, to evaluate our clustering results. However, we decided to abandon this idea as the metric calculation implementation provided by Scikit-Learn would take longer than the whole pipeline to finish. Then, we used Scanpy to run a differential expression analysis on the clustering results and also to investigate the expression levels of genes known to be related to breast cancer [Grunda et al., 2012].

With these preliminary results in hand, I presented the research to an examining board. A specialist in scRNA-seq processing who was a member of the board mentioned that grouping cells by

breast cancer type does not help much with discovering potential treatment targets, and suggested us to group by cell type. Although the datasets small dataset did have cell types to be used as labels, the medium and large datasets did not, at least not explicitly. In order to be able to evaluate results on the medium and large datasets, we based ourselves on the labels provided by a follow-up study [Chen et al., 2022].

Prior to this approach, we were using all 430,000 cells in the training process. However, because we needed labels to evaluate our clustering results, we had to limit ourselves to using the same groups of cells used in the reference study [Chen et al., 2022]. On the other hand a benefit of this approach is that, because we are using the same groups of cells as the reference study, the cell filtering task is already done. This allowed us to focus on other steps of the pipeline, such as dimensionality reduction and clustering.

Once we had results for all pipelines, the pipelines using Louvain seemed to yield the best results, since their Homogeneity Score were close to 1.0. However, after further analysis, we realized that these results were not helpful as these pipelines created one cluster for each cell. We could not have a final conclusion about the cause of this behavior, but we suspect this was due to the way the gene expression matrix was converted into an adjacency matrix. We performed this conversion by simply calculating the Euclidean distance between cells, while it seems libraries such as Seurat and Scanpy use KNN to limit the number of neighbors a cell can have.

VI.2 Contributions, Limitations and Future Work

This section describes the contributions of this research, the limitations, and ideas for future work to expand this research. The main contribution of our research is the highlight of how changing techniques in each step of the pipeline influences the final result. Also, the ablation study evidences the impact of each step in the final result.

A limitation of our work is that all members of the team are computer scientists. We can talk about different dimensionality reduction and clustering techniques, but we do not have the expertise to interpret results from a biological standpoint. Another limitation of the study is that the implementation of techniques used was not suited to work with efficient representations of sparse data. Implementing such techniques to work with sparse matrix representation could allow us to work with larger datasets, given computational constraints.

There are a few ideas that could be used as future work. One of them is to train decision tree models against the clustering results for better explainability, since decision trees would evidence which features were most important for each cluster of cells. Another idea for future work would be the hyperparameter tuning of each of the models, which could lead to better results. Also, further techniques could be explored, such as the Leiden algorithm for clustering [Traag et al., 2019].

There is also a recently started project investigating the use of semi-supervised clustering techniques [da Silva, 2006; Cai et al., 2023]. Such techniques revolve around applying restrictions, or suggestions, in the clustering step. This would allow infusing expert knowledge early into the scRNA-seq pipeline.

Bibliography

Andrews, T. S., Kiselev, V. Y., McCarthy, D., and Hemberg, M. (2021). Tutorial: guidelines for the computational analysis of single-cell rna sequencing data. *Nature Protocols*, 16. 19, 20, 37, 38

Artac, M., Jogan, M., and Leonardis, A. (2002). Incremental pca for on-line visual learning and recognition. In *2002 International Conference on Pattern Recognition*, volume 3, pages 781–784 vol.3. 22

Arthur, D. and Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Technical Report 2006-13, Stanford InfoLab. 25

Asada, K., Takasawa, K., Machino, H., Takahashi, S., Shinkai, N., Bolatkan, A., Kobayashi, K., Komatsu, M., Kaneko, S., Okamoto, K., et al. (2021). Single-cell analysis using machine learning techniques and its application to medical research. *Biomedicines*, 9(11):1513. 15, 31

Berisha, V., Krantsevich, C., Hahn, P. R., Hahn, S., Dasarathy, G., Turaga, P., and Liss, J. (2021). Digital medicine and the curse of dimensionality. *npj Digital Medicine*, 4(1). All Open Access, Gold Open Access, Green Open Access. 38

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10). 27

Cai, J., Hao, J., Yang, H., Zhao, X., and Yang, Y. (2023). A review on semi-supervised clustering. *Information Sciences*, 632:164–200. 66

Chen, Y., Pal, B., Lindeman, G. J., Visvader, J. E., and Smyth, G. K. (2022). R code and downstream analysis objects for the scrna-seq atlas of normal and tumorigenic human breast tissue. *Scientific Data*, 9(1):96. 45, 65

Chung, W., Eum, H. H., Lee, H.-O., Lee, K.-M., Lee, H.-B., Kim, K.-T., Ryu, H. S., Kim, S., Lee, J. E., Park, Y. H., et al. (2017). Single-cell rna-seq enables comprehensive tumour and immune cell profiling in primary breast cancer. *Nature communications*, 8(1):1–12. 15, 41

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., and Mortazavi, A. (2016). A survey of best practices for rna-seq data analysis. *Genome Biology*, 17(1):13. 15

da Silva, E. B. (2006). *Agrupamento Semi-Supervisionado de Documentos XML*. PhD thesis, UFRJ, https://www.pesc.coppe.ufrj.br/uploadfile/publicacao/2018.pdf. 66

Devino, M., Belloze, K., and Bezerra, E. (2022). Comparison of machine learning pipelines for gene expression matrices. In Scherer, N. M. and de Melo-Minardi, R. C., editors, *Advances in Bioinformatics and Computational Biology*, pages 32–37, Cham. Springer Nature Switzerland. 64

Ding, S., Chen, X., and Shen, K. (2020). Single-cell rna sequencing in breast cancer: Understanding tumor heterogeneity and paving roads to individualized therapy. *Cancer Communications*, 40(8):329–344. 14, 19

Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210. 41

Grunda, J. M., Steg, A. D., He, Q., Steciuk, M. R., Byan-Parker, S., Johnson, M. R., and Grizzle, W. E. (2012). Differential expression of breast cancer-associated genes between stage- and age-matched tumor specimens from african- and caucasian-american women diagnosed with breast cancer. *BMC Research Notes*, 5:1–13. 16, 40, 64

Guo, W., Wang, D., Wang, S., Shan, Y., Liu, C., and Gu, J. (2021). Sccancer: A package for automated processing of single-cell rna-seq data in cancer. *Briefings in Bioinformatics*, 22(3). 31, 32, 35

Halko, N., Martinsson, P. G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288. 22

Han, J., Pei, J., and Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann. 21, 22, 24, 27, 28, 29

Hanahan, D. (2022). Hallmarks of cancer: New dimensions. *Cancer Discovery*, 12:31–46. 14, 19

Hanahan, D. and Weinberg, R. A. (2000). The hallmarks of cancer. *Cell*, 100. 19

Hanahan, D. and Weinberg, R. A. (2011). Hallmarks of cancer: The next generation. *Cell*, 144. 19

Hao, Y., Hao, S., Andersen-Nissen, E., III, W. M. M., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zagar, M., Hoffman, P., Stoeckius, M., Papalexi, E., Mimitou, E. P., Jain, J.,

Srivastava, A., Stuart, T., Fleming, L. B., Yeung, B., Rogers, A. J., McElrath, J. M., Blish, C. A., Gottardo, R., Smibert, P., and Satija, R. (2021). Integrated analysis of multimodal single-cell data. *Cell*. 31, 32

Jia, S. and Hu, P. (2021). Chrnet: A re-trainable chromosome-based 1d convolutional neural network for predicting immune cell types. *Genomics*, 113(4):2023–2031. 33

Kim, B.-H., Yu, K., and Lee, P. (2020). Cancer classification of single-cell gene expression data by neural network. *Bioinformatics*, 36(5):1360–1366. 33

Kim, T., Lo, K., Geddes, T., Kim, H., Yang, J., and Yang, P. (2019). Screclassify: Post hoc cell type classification of single-cell rna-seq data. *BMC Genomics*, 20. 33, 34

Li, G. W. and Xie, X. S. (2011). Central dogma at the single-molecule level in living cells. *Nature*, 475. 14

Li, X. and Wang, C.-Y. (2021). From bulk, single-cell to spatial rna sequencing. *International Journal of Oral Science*, 13(1). 15, 19

Liu, H.-P., Wang, D., and Lai, H.-M. (2022). Can we infer tumor presence of single cell transcriptomes and their tumor of origin from bulk transcriptomes by machine learning? *Computational and Structural Biotechnology Journal*, 20:2672–2679. 33

Liu, J., Fan, Z., Zhao, W., and Zhou, X. (2021). Machine intelligence in single-cell data analysis: Advances and new challenges. *Frontiers in Genetics*, 12. 31

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137. 25

Luxburg, U. V. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17. 26, 27

Lytal, N., Ran, D., and An, L. (2020). Normalization methods on single-cell rna-seq data: An empirical survey. *Frontiers in Genetics*, 11. All Open Access, Gold Open Access, Green Open Access. 37

Maaten, L. V. D. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9. 23, 24

McInnes, L., Healy, J., Saul, N., and Großberger, L. (2018). Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3. 23

Mädler, S., Julien-Laferriere, A., Wyss, L., Phan, M., Sonrel, A., Kang, A., Ulrich, E., Schmucki, R., Zhang, J., Ebeling, M., Badi, L., Kam-Thong, T., Schwalie, P., and Hatje, K. (2021). Besca,

a single-cell transcriptomics analysis toolkit to accelerate translational research. *NAR Genomics and Bioinformatics*, 3(4). 31

Nayak, R. and Hasija, Y. (2021). A hitchhiker's guide to single-cell transcriptomics and data analysis pipelines. *Genomics*, 113(2):606–619. 31

Pal, B., Chen, Y., Vaillant, F., Capaldo, B. D., Joyce, R., Song, X., Bryant, V. L., Penington, J. S., Di Stefano, L., Tubau Ribera, N., et al. (2021). A single-cell rna expression atlas of normal, preneoplastic and tumorigenic states in the human breast. *The EMBO journal*, 40(11):e107333. 16

Poirion, O., Zhu, X., Ching, T., and Garmire, L. (2016). Single-cell transcriptomics bioinformatics and computational challenges. *Frontiers in Genetics*, 7(SEP). 31, 37

Ren, L.-R., Gao, Y.-L., Liu, J.-X., Zhu, R., and Kong, X.-Z. (2020). L2,1-extreme learning machine: An efficient robust classifier for tumor classification. *Computational Biology and Chemistry*, 89. 33

Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. 28

Srinivasan, S., Leshchyk, A., Johnson, N., and Korkin, D. (2020). A hybrid deep clustering approach for robust cell type profiling using single-cell rna-seq data. *RNA*, 26(10):1303–1319. 34

Steinley, D. (2004). Properties of the hubert-arabie adjusted rand index. *Psychological Methods*, 9. 29

Sung, H., Ferlay, J., Siegel, R., Laversanne, M., Soerjomataram, I., Jemal, A., and Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer Journal for Clinicians*, 71(3):209–249. 14

Torang, A., Gupta, P., and Klinke, D.J., I. (2019). An elastic-net logistic regression approach to generate classifiers and gene signatures for types of immune cells and t helper cell subsets. *BMC Bioinformatics*, 20(1). 33

Traag, V., Waltman, L., and van Eck, N. (2019). From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1). 65

Vieth, B., Parekh, S., Ziegenhain, C., Enard, W., and Hellmann, I. (2019). A systematic evaluation of single cell rna-seq analysis pipelines. *Nature Communications*, 10(1). 31

Wagner, F. (2020). Monet: An open-source python package for analyzing and integrating scrna-seq data using pca-based latent spaces. *bioRxiv*. 31, 32

Wang, L., Liu, F., Du, L., and Qin, G. (2021). Single-cell transcriptome analysis in melanoma using network embedding. *Frontiers in Genetics*, 12. 34

Wolf, F., Angerer, P., and Theis, F. J. (2018). Scanpy: Large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1). 31, 32

Xiang, C.-X., Liu, X.-G., Zhou, D.-Q., Zhou, Y., Wang, X., and Chen, F. (2022). Identification of a glioma functional network from gene fitness data using machine learning. *Journal of Cellular and Molecular Medicine*, 26(4):1253–1263. 34

Yang, M., Weissman, S., Yang, W., Zhang, J., Canaann, A., and Guan, R. (2018). Misc: Missing imputation for single-cell rna sequencing data. *BMC Systems Biology*, 12. 32

Ye, X., Zhang, W., Futamura, Y., and Sakurai, T. (2020). Detecting interactive gene groups for single-cell rna-seq data based on co-expression network analysis and subgraph learning. *Cells*, 9(9):1–15. 34

Zeng, D., Ye, Z., Shen, R., Yu, G., Wu, J., Xiong, Y., Zhou, R., Qiu, W., Huang, N., Sun, L., Li, X., Bin, J., Liao, Y., Shi, M., and Liao, W. (2021). Iobr: Multi-omics immuno-oncology biological research to decode tumor microenvironment and signatures. *Frontiers in Immunology*, 12. 31

Zhang, W., Jin, W., and Yang, J. (2020). Methods of identifying cell type from single cell rna-seq data and the interpretation. pages 1672–1679. 33

Zhang, Z., Wang, Z.-X., Chen, Y.-X., Wu, H.-X., Yin, L., Zhao, Q., Luo, H.-Y., Zeng, Z.-L., Qiu, M.-Z., and Xu, R.-H. (2022). Integrated analysis of single-cell and bulk rna sequencing data reveals a pan-cancer stemness signature predicting immunotherapy response. *Genome Medicine*, 14(1). 35

Zhao, S., Ye, Z., and Stanton, R. (2020). Misuse of rpkm or tpm normalization when comparing across samples and sequencing protocols. *RNA*, 26. 38

Appendix A Wall and Processing Times

		small		medium		large	
		WT	PT	WT	PT	WT	PT
p1	v1	00:00:06	00:05:01	00:53:41	48:28:15	02:14:08	122:08:19
	v2	**00:00:05**	**00:00:43**	00:03:19	00:53:33	**00:06:50**	**02:08:03**
	v3	00:00:06	00:00:45	**00:02:57**	**00:43:22**	00:07:57	02:52:09
	v4	00:00:06	00:04:40	00:33:13	29:30:04	02:20:09	127:35:21
p2	v1	00:00:06	00:04:38	N/A	N/A	N/A	N/A
	v2	**00:00:03**	**00:00:40**	00:29:05	01:21:44	01:45:44	**05:19:30**
	v3	**00:00:03**	00:00:42	**00:27:58**	**01:11:24**	**01:39:38**	04:56:07
	v4	00:00:53	00:48:33	01:45:26	54:23:33	05:16:06	86:01:34
p3	v1	00:00:05	00:03:55	01:04:54	53:10:15	N/A	N/A
	v2	**00:00:03**	**00:00:03**	**03:22:32**	**03:22:32**	N/A	N/A
	v3	**00:00:03**	**00:00:03**	03:22:34	03:22:33	N/A	N/A
	v4	00:00:04	00:03:33	00:47:11	36:09:18	N/A	N/A
p4	v1	00:00:04	00:03:38	00:30:46	27:07:54	01:13:19	65:51:53
	v2	**00:00:02**	00:00:03	00:26:20	00:26:20	01:11:50	**01:11:50**
	v3	**00:00:02**	**00:00:02**	**00:26:10**	**00:26:10**	01:13:12	01:13:12
	v4	00:00:04	00:03:38	00:28:00	24:37:11	**01:03:30**	56:38:24
p5	v1	00:00:06	00:01:32	00:03:11	00:37:09	**00:04:55**	**01:08:12**
	v2	00:00:06	**00:00:43**	00:03:09	**00:47:39**	00:05:53	01:48:36
	v3	**00:00:05**	00:00:46	**00:02:51**	00:39:44	00:06:10	02:02:45
	v4	00:00:06	00:01:27	00:06:05	01:23:46	00:12:58	02:58:58
p6	v1	00:00:06	00:01:09	00:06:53	01:46:40	**00:23:43**	06:34:14
	v2	**00:00:03**	00:00:40	00:29:25	01:37:33	01:49:42	06:31:04
	v3	**00:00:03**	**00:00:39**	**00:28:13**	**01:11:27**	01:44:19	**05:54:52**
	v4	00:00:06	00:01:06	00:10:47	02:27:05	00:26:57	08:09:01
p7	v1	00:00:09	00:00:26	**00:09:24**	**00:41:54**	N/A	N/A
	v2	**00:00:03**	**00:00:03**	03:23:52	03:23:52	N/A	N/A
	v3	**00:00:03**	**00:00:03**	03:22:37	03:22:38	N/A	N/A
	v4	00:00:05	00:00:19	00:12:44	01:28:53	N/A	N/A
p8	v1	00:00:05	00:00:20	**00:03:31**	00:52:43	**00:06:27**	02:23:48
	v2	**00:00:02**	00:00:08	00:26:13	**00:26:13**	01:11:12	01:11:12
	v3	**00:00:02**	**00:00:02**	00:26:13	**00:26:13**	01:10:40	**01:10:39**
	v4	00:00:05	00:00:18	00:07:46	01:41:58	00:13:33	03:59:35
p9	v1	00:00:47	00:12:36	N/A	N/A	N/A	N/A
	v2	**00:00:05**	**00:00:44**	N/A	N/A	N/A	N/A
	v3	**00:00:05**	**00:00:44**	N/A	N/A	N/A	N/A
	v4	00:00:49	00:13:02	N/A	N/A	N/A	N/A
p10	v1	00:00:48	00:12:55	N/A	N/A	N/A	N/A
	v2	**00:00:03**	**00:00:38**	N/A	N/A	N/A	N/A
	v3	**00:00:03**	00:00:39	N/A	N/A	N/A	N/A
	v4	00:01:22	00:42:39	N/A	N/A	N/A	N/A
p11	v1	00:00:46	00:11:21	N/A	N/A	N/A	N/A
	v2	**00:00:01**	00:00:02	N/A	N/A	N/A	N/A
	v3	**00:00:01**	**00:00:01**	N/A	N/A	N/A	N/A
	v4	00:00:49	00:12:13	N/A	N/A	N/A	N/A
p12	v1	00:00:47	00:11:44	N/A	N/A	N/A	N/A
	v2	**00:00:02**	00:00:07	N/A	N/A	N/A	N/A
	v3	**00:00:02**	**00:00:02**	N/A	N/A	N/A	N/A
	v4	00:00:44	00:11:07	N/A	N/A	N/A	N/A

Table A.1: Ablation study wall and processing times. Four variations of each pipeline were executed. In variation v1, the scaling step was removed. In v2, the dimensionality reduction step. In v3, both scaling and dimensionality reduction steps were removed. Lastly, in v4, all steps in the pipeline were executed. Columns WT and PT denote the wall and process times, respectively. Cells with "N/A" denote the experiment did not finish successfully.

Appendix B Clustering Results Visual Representations

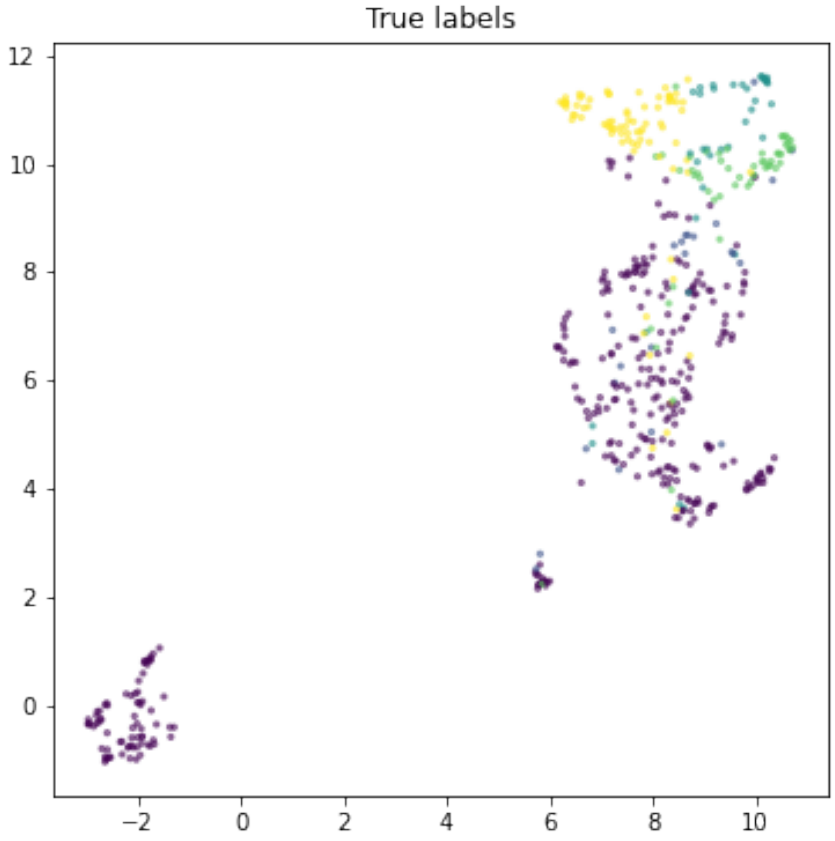


Figure B.1: Toy dataset UMAP true labels.

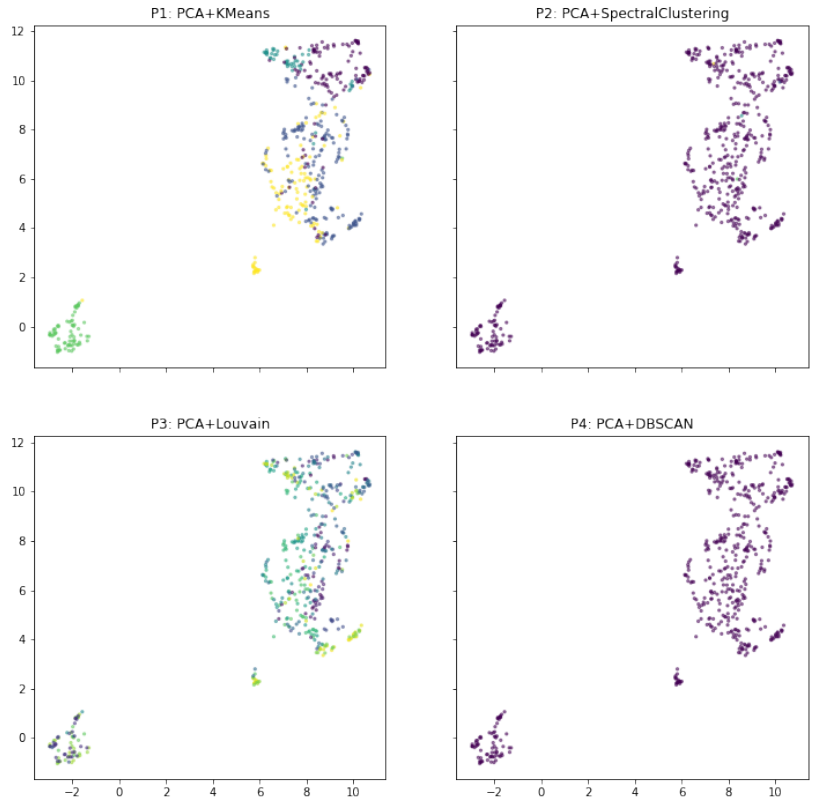


Figure B.2: Toy dataset UMAP for pipelines P1, P2, P3 and P4.

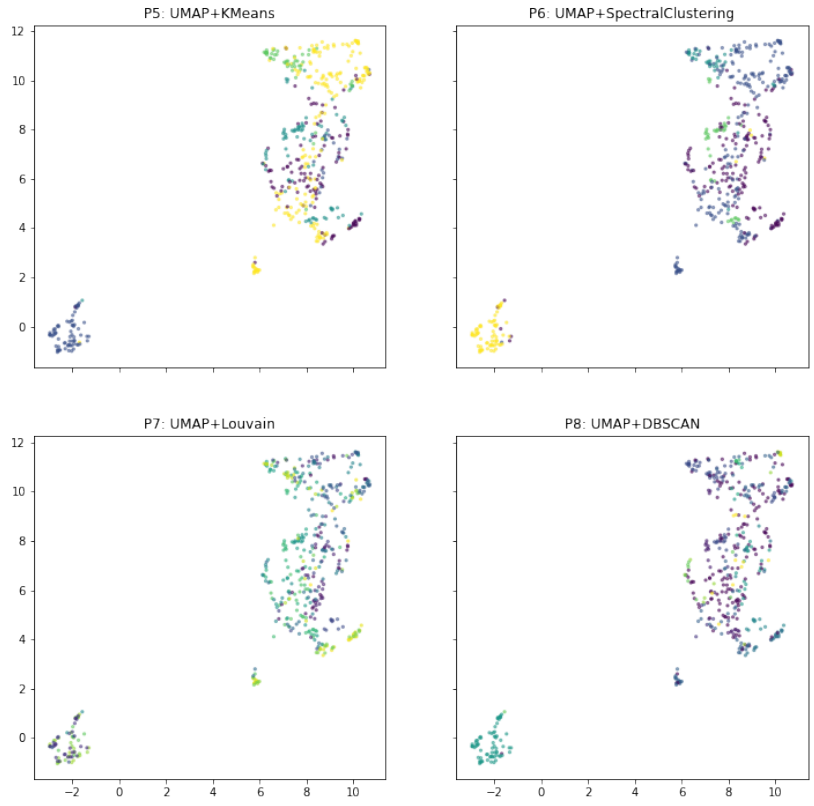


Figure B.3: Toy dataset UMAP for pipelines P5, P6, P7 and P8.

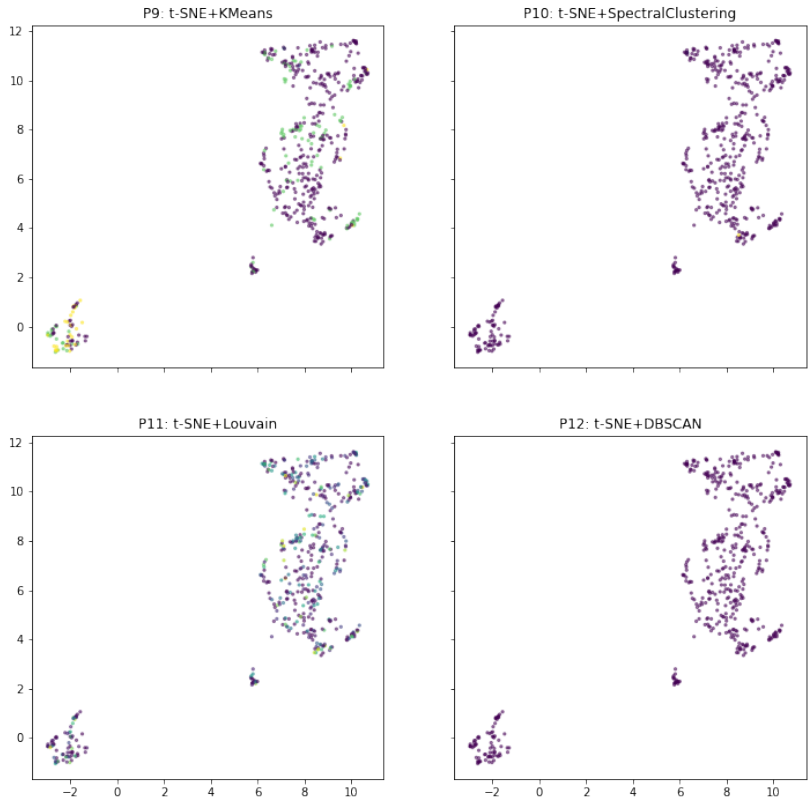


Figure B.4: Toy dataset UMAP for pipelines P9, P10, P11 and P12.

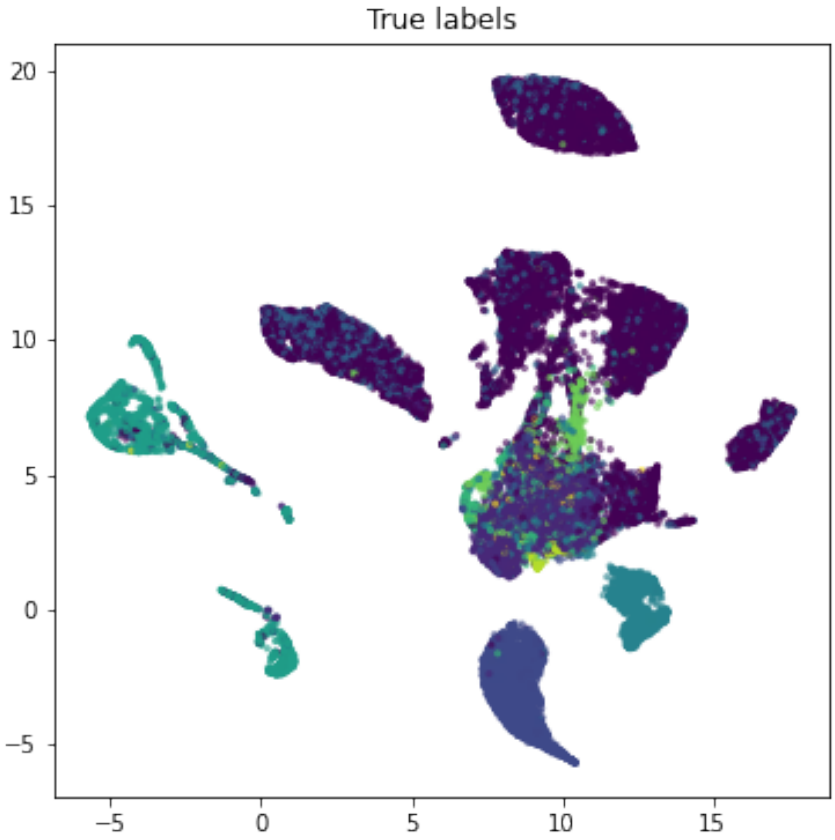


Figure B.5: Medium dataset UMAP true labels.

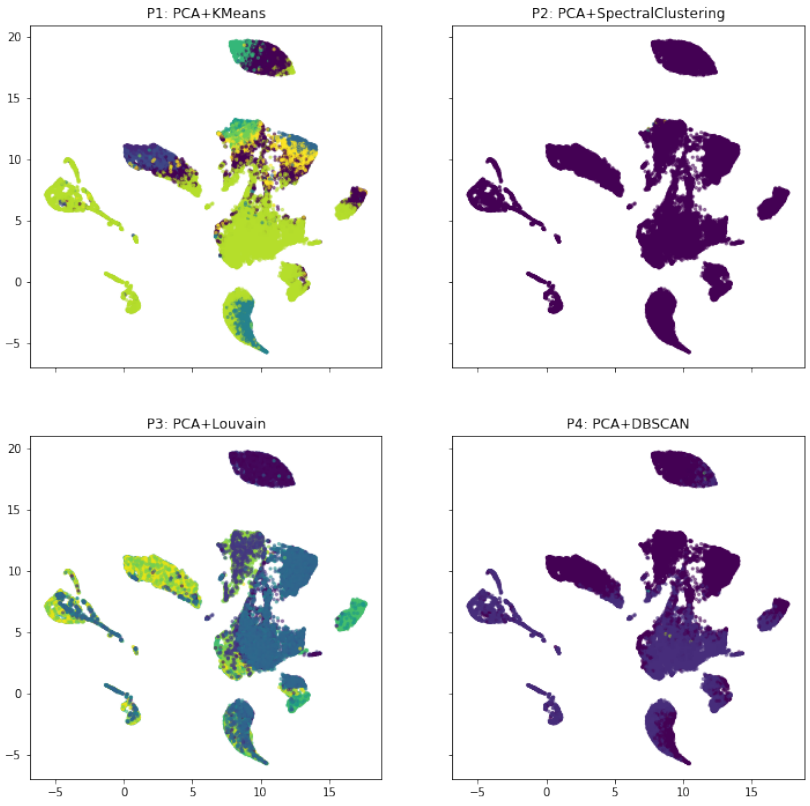


Figure B.6: Medium dataset UMAP for pipelines P1, P2, P3 and P4.

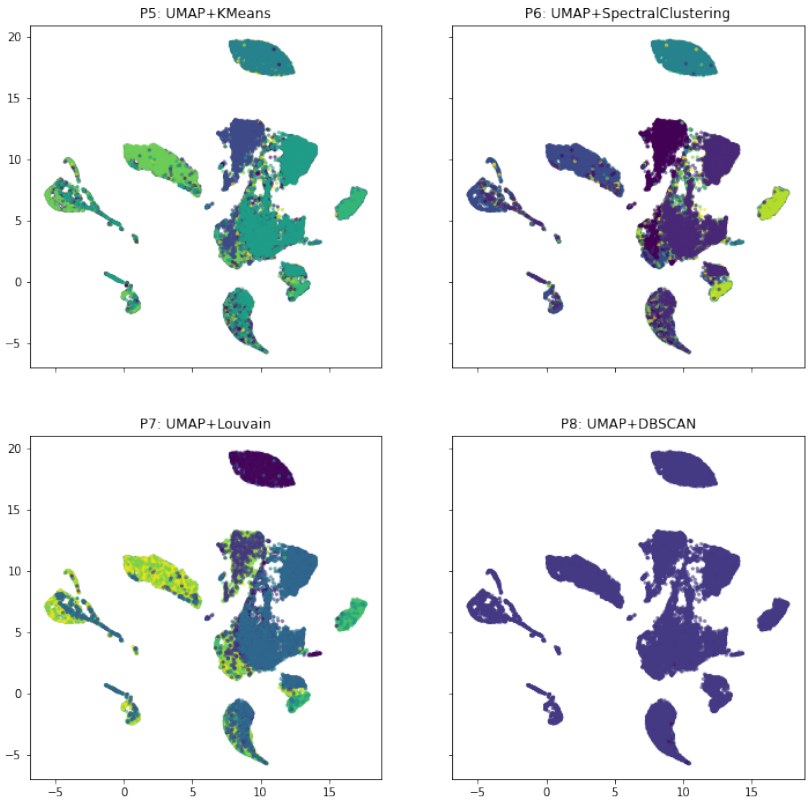


Figure B.7: Medium dataset UMAP for pipelines P5, P6, P7 and P8.

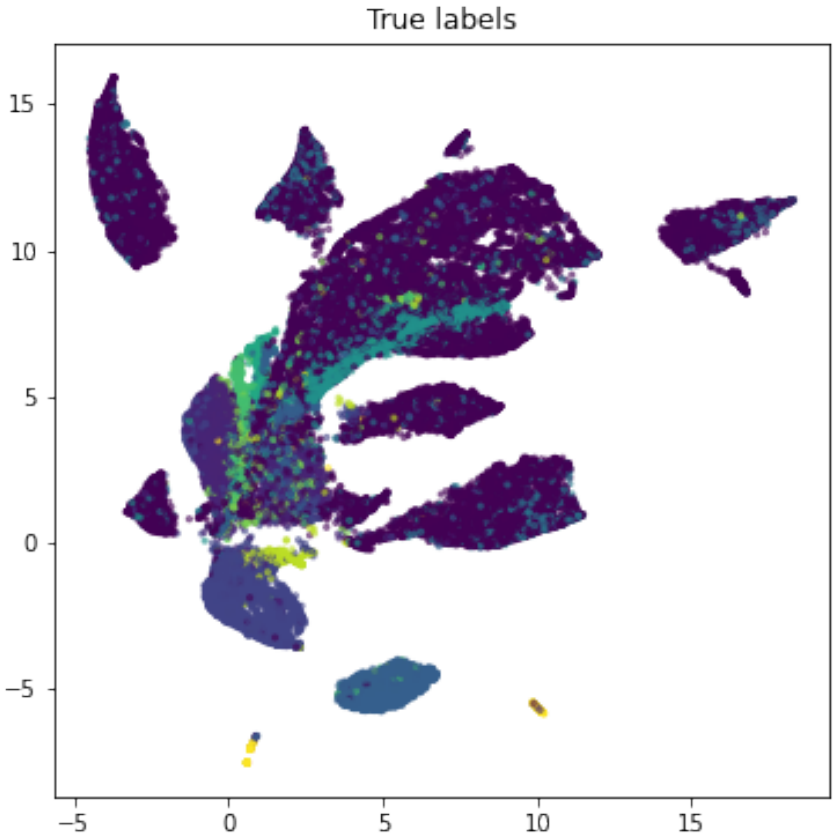


Figure B.8: Large dataset UMAP true labels.

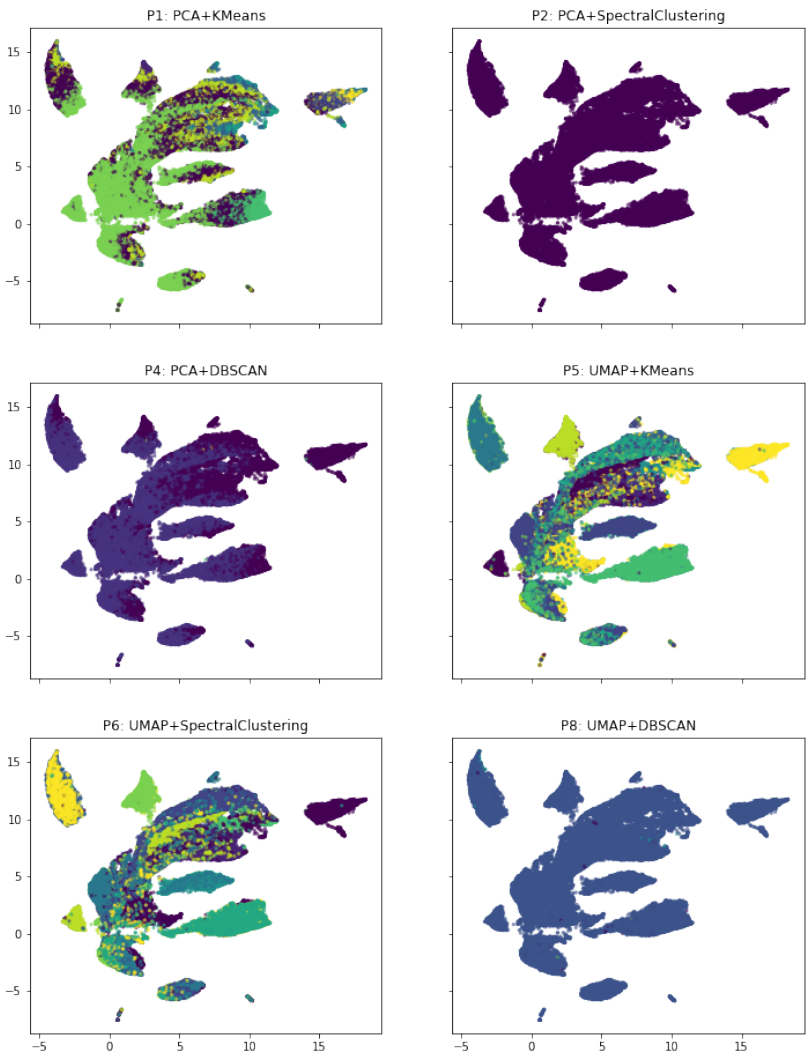


Figure B.9: Large dataset UMAP for pipelines P1, P2, P4, P5, P6 and P8.

Appendix C Expression of Known Genes

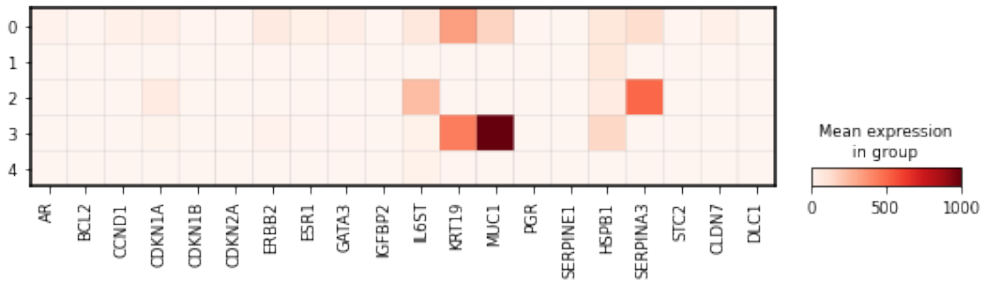


Figure C.1: Expression of genes related to breast cancer on Pipeline P2, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

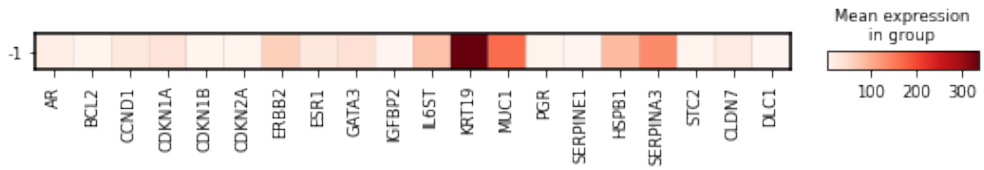


Figure C.2: Expression of genes related to breast cancer on Pipeline P4, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

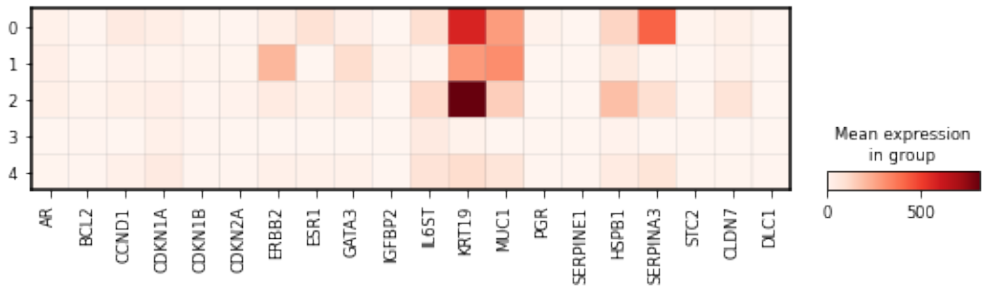


Figure C.3: Expression of genes related to breast cancer on Pipeline P5, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

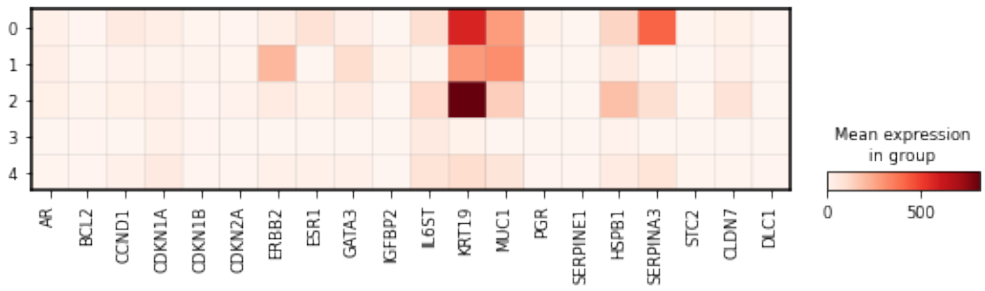


Figure C.4: Expression of genes related to breast cancer on Pipeline P6, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

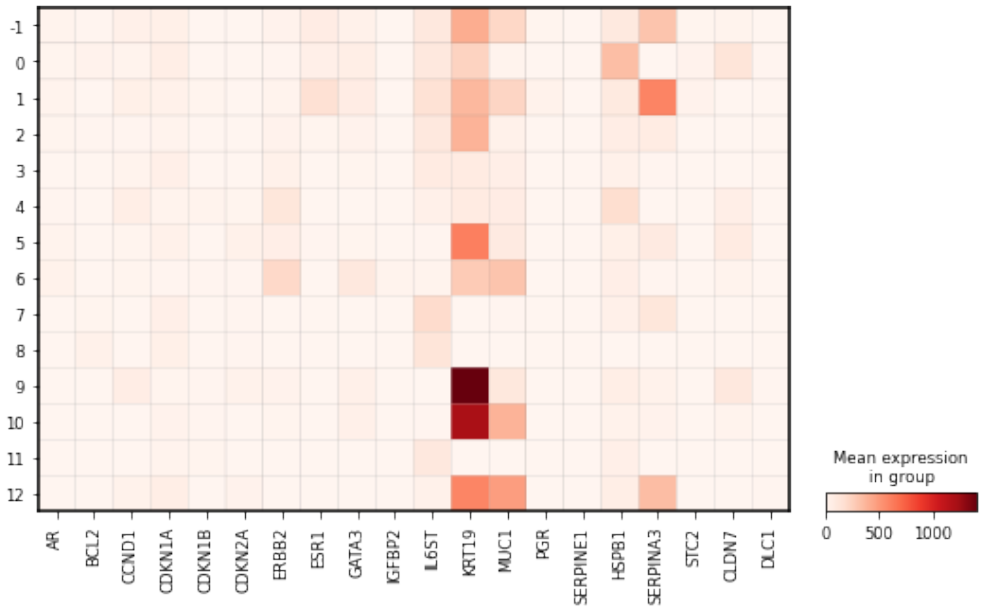


Figure C.5: Expression of genes related to breast cancer on Pipeline P8, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

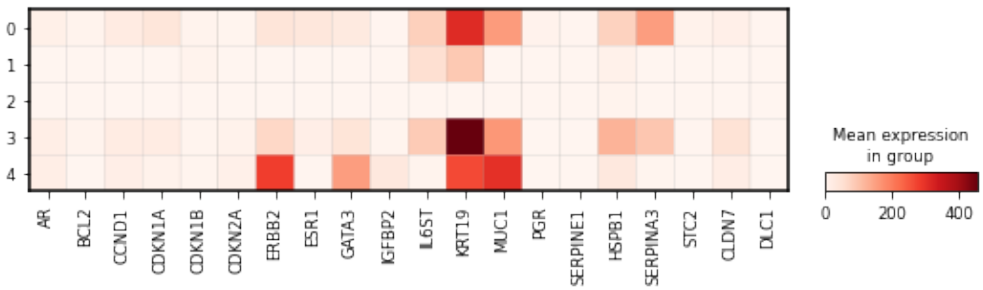


Figure C.6: Expression of genes related to breast cancer on Pipeline P9, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

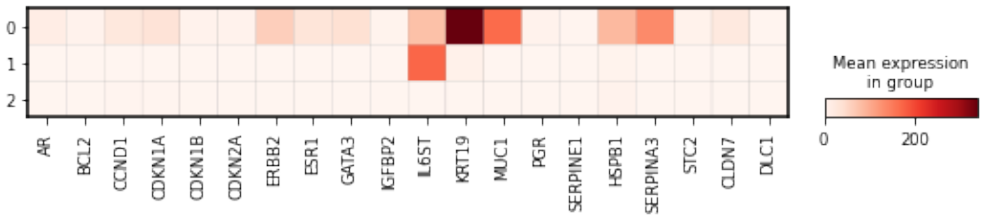


Figure C.7: Expression of genes related to breast cancer on Pipeline P10, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

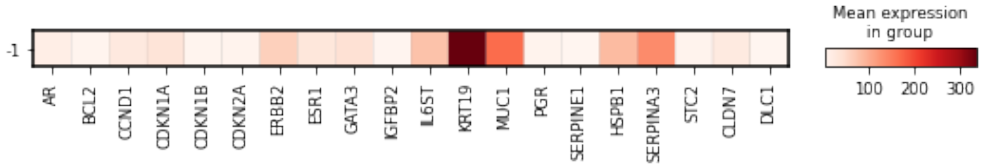


Figure C.8: Expression of genes related to breast cancer on Pipeline P12, small dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

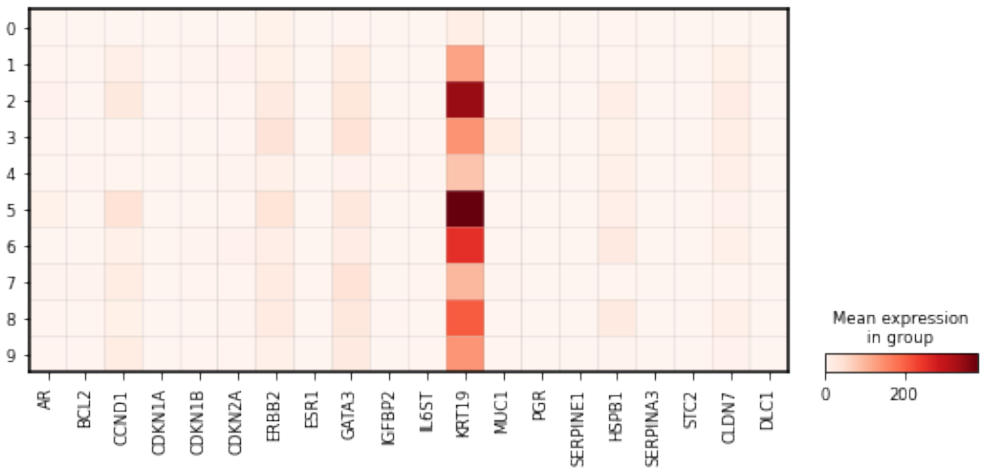


Figure C.9: Expression of genes related to breast cancer on Pipeline P2, medium dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

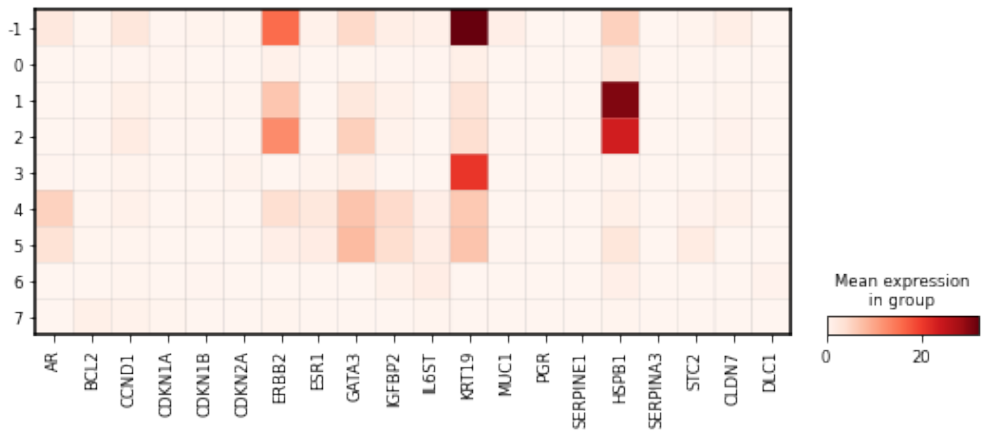


Figure C.10: Expression of genes related to breast cancer on Pipeline P4, medium dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

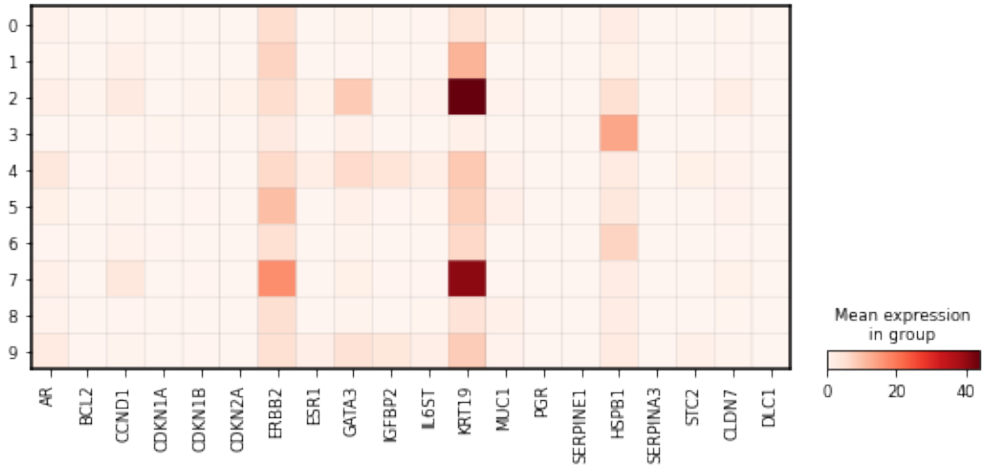


Figure C.11: Expression of genes related to breast cancer on Pipeline P5, medium dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

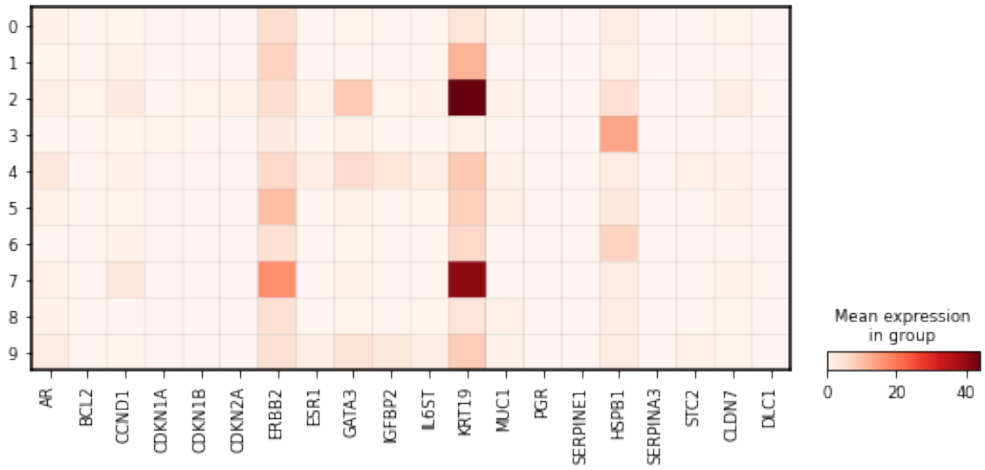


Figure C.12: Expression of genes related to breast cancer on Pipeline P6, medium dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

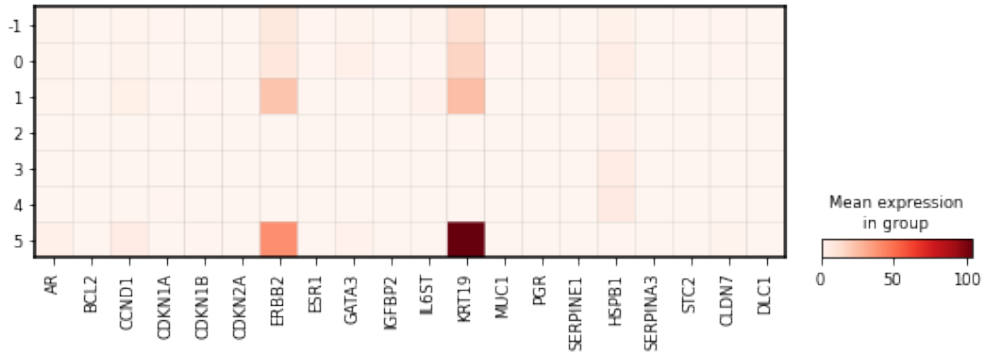


Figure C.13: Expression of genes related to breast cancer on Pipeline P8, medium dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

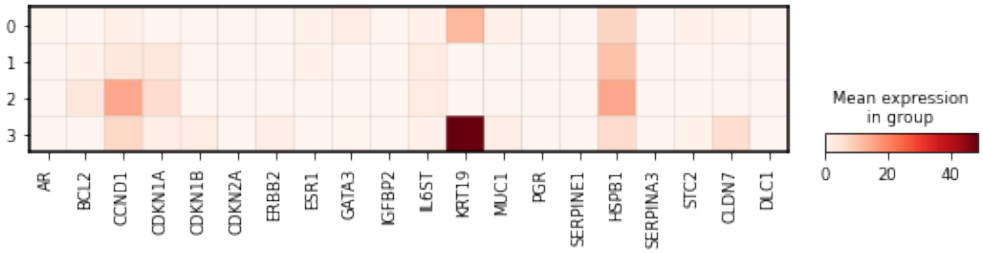


Figure C.14: Expression of genes related to breast cancer on Pipeline P2, large dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

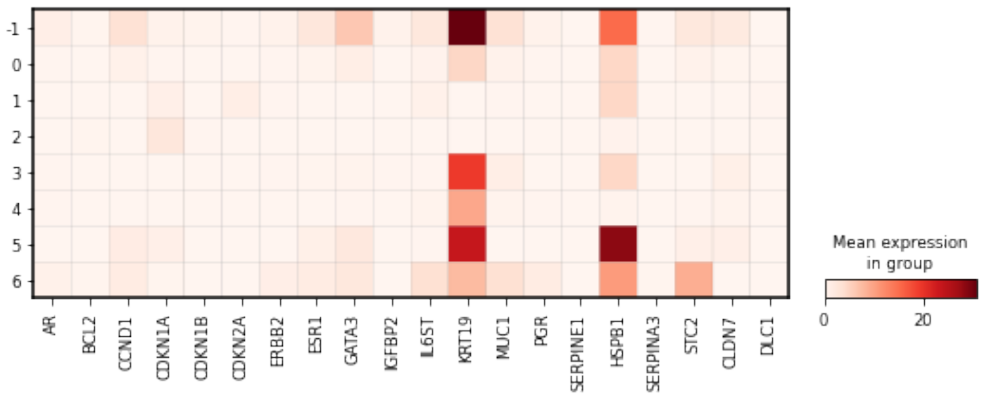


Figure C.15: Expression of genes related to breast cancer on Pipeline P4, large dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

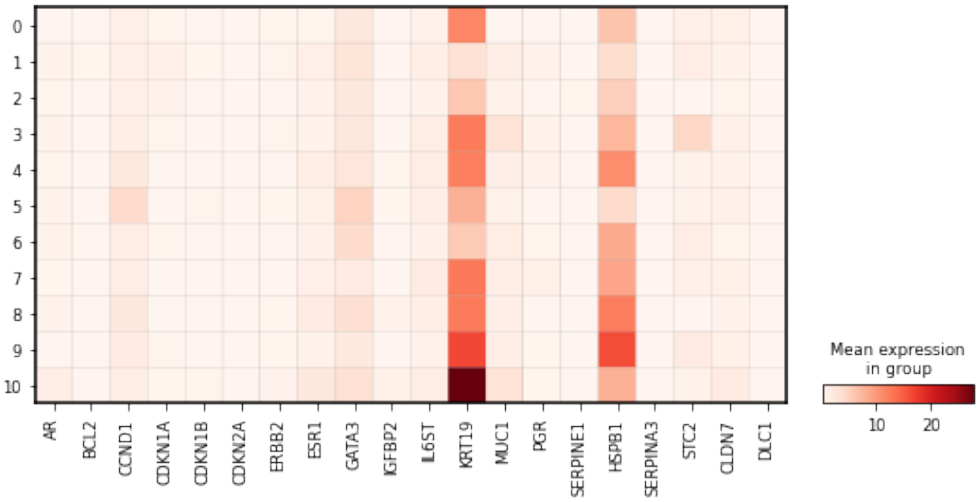


Figure C.16: Expression of genes related to breast cancer on Pipeline P5, large dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

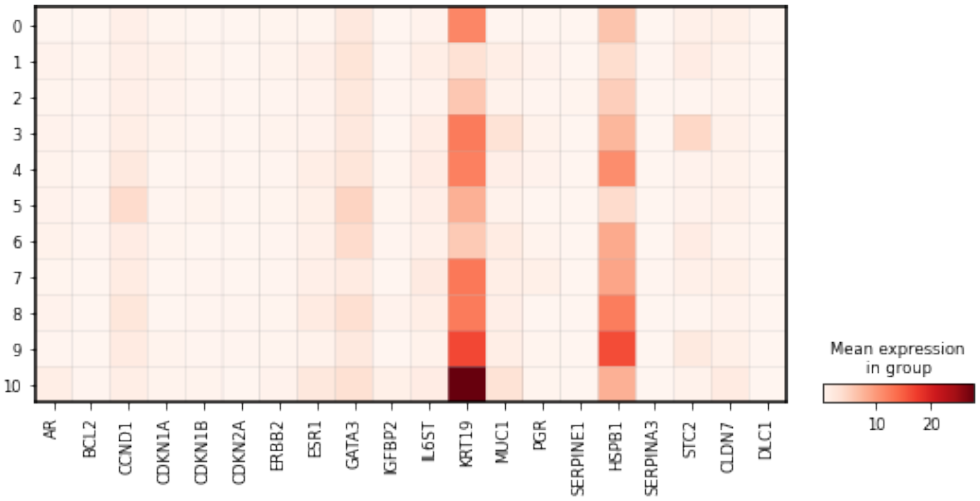


Figure C.17: Expression of genes related to breast cancer on Pipeline P6, large dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.

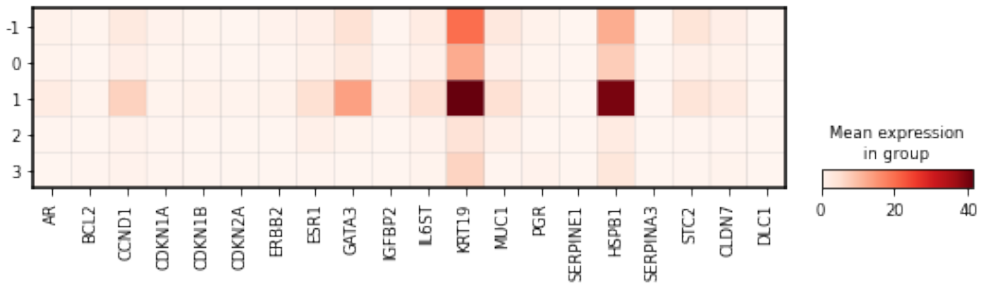


Figure C.18: Expression of genes related to breast cancer on Pipeline P8, large dataset. Rows represent clusters, and columns represent genes known to be related to breast cancer. Each cell represents the mean expression level of a gene in the cluster.