University of Cape Town

Course Code

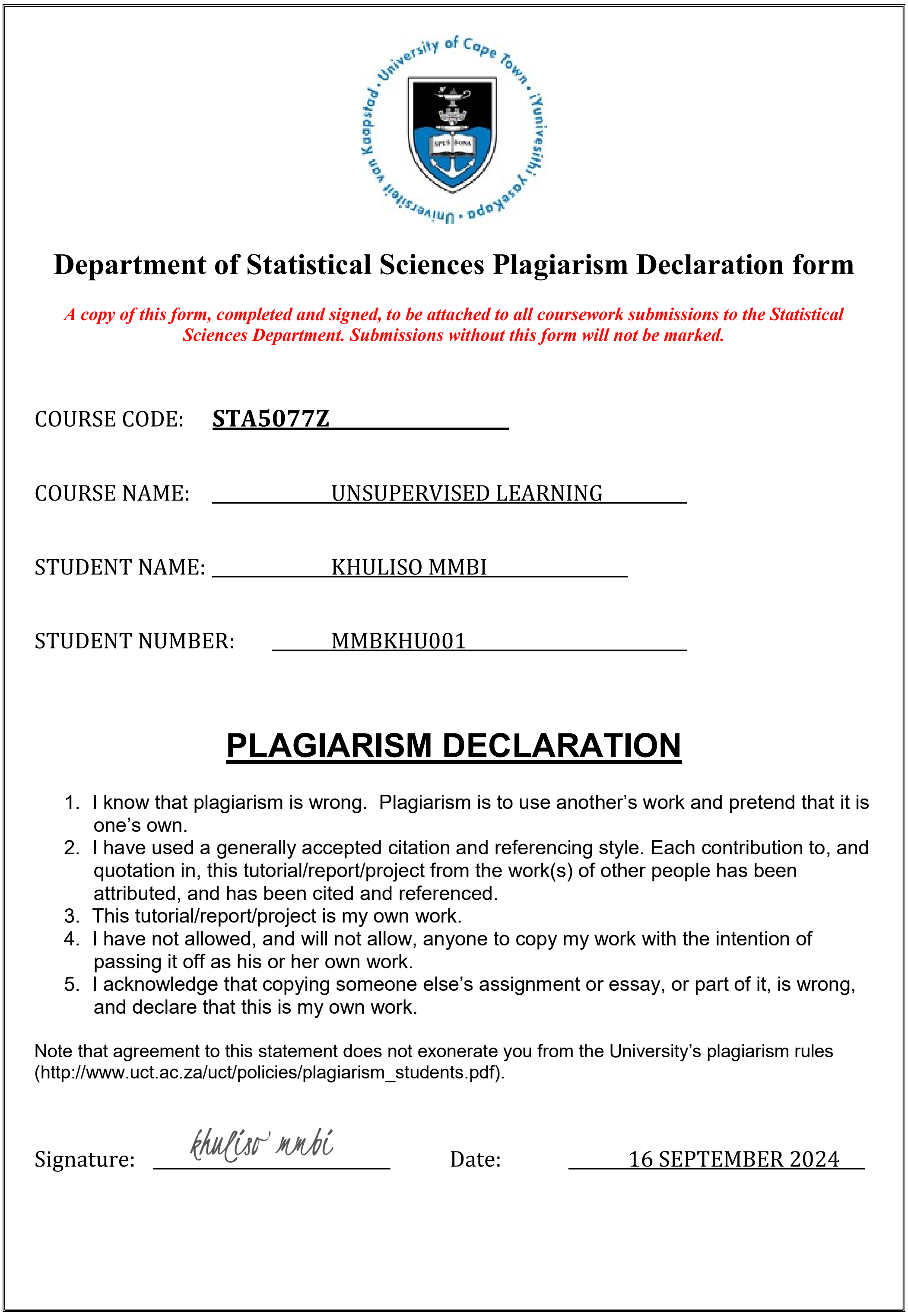
STA 5077Z

Dimensional Reduction Techniques

Author:
Khuliso Mmbi

Student Number:
MMBKHU001

October 14, 2024



Department of Statistical Sciences Plagiarism Declaration form

A copy of this form, completed and signed, to be attached to all coursework submissions to the Statistical Sciences Department. Submissions without this form will not be marked.

COURSE CODE: **STA5077Z**

COURSE NAME: UNSUPERVISED LEARNING

STUDENT NAME: KHULISO MMBI

STUDENT NUMBER: MMBKHU001

PLAGIARISM DECLARATION

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used a generally accepted citation and referencing style. Each contribution to, and quotation in, this tutorial/report/project from the work(s) of other people has been attributed, and has been cited and referenced.
3. This tutorial/report/project is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.
5. I acknowledge that copying someone else's assignment or essay, or part of it, is wrong, and declare that this is my own work.

Note that agreement to this statement does not exonerate you from the University's plagiarism rules (http://www.uct.ac.za/uct/policies/plagiarism_students.pdf).

Signature: khuliso mmbi

Date: 16 SEPTEMBER 2024

Contents

1 Introduction

Unsupervised learning is an important part of machine learning used to uncover hidden insights from unlabelled data. It can be divided into three main sections: clustering, association rules, and dimensionality reduction.

The aim of this assignment is to utilize dimensionality reduction techniques to analyze two datasets. The first part focuses on examining the flow cytometry dataset to visualize various cell types, while the second part aims to predict tuberculosis (TB) progression using protein expression data. Through these sections, this assignment seeks to demonstrate the practical applications of dimensionality reduction in biomedical research and its significance in improving data interpretability and predictive accuracy.

2 Flow Cytometry Data

Flow cytometry is an important tool in cell biology that is used to analyze the flow of single cells by measuring visible light scatter and fluorescence to characterize cell types(McKinnon, 2019, p. 2). However, the high dimensional nature of the data makes it hard to interpret. The aim of this section is to utilize various dimensionality reduction techniques to effectively visualize the data from flow cytometry experiments.

2.1 Explanatory Data Analysis

2.1.1 Data description

The flow Cytometry dataset has 100000 observations across 19 features with no missing values and no duplicates. The features include:

- Major cell type: Comprised of the primary cell type
- Minor cell type: Subset of the major cell
- Marker Expression levels: the rest of the columns represent the expression levels of different markers

2.1.2 Summary statistics

Table 1 below shows the summary statistics for all the numeric variables:

Key insights:

- The maximum expression level is 7.6033 for ccr7 and the minimum is -0.4027. This is an indication of the variability in the dataset.
- There is a relatively skew distribution amongst most features which can be seen by mean and median values being close to each other, for instance cd161(mean of 2.06) and median of 2.4, however, other variables such as cd8 and cd4 show a slightly skewed distribution.
- Overall, the range in these summary statistics, with some values low and other low, reveals a insight into the dynamics of the immune cell dynamics.

Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
cd3	-0.4027	4.3669	4.8852	4.5670	5.3026	7.1827
cd4	-0.4140	1.9640	2.8440	3.3960	4.9420	7.1370
cd8	-0.3637	1.8698	2.4764	3.1307	4.7643	7.0217
cd19	-0.3628	1.6720	2.0669	2.2479	2.4983	6.8787
cd20	-0.4120	1.6700	2.0640	2.2490	2.5040	7.0220
cd16	-0.4000	1.6770	2.0800	2.2020	2.5250	7.1390
cd56	-0.4401	1.6533	2.0401	2.1482	2.4541	7.4206
cxcr3	-0.0282	2.5565	2.9843	2.9816	3.4091	6.7403
ccr6	-0.3986	1.6457	2.0229	2.0813	2.4162	6.5758
ccr4	-0.2574	1.6398	2.0179	2.0480	2.4059	5.8497
igd	-0.6552	1.6627	2.0479	2.2021	2.4663	6.8922
tcrgd	-0.4521	1.6363	2.0059	2.0356	2.3862	6.9373
va7.2	-0.3955	1.6417	2.0150	2.0678	2.4001	6.7196
cd161	-0.5029	1.6373	2.0139	2.0659	2.4008	6.8901
cd27	-0.0702	2.7695	3.4201	3.5426	4.3269	7.2053
cd45ra	-0.0724	1.9816	2.6441	3.0230	4.1341	7.2001
ccr7	-0.1945	2.3696	3.2802	3.5231	4.8260	7.6033

Table 1: Descriptive Statistics of Immune Cell Data

2.1.3 Correlation Matrix

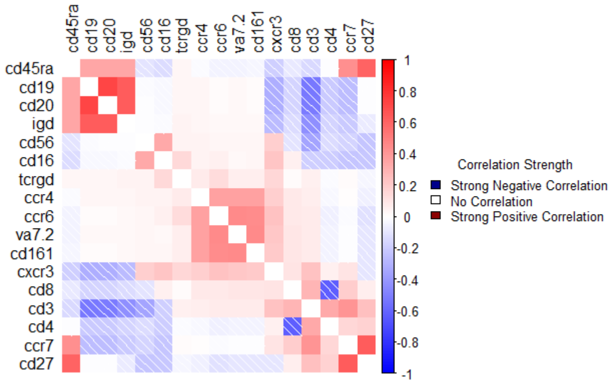


Figure 1: Correlation matrix

Figure 1 above summarises the correlation between the numeric variables with string correlation represented by a dark red colour and no correlation by a dark blue colour.Most variables exhibit weak correlations with each other, as indicated by the pale colours, however, there are some variables that are highly correlated such as igd and cd19 and cd20. On the other hand, some variables such as cd4 and cd8 have weak correlation.

2.1.4 Visualisation of‘ cell types

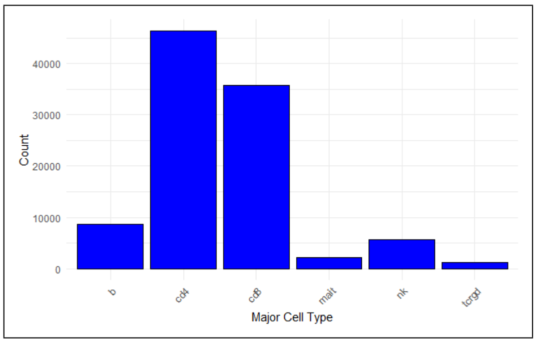


Figure 2: Minor Cell type

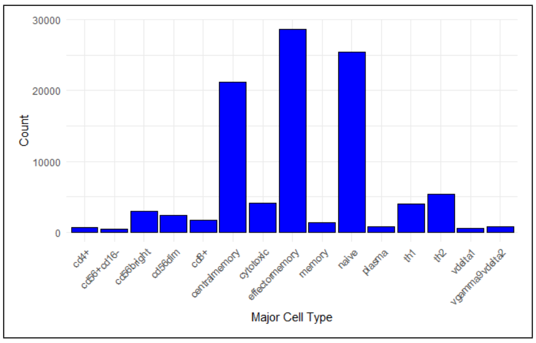


Figure 3: Major cell type

The histograms above show the distribution for the Major and Minor cell types, both of which shows uneven distributions. From Figure 2, depicting Minor cell types, categories such as cd4 and cd8 have noticeably high values, while other cell types like mait and tscm have low values. The Major cell type graph, Figure 3, has high values for effector memory, central memory, and naive cells, contributing the most to the uneven distribution of the data.

2.2 Dimension Reduction

Dimension reduction is an unsupervised learning technique that is used to represent a dataset using lower dimensions (number of features) whilst still capturing the data’s original properties (Jolliffe and Cadima, 2016). This section will explore the different dimension techniques using the Flow Cytometry Dataset namely Principal Component Analysis, Self Organising Maps, t-SNE and Autoencoders.

2.2.1 Distribution of the datasets

Figure 4 below shows how the data is distributed. There are 6 major cell types made up of the different minor cells, the highest is cd4 and cd8, so we expect them to be more visible when we apply dimensional techniques. For minor cells, naive, effective memory and cd8+ seems to be more dominant. therefore when we visualise the minor cells we expect to see 15 clusters. In all the 3 techniques to be explored in the following sections, the aim is to see which technique is able to clearly cluster the major cells into 6 clusters and the minor cells into their respective clusters as well and the contribution of the expression levels.

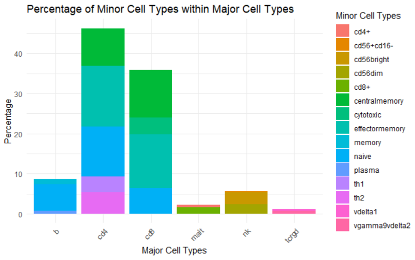


Figure 4: Correlation matrix

2.2.2 Principal Component Analysis

Principal Component Analysis (PCA) is a linear dimensionality reduction that works by finding a low-dimensional representation of the data set that retains as much as possible of the variation, thereby summarizing the data effectively and while minimizing loss of information (James et al., 2013, p. 375). The average correlation between all the numeric variables was 0.0375, which is significantly less than 0.3. This suggests that the variables in our dataset are not strongly linearly correlated. Despite this, PCA was applied with scaling off as the observations were already in the same scale as shown in the summary statistics above.

Parameter optimisation

- The scree plot below in Figure 5 below shows the variance for all the 17 components found on PCA. The optimal components were found to be six, when using the idea that all variables equally contribute to the variance. However, for this paper, we will focus on the first 2 components, PCA1 and PCA2 of which together, they explain 35.61% of the total variance. In order to get to 80% explained variance, more components are needed and therefore we can expect some information to be lost.

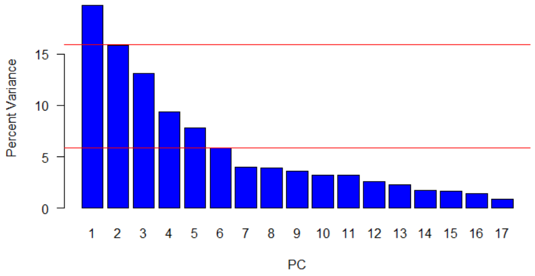


Figure 5: Scree plot

Run time analysis

The run time for PCA was PCA 0.2778 seconds.

Cell separation

- The plots below illustrate the separation of cells achieved when using PCA. For the major cells in Figure 6, some of the major cells such as b, cd4 and mk are well separated, however, the significant overlap between all the cells could suggest that PCA may not be fully capturing the underlying variance in the data.
- For minor cells in Figure 7, the separation is more distinct in cells such as cd8 and cd4, showing that PCA can differentiate between these cell types. However, for some cells such as as central memory and cytotoxic cells, the boundaries are not as clear, which might indicate that more complex structures or relationships exist between these cell types that PCA alone may not fully reveal.
- Overall, the overlaps in both the major cells and the minor cells suggest that PCA might be missing the nonlinear differences that exist between these cells.

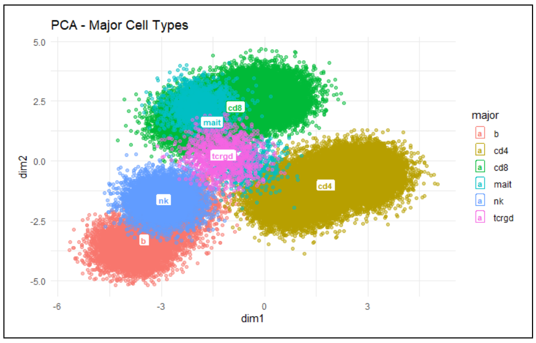


Figure 6: Major Cell type

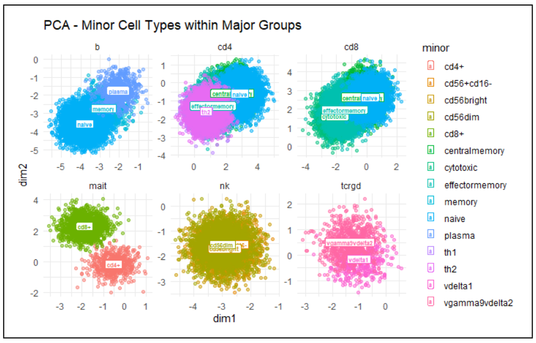


Figure 7: Minor cell type

2.2.3 t-Distributed Stochastic Neighbor Embedding (t-SNE)

The second technique explored was the t-SNE technique which is a non linear dimensionality reduction technique. t-SNE reduces the number of dimensions in the data while keeping similar points close to each other. This technique is useful for visualizing complex data that doesn't naturally split into distinct groups.

Parameter Optimisation

Grid search was used to find the optimal parameters of which the values were:

- dims = 2
- perplexity = 30
- max iter = 1000

Run time analysis: The run time for the model was 651.6807 seconds

Cell separation

- The major cells in Figure 8 are well separated into their distinct clusters. This suggests that t-SNE is was effective in capturing the non-linear relationships in the data.
- From the minor cells, Figure 9, some cells such as cd8, cd4 and memory are well separated whist cells such as effectormemory and th2 are not well separated. This indicates that t-SNE is unable to distinguish between cells that may have similar characteristics.

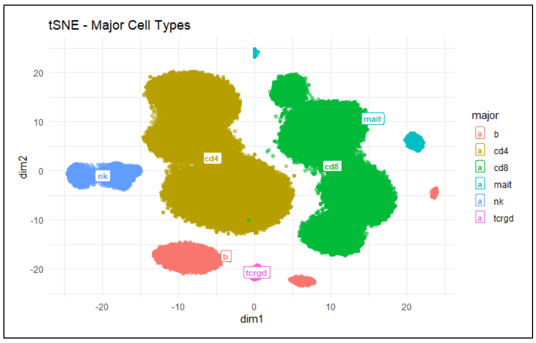


Figure 8: Major Cell type

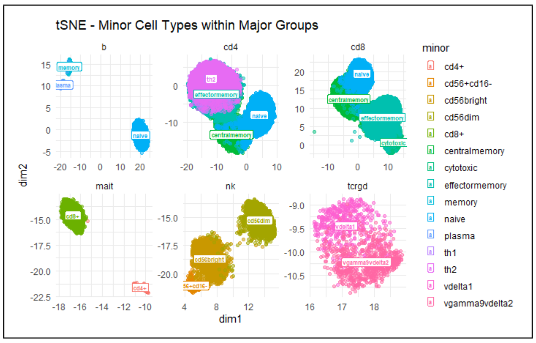


Figure 9: Minor cell type

2.2.4 Autoencoders

Autoencoders are a type of neural network that reduce dimensionality by encoding data into a lower-dimensional latent space and then decoding it to reconstruct the original data. This process allows the model to capture both linear and non-linear relationships within the data, making it effective for compressing and reconstructing complex patterns.

Optimal parameters:

- Hidden Layers: hidden = 50, 20, 50
- Epochs 50
- Activation = Tanh"

Run time analysis

The run time was 66.09111 seconds.

Cell Separation

Major cells (Figure 10)

- The separation between the clusters is not that distinct. Clusters such as cd4, cd8 and mat show a more visible overlap which suggests that the model struggled to differentiate between these groups. Although the model captured some structure of the original data, as we can see the different groups, the overlap indicates that it was not that effective in maintaining clear boundaries between the major cells.

Minor cells (Figure 11)

- The model also struggles to differentiate between the minor cells as seen by the overlap in all the cells. Cells such as cd4 and naive were poorly separated, as well as cd56bright and nk. This suggests that autoencoders was not able to distinguish finer relationships between cell types.

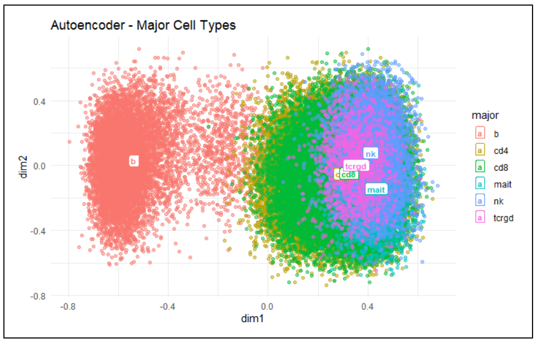


Figure 10: Major Cell type

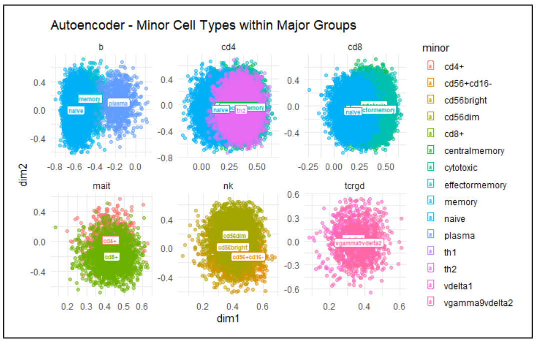


Figure 11: Minor cell type

2.2.5 Self Organising Maps

Self-organizing maps (SOM) are a non-linear dimensional reduction technique. They aim to preserve the structire of data by capturing non-linear relationships. This is achieved by adjusting neurons to fit the data points, thus creating a more simplified representation of the data (Vesanto & Alhoniemi, 2000, p.1).

For the SOM model, categorical variables were one-hot encoded and the data was standardized to ensure accuracy. Optimal parameters were determined using grid search, with the following values being selected:

Parameters:

- Learning rate (alpha): 0.05 - 0.01
- Iterations : 1000
- SOM grid : 4 x 4 nodes with hexagonal typology

Training duration and Counts plot

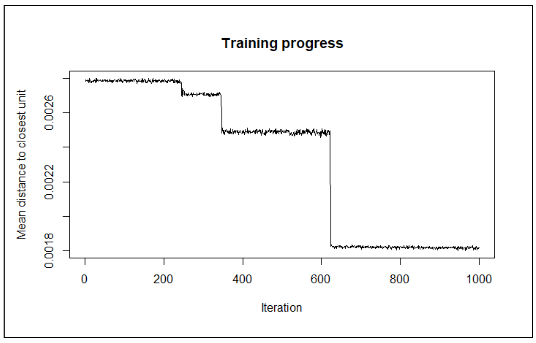


Figure 12: Training duration

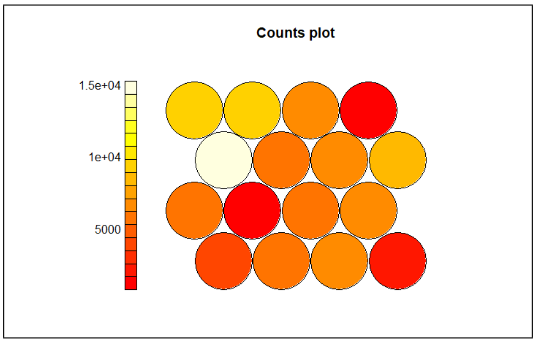


Figure 13: Counts plot

- Figure 12 above shows the training progress using the above parameters. The model converged at around 650 iteration as indicated by the mean distance to closest unit not changing significantly after this point which suggests that the SOM has reached a stable point.
- Figure 13 on the other hand shows the distributipn of data points in a 4 x 4 som grid. The total observations represented is 240000. From the plot, majority of the nodes have values between 5000 and 10000(1e+04) and with one node with more than 15000 data points. This could suggest an uneven distribution in the node as a result of the model trying to preserve the structure of the data.

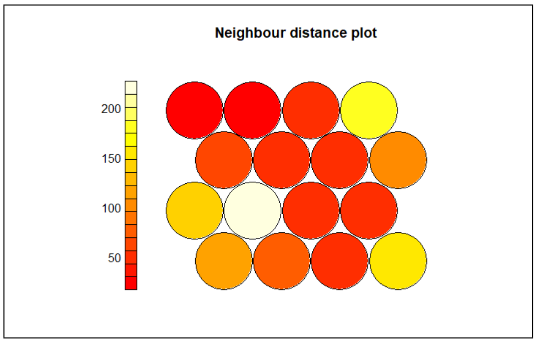


Figure 14: Neighbour distance

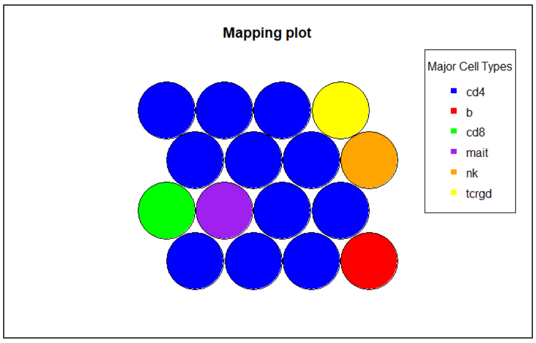


Figure 15: Mapping plot

- The neighbor distance plot (Figure 14) illustrates the distances between the nodes, with red indicating nodes that have shorter distances and may contain similar data, while yellow indicates the dissimilarity of the data points within those nodes. In the plot, the red nodes are closer to each other, which indicates well-defined clusters in the data. In contrast, the yellow nodes, which are further apart, show that the model successfully separated them from dissimilar data. There is also an overall good separation between the yellow and red nodes, suggesting that there is less overlap in the data.
- The Mapping plot in figure 15, shows how the data is mapped in the SOM grid. The are a lot of red nodes which suggests that the majority of the data is clustered in those areas. The other colours such as yellow, orange and green are a representation of the distinct clusters that are less frequent, however, there are separated from the other nodes. This is a good indication that SOM was able to differentiate the different classes within the data effectively.

Quality Plot

- The quality plot (Figure 16) shows the quantization error for each node. The red nodes, shows low quantization error where SOM is able to fit the input data, whilst yellow and white nodes show a higher quatization error. From the plot, there is a mixture of performance across the map, with some areas (red) performing well and others (white/gray) showing higher error and room for improvement. The variation suggests that while the SOM has successfully mapped certain areas of the dataset, other regions might need further tuning or a more complex model to reduce the quantization error.

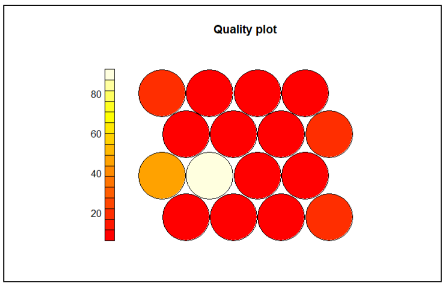


Figure 16: Quality Plot

2.3 Evaluation

Method	Silhouette_Major	Silhouette_Minor	DB_Index_Major	DB_Index_Minor	Runtime
PCA	0.359	-0.345	1.386	7.905	0.278
t-SNE	0.237	-0.155	1.305	11.337	651.681
Autoencoder	-0.041	-0.217	7.519	20.422	66.091
SOM	0.311	-0.076	0.595	NaN	94.539

Table 2: Performance of Various Methods

Table 2 above shows how the different dimensionality reduction technqinues performed measued by the Silhoette score, DB Index and the run time.

PCA

- Silhoette scores - the model managed to separate the major cells (0.359), however, the negative score for minor (-0.345) shows that the model could differentiate between the minor cells. This could be due to it's inability to capture non-linear relationships.
- DB index - The minor clusters were less compact as seen by the high DB score (7.905) as compared to the major cells (1.386)
- The run time was 0.278, which is effective given the observations in our dataset.

t-SNE

- Silhoette scores - The model performed moderately for for major cells (0.237) and poor performance for minor cells (-0.155). Although, this is better than for PCA, the model struggled to clearly separate the cells.
- DB Index -t-SNE achieved the best (lowest) DB index for major cells (1.305), indicating well-defined clusters. However, it performed poorly for minor cells (11.337), suggesting overlapping or poorly defined clusters for cell subtypes.
- Run time - the run time of 651.681 was the lowest, which could show that t-SNE does not have work well with large datasets.

Autoencoders

- Siloette score - The model was unable to separate both the major cells and the minor cells. This can be seen by the negative silhoette score of -0.041 and -0.217 respectively.
- DB Index - The high DB index values for both major (7.519) and minor (20.422) cells indicate that autoencoder-based dimensionality reduction resulted in overlapping or poorly defined clusters.
- Run time was 66.091 which was moderately fast, which could indicate a balance between performance and cmputational costs.

SOM

- Silhoette scores - The model performed well for major cells (0.311) and showed the best performance among all methods for minor cells (-0.076), though still negative. This suggests that SOM was most effective in preserving both broad and fine distinctions between cell types.

- DB Index - the DN index for the major cells was 0.595 which was the lowest, this indicates that the clusters are well defined and separated. However, the model could not produce any results for minor cells, which may indicate that some clusters may have contained enough data sets as supported in the counts plots above
- Run time of 94.539 was good, which could indicate also a good balance between performance and computational costs.

Overall Evaluation

PCA

- The model was the best in terms of run time (0.278 seconds) and also in terms of separating the major cells; however, it struggled with separating the minor cells. This suggests that PCA may be good when working with data were time is crucial.

t-SNE

- t-SNE was able to separate the major cells and also managed to separate the minor cells better than PCA. Due to the run-time, the model may not be best when working with large datasets and run time is a constraint.

Autoencoders

- The model struggled to separate both the minor and major cell types. The limitations could have been in the parameters chosen.

SOM

- SOM showed the best performance amongst all the other models, especially in separating the major cell types. The model may be best when working with datasets where a detailed or clear distinction of variables is important.However, compared to PCA and t-SNE, interpretation may be less informative.

2.4 Conclusion

In conclusion, the evaluation showed the strength of each dimensionality reduction technique and the choice depends on the requirements of the analysis. PCA was best used when quick and broad-level results are required. For a more detailed analysis with moderate computational costs, SOM proved to be the best. On the other hand, where run time is not a constraint and a distinct differentiation of variables is required, then t-SNE would be the best choice. The poor performance of autoencoders however, could suggest that they may not have been well-suited for this particular dataset without additional tuning.

3 Predictive Modelling of TB Progression Using Protein Expression Data

Predicting the progression of tuberculosis (TB) is important for timely intervention and treatment. This sections uses the protein expression data from a cohort of individuals, with the aim of differentiating between those who developed TB and those who did not within a year. This will be achieved by using various cross-validation techniques and dimensionality reduction methods to evaluate how these approaches influence the predictive accuracy of a logistic regression model with LASSO regularization. The results of this analysis will contribute to a better understaning of TB and its underlying causes.

3.1 Explanatory Data Analysis

3.1.1 Data description

The Protein expression dataset has 4 features:

- Sample id: Unique identifier for each individual.
- tb: TB status (yes or no).
- protein: Protein name.
- value: Protein expression level.

This dataset contains 406,560 observations, with no missing values and no duplicates. There are 154 distinct sample IDs and 154 unique protein names. The distribution of TB status is as follows: 74,560 individuals tested positive for TB, while 132,000 individuals tested negative. The scatter plot below in figure 17 shows the distribution of the value: protein expression level. The Protein expression levels range from a 1.244 (min) to 5.289 (max) with a mean and median of 4.099 and 4.084 respectively. This indicates that the data is slightly skewed to the right. There is a tendency yowards higher values as indicated by the Q1 of 3.8888 and Q3 of 4.303.

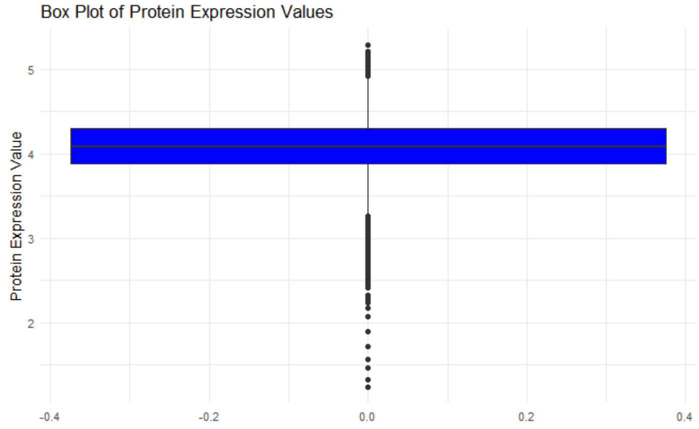


Figure 17: Protein expression levels

3.2 Data Processing

The dataset has 154 individuals with 104 TB Negative (No) and 50 TB Positive (Yes). Given the imbalance, 3 methods will be employed to balance the data:

- **Not balanced:** The original dataset used as it is
- **Manually balanced:** Minority class (TB Positive) was over-sampled to match the majority class (TB Negative).
- **SMOTE:** Through generating synthetic samples from the minority class and balancing it without duplicates.

In addition, the structure of the protein dataset was further analyzed to ensure accuracy of the results and there were no missing values and no duplicates. Character columns were converted to factor variables in preparation for modelling. Further more, the dataset was reshaped from a long format to a wide format. The data was split into training and testing set, of which 80% was for training and 20% for testing.

3.3 Dimension reduction techniques

3.3.1 Pre - Processing

Before applying any dimensional reduction techniques, the logistic regression model with LASSO regularisation was explored. Figure 18 below shows the AUC of 0.735 which was achieved by the model.

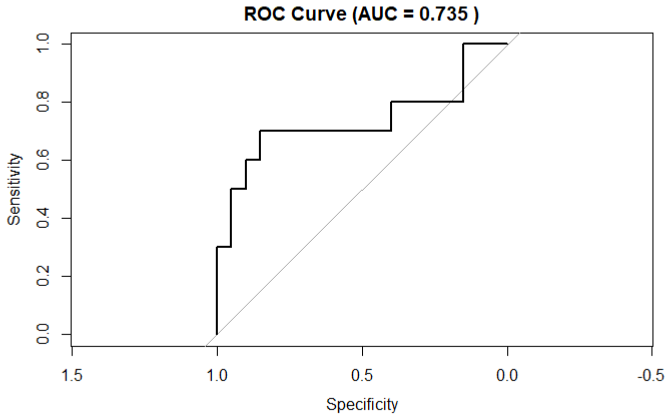
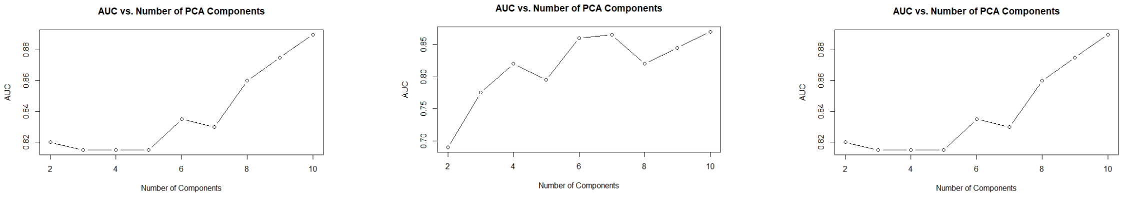


Figure 18: AUC for Logistic regression with Lasso

3.3.2 Principal Component Analysis (PCA)

The first technique employed was the PCA. This was done correlation matrix to ensure equal weight between the variables. PCA was explored using different numbers of components to optimize the Area Under the Curve (AUC) for a logistic regression model with LASSO regularization.

The three diagrams in figure 19 shows the number of components and their respective AUC for PCA when using the different balancing techniques. The optimal number of components for unbalanced data and SMOTE was 10 however the manually balanced had 7 as the optimal number



Unbalanced data Manually balanced SMOTE

Figure 19: AUC vs number of components using different balancing techniques

of components. The logistic regression model was fitted using these optimal number of principal components. The table below shows the performance metrics of PCA achieved for each technique.

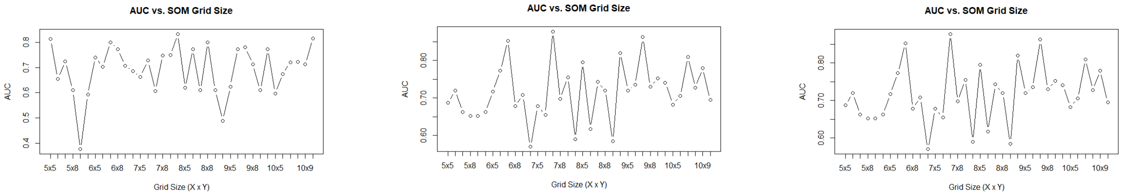
Method	**AUC**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**FP**	**FN**
Unbalanced	0.84	0.83	1.00	0.50	0.67	0	5
Manually Balanced	0.835	0.80	0.70	0.70	0.70	3	3
SMOTE	0.895	0.83	1.00	0.50	0.67	0	5

Table 3: Performance Metrics for Different Balancing Methods

From the Table 3 above , the PCA seems to be over-fitting when using the unbalanced dataset and SMOTE, this is evident from the Precision of 1 and a low recall of 0.5. However, for the manually balanced dataset, the precision was 0.7 which shows that the model is able to distinguish between positives and negative TB classes. The AUC was high for both techniques, with SMOTE being the highest (0.895). Overlall the manually balanced PCA produced the best results of the three as can be seen by all the metrics.

3.3.3 Self Organising Maps (SOM)

The SOM technique was explored using several grid sizes ranging from 5x5 to 10x9 to find the optimal grid size. AUC was used as a measure of performance as can be seen by Figure 20 below where the optimal grid size for unbalanced data is 7 x 10 and for manually balanced and SMOTE being 7 x 7.



Unbalanced data Manually balanced SMOTE

Figure 20: AUC vs Grid sizes using different balancing techniques

Table 4 summarises the performance metrics of PCA after fitting the optimal grid sizes on a logistic model with Lasso. The Manually balanced and SMOTE models had the same results for all the metrics. The model had a precision of 1 when using both techniques, however, only the unbalanced dataset had a higher AUC (0.857). Recall and F1 score were low when using both techniques.

Method	AUC	Accuracy	Precision	Recall	F1 Score	FP	FN
Unbalanced	0.857	0.723	1.00	0.20	0.33	0	8
Manually Balanced	0.68	0.70	1.00	0.10	0.18	0	9
SMOTE	0.68	0.70	1.00	0.10	0.18	0	9

Table 4: Performance Metrics for Different Balancing Methods

3.3.4 T-Distributed Stochastic Neighbor Embedding (t-SNE)

The t-SNE technique was examined with perplexity values ranging from 5 to 9 to find the best setting for maximizing the model's AUC. A perplexity of 6 emerged as the optimal value for all three techniques. Figure 21 shows the AUC and Perplexity achieved. AUC values fluctuated across different perplexity settings, creating a peak in the graph. The highest AUC, 0.74, was recorded at a perplexity of 6, indicating that this was the ideal setting for t-SNE in this case.

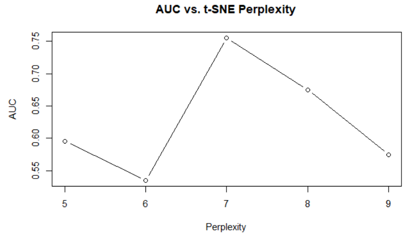


Figure 21: AUC vs Perplexity using different balancing techniques

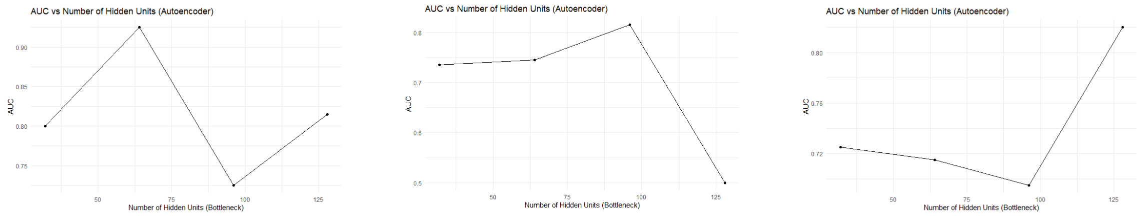
The Logistic model with Lasso was done for all the datasets, and all the models had the same results for the performance metrics. Table 5 shows the results. The performance metrics for the t-SNE algorithm with a perplexity of 6 indicate that while the t-SNE algorithm achieved a reasonable AUC of 0.55, the overall performance in terms of accuracy (0.63), precision (0.45), recall (0.50), and F1 score (0.48) suggests that there is room for improvement. The relatively low precision and recall indicate that the model may struggle with correctly identifying positive cases, leading to a higher number of false positives (6.00) and false negatives (5.00). Overall, these metrics highlight the need for further optimization to improve the model's ability to correctly identify positive cases.

AUC	Accuracy	Precision	Recall	F1 Score	FP	FN
0.55	0.63	0.45	0.50	0.48	6.00	5.00

Table 5: t-NSE metrics

3.3.5 Autoencoders

Autoencoders were examined with various hidden layer sizes to identify the best configuration. Figure 22 below illustrates the different hidden layer sizes along with their corresponding AUC values. The ideal hidden layer size was determined to be 96, achieving an AUC of 0.915 for teh unbalanced dataset, for Manually balanced, hidden layer size was 94 with an AUC of 0.82 and lastly SMOTE had 128 hidden layer size and an AUC of 90. The logistic regression model with Lasso was then re-evaluated using these optimal parameters, and Table 6 presents the performance metrics.



Unbalanced data Manually balanced SMOTE

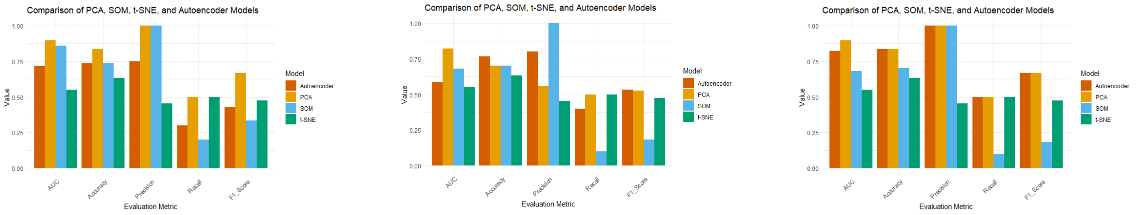
Figure 22: AUC vs Hidden units different balancing techniques

From table 6, the unbalanced dataset, the model has a high AUC of 0.83 and accuracy of 0.80 but suffers from lower recall (0.60), indicating missed positive cases (4 FN). In the Manually Balanced dataset, accuracy improves to 0.83, precision reaches 1, but recall drops to 0.50, meaning more false negatives (5 FN). When sing SMOTE, the AUC slightly decreases to 0.755, accuracy remains 0.833, precision is perfect (1.00), but recall is still low (0.50) with the same false negatives (5 FN).

Method	**AUC**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**FP**	**FN**
Unbalanced	0.83	0.80	0.75	0.60	0.67	2	4
Manually Balanced	0.72	0.83	1	0.50	0.67	0	5
SMOTE	0.755	0.833	1.00	0.50	0.67	0	5

Table 6: Performance Metrics for Different Balancing Methods

3.4 Comparison and Analysis



Unbalanced data Manually balanced SMOTE

Figure 23: Metrics

The aim of this section is to evaluate the performance of the dimensional reduction techniques employed in the previous sections. This will be done using Figure 9999 above, where each model will be evaluated on its performance in the 3 datasets (unbalanced, manually balanced and SMOTE).

Unbalanced

- All the models achieved high AUC of above 0.7 except for t-nse. However, it seems like PCA and SOM might have over-fitted, this is evident from the Precision of 1 which is accompanied by a low recall. All the models had high accuracy, however, the recall and F1-score were low which indicates that there might be some difficulties in consistently identifying true positive cases. Overall, Autoencoders proved to be the most reliable technique, followed by t-SNE which is highly accurate and precise. SOM and PCA did not perform well given their precision and recall values.

Manually balanced

- PCA seems to have improved as can be seen by the improvement in the Precision, however, the Recall remained low. t-SNE had the same results which suggests that the mode performed the same way. The AUC, Recall and F1-score for SOM decreased and the precision still remained at 1 suggesting that the model is still over-fitting but as compared to the unbalanced data, the model worsened from being manually balanced. For Autoencoders, the AUC, all the metrics improved except for AUC which slightly decreased to just over 0.5. Overall, both PCA and Autoencoders performed well best in terms of AUC, Accuracy and F1 score.

SMOTE

- When using SMOTE, PCA, Som and Autoencoders achieved a precision of 1, which suggests they are overfiiting. Additionally, the Recall and F1 score were low. On the other hand, t-SNE still mainted the same metrics as from the other data sets. Overall, tnse seemed to be the best performer in this data set

Summary

- PCA performed well on the manually balanced data
- SOM overfitted in all the datasets, however all the other metrics were high in the unbalanced dataset
- t-SNE achieved the same results across all the different methods
- Autoencoders over-fitted on the SMOTE dataset, however, it performed well in the unbalanced and manually balanced dataset.

3.5 Model Evaluation

This section will evaluate each technique by looking at the strength and weaknesses and why they the way they did in the different datasets.

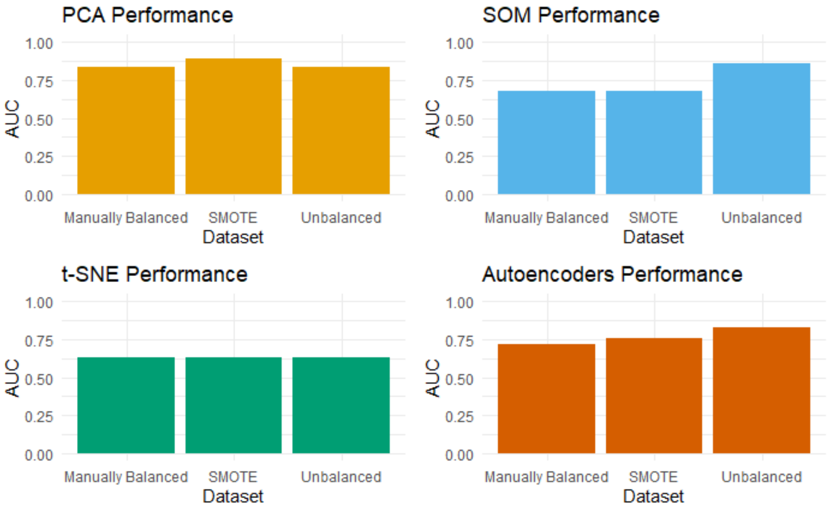


Figure 24: AUC

PCA

- PCA consistently performed well across all three datasets, with AUC values close to 0.75. This suggests that PCA preserves the variance necessary for the classification task, even when the data is imbalanced or synthetically balanced using SMOTE. The strength of PCA lies in its simplicity and efficiency, but one of its limitations is that it may not capture non-linear relationships, which could be important for TB progression prediction.

SOM

- The SOM had high AUC for the Unbalanced dataset, which indicates that it might not be robust where rhe distribution of the data is skewed.

t-SNE

- t-SNE showed a moderate performance across all the datasets, with a consistent AUC of above 0.5. This showed that it is able to effectively reduce the data to a lower-dimensional space regardless of whether the data is balanced or not.

Autoencoders

- Autoencoders performed well in all datasets, with high AUC values. This indicates that it can capture the linear and non-linear patterns in the datasets. The drawback for it was that the run time was long and it required alot of parameter tuning and network architecture.

3.6 Conclusion

Among the four methods, PCA and Autoencoders showed the most promise in terms of preserving relevant data structure and providing strong predictive performance, as reflected by their high

AUC values. t-SNE and SOM, while useful for visualization and capturing non-linear structures, did not perform as well in the predictive modeling task.

4 Conclusion

In conclusion, the unsupervised learning techniques used in this assignment, including PCA, t-SNE, Autoencoders, and SOM, each showed distinct strengths and weaknesses depending on the task and dataset used. These methods highlighted the importance of dimensionality reduction in simplifying complex data, improving interpretability, and enhancing predictive accuracy in biomedical research

5 References

1. Alizadehsani, R., et al., 2013. A data mining approach for diagnosis of coronary artery disease. *Computer Methods and Programs in Biomedicine*, 111(1), pp.52-61.

2. Jolliffe, I.T. and Cadima, J., 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), p.20150202. doi:10.1098/rsta.2015.0202.

3. James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An introduction to statistical learning: with applications in R*. Springer Texts in Statistics.

4. McKinnon, K.M., 2019. Flow Cytometry: An Overview. *Vaccine Branch, National Cancer Institute, National Institutes of Health, Bethesda, MD.* Available at: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5939936/pdf/nihms918259.pdf` [Accessed 26 September 2024].

5. Vesanto, J. and Alhoniemi, E., 2000. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3), pp.586-600. Available at: `https://pubmed.ncbi.nlm.nih.gov/18249787/` [Accessed 13 October 2024].