
# Using Transformer-based Neural Networks for classifying cellular states in Glioblastoma

## An assessment of scBERT in annotated cellular state classification of Glioblastoma

Master's thesis in Complex Adaptive Systems

Ronja Hedberg

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2024

www.chalmers.se

Master's thesis 2024

# Using Transformer-based Neural Networks for classifying cellular states in Glioblastoma

An assessment of scBERT in annotated cellular state classification of Glioblastoma

RONJA HEDBERG


CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
*Division of Applied Mathematics and Statistics*
Chalmers University of Technology
Gothenburg, Sweden 2024

Using Transformer-based Neural Networks for classifying cellular states in Glioblastoma
An assessment of scBERT in annotated cellular state classification of Glioblastoma
RONJA HEDBERG



Co-supervisor: Alejandro Lozada Cortés, Mathematical Sciences
Supervisor: Rebecka Jörnsten, Mathematical Sciences
Examiner: Rebecka Jörnsten, Mathematical Sciences

Master's Thesis 2024
Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Using Transformer-based Neural Networks for classifying cellular states in Glioblastoma
An assessment of scBERT in annotated cellular state classification of Glioblastoma
RONJA HEDBERG
Department of Mathematical Sciences
Chalmers University of Technology

# Abstract

By taking inspiration from the progress made in Natural Language Processing with the use of Transformer-based Neural Networks, similar approaches have been proposed for single-cell RNA-sequencing data in hope of capturing complex gene-to-gene interactions. One such approach is the pre-trained single-cell bidirectional encoder (scBERT), whose architecture and pre-training follows its Natural Language counterpart, BERT. Unlike BERT, scBERT was pre-trained for masked gene expression prediction using single-cell datasets comprising over 1.5 million single-cell RNA-sequencing profiles. This thesis performs an initial assessment of the use of scBERT with novel single-cell data. In classifying annotated cellular states of Glioblastoma, the inclusion of scBERT showed overall limited advantages compared to using the gene expression directly. However, through the simulation of different scenarios, this thesis provides preliminary evidence in favor of the use of scBERT in the lack of ample signal (low number of expressed genes, and scarce number of training examples). This showcases the potential benefits of using the gene representations of massive single-cell Transformer-based models, especially when little information is available, which is frequently the case when working with in-house data or heavily underrepresented cellular states.

Keywords: Machine Learning, scRNA-seq, Transformer, Cellular states, Glioblastoma, Cancer, Natural Language Processing, Encoder.

# Acknowledgements

I would like to begin by thanking my supervisor, Professor Rebecka Jörnsten, who made this thesis possible by welcoming me into the research team, offering guidance, invaluable insight and inspiration.

I would also like to extend my gratitude to my co-supervisor, Alejandro Lozada Cortés, who has provided indispensable counsel and support for everything from programming to proof-reading, and who has made an amazing contribution as a sounding board.

A big thank you to Nelander Lab at Uppsala University, for allowing me to use their data for this project, with an extra thank you to Ida Larsson, who patiently answered every question in my emails.

Many thanks to Ziming Wang, who held an amazing crash course on the computer cluster that was used, and to Adam Malik, who showed me how the file format for the data works. Much time would have been lost without them.

Lastly I want to thank Arvid, who not only provided writing feedback but also ensured I didn't have too much blood in my caffeine stream.

Ronja Hedberg, Göteborg, May 2024

# Nomenclature

Below is the nomenclature of sets, parameters, and variables that have been used throughout this thesis.

## Sets

| | |
|---|---|
| $\mathcal{S}_k$ | The set of genes associated with cellular state $k$ |
| $\mathcal{A}_k$ | A random subset of the complement of $\mathcal{S}_k$ |

## Parameters

| | |
|---|---|
| $\mathbf{X}$ | Data matrix of gene expressions with dimensions $M \times N$ |
| $N$ | The amount of cells in the data matrix |
| $M$ | The number of genes in data matrix |
| $K$ | The number of possible cellular states |

## Variables

| | |
|---|---|
| $x_{j,i}$ | The expression of gene $j$ in cell $i$ |
| $m_{i,j}$ | The amount of class $i$ being classified as class $j$ |
| $c_i^k$ | The score of cellular state $k$ in cell $i$ |

| | |
|---|---|
| $C_i$ | The cellular state assigned to cell $i$ |

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

Glioblastoma is a fast growing and aggressive form of brain cancer that often spreads throughout the brain matter in a pattern similar to a spider web, which makes it very difficult to remove completely by surgery, since such an intervention would be detrimental for the brain [1]. In hope of being able to stop the spread of this type of cancer, clinical trials are conducted with agents[1] that target specific signalling pathways[2] to hinder tumour growth.

Glioblastoma is a heterogeneous form of cancer, which makes treatment difficult. There has been shown that four different cellular states drives this heterogeneity in malignant cells [3].

In recent years there have been great development within Natural Language Processing with the help of transformer neural networks, e.g. OpenAI's ChatGPT-4 [4] and Google's BERT [5]. These models have shown to be very versatile in their applications and suitable for a range of different language tasks. These base structures from transformer neural networks have shown potential in bio-related tasks as well [6], which due to their general nature could be used for a multitude of assignments.

This project focuses on the transformer model scBERT [6], its capacity to determine the cellular states that are specific for Glioblastoma cells, and evaluates how the encoder structure affects the performance of the model.

[1] Certain medicines, vitamins, minerals, or food supplements [2].

[2] Describes a series of chemical reactions in which a group of molecules in a cell work together to control a cell function, such as cell division or cell death [2].

# 2

# Theory

With the help of Next-Generation Sequencing, the production of substantial volumes of single-cell RNA-sequencing data has become increasingly feasible [7]. Taking inspiration from the achievements of transformer neural networks in Natural Language Processing, these data-intensive algorithms have started to be utilized within biology as well.

## 2.1 Single-cell RNA-sequence data and gene expression

All living things on Earth are made out of cells, which consecutively have sequences of DNA-bases that carry hereditary information known as genes [8]. Human cells, for example, are estimated to have between 20 000 and 25 000 different genes [8]. The activity status of the genes within a cell (i.e. which genes are active at any given time) can help characterize cellular states: inner configurations of the cell that can change over time [9]. Specifically in brain tumor cells, distinct cellular states have been found, making it valuable to study how they are formed and how they change in the context of cancer research [10].

Single-cell RNA-sequencing is a technique used to measure gene activity at a cellular level that allows studying cellular states. In a laboratory setting, individual cells are isolated from a tissue sample, and their mRNA captured. A following reverse transcription of the captured RNA yields complementary DNA, which is then tagged with a cell ID and amplified. The digital data from these cells is efficiently generated through the method of Next-Generation Sequencing (NGS), a technique capable of sequencing large volumes of DNA effectively [7].

A cell line, defined as a set of cells derived from a singular biological sample, and its gene expressions, can be represented by a two dimensional data matrix. One

dimension of such a matrix represents the genes, whose names are standardised to allow for cross-cell-line comparisons, while its other dimension represents the individual cells, and the numerical values of the entries represents the gene expressions where zero symbolizes inactivity. Due to the amount of inactive genes in the cell population, this matrix is also sparse.

Analysing the gene expressions within a cell makes it possible to classify them in different cellular states, depending on how the genes are expressed.

## 2.2 Transformer neural networks

Although machines have surpassed humans in calculation efficiency long ago, enabling computations that would have been infeasible if done by hand, mastering human language remains a challenge for them. While numerical calculations can be executed sequentially, natural language is less straight forward. Context is very important, for example. There is a large difference in the meaning of the word "rock" in "It was her favourite rock song" compared to "He found a rock at the beach". Furthermore, language is ever evolving with new words emerging and others fading away, the meaning of words might also vary depending on the speakers dialect. Add into this that there is 7 164 known languages spoken in the world, all with their own quirks and rules, and most can understand why natural language is a daunting task for a computer [11].

To start making language comprehensible for a computer one needs to begin by creating a mathematical representation of it. In the context of written Natural Language Processing (NLP), this is done by turning text into vectors. The first step in achieving this is tokenization, where the text is broken up in smaller parts. Numerical embeddings are then created by vectorizing the tokens, with for example word2vec, which maps the meaning of the words such that similar words have similar vectors. For example, with word2vec you can get the vector closest to the vector $vector(queen)$ by doing the following operations $vector(king) - vector(man) + vector(woman)$ [12].

In recent years, the transformer neural network has become a widely adopted architecture for Natural Language Processing [13]. As illustrated in Figure 2.1, its fundamental components compromise of encoder and decoder blocks.

Each block has a combination of positional encoding, attention heads, and feed forward networks. The idea is that the attention mechanism maps a query and a set of key-value pairs to an output, where the query, keys, values and outputs are all vectors [14]. An attention map is created by multiplying the query with the key, which lists which tokens are important for the query. The attention map in turn is multiplied with the values and produce a weighted sum of the embeddings based on the relevancy of the words [15]. The output from the attention mechanism is fed into a position-wise feed-forward network which consists of two linear, fully connected layers [14].

The encoder condense the important parts of the input, and the decoder use this knowledge from the encoder to make it into a comprehensible output.

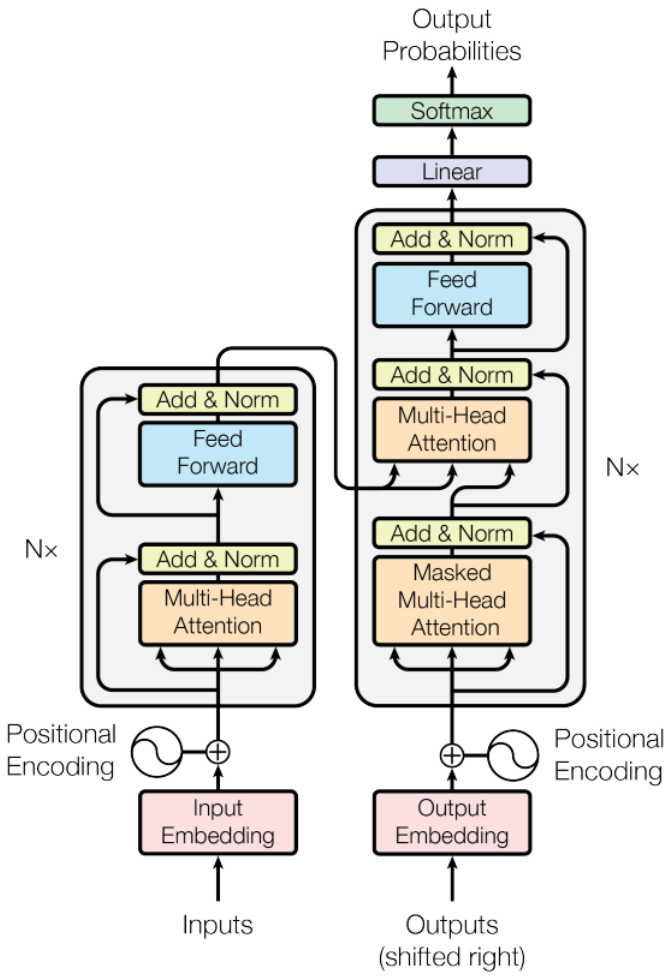


**Figure 2.1:** Transformer-model architecture with the encoder block on the left and the decoder block on the right from "Attention is all you need" [14].

To initiate the processing pipeline, the input is transformed into an embedding be-

fore being fed into the encoder. This entails representing the input through numerical vectors, for example via word2vec [12], accompanied by positional encoding to encapsulate information about the input token's absolute and relative positions [14].

The encoder then converts these inputs into tensors imbued with contextual information. The decoder has a Masked Multi-Head Attention layer that only allows it to consider the the tokens before the token it wants to predict, and not the ones coming after. The output from the Masked Multi-Head Attention gets normalized and sent forward to a regular Multi-Head Attention layer, that takes the output from the encoder which is used as the Key and Value and the output from the Masked Multi-Head Attention as Query Matrix. The decoder then predicts the next token based on previous tokens and the attention from the encoder, repeating the process until the end of the sentence has been reached, producing an interpretable output [14].

## 2.3 BERT: Bidirectional Encoder Representation Transformer

One example of a transformer architecture that has been successful in Natural Language Processing is the Bidirectional Encoder Representation Transformer (BERT) [5]. One thing that separates BERT from a normal transformer architecture is that it does not have a decoder component, meaning it is an encoder-only model. Additionally it is bidirectional, which means that in contrast to the original transformer it can take context from both sides of the token it wants to predict, instead of only seeing the previous tokens. Furthermore, BERT introduces two new tokens: CLS and SEP [5].
The CLS-token denotes the start of an input sentence, as well as capturing the overall representation of the input, which later can be used in a downstream classification tasks due to its position being fixed and its attention adapting depending on the task at hand. If the task it's trained for would be to automatically sort out offensive messages between players in an online game, the CLS-token would have more attention on, for example, derogatory language [16].
The SEP-token denotes the boundary between segments, for example between a question and the answer, allowing the model to accurately capture the relative positions between the segments in its positional encoding, as well as distinguishing the different parts of the input [17].

BERT consists of two pre-training stages, Masked Language Modelling and Next Sentence prediction, which allows it to gain a general understanding of Natural Language, that later can be fine-tuned for specific tasks. This means that instead of retraining the whole model for each specific task, a necessity with a regular transformer model, BERT gets fine-tuned into the specific task it is supposed to perform while the base model remains unaltered [5].

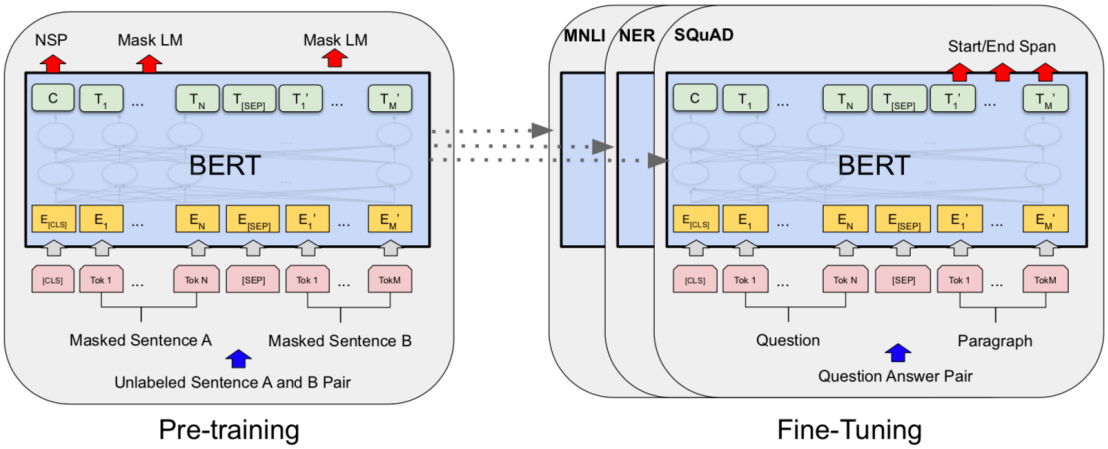


**Figure 2.2:** Pre-training and Fine-Tuning for BERT. The model architecture is the same apart from the output layers. Figure from "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" [5].

The masked modelling consists of three parts: a token can either be replaced by [MASK], a random token, or remain unaltered. Usually, 15% of the tokens get randomly chosen for the masked modelling, of which 80% get replaced by [MASK], 10% by a random token, and 10% remains unaltered [18].

If the fourth token in the input "This is my dog Bertil. He is a good boy." were to be chosen, the different options could would be:

"This is my [MASK] Bertil. He is a good boy."
"This is my 'empire' Bertil. He is a good boy."
"This is my dog Bertil. He is a good boy."

In next sentence prediction modelling, the goal is for the model to predict if the second sentence has any connection to the first. Consider these two examples:

"This is my dog Bertil. He is a good boy."
"This is my dog Bertil. It is sunny today."

In the first example the sentences are connected, unlike the second example.

Due to the pre-training phase described above, BERT has a deep contextual understanding of Natural Language. To be able to use the model for practical purposes, however, it needs to be fine-tuned. This is done by swapping out the input and output to those of the desired task and fine-tuning the parameters end-to-end, as illustrated by Figure 2.2. This is much less computationally expensive than retraining the full model, and has been shown to achieve high performance [5].

## 2.4 scBERT: From Language to Biology

Given the success of transformers in Natural Language Processing, there has been interest in applying these architectures to biological data [19]. By conceptualizing cells as sentences and gene expressions as tokens, the goal of such an approach is to discover patterns between gene co-expressions in the data.

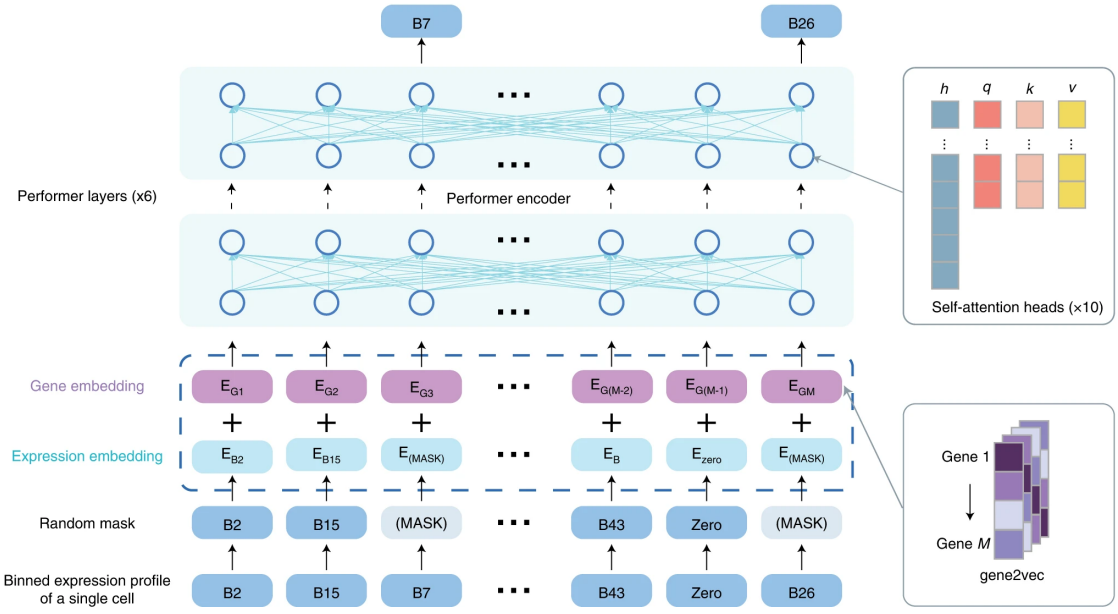


**Figure 2.3:** Illustration of the embedding process fed into scBERT and output of the learned representations. The non-zero values of the binned expressions gets randomly masked, and then gets added to the corresponding gene embedding from gene2vec. The input embedding is then fed through the performer encoder to estimate the masked expressions. The figure is from "scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data" [6].

This is the idea behind the transformer model single-cell BERT (scBERT) [6], which

draws inspiration from the BERT language model with one key difference: instead of written text, it is trained on massive amounts of masked single-cell RNA-seq data.

As illustrated in Figure 2.3, the input is generated by discretizing the gene expressions within each cell through binning. This makes sure that cross-gene comparisons are possible, by focusing on the relative distribution within a gene instead of their absolute values. Due to the sparse nature of single-cell RNA-data, only non-zero gene expressions are binned and subsequently randomly masked [6].

The resulting expression embeddings are then added with the gene embeddings, which are the binned values of gene2vec [20], a 200-dimensional vectorized representation of genes. This results in an input of dimension $M \times N \times 200$, where $M$ is the amount of genes in the dataset and $N$ the number of cells. The input is then ready to be fed into the network which is composed of performer blocks, each housing six performer layers, with each layer incorporating ten self-attention heads [6].

The data used to pre-train scBERT was collected from the PanglaoDB dataset which has integrated 209 human single-cell datasets comprising 74 tissues with 1,126,580 cells originating from different experimental sources via various platforms [21], and from the Heart dataset which contains 451,513 cells from 11 cell types by four different sequencing platforms (Harvard-Nuclei, Sanger-Nuclei, Sanger-Cells, and Sanger-CD45) [22].

During the pre-training, as illustrated by Figure 2.4, the output gets reconstructed into gene expressions and a reconstruction loss is calculated as the cross-entropy loss shown in Equation (2.1), where $K$ is the set of masked expressions, $y_{j,i} \in \{0, 1\}$ and $p_{j,i} \in (0, 1]$ is the true and the predicted expression for gene $j$ in cell $i$ respectively.

$$L_{reconstruction} = -\sum_{i=1}^{N} \sum_{j \in K} y_{j,i} \log\left(p_{j,i}\right) \tag{2.1}$$

The pre-training process allows scBERT to gain a general understanding of gene-to-gene interaction in single-cell RNA-data, similar to how BERT understands contextual information in Natural Language. By performing a 1D-convolution of the output instead of reconstruction, it regains its input dimension of $M \times N$ and is now ready to be used in the fine-tuning classifier.

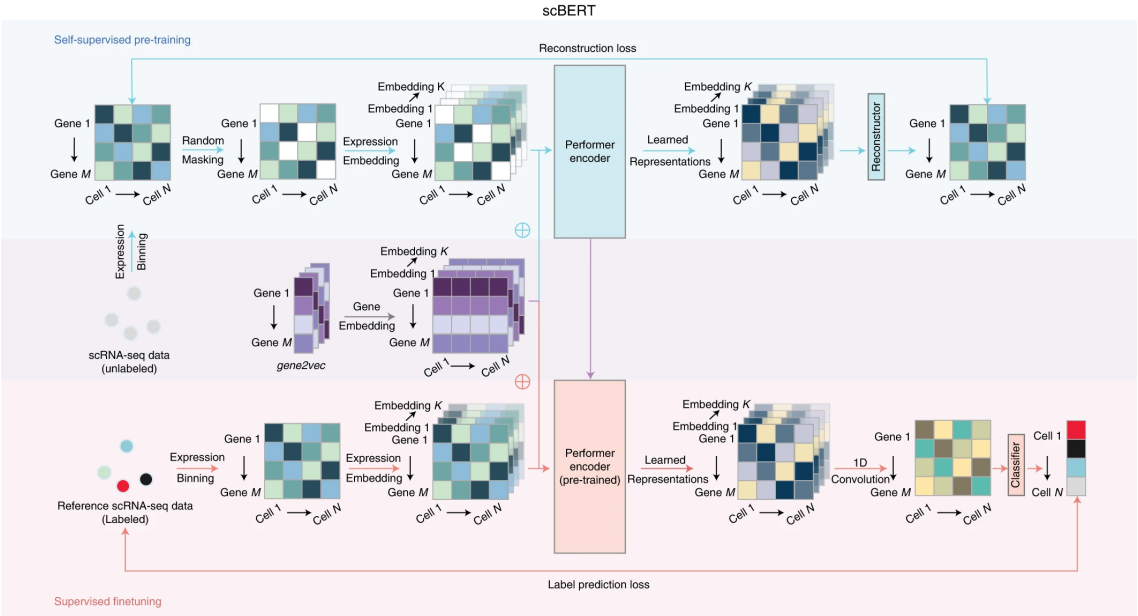


**Figure 2.4:** The pre-training and fine-tuning process illustrated for scBERT. To create the input embedding the expression matrix gets binned, during the pre-training process there is an additional step, where a random selection of non-zero values gets masked. The resulting matrix then gets replicated $K$ times, such that the input gets the dimension $M \times N \times K$, to then get added with the gene embeddings from gene2vec. During the pre-training the output embeddings gets reconstructed into the original dimension $M \times N$, and then the recontruction loss is calculated, as shown in Equation (2.1). For the fine-tuning process the embedding goes through a 1D-Convolution into the dimension $M \times N$, to then be used in the classifier. After classification the prediction loss is calculated, as shown in Equation (2.2). The figure is from "scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data" [6].

For fine-tuning, scBERT incorporates a three-layer neural network classification head, yielding probability estimates for each classification. The model's performance during fine-tuning is evaluated through the cross-entropy loss, as expressed by Equation (2.2)

$$L_{pred} = -\sum_{i=1}^{N} z_i \log(q_i) \tag{2.2}$$

where $z_i$ denotes the true target for cell $i$, and $q_i$ represents the probability score assigned to the corresponding class for cell $i$.

## 2.5 Cellular state classification

In the case of Glioblastoma, it has been suggested that malignant cells exist in a limited set of cellular states: neural-progenitor-like (NPC-like), oligodendrocyte-

progenitor-like (OPC-like), astrocyte-like (AC-like), and mesenchymal-like (MES-like) [3]. In addition, there are two variants of the NPC-like and MES-like states respectively, creating six different cellular states: MES1, MES2, NPC1, NPC2, AC and OPC [3].

Although cellular states are determined by complex relationships between networks of genes and their activity, a first approximation of classifying cells into different cellular states takes a purely expression-driven statistical approach. In this framework, each state is linked with a curated list of genes that is then used to calculate the average expression of the genes associated with a particular cellular state in each cell.

This average is then compared to the average expression of a randomly selected set of genes active within each cell. Each cell is then annotated with the cellular state that attains the highest score compared to the random gene set, as shown in Equations (2.3) and (2.4).

$$c_i^k = \sum_{j \in \mathcal{S}_k} \frac{x_{j,i}}{|\mathcal{S}_k|} - \sum_{j \in \mathcal{A}_k} \frac{x_{j,i}}{|\mathcal{A}_k|}, \ \mathcal{A}_k \subset \mathcal{S}_k' \tag{2.3}$$

$$C_i = \max_{k \in K} \left\{ c_i^k \right\} \tag{2.4}$$

where $c_i^k$ is the score for cellular state $k$ in cell $i$, $\mathcal{S}_k$ is the set of genes associated with cellular state $k$, $x_{j,i}$ is the expression for gene $j$ in cell $i$, $|\mathcal{S}_k|$ is the number of genes in set $\mathcal{S}_k$, $\mathcal{A}_k$ is a random subset of the complement of $\mathcal{S}_k$, $K$ is the number of possible cellular states, and $C_i$ is the final cellular state assigned to cell $i$.

## 2.6 Performance metrics

The most common metrics to measure the performance of a model include Accuracy, Precision, Recall and $F_1$ score.

In Figure 2.5, the metrics used to estimate a classifier's performance are shown. A true positive (TP) is defined as a positive label being labelled positive, a false negative (FN) is a positive label being labelled negative, a false positive (FP) is a negative label being labelled positive and a true negative (TN) is a negative label being correctly labelled negative. The calculations below are referring to these definitions.

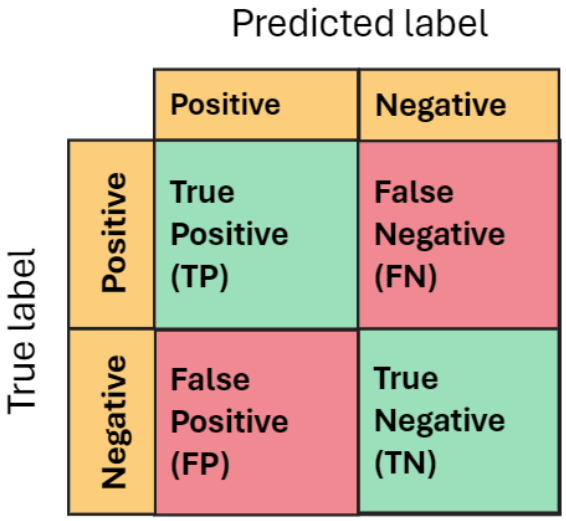


**Figure 2.5:** Classification categories used to calculate performance.

The Accuracy in Equation (2.5) gauges the proportion of correct predictions relative to the total predictions made.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.5}$$

Precision, denoted by Equation (2.6), measures the rate of correctly identifying positive instances, making it relevant for scenarios where minimizing false positives is critical.

$$Precision = \frac{TP}{TP + FP} \tag{2.6}$$

Recall, defined by Equation (2.7), evaluates the rate of correctly identified instances of positive values compared to the total volume of positive predictions, making it relevant in scenarios where minimizing false negatives is a priority.

$$Recall = \frac{TP}{TP + FN} \tag{2.7}$$

A balanced compromise between Precision and Recall is often sought, something the $F_1$ score achieves through the formula in Equation (2.8) which calculates the harmonic mean between them.

$$F_1 = \frac{2}{Precision^{-1} + Recall^{-1}} \tag{2.8}$$

In many classification tasks, there are more than one positive and one negative class that need labelling.
Consider the example in Table 2.1, which has $K$ different classes.

| | class 1 | class 2 | class 3 | ... | class $K$ | $n_i$ |
|---|---|---|---|---|---|---|
| class 1 | 30 | 10 | 5 | ... | 7 | $\sum_{j=1}^{K} m_{1,j}$ |
| class 2 | 15 | 40 | 10 | ... | 3 | $\sum_{j=1}^{K} m_{2,j}$ |
| class 3 | 5 | 5 | 20 | ... | 0 | $\sum_{j=1}^{K} m_{3,j}$ |
| ... | ... | ... | ... | ... | ... | ... |
| class $K$ | 12 | 3 | 4 | ... | 45 | $\sum_{j=1}^{K} m_{K,j}$ |

**Table 2.1:** Example table of prediction matrix of $K$ classes, where $n_i$ denotes the number of data points of each class $i$ and $N = \sum_{i=1}^{K} n_i$ is the total amount of data points in the dataset.

If $m_{i,j}$ is the number of class $i$ predicted as class $j$ (e.g. $m_{1,2} = 10$ according to Table 2.1), the Accuracy can be calculated for the whole dataset as shown in Equation (2.9),

$$Total\ Accuracy = \frac{1}{N} \sum_{i=1}^{K} m_{i,i} \tag{2.9}$$

whereas Precision, Recall and $F_1$ need to be calculated individually for each class $i$, as shown in Equation (2.10), (2.11) and (2.12).

$$Precision(i) = \frac{m_{i,i}}{\sum_{j=1}^{K} m_{j,i}} \tag{2.10}$$

$$Recall(i) = \frac{m_{i,i}}{\sum_{j=1}^{K} m_{i,j}} \tag{2.11}$$

$$F_1(i) = \frac{2}{Precision(i)^{-1} + Recall(i)^{-1}} \tag{2.12}$$

By calculating the average, with or without weights, of these metrics it is possible to get a representation of the performance of the whole dataset.

In the case of Table 2.1, the average of $F_1$ without weights, denoted macro average $F_1$, would be calculated in accordance to Equation (2.13) and the weighted average would be calculated by Equation (2.14).

$$Macro\ Average\ F_1 = \frac{1}{K} \sum_{i=1}^{K} F_1(i) \tag{2.13}$$

$$Weighted\ Average\ F_1 = \frac{1}{N}\sum_{i=1}^{K} n_i F_1(i) \tag{2.14}$$

It is also possible to calculate the micro $F_1$, where all true positives, false negatives and false positives are calculated first, in accordance with Equations (2.15), (2.16) and (2.17).

$$TP_{Total} = \sum_{i=1}^{K} m_{i,i} \tag{2.15}$$

$$FP_{Total} = \sum_{j=1}^{K}\left(\sum_{i=1}^{K} m_{j,i}\right) - \sum_{i=1}^{K} m_{i,i} \tag{2.16}$$

$$FN_{Total} = \sum_{i=1}^{K}\left(\sum_{j=1}^{K} m_{j,i}\right) - \sum_{i=1}^{K} m_{i,i} \tag{2.17}$$

In the case where every instance can be assigned one and only one label, the micro $F_1$ is equal to the accuracy, which is the case for this project.

# 3

# Methods

The Nelander lab at Uppsala University [23] provided the single-cell RNA-sequencing data of Glioblastoma. The pre-processing of the cell-lines includes quality assessment of the cells, normalization and gene filtering. The cells were then classified into one of six cellular states, and combined into one large dataset. Gene matching with gene embeddings from gene2vec was performed and the data was then prepared for the fine-tuning process.

## 3.1 Data

The single-cell RNA-sequencing data used for this project were received from Nelander lab at Uppsala university [23] and contains 20 cell-lines of glioblastoma. The cell lines are derived from seven primary tumors being cultured on a culture disc or mouse xenograft, i.e. grown in a mouse. In total, twelve of the cell-lines are from xenograft. Gene2vec was used as gene embeddings, which are available to download from their git repository [20].

## 3.2 Pre-processing

Utilization of single-cell data requires a pre-processing stage, both for discerning poor-quality cells and eliminating genes that carry uninformative signal or simply biological noise. In addition, normalization is essential to ensure the comparability of gene expressions across different cells.

Given that the single-cell RNA sequencing data are collected in a laboratory environment, it is necessary to exclude cellular data that may be contaminated or in other ways unsuitable for further analysis.

A standardized approach is to begin evaluation of each cell. If a cell contains few

active genes, it could mean that the sample might have been empty. Therefore each cell with less than 500 active genes are omitted.
If a large proportion of the total gene expressions in a cell comes from mitochondrial genes, it is indicative of cell degradation. Hence, all cells where the mitochondrial genes are responsible for more than 20% of the total expressions are omitted as well.

To facilitate meaningful comparisons between cells, the expression data undergoes normalization. Each cell undergoes a normalization such that their total gene expressions equals 10 000, as illustrated in Equation (3.1). Each normalized gene expression $x$ is then logarithmized with the natural logarithm $\log(x+1)$, as shown in Equation (3.2).

$$\hat{x}_{j,i} = 10^4 \frac{x_{j,i}}{\sum_{k=1}^{M} x_{k,i}} \tag{3.1}$$

$$x_{j,i}^{New} = \log(\hat{x}_{j,i} + 1) \tag{3.2}$$

Genes that are active in less than 50 cells are considered noise in the data and are also excluded from the dataset. To prohibit genes with a high variation from dominating the learning process, while still preserving the distribution structure, scaling of the data is performed. The standard deviation of each gene's expressions is calculated, and then their expressions get divided with their respective standard deviation. If $\sigma_j$ denotes the standard deviation of gene $j$, each gene expression gets updated in accordance to Equation (3.3).

$$x_{j,i}^{New} = \frac{x_{j,i}}{\sigma_j} \tag{3.3}$$

This results in all genes having a standard deviation of one.

### 3.2.1 Cellular state classification

By looking on how prominent the expressions are from certain genes in individual cells, it is possible to classify each cell according to its state. Six different cellular states were used based on the article "An Integrative Model of Cellular States, Plasticity, and Genetics for Glioblastoma" [3]. Each cellular state has a list of genes representing that state, and by taking the average expression of the gene representation from each cell and subtract it with the average expression from a subset of the remaining genes, each state gets a score. Each cell was then classified

by assigning the state corresponding to its top score, as explained in section 2.5.

### 3.2.2 Combining the datasets

In order to run all cell-lines through scBERT, the cell-lines needed to be combined into a single file. Each cell-line got pre-processed individually, therefore ending up with slightly different genes present. Since combining the cell-lines into a large dataset requires the presence of the same genes in all cell-lines, the cell-lines got reduced to only contain the intersecting set of genes.

The gene2vec embeddings were downloaded from the Github associated with the article "Gene2vec: distributed representation of genes based on co-expression" [20], and an intersection of genes between the gene2vec-file and the combined datasets were performed, so that the gene2vec embeddings matched the corresponding genes present in the dataset.

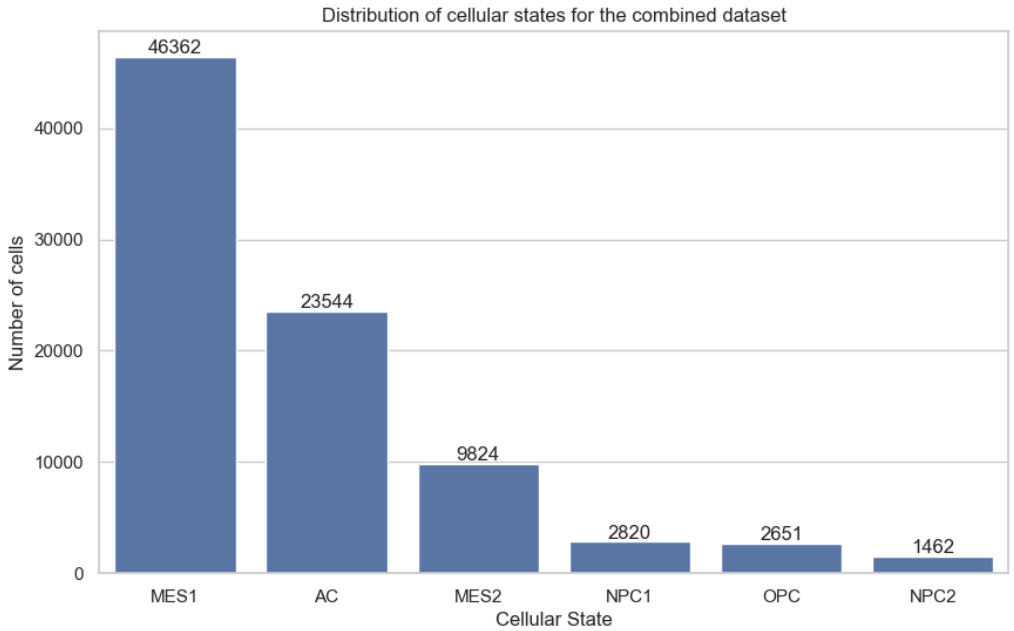


**Figure 3.1:** Plot showing the distribution of cellular states in the combined dataset.

| **Cellular State** | MES1 | AC | MES2 | NPC1 | OPC | NPC2 |
|---|---|---|---|---|---|---|
| **Percentage** | 53.50% | 27.17% | 11.34% | 3.25% | 3.06% | 1.69% |

**Table 3.1:** The percentage of each cellular state in the dataset.

After this process, the combined dataset contained a total of 86 663 cells and 8 759 genes that were run through scBERT, with the distribution of cellular states illustrated in Figure 3.1 and Table 3.1.

### 3.2.3 Reducing cellular state-specific genes

Due to the uneven distribution of cellular states in the dataset, as illustrated by Table 3.1, additional datasets were created in which 25%, 50%, 75% and 100% of the genes associated with the cellular state with the largest representation, MES1, were removed to see how this affected performance. The same was done with the cellular state with the smallest representation, NPC2.

| **Percentage removed** | 0% | 25% | 50% | 75% | 95% | 99% |
|---|---|---|---|---|---|---|
| **Genes present** | 8 759 | 6 557 | 4 360 | 2 205 | 434 | 88 |

**Table 3.2:** Table of random removal of genes in the dataset, where the top row shows the percentage of total genes that were removed and the bottom row shows how many genes were present in the dataset afterwards.

In addition datasets were created where a random total subset of genes were removed, shown in Table 3.2, in order to test how far it was possible to push the model.

## 3.3 Fine-tuning

When fine-tuning, 80% of the data are reserved for training and 20% reserved for validation. The batch size is set to three. Total number of epochs are 100, but if no higher accuracy has been reached in the last ten epochs the fine-tuning stops.

The fine-tuning consists of three linear layers, and uses RELU as an activation function. The input layer consists of $M$ (the number of genes) neurons, the first hidden layer consists of 512 neurons, the second hidden layer consists of 128 neurons and the output dimension is equal to the amount of labels in the dataset, in this case six.

Since the fine-tuning mechanism is its own neural network with the same input dimensions as the original input, a comparison have been made between running

the input straight into the fine-tuning network, without any processing through the scBERT-encoder, and using the output from the scBERT-encoder as intended. This is to see whether scBERT has any biological context that is beneficial for this relatively small classification task.
All fine-tuning have been using the same seed to ease comparison between the two methods.

The weights from the epoch with the highest validation accuracy are chosen as the best model. Considering the unbalanced dataset, extra attention is given to the macro average $F_1$-performance, shown in Equation (2.13), since it gives equal attention to all labels without taking their respective proportion of the dataset into consideration.

## 3.4 Limitations

Due to time constraint and the computational resources required for the task, each experiment has been run only once. Therefore less weight is added to the individual results, and more focus lies in the general behaviour and results of the model.
A simplification was done in the statistical method for the cellular state classification. It was done in a very naive manner, where every cell was classified into one singular cellular state, while in reality cells moves between the different states and therefore can lie in between states.

# 4

# Results and Discussion

The scBERT-encoder is tested in a few scenarios, both with random removal of genes and targeted removal, i.e. cellular state-specific genes. Its performance is also compared with other cellular state-classification tasks, pertaining to different types of cancer. The overall performance does not seem to benefit from the use of the scBERT-encoder, but the performance of the underrepresented cellular states seem to gain from it.

## 4.1 Benchmark

The full dataset were run through the model, both by running it through the scBERT-encoder and feeding the input embeddings straight into the fine-tuner.

| **Datasets group** | **Datasets name** | **Species** | **Cancer type** |
|---|---|---|---|
| Tumor immune microenvironment (TME) | BRCA | Human | Breast cancer |
| | HNSC | Human | Head and neck cancers |
| | NSCLC | Human | Non–small-cell lung cancer |
| TIT | LIHC | Human | Liver cancer |
| | CRC | Human | Colorectal cancer |
| | NSCLC | Human | Non–small-cell lung cancer |

**Table 4.1:** Datasets from the Gene Expression Omnibus database. Breast cancer (BRCA), head and neck cancers (HNSC), non–small-cell lung cancer (NSCLC), colorectal cancer (CRC) and liver cancer (LIHC) [24].

The results are compared in Figure 4.1 with cellular classification in other cancers from “Evaluation of machine learning approaches for cell-type identification from

single-cell transcriptomics data" [24]. The TME and TIT dataset were collected from the Gene Expression Omnibus database and are presented in Table 4.1. These cells were all collected from humans, while the cells present in the Glioblastoma (GBM) dataset is collected from both humans and mice.

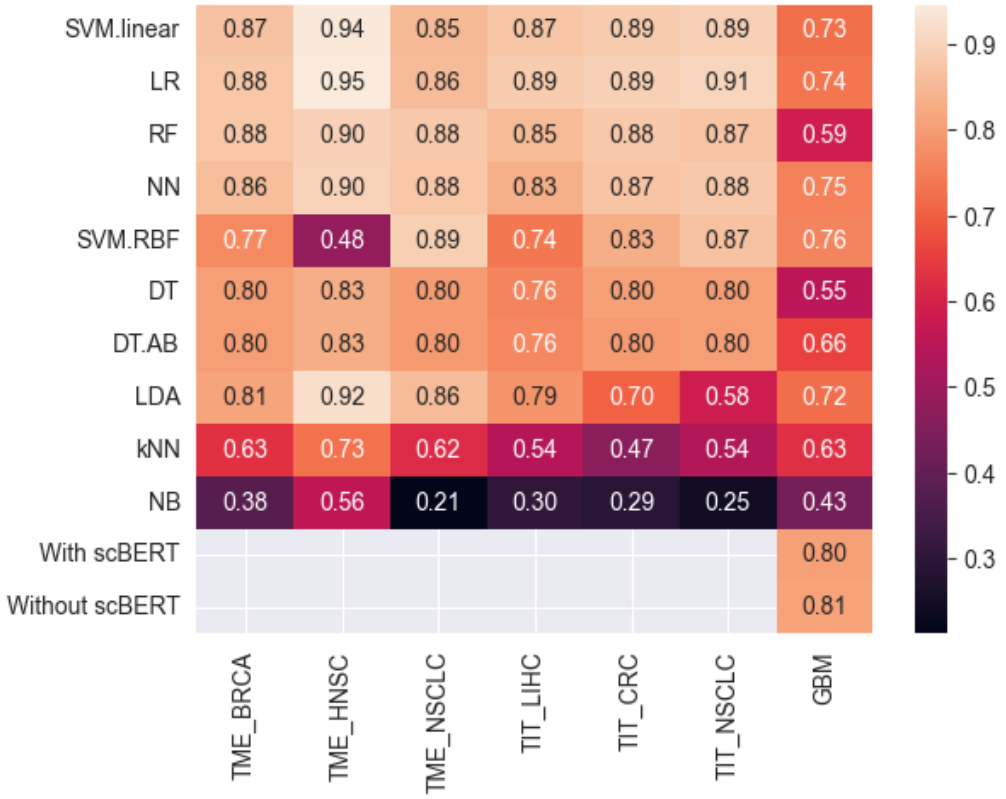


**Figure 4.1:** Micro $F_1$ for different datasets and machine learning methods. Linear Support Vector Machine (SVM.linear), Logistic Regression (LR), Random Forest (RF), Neural network (NN), Support Vector Machine with RBF kernel (SVM.RBF), Decision Trees (DT), Decision Tree Regres sion with AdaBoost (DT.AB), Linear Discriminant Analysis (LDA), k Nearest Neighbors (kNN) and Naive Bayes (NB) [24].

In Figure 4.1 it can be seen that the scBERT-models (with and without the encoder) performance is approximately average, compared to the other models performance with the datasets TME and TIT. However, using the same process for testing these models as described in "Evaluation of machine learning approaches for cell-type identification from single-cell transcriptomics data" [24], for the Glioblastoma dataset, they perform worse than both scBERT-models. This could indicate that the TME and TIT datasets are less complex to classify, and it could be interesting to try classifying them with scBERT to see if it would reach a higher performance for them as well.

## 4.2 Random removal of genes

Random genes were removed from the dataset, 0%, 25%, 50%, 75%, 95%, and 99%. The number of genes still present for each percentage can be viewed in Table 3.2. They were put through fine-tuning, both with and without the scBERT-encoder, to see how the presence of the encoder affected the performance.

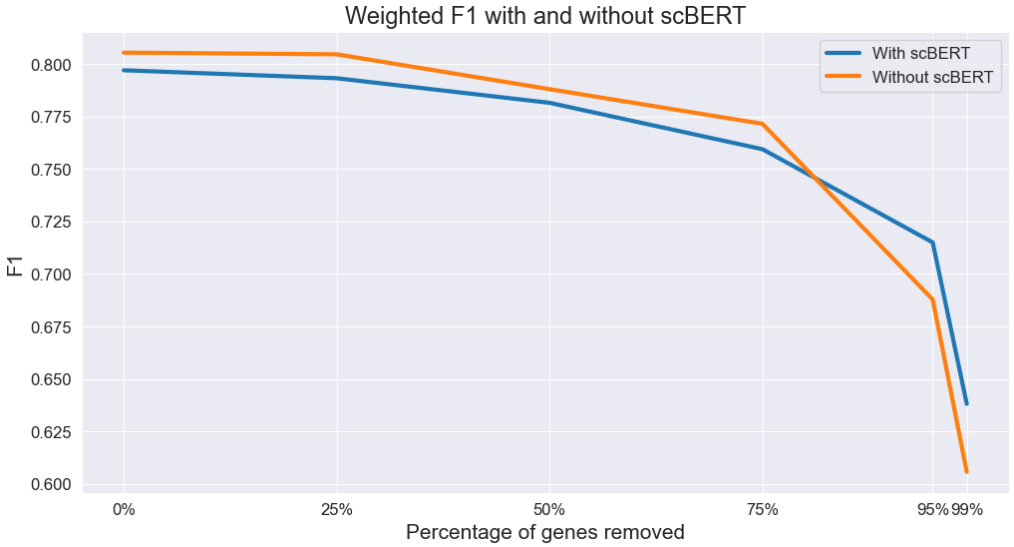


**Figure 4.2:** Weighted average F1 of the model with and without scBERT, where random genes have been removed.

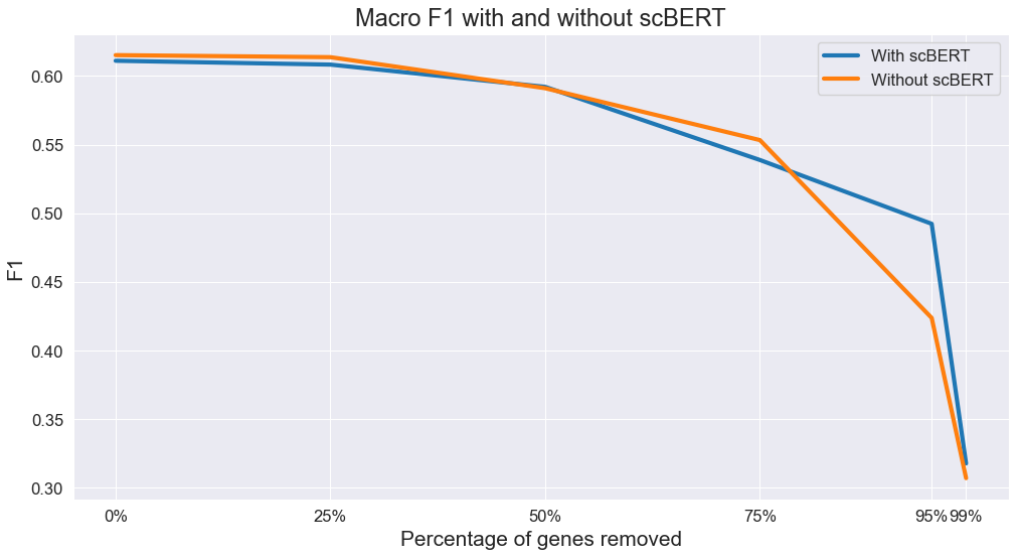


**Figure 4.3:** Macro average F1 of the model with and without scBERT, where random genes have been removed.

Figure 4.2 shows the weighted average $F_1$ of the top epoch for each percentage removed. The results with and without the encoder were fairly similar, with the data that had been run through the scBERT-encoder performing slightly worse than without until a considerable percentage (95% and 99%) of the genes had been removed.

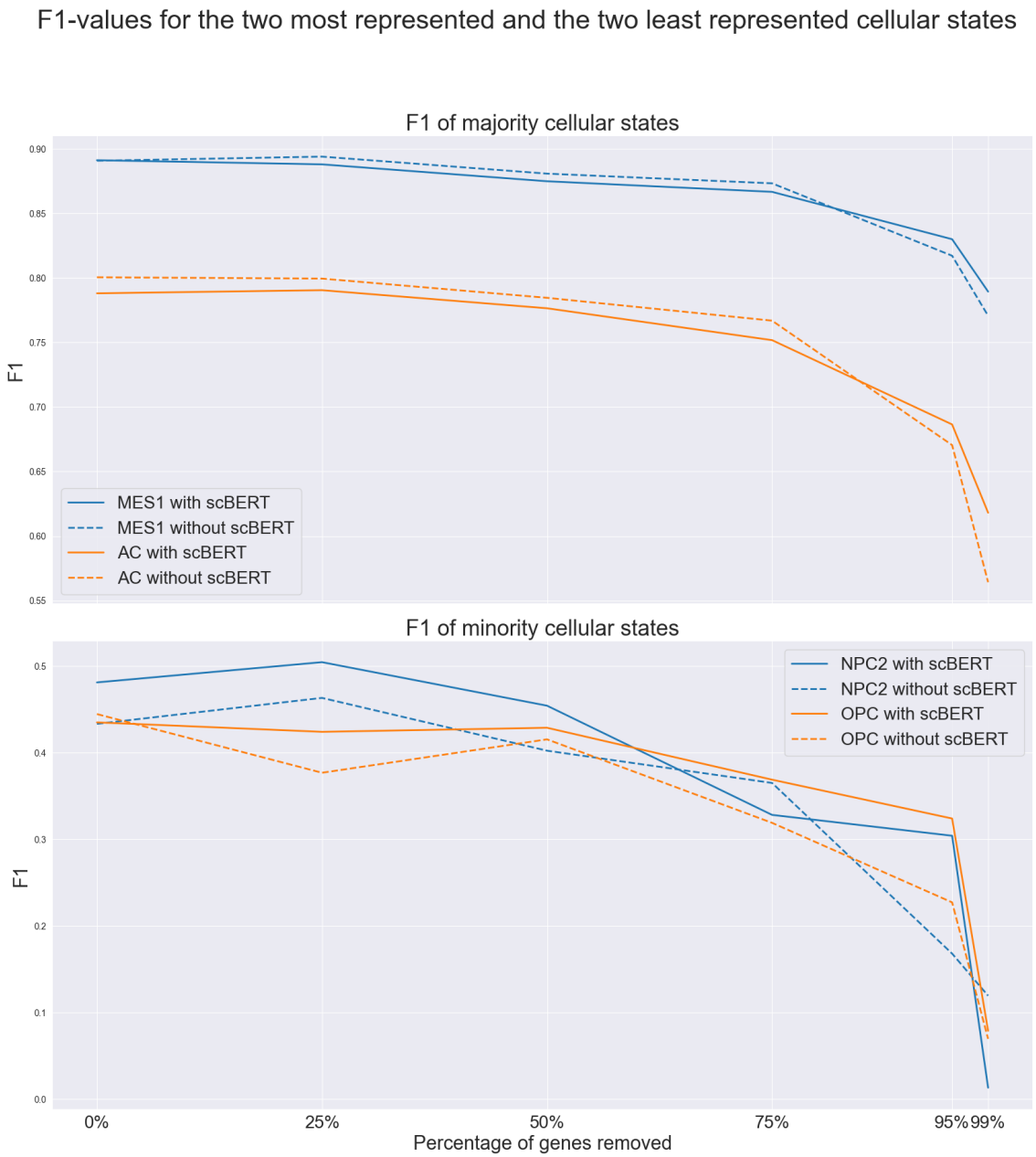


**Figure 4.4:** $F_1$ of MES1 and AC, the two cellular states with the largest representation and NPC2 and OPC, the two cellular states with the smallest representation, with and without scBERT where random genes have been removed.

The drop of the $F_1$-score in general is surprisingly low, even when 99% of the genes were removed, the $F_1$-score for both methods were above 60%, but this could be due

to the fact that 53.5% of the cellular states consists of MES1.

The macro average of $F_1$, shown in Figure 4.3, show a similar trend and were plotted in a similar manner. The macro average of $F_1$ gains more benefit from using scBERT when a larger percentage of the genes were removed compared to the weighted average of $F_1$.

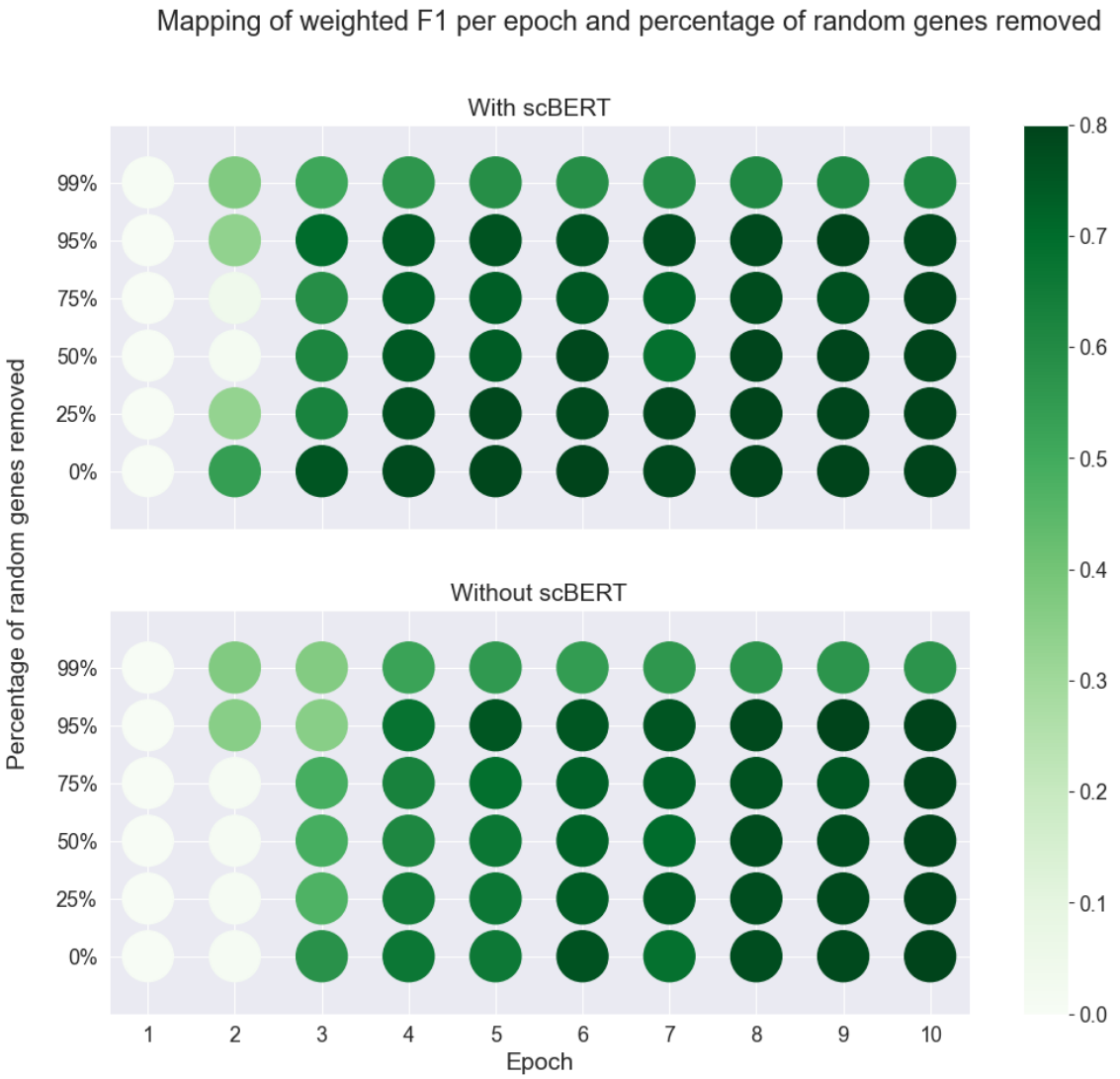


**Figure 4.5:** The resulting weighted average $F_1$ per epoch and percentage of removed genes for both embeddings that have gone through scBERT and without, where a darker green indicates higher $F_1$.

By looking at the $F_1$-score of the two cellular states with the smallest and largest representation in the dataset, NPC2 and OPC and MES1 and AC respectively, in Figure 4.4, the minority cellular states NPC2 and OPC seems to benefit from using

the scBERT-encoder. This might indicate that the information the scBERT-encoder provides slightly makes up for the small amount of data available for these cellular states. This is consistent with the general results of the $F_1$-scores, where the use of the encoder seems beneficial when a lot of genes are removed. The $F_1$-score for the cellular states MES1 and AC, that together makes up approximately 80% of the cellular states in the dataset, unsurprisingly shows a similar trend as the overall performance of the model, where the usage of the scBERT-encoder performs slightly worse than without, until a considerable amount of genes are removed.

In Figure 4.5 the weighted average $F_1$ vs. percentage of genes removed and the first 10 epochs is shown. The model in general gets a higher $F_1$ value with fewer epochs when the input has been processed through the scBERT-encoder first. It is possible that the encoder provides some information that makes the fine-tuning reach higher performance on fewer epochs.

The cellular state classification is a fairly simple algorithm, which can make the encoder from scBERT redundant in its training. To reach a proper conclusion in its performance it would be beneficial to train it several times and it would also be of interest to test it with a more complex classification task. It might also be worth exploring its performance on several different cancer types, for example those represented in the benchmarking in Section 4.1, since heterogeneity varies between different types of cancer [25].

## 4.3 Removal of cellular state-specific genes

To test how well the model performs when the data that underlies the actual classification is removed, cellular state-specific genes were removed. This was done with two cellular state, the one with the largest representation, MES1, and with the smallest representation, NPC2. It was done in increments of 25%, i.e. 0%, 25%, 50%, 75% and 100% of the genes associated with that particular cellular state were removed.

Even though cellular state-specific genes were removed, the total amount of genes removed were very few. In the case of removing 100% of the MES1-specific genes, a total of 31 genes were removed, and in the case of removing 100% of NPC2-specific genes, a total of 25 genes were removed.

### 4.3.1 Removal of MES1-specific genes

Cellular-state specific genes were removed for MES1, which is the cellular-state with the highest representation in the dataset, and then run through the fine-tuning network with and without scBERT.

In Figure 4.6 the $F_1$-scores for MES1 for each increment in MES1-specific genes removal are shown. Even though a maximum of 31 genes were removed, there is a slight decrease in the $F_1$-score for MES1. In general the scBERT-encoder didn't provide any benefit to the result, even when 100% of the MES1-specific genes were removed.
This could indicate that there might be redundancy in the remaining genes, where the pre-training of scBERT-encoder is unnecessary for the model to complement the gene-to-gene interaction.

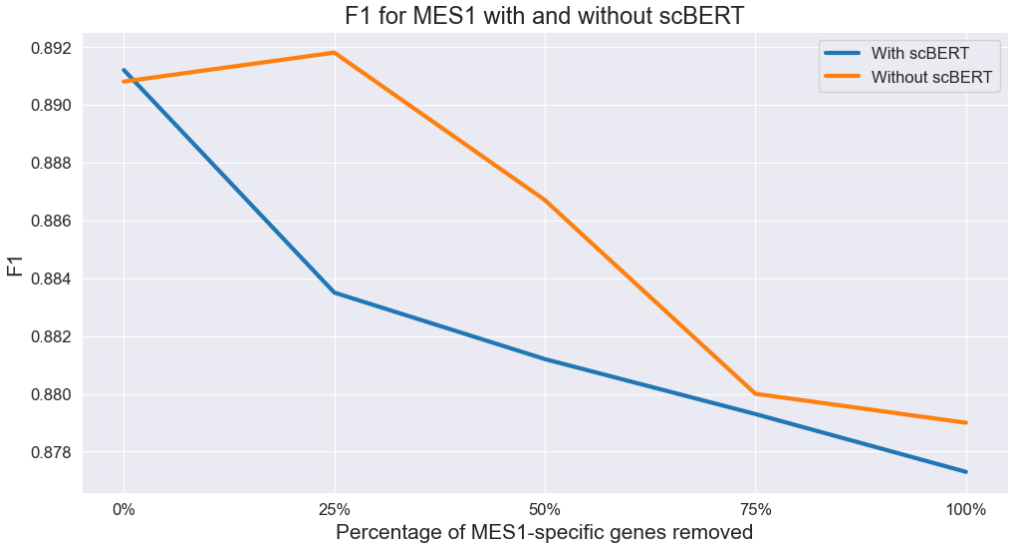


**Figure 4.6:** $F_1$-score of model for MES1 with and without scBERT when MES1-specific genes were removed.

There is also a possibility that since the cellular state MES1 is overrepresented in the dataset, with over 50% of the cells, that the model has a bias for classifying cells as MES1 whenever possible.

### 4.3.2 Removal of NPC2-specific genes

Cellular state-specific genes for NPC2, the cellular state of only 1.69% of the cells in the dataset, were removed. In Figure 4.7 the $F_1$-score for NPC2 for every increment is shown.

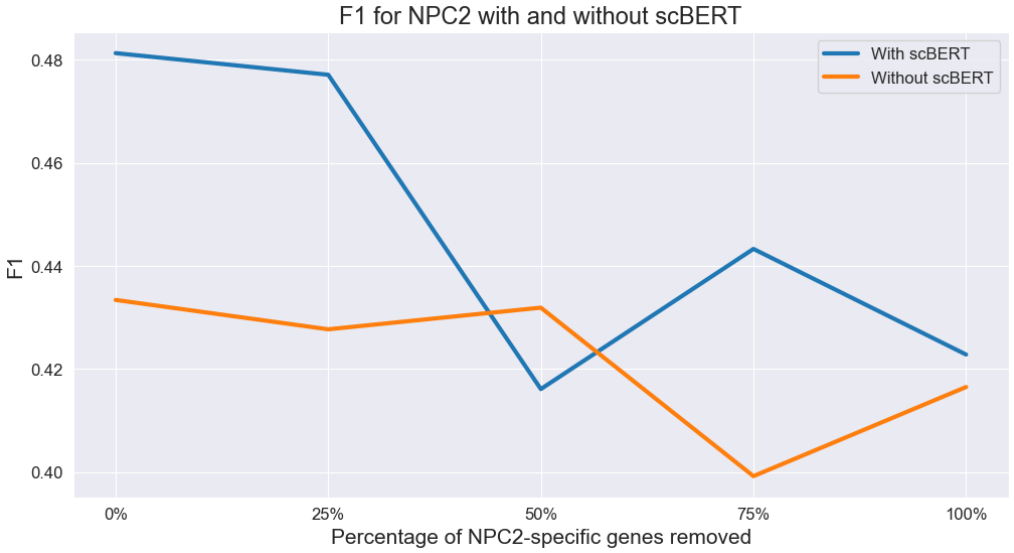


**Figure 4.7:** $F_1$-score of model for NPC2 with and without scBERT when NPC2-specific genes were removed.

Similar to the random removal of genes, shown in Figure 4.4, the performance for NPC2 seems to benefit from use of the scBERT-encoder. The scores decreased more than for MES1 in Figure 4.6, a reason could be that there is no benefit of classifying cells it is uncertain of as NPC2, due to the underrepresentation. There are still cells that are classified as NPC2 when all the NPC2-specific genes are removed, which might indicate that there is redundancy in the gene expressions, especially since there is no gain for the model to randomly assign the label NPC2 to a cell, compared to MES1.

# 5

# Conclusion

Although the integration of scBERT's encoder in classifying annotated cellular states in Glioblastoma showed minimal improvement in overall performance, it seems to demonstrate the ability to compensate for scenarios where the gene count is very low and for underrepresented cellular states. This might indicate that scBERT's prior knowledge is utilized in these specific situations, and it could therefore be of interest to test it on more complex tasks. It should also be possible to modify the fine-tuning into taking the cells movement between different cellular states into account, which could give a more realistic representation.

Due to limitations in time and computational resources, each experiment was conducted only once. Therefore additional runs are necessary to be able to properly evaluate each individual result. It seems to perform well compared to some other machine learning techniques, as illustrated in Figure 4.1, hence it could be of interest to test its performance on other types of cancer as well to see if the trend holds.

Potential appears to exist in the usage of the Transformer architecture for single-cell RNA-sequencing data, and hopefully it continues to evolve in the future.

# Bibliography


[1] National Cancer Institute, *Glioblastoma—Unraveling the Threads: A Q&A with Drs. Mark Gilbert and Terri Armstrong of the NIH Neuro-Oncology Branch*, https://www.cancer.gov/news-events/cancer-currents-blog/2017/glioblastoma-research-making-progress [Accessed: 10-11-2023], 2017.

[2] National Cancer Institute, *NCI dictionary of cancer terms*, https://www.cancer.gov/publications/dictionaries/cancer-terms/, Accessed: (9/4-2024), 2024.

[3] C. Neftel, J. Laffy, M. Filbin, *et al.*, "An integrative model of cellular states, plasticity, and genetics for glioblastoma," *Cell*, vol. 178, no. 4, Aug. 2019. DOI: 10.1016/j.cell.2019.06.024. [Online]. Available: https://doi.org/10.1016/j.cell.2019.06.024.

[4] OpenAI, J. Achiam, S. Adler, *et al.*, "Gpt-4 technical report," 2023. DOI: 10.48550/ARXIV.2303.08774. [Online]. Available: https://arxiv.org/abs/2303.08774.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[6] F. Yang, W. Wang, F. Wang, *et al.*, "scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data," *Nature Machine Intelligence*, vol. 4, no. 10, pp. 852–866, Oct. 2022. DOI: https://doi.org/10.1038/s42256-022-00534-z.

[7] A. Haque, J. Engel, S. A. Teichmann, and T. Lönnberg, "A practical guide to single-cell rna-sequencing for biomedical research and clinical applications," *Genome Medicine*, vol. 9, no. 1, Aug. 2017, Article 75. DOI: https://doi.org/10.1186/s13073-017-0467-4.

[8] MedlinePlus, *What is a gene?* https://medlineplus.gov/genetics/understanding/basics/gene/, Accessed: (28/3-2024), 2024.

[9] M. Bernstein N., *On cell types and cell states.* https://mbernste.github.io/posts/cell_types_cell_states/, Accessed: (28/3-2024), 2021.

[10] I. Larsson, E. Dalmo, R. Elgendy, *et al.*, "Modeling glioblastoma heterogeneity as a dynamic network of cell states," *Molecular Systems Biology*, vol. 17, no. 9, e10105, 2021. DOI: https://doi.org/10.15252/msb.202010105. eprint: https://www.embopress.org/doi/pdf/10.15252/msb.202010105. [Online]. Available: https://www.embopress.org/doi/abs/10.15252/msb.202010105.

[11] Ethnologue, *How many languages are there in the world?* https://www.ethnologue.com/insights/how-many-languages/, Accessed: (28/3-2024), 2024.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. DOI: 10.48550/ARXIV.1301.3781. [Online]. Available: https://arxiv.org/abs/1301.3781.

[13] N. Patwardhan, S. Marrone, and C. Sansone, "Transformers in the real world: A survey on nlp applications," en, *Information*, vol. 14, no. 4, p. 242, Apr. 2023, ISSN: 2078-2489. DOI: 10.3390/info14040242.

[14] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," 2017. DOI: 10.48550/ARXIV.1706.03762. [Online]. Available: https://arxiv.org/abs/1706.03762.

[15] M. Nabil, *Unpacking the query, key, and value of transformers: An analogy to database operations*, https://www.linkedin.com/pulse/unpacking-query-key-value-transformers-analogy-database-mohamed-nabil/, Accessed: (28/3-2024), 2023.

[16] H2O.ai, *Classify token ([CLS])*, https://h2o.ai/wiki/classify-token/, Accessed: (2/5-2024), 2024.

[17] H2O.ai, *Separate token ([SEP])*, https://h2o.ai/wiki/separate-token/, Accessed: (2/5-2024), 2024.

[18] M. Gupta, *BERT: A deeper dive*, https://medium.com/data-science-in-your-pocket/bert-a-deeper-dive-7412db7c1a93, Accessed: (8/4-2024), 2024.

[19] S. R. Choi and M. Lee, "Transformer architecture and attention mechanisms in genome data analysis: A comprehensive review," en, *Biology*, vol. 12, no. 7, p. 1033, Jul. 2023, ISSN: 2079-7737. DOI: 10.3390/biology12071033.

[20] J. Du, P. Jia, Y. Dai, C. Tao, Z. Zhao, and D. Zhi, "Gene2vec: Distributed representation of genes based on co-expression," *BMC Genomics*, vol. 20, no. 1, Feb. 2019, Article 82. DOI: https://doi.org/10.1186/s12864-018-5370-x.

[21] O. Franzén, L.-M. Gan, and J. L. M. Björkegren, "Panglaodb: A web server for exploration of mouse and human single-cell rna sequencing data," en,

*Database*, vol. 2019, Jan. 2019, ISSN: 1758-0463. DOI: 10.1093/database/baz046. [Online]. Available: https://academic.oup.com/database/article/doi/10.1093/database/baz046/5427041.

[22] M. Litviňuková, C. Talavera-López, H. Maatz, *et al.*, *Cells of the adult human heart*, https://explore.data.humancellatlas.org/projects/ad98d3cd-26fb-4ee3-99c9-8a2ab085e737, Accessed: (3/4-2024), 2021.

[23] *Nelander lab*, https://nelanderlab.org/, Accessed: (23/5-2024), 2024.

[24] Y. Huang and P. Zhang, "Evaluation of machine learning approaches for cell-type identification from single-cell transcriptomics data," en, *Briefings in Bioinformatics*, bbab035, Feb. 2021, ISSN: 1467-5463, 1477-4054. DOI: 10.1093/bib/bbab035.

[25] G. H. Heppner, W. R. Shapiro, and J. K. Rankin, "Tumor heterogeneity," *Pediatric Oncology 1: with a special section on Rare Primitive Neuroectodermal Tumors*, pp. 99–116, 1981.

**DEPARTMENT OF MATHEMATICAL SCIENCES**
**CHALMERS UNIVERSITY OF TECHNOLOGY**
Gothenburg, Sweden
www.chalmers.se


CHALMERS
UNIVERSITY OF TECHNOLOGY