Predicting Cell Types in Non-Human Single Cell RNA Sequencing Data

Sean Connelly **Ryan Videgar-Laird**

Stephanie Ting

1 Introduction

The biology that underlies disease is heterogeneous, with multiple cell types and pathways within those cell types that contribute to the development of disease. All cells have DNA, the genetic code, that is transcribed into messenger RNA (mRNA), called gene expression. After these genes are expressed, the mRNA is then translated into proteins, which go on to perform different functions. mRNA can be expressed at different levels, depending on which proteins are needed by the given cell type. These different levels of mRNA expression can characterize cells and allow further analysis of their contribution to disease biology [1].

The mRNA produced can be quantified at the cellular level, called single cell RNA sequencing (scRNAseq), to obtain a 'fingerprint' of different cell types and can be compared between normal or disease conditions to find how these cell types change. scRNAseq data is processed into a gene expression matrix, which has dimensions of m cells x n genes. After performing quality control steps, normalizing and scaling this gene expression matrix, this matrix undergoes dimensionality reduction to compress this dataset that contains many features (genes) to see how these cells relate to one another. The patterns of how cells vary compared to one another can be clustered using any algorithm of one's choosing. The resulting clusters represent certain types of cells that may be characterized and classified [2].

Classifying single cells has mainly focused on utilizing three categories of techniques. The first class of techniques annotate these clusters through comparing the differential expression (greater number of counts) of the genes in a given cluster with every other cluster through statistical tests, like the Wilcoxon rank sum test [3]. These differentially abundant genes in each cluster can be compared to experimentally validated genes that characterize cell types to identify the label for that cluster. The second class of techniques correlates the newly generated dataset with a reference dataset to annotate cell clusters [4]. This reference dataset can be other single cell RNA sequencing data or bulk RNA sequencing of specific cell types individually or over a time series. Lastly, the third class of techniques uses supervised learning, which utilizes previously generated data to train a model to classify cell types. Some examples of potential models to implement could be logistic regression, support vector machine or random forest, to name a few [5].

Many of these tasks have been focused on human data, where there has been much focus on diseases that affect many people domestically, but not much focus on those that have broad impacts globally. One such disease is malaria, a parasitic infection that resulted in 247 million cases in 2021 and causes a large burden of morbidity and mortality globally [6]. Malaria has a complex life cycle, which starts in the mosquito vector to result in infection of a human host. When individuals are sick with malaria, the parasite replicates in the blood cells and causes these blood cells to lyse when the parasite replicates. These replication stages have unique gene expression patterns and single cell RNA sequencing provides a key benefit to further understand how these genes change per cell stage. These same concepts can be applied to other neglected diseases and hosts. We would like to build a powerful, simple and efficient model that can classify cell types on these sources of data.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Previous works have used attention weights from transformer based models to identify biomarkers from cell type prediction [7] [8]. However, these methods rely on large amounts of training data and manually curated expert knowledge that are not available for organisms such as malaria. In our study we attempt to create a model with comparable accuracy that does not require either of these. Our approach is to use these methods as a foundation to build a simpler model that is less computationally expensive and works on non-human, non-mouse datasets. Our model will also not rely on manual curation of knowledge, which is often inconsistently annotated and formatted.

1.1 Related Work

Two previous studies have introduced methods that successfully use attention based models to predict cell type in scRNAseq. The first one, TOSICA, uses a three part approach of cell embedding, multi-head self attention, and then cell type classification [7]. TOSICA is highly accurate for both mouse and human data. This model, however, relies on curated expert knowledge in the cell embedding stage to provide pathway information critical to identifying cell types. The second model, scBERT, uses an adaptation of BERT, a natural language processing model on scRNAseq where the transformer architecture is replaced with a performer in order to better handle larger scale input data [8]. This model is trained with data from millions of cells from a variety of different biological contexts. scBERT not only predicts cell type with very high accuracy, it also has interpretability in that the prediction rediscovers known biomarker genes of each cell type. However, scBERT does require an immense amount of training data and has only been tested on human data. It is also a complex model that is computationally intensive.

2 Methods

2.1 Data Processing

Anndata [9] and Scanpy /citewolfSCANPYLargescaleSinglecell2018 were used to create efficient data objects and follow best practices for scRNA-seq analysis. The datasets represented in Table 1 have to be accessed from their data repositories to obtain the m x n matrix. We implemented quality control filtering and normalization on these datasets through thresholding cells to only those with 200 or more expressed genes and filtering genes to those expressed in 3 cells or more. Counts were then normalized so that all cells had the same total count and log transformed (see https://scanpy.readthedocs.io/en/stable/generated/scanpy.pp.log1p.html). These quality control (QC) steps help reduce noise from experimental prep from sources, such as low quality cells and environmental contamination, and stabilize variance across cells [5]. The output is a gene expression matrix X, with dimensions m cells by n genes. We aimed to predict the different life stages of the malarial parasite (4 total classes: rings, early trophs, late trophs, and schizonts) and cell types reflected in the liver cancer cell atlas (7 total classes: B cells, CAFs, Malignant cells, T cells. TAMs, TECs, and unclassified).

Let y represent total gene counts for a given cell, and s_c represent the size factor for that cell. Counts are then shifted by:

$$f(y) = log\left(\frac{y}{s_c} + y_0\right)$$

The size factor for a given cell designed to account for variation in cell size (larger cells typically contain more genetic transcripts) and sampling effects. Let L represent the median raw count depth across all cells. Then s_c is calculated across all genes, g, per cell:

$$s_c = \frac{\sum_g y_{gc}}{L}$$

2.2 Logistic Regression

We first employed multinomial logistic regression, as implemented in the package scikit-learn [10]. The model predicts probability $\hat{P}(y_i = k|x_i)$, out of k different classes. The probabilities are determined using this softmax function:

$$\hat{P}(y_i = k|x_i) = \frac{\exp(x_i W_k + W_{0,k})}{\sum_{l=0}^{K-1} \exp(x_i W_l + W_{0,l})}$$

where W is a matrix of coefficients with each row w_k corresponds to class k and $W_{0,k}$ is the bias term for class k. The gene expression vector for each cell is detonated by x_i

The optimization objective is to minimize the negative log-likelihood with an added L_2 regularization term to mitigate overfitting:

$$\min_W -C \sum_{i=1}^{n} \sum_{k=0}^{K-1} [y_i = k] \log(\hat{P}(y_i = k|X_i)) + \frac{1}{2}\lambda ||W||_F^2$$

The Iverson bracket $[P]$ evaluates to 1 if predicate P is true and 0 otherwise. The regularization term $\frac{1}{2}\lambda ||W||_F^2$ is the L_2 norm of the weight matrix W, where λ is the regularization strength, helping to control the complexity of the model by penalizing large weights.

2.3 SVM

For the training matrix X_{train} and a vector of labels y, SVM funds w and b such that the prediction of sign($w^T x + b$) is correct for all x in X_{train}. The hinge loss function is optimized by the following primal problem, as detailed in [10]:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \max(0, 1 - y_i(w^T \phi(x_i) + b))$$

where w are the weights, b is the bias, C is the regularization parameter, x_i is the i-th training example, y_i is the label of the i-th training example, and ϕ is the identity function.

2.4 GCN

Graph Convolutional Networks (GCNs) are utilized to capture the dependencies between cells in the gene expression matrix by considering the graph structure. A GCN learns hidden layer representations that encode both local graph structure and features of nodes. The layer-wise propagation rule, described in detail in [11] and implement through the Spektral package [12], is defined as:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)})$$

where $H^{(l)}$ is the matrix of activations at the l-th layer, $\hat{A} = A + I$ is the adjacency matrix of the graph with added self-connections, $\hat{D}$ is the degree matrix of $\hat{A}$, $W^{(l)}$ is a layer-specific trainable weight matrix, and $\sigma(\cdot)$ is a non-linear activation function ReLU. The input to the GCN model consists of a gene expression matrix X, which is (m, n) and a weighted adjacency matrix with shape (m, m), generated from computing a neighborhood graph of cells with the function scanpy.pp.neighbors. The model is trained in a 70

2.5 Transformer

Our goal for each model was to utilize the performer and transformer architecture with a novel embedding that did not rely on expertly curated knowledge. The gene expression for each cell was embedded through multiplying the gene expression matrix with $d_{model}l$ (performer: 128,transformer: 48)-dimensional embedding, consisting of learnable and randomly initialized embeddings with the counts per each gene concatenated to the end of each embedding to reach the total amount of layers

in each embedding. The positional encoding for each gene was not needed, as gene expression is coordinated in cellular programs but not ordered in a given cell.

With our unique embedding, we split each embedding into 70

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_{model}}})V$$

Multi-head attention is implemented as below:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$, $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

The fully-connected feed-forward network consists of a linear transformation, a ReLU activation, and another linear transformation:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

where $W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$, $b_1 \in \mathbb{R}^{d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$, $b_2 \in \mathbb{R}^{d_{model}}$. The input and output have dimension d_{model} dimension of the hidden layer d_{ff} is set to 200. All sub-layers in the model output a d_model dimensional output. We used cross entropy loss as our loss function. We average the output across these transformer encoder layers and use a linear classification layer to generate a (m, k) matrix of classification predictions.

2.6 Performer

The performer architecture [13] approximates full-rank attention through Fast Attention Via positive Orthogonal Random features (FAVOR+). The attention matrix A is approximated using the form:

$$A(i, j) = K(q_i^T, k_j^T)$$

with q_i and k_j being the query and key row vectors, respectively. The kernel K is approximated using the following expectation:

$$K(x, y) = \mathbb{E}\left[\phi(x)^T \phi(y)\right]$$

where ϕ is the random feature map. For the lower rank randomized matrices Q' and K', with rows $\phi(q_i^T)^T$ and $\phi(k_i^T)^T$, respectively, the approximation is:

$$\text{Att}_\phi(Q, K, V) = D^{-1}(Q'(\mathbb{K}')^T)V), \quad D = \text{diag}(Q'(\mathbb{K}')^T 1_l)$$

For the lower rank randomized matrices Q' and K', with rows $\phi(q_i^T)^T$ and $\phi(k_i^T)^T$, respectively, the fast attention approximation is: $\widehat{Att_\leftrightarrow}(Q, K, V) = \hat{D}^{-1}(Q'((\hat{K}')^T V)), \quad \hat{D} = \text{diag}(Q'((\hat{K}')^T \mathbf{1}_L))$

Lastly for the Orthogonal Random features portion, the random feature map ϕ for functions $f1, \ldots, f_l$, function $h(x)$ and the deterministic vectors $w_1, \ldots, w_m \sim^{iid} D$ for some distribution is defined below:

$$\phi(x) = \frac{h(x)}{\sqrt{m}}\left[f_1(w_1^T x), \ldots, f_1(w_m^T x), \ldots, f_l(w_1^T x), \ldots, f_l(w_m^T x)\right]$$

For performer, $d_{model} = 64$ and 50 epochs, but dimensions of performer for the other layers are the same as the transformer model. As above, 4 layers for the performer encoder are used. Each layer has two-head self-attention and a 4 layer fully connected feed forward network. The output is averaged across performer encoder layers and input into a linear classifier to result in a (m, k) matrix of classification predictions.

2.7 Performance Metrics

We evaluated the performance of the models through the F1 score, averaged per each class for each model:

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

where precision is defined as $TP/(TP + FP)$ and recall is defined as $TP/(TP + FN)$

3 Results

3.1 Traditional Methods

We began with two scRNAseq dataset (P. falciparum) from the Malaria Cell Atlas [14] and data from the liver cancer cell atlas [15] to test out different supervised classification models. We began with these two datasets to compare a small non-human parasite dataset with a large human data, which we anticipated would perform well, as it has many cells and genes for each class. In addition, standard methods have been shown to perform well on this dataset [16] so it serves as a good baseline for later comparisons.

Principal Component Analysis was performed to visualize how the given cells vary compared to one another. The cells are well stratified by developmental stage (Figure 1). To test the performance of two classic models for classification tasks, we used logistic regression (LR) and support vector machines (SVM), implemented through Scikit-learn [10]. To incorporate more graph based information in our model, we used an implementation of a graph convolutional network in the Spektral package [12], based on the work of Kipf and Welling [11]. The data was split into a 70% training and 30% test split. The input to the model was the filtered and normalized gene expression matrix along with the cell type labels.

For each model, we labeled the cells in the PCA as correct (blue) or incorrect (orange). Incorrectly predicted cells were spread throughout the PCA and did not concentrate within a certain cell type across all models (Figure 2).

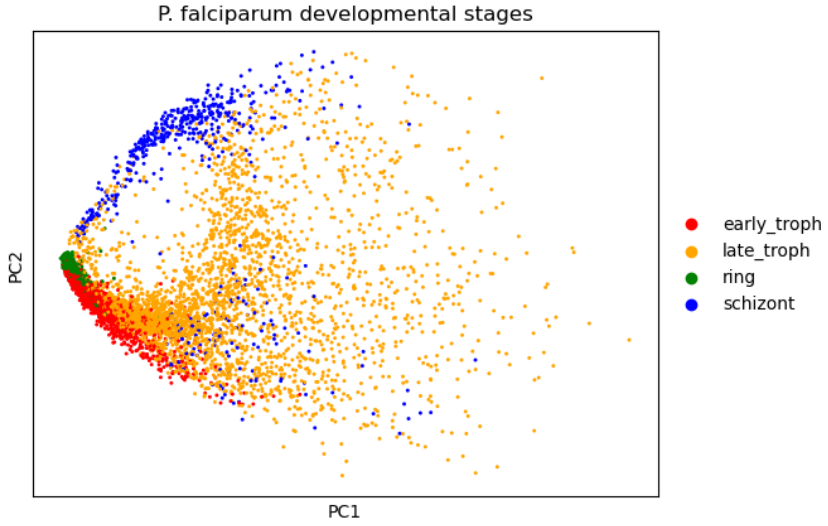
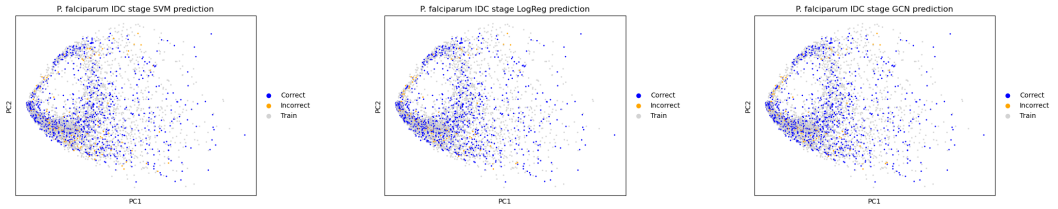


Figure 1: Principal components analysis of *P. falciparum* single cell RNA sequencing data, colored by lifecycle stage.



(a) PCA of predictions from SVM model.

(b) PCA of predictions from logistic regression model.

(c) PCA of predictions from graph convolutional network model.

Figure 2: Model performance visualized on PCA plots.

3.2 Transformer Models

<Talk about perforance of them>

We computed the F-1 score across all methods. Each model achieved F1 scores above 0.80 across at least two cell types. The transformer model is an expection <talk about high training loss variance, might need to adjust learning rate>(Figure 3).

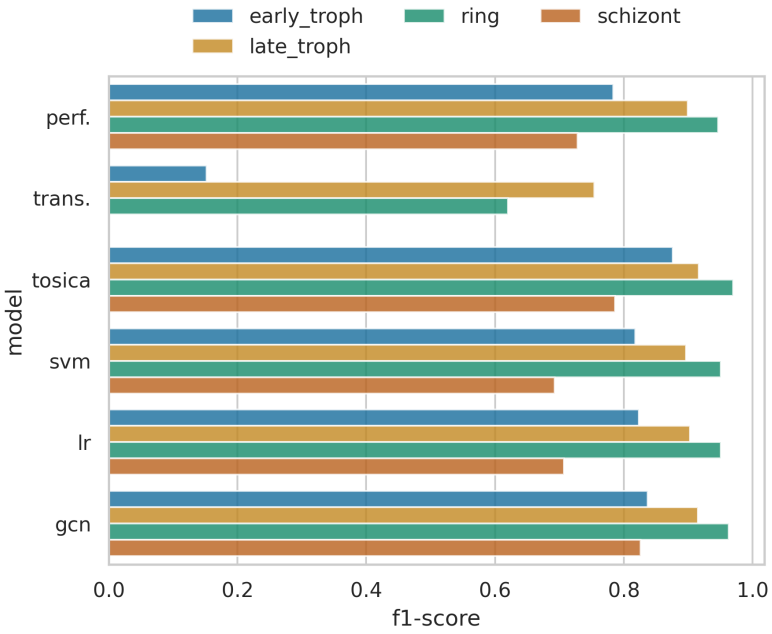
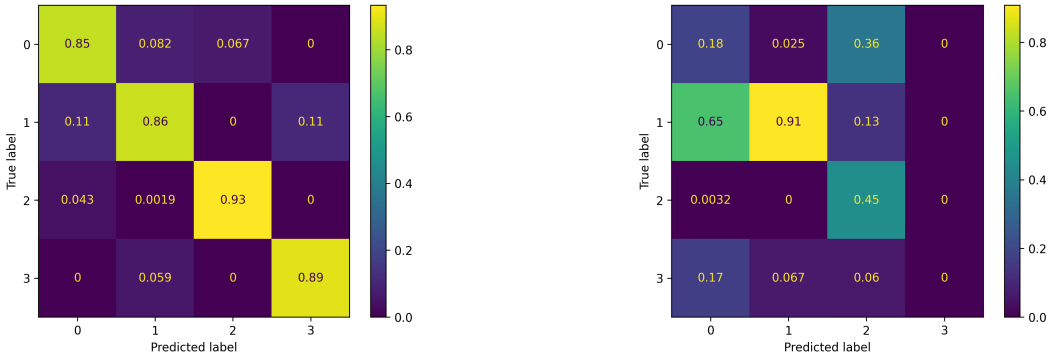


Figure 3: Prediction accuracy of models on *P. falciparum* single cell RNA sequencing data.



(a) PerformerClassifier

(b) TransformerClassifier

Figure 4: Confusion matrices comparing our Performer and Transformer based classifiers in *P. falciparum*. Classes 0-3 represent: early troph.; late troph.; ring; and schizont stages, respectively

3.3 Datasets

We collated a list of scRNAseq datasets, which represent non-human, non-mouse datasets from a wide array of organisms and reference human and mouse datasets (Table 1). The organisms include malaria, parasitic worm *Schistosoma mansoni*, the cause of African sleeping sickness *Trypanosoma brucei*, tick borne parasite *Babesia microti*, zebrafish, and non-human primates. Through training and testing our model on a variety of organisms, we hope to create a model that transcends the need

for manual knowledge and can find biomarkers of different cell types to better characterize these diseases.

Table 1: Summary of datasets that will be used in this study.

Host	Source	Reference	Cells
Parasite	GitHub	Howick et al. [14]	6,737
Human	GEO: GSE151530	Ma et al. [17]	56,721

4 Conclusion

Biomarker discovery is an inherently difficult procedure. An ideal biomarker is: generalizable across disease state and patient demographics; easy to measure (e.g. a simple blood draw); robust; and based upon a strong biological foundation related to the underlying disease etiology. Identification typically occurs through an iterative process of hypothesis generation followed by costly experimental validation. The procedure is often more akin to the 'art' of gambling, rather than science. Biology is immensely complex. Most potential biomarkers identified through computational methods quickly fall apart once interrogated in the lab. Additionally, there is a general feeling much of the low-hanging fruit (e.g. BRCA1/2 in breast cancer) has already been plucked. Therefore optimistic researchers frequently hope for new methods that better handle biological intrices and afford more precision.

Naturally, this has led many to explore the use of transformer-based models due to their massive success in generative AI. To help address the utility of such models in genomic biomarker discovery, your authors skeptically jumped on the (perhaps already fading) transformer and attention bandwagon. We were motivated by the broad observation that biological models being published, such as scBERT or TOSICA, seem to miss the overall point and context of their ideal use. In pursuit of raw predictive power, they tend to build overly complex models that rely on huge datasets and external priors, such as scBERT's use of gene2vec [18]. While performant, such models may miss the chance to generate novel encodings that could help better inform downstream hypothesis generation. They are also less applicable to smaller, often niche datasets, such as malaria, which do not have large amounts of training data available, nor do they have curated knowledge available for masking.

We successfully adapted two new transformer models for single-cell type label prediction. Our models have comparable accuracy to previous work without relying on extremely large training datasets or manually curated knowledge, which will be instrumental in analyzing single cell RNAseq data in less well-studied organisms. However, we remain skeptical, as these models took considerably more computational resources to achieve performance on par with 'traditional', and more directly interpretable, methods such as logistic regression.

References

[1] Ignacio San Segundo-Val and Catalina S. Sanz-Lozano. Introduction to the Gene Expression Analysis. *Methods in Molecular Biology (Clifton, N.J.)*, 1434:29–43, 2016. ISSN 1940-6029. doi: 10.1007/978-1-4939-3652-6_3.

[2] Aisha A. AlJanahi, Mark Danielsen, and Cynthia E. Dunbar. An Introduction to the Analysis of Single-Cell RNA-Sequencing Data. *Molecular Therapy. Methods & Clinical Development*, 10: 189–196, September 2018. ISSN 2329-0501. doi: 10.1016/j.omtm.2018.07.003.

[3] F. Wilcoxon. Individual comparisons of grouped data by ranking methods. *Journal of Economic Entomology*, 39:269, April 1946. ISSN 0022-0493. doi: 10.1093/jee/39.2.269.

[4] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive Integration of Single-Cell Data. *Cell*, 177(7):1888–1902.e21, June 2019. ISSN 1097-4172. doi: 10.1016/j.cell.2019.05.031.

[5] Lukas Heumos, Anna C. Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D. Lücken, Daniel C. Strobl, Juan Henao, Fabiola Curion, Herbert B. Schiller,

and Fabian J. Theis. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, 24(8):550–572, August 2023. ISSN 1471-0064. doi: 10.1038/s41576-023-00586-w.

[6] World Health Organization. *World Malaria Report 2022*. World Health Organization, December 2022. ISBN 978-92-4-006489-8.

[7] Jiawei Chen, Hao Xu, Wanyu Tao, Zhaoxiong Chen, Yuxuan Zhao, and Jing-Dong J. Han. Transformer for one stop interpretable cell type annotation. *Nature Communications*, 14(1):223, January 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-35923-4.

[8] Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nature Machine Intelligence*, 4(10):852–866, October 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00534-z.

[9] Isaac Virshup, Sergei Rybakov, Fabian J. Theis, Philipp Angerer, and F. Alexander Wolf. Anndata: Annotated data, December 2021.

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. ISSN 1533-7928.

[11] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017.

[12] Daniele Grattarola and Cesare Alippi. Graph Neural Networks in TensorFlow and Keras with Spektral, June 2020.

[13] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking Attention with Performers, November 2022.

[14] Virginia M. Howick, Andrew J. C. Russell, Tallulah Andrews, Haynes Heaton, Adam J. Reid, Kedar Natarajan, Hellen Butungi, Tom Metcalf, Lisa H. Verzier, Julian C. Rayner, Matthew Berriman, Jeremy K. Herren, Oliver Billker, Martin Hemberg, Arthur M. Talman, and Mara K. N. Lawniczak. The Malaria Cell Atlas: Single parasite transcriptomes across the complete Plasmodium life cycle. *Science*, 365(6455):eaaw2619, August 2019. doi: 10.1126/science.aaw2619.

[15] Lichun Ma, Limin Wang, Subreen A. Khatib, Ching-Wen Chang, Sophia Heinrich, Dana A. Dominguez, Marshonna Forgues, Julián Candia, Maria O. Hernandez, Michael Kelly, Yongmei Zhao, Bao Tran, Jonathan M. Hernandez, Jeremy L. Davis, David E. Kleiner, Bradford J. Wood, Tim F. Greten, and Xin Wei Wang. Single-cell atlas of tumor cell evolution in response to therapy in hepatocellular carcinoma and intrahepatic cholangiocarcinoma. *Journal of Hepatology*, 75 (6):1397–1408, December 2021. ISSN 0168-8278. doi: 10.1016/j.jhep.2021.06.028.

[16] Swarnim Shukla, Soham Choudhuri, Gayathri Priya Iragavarapu, and Bhaswar Ghosh. Supervised learning of Plasmodium falciparum life cycle stages using single-cell transcriptomes identifies crucial proteins. *Journal of Bioinformatics and Systems Biology*, 06(01), 2023. ISSN 26885107. doi: 10.26502/jbsb.5107047.

[17] Lichun Ma, Sophia Heinrich, Limin Wang, Friederike L. Keggenhoff, Subreen Khatib, Marshonna Forgues, Michael Kelly, Stephen M. Hewitt, Areeba Saif, Jonathan M. Hernandez, Donna Mabry, Roman Kloeckner, Tim F. Greten, Jittiporn Chaisaingmongkol, Mathuros Ruchirawat, Jens U. Marquardt, and Xin Wei Wang. Multiregional single-cell dissection of tumor and immune cells reveals stable lock-and-key features in liver cancer. *Nature Communications*, 13 (1):7533, December 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-35291-5.

[18] Jingcheng Du, Peilin Jia, Yulin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. Gene2vec: Distributed representation of genes based on co-expression. *BMC genomics*, 20(Suppl 1):82, February 2019. ISSN 1471-2164. doi: 10.1186/s12864-018-5370-x.