A STUDY OF META-LEARNING AND TRANSFER LEARNING APPROACHES FOR CLUSTERING OF SINGLE CELL DATA

A Project Report

submitted by

RISHAB MUNJAL

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

COMPUTATIONAL BIOLOGY
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

26th May 2022

THESIS CERTIFICATE

This is to certify that the thesis titled **APPLICATIONS OF META-LEARNING AND TRANSFER LEARNING TO ANALYSIS OF SINGLE CELL DATA** , submitted by **Rishab Munjal**, to the Indraprastha Institute of Information Technology, Delhi, for the award of the degree of **Master of Technology**, is an original research work carried out by him under our supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree. The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

Dr. Debarka Sengupta
Thesis Supervisor
Associate Professor
Dept. of Computational Biology and
Computer Science
IIIT Delhi, 110020

Place: New Delhi

Date: 26th May 2022

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Debarka Sengupta for his constant support,supervision and guidance. His valuable ideas and insightful comments ensured that the work I did was relevant and insightful. I would also like to express deep gratitude to Ms.Namrata Bhattacharya for her valuable suggestions and ideas during the development of TranSCend. Finally, I am highly indebted to Dinesh Joshi and Kiran Sethi for their valuable inputs and contributions towards TranSCend.

ABSTRACT

KEYWORDS: scRNA-seq ; Clustering ; Transfer Learning ; Meta Learning

Single cell RNA-seq data is an important source for profiling cellular heterogeneity. Clustering is an important step in any single cell pipeline because it allows us to discover unknown cell types.Furthermore,it is possible for data generated in cell studies to be contaminated with cells from other tissues or organs, a fact commonly known as tissue heterogeneity.Failures in detection of tissue heterogeneity affect data interpretability and reproducibility. Efficient clustering approaches aid the study of tissue heterogeneity. Recently, transfer learning approaches like Xu *et al.* (2021) and Sun *et al.* (2015) have shown superior performance in clustering single cell data. These approaches leverage information learned from a source dataset to cluster cells in a target dataset.In this work, we introduce an alternative approach for clustering single cell data based on meta learning. In a nutshell, given data from n tasks $T_1, T_2, ..., T_n$ meta learning aims to solve a new task T_test quickly. Several meta learning methods were applied to single cell data and their performance was compared against two transfer learning based methods namely SCANVI(Xu *et al.* (2021)) and CORAL(Sun *et al.* (2015)). We also tested performance in a more challenging cross species setting where the source data and target data come from different organisms. We also introduce TranSCend, a webserver and online repository dedicated to transfer learning.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **i**

ABSTRACT **ii**

LIST OF FIGURES **v**

ABBREVIATIONS **vi**

1 INTRODUCTION **1**

1.1 RELATED WORK 1

2 META LEARNING **4**

2.1 Model Agnostic Meta Learning 4

2.2 Matching Network 6

2.3 Neural Complexity Measures 7

3 METHODOLOGY **10**

3.1 Datasets 10

3.1.1 Pancreas 10

3.1.2 Lung 10

3.1.3 Mouse and Human 11

3.2 Preprocessing 11

3.3 Experiments 12

3.3.1 MAML 12

3.3.2 Matching Network 13

3.3.3 Neural Complexity 13

3.4 Baselines 14

3.4.1 scANVI 14

3.4.2 CORAL 14

3.5 Metrics 14

3.5.1 Accuracy . . . 15
3.5.2 ARI . . . 15
3.5.3 NMI . . . 15
3.5.4 Confusion Matrix . . . 16

4 RESULTS **17**
4.1 Performance on Pancreas Dataset . . . 17
4.2 Performance on Lung Dataset . . . 18
4.3 Mouse and Human . . . 19

5 TRANSCEND **20**
5.1 Tools . . . 20
5.1.1 Single Cell Variational Inference . . . 20
5.1.2 Transformer Variational Autoencoder . . . 21
5.1.3 Single Cell Annotation Using Variational Inference . . . 21
5.1.4 Total Variational Inference . . . 21
5.1.5 Single Cell Embedded Topic Model . . . 21
5.1.6 scRNA . . . 22
5.1.7 Learning With Autoencoder . . . 22
5.2 Datasets . . . 23
5.2.1 Pancreas . . . 23
5.2.2 Lung, Oesophagus and Spleen . . . 23
5.2.3 Peripheral Blood Mononuclear Cells . . . 24
5.2.4 Prostate . . . 24
5.2.5 Kidney . . . 24
5.3 Contribution . . . 24

6 Conclusion **25**

LIST OF FIGURES

2.1 Model Agnostic meta Learning optimizes representation θ that quickly adapts to new tasks 5

3.1 Flowchart of preprocessing pipeline common to all experiments . . 12

3.2 Flowchart of Training and Inference for MAML 13

3.3 Flowchart of Training and Inference for Matching Networks 13

3.4 Flowchart of Training and Inference for Neural Complexity Measures 13

4.1 Performance in terms of ARI,NMI, and accuracy for various models on the Pancreas dataset. 17

4.2 Performance in terms of ARI,NMI, and accuracy for various models on the Lung dataset. 18

4.3 Performance in terms of ARI,NMI, and accuracy for various models on the Mouse and Human dataset. 19

ABBREVIATIONS

iiitd	Indraprastha Institute of Information Technology, Delhi
MAML	Model Agnostic Meta Learning
MAN	Matching Networks
NC	Neural Complexity Measures
scVI	Single Cell Variational Inference
scETM	Single Cell Embedded Topic Model
scRNA-seq	Single Cell RNA sequencing
LATE	Learning using Auto Encoder
scANVI	Single Cell Annotation Using Variational Inference
trVAE	Transformer Variational Inference
totalVI	Total Variational Inference

CHAPTER 1

INTRODUCTION

Single cell RNA-seq data is an important source for profiling cellular heterogeneity. Clustering is an important step in any single cell pipeline because it allows us to discover unknown cell types. More often than not, data generated in cell studies to be contaminated with cells from other tissues or organs, a fact commonly known as tissue heterogeneity. Tissue heterogeneity, if not properly addressed, has adverse outcomes for data interpretability and reproducibility. This is where clustering can play a vital role. Efficient clustering approaches are instrumental in the study of tissue heterogeneity. This has resulted in a plethora of methods aimed at solving the clustering problem. In the following section, we discuss some of these approaches.

1.1 RELATED WORK

Park and Zhao (2018) introduce a spectral clustering approach for clustering single cell data. Spectral clustering(SC) relies on eigenvectors of the data matrix for clustering. It is especially easy to implement using modern linear algebra libraries. The authors modify the SC framework by imposing a sparse structure on the target matrix. They also utilize multiple doubly stochastic affinity matrices to construct a robust similarity matrix.

Li *et al.* (2020) highlight the computational challenges imposed by batch effects and the ever increasing size of scRNA-seq data. The authors claim that the latter is especially pressing because many existing clustering methods cannot be scaled to large datasets while the former if not dealt with properly leads to complications in downstream analysis and false interpretation of results. Their approach iteratively optimizes the clustering objective function. It is able to remove batch effects provided the differences between batches are not significant compared to the true biological variations. Their approach DESC uses a neural network based autoencoder architecture to map the

original data to a low dimensional latent space. As discussed above, this is done by iteratively optimising the objective function. The procedure works by moving each cell to the nearest cluster centroid. The authors test the scalability of their approach by testing it on the 1.3 Million Mouse dataset generated by 10x Genomics. Compared to other approaches such as Seurat 3.0(But (2018)) whose run times increase exponentially with the number of cells, the run time of DESC increases linearly with the number of cells.

In their work, Wang *et al.* (2021) motivate the development of a novel model architecture by stating that scRNA-seq analysis suffers from major challenges like sequencing sparsity and complex differential patterns in gene expression. They report on the shortcomings of traditional clustering algorithms like SEURAT(But (2018)), MAGIC(van Dijk *et al.* (2018)) and Phenograph(Levine *et al.* (2015)). In a nutshell, these approaches use a K nearest neighbour graph to model cell-cell relationships. This constitutes an oversimplification of the complex cell and gene interactions. They propose an alternative approach based on Graph Neural Networks which have deconvoluted node relationships in a graph through neighbor information propagation in a deep learning architecture.

Xu *et al.* (2021) introduce a semi supervised version of the scVI Lopez *et al.* (2018) called scANVI that leverages existing cell annotations. scVI is able to model the underlying data distribution from gene expression values using stochastic optimization and deep neural networks. The scVI model is applicable to a variety of tasks like batch correction, clustering and visualization. scANVI works by transferring annotations between a source dataset for which annotations are available and a target dataset for which annotations need to be predicted. It scales to large datasets. It provides a completely probablistic interpretation of scRNA-seq data which helps control for technical factors of variation such as over-dispersion, library size discrepancies and zero inflation. On the basis of extensive experimentation, the authors claim that the scVI and scANVI compare favourably to existing state of the art methods for data integration and cell state annotation in terms of accuracy, scalability, and adaptability to challenging settings.

Gan *et al.* (2022) draw extensively on previous work and combine many important ideas into a single architecture namely scDSC.The proposed model consists of a Zero-Inflated Negative Binomial (ZINB) model-based autoencoder, a graph neural network (GNN) module and a mutual-supervised module. The authors highlight the main

challenges faced while clustering scRNA-seq data namely noise impacts, high dimensionality and pervasive dropout events. They present a thorough analysis of the performance of their proposed model on six real world datasets and demostrate that scDSC outperforms the baselines considered in the paper. The performance is compared using clustering accuracy , ARI and NMI.

CHAPTER 2

META LEARNING

The field of meta-learning takes a less conventional approach to solve common machine learning problems like clustering. It is synonymous with learning to learn. The field of meta-learning has gained popularity in the recent years (Hospedales *et al.* (2020)). In stark contrast with conventional Artificial Intelligence methods, meta-learning tries to augment the learning algorithm by assimilating experience from multiple episodes (Hospedales *et al.* (2020)). Of special interest to this work and possibly others that follow are the applications of meta-learning towards solving conventional challenges in deep learning such as generalization. A simple way of thinking about meta-learning is the following : The aim of meta-learning is to come up with a general purpose learning algorithm that generalizes across several tasks. Ideally such an algorithm should also be able to solve each new task better than the previous task. The problem of catastrophic forgetting i.e. poor performance over previous tasks as data from newer tasks is encountered also poses a challenge to meta-learning algorithms indeed any machine learning algorithms. It is not our goal here to give an exhaustive account of meta-learning, therefore we shall restrict ourselves to three algorithms that are the focus of this work. In what follows, we will give a brief account of these algorithms.

2.1 Model Agnostic Meta Learning

The MAML algorithm (Finn *et al.* (2017)) is arguably one of the most landmark algorithms in the field of meta-learning. We begin our account of it by examining the meaning of "Model Agnostic". MAML is model agnostic in the sense that any model that is trainable with gradient descent is compatible with MAML. This allows for a wide variety of models to be used with it. Also, it is applicable to a wide range of learning problems, viz. classification, regression and reinforcement learning. As the reader might recall, the goal of meta-learning is to train a model on several tasks, such that any future tasks can be solved with a minimum number of examples. MAML is

designed with this goal in mind. According to the authors, MAML achieves state of the art accuracy on two few shot image datasets. Finally, MAML is compatible with a variety of loss functions, from differentiable supervised losses to non-differentiable reinforcement learning objectives.

The general MAML algorithm is described below :

Algorithm 1 Model Agnostic Meta Learning

Require: p($\mathcal{T}$) : distribution over tasks
Require: α, β : learning rate hyperparameters
  randomly initialize θ
  while not done **do**
    sample batch of tasks $\mathcal{T}_i$ from p($\mathcal{T}$)
    for all tasks $\mathcal{T}_i$ **do**
      Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ for K examples
      Adapt model parameters using gradient descent $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
  end while

The aim is to find model parameters that are sensitive to changes in a particular task such that small perturbations in these parameters can bring large gains in the performance over any task sampled from p($\mathcal{T}$) , when the parameters are altered in the direction of the gradient of the loss on that task. This is illustrated in the figure below

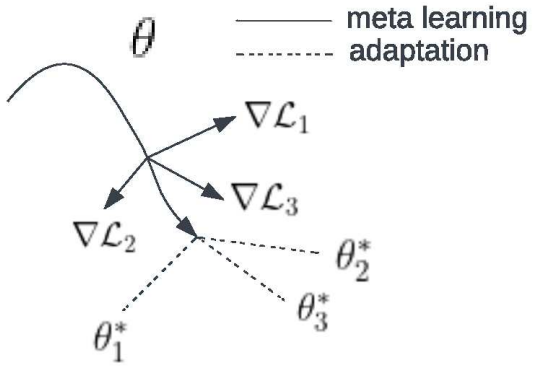

Figure 2.1: Model Agnostic meta Learning optimizes representation θ that quickly adapts to new tasks

To conclude this section, we describe the formulation of the loss function used for the experiments. This is the well known cross entropy loss :

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = \sum_{x^{(i)}, y^{(i)} \sim \mathcal{T}_i} y^{(j)} log f_\phi(x^{(j)}) + (1 - y^{(j)}) log(1 - f_\phi(x^{(j)}))$$

2.2 Matching Network

Matching networks (Vinyals *et al.* (2016)) are an example of a metric based meta learning approach. These models employ attention any memory to enable fast learning. The network is typically trained by showing it small number of examples per task and switching the task for every minibatch. This is done to mimic the conditions the model will face at test time where the requirement will be rapid generalization from a few examples. The authors lay special emphasis on one shot learning. One shot learning refers to the number of examples per class used to train the model at any given time; namely one.

The setup is as follows : We are given a support set S consisting of data. The model then generates a function c_S for each S. This is a mapping S -> $c_S(.)$. The special feature of matching networks is that such a network when fully trained is able to produce accurate labels for unobserved datapoints with necessitating a change in the network in any way.

To make the previous discussion more concrete, let us look closely at the problem that Matching Networks solve. Given a support set S, containing k examples where each example is a tuple (x_i, y_i) Where x_i and y_i are data and labels respectively, the objective is to map from S to classifier $c_S(\hat{x})$. This classifier when given an unknown data sample $\hat{x}$ will produce a probability distribution over the set of possible output classes $\hat{y}$. The support set changes every minibatch which allows the model to learn to generalize to new data quickly and with the aid of a small number of samples. The mathematical form of the model is given below :

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

Here x_i and y_i are labels from the support set. $\hat{x}$ is an unknown input and $\hat{y}$ is the predicted class. The symbol a refers to an attention mechanism that is discussed below.

$$a(\hat{x}, x_i) = \frac{e^{c(f(\hat{x}),g(x_i))}}{\sum_{j=1}^{k} e^{c(f(\hat{x}),g(x_j))}}$$

The above equation gives a precise formulation of the attention mechanism. In a nutshell, to calculate attention, the unknown input is passed through an embedding neural network f typically a MLP and the support set is passed through another similar embedding neural network g. The resulting embeddings are passed through a cosine distance function to calculate similarity followed by a softmax function over the outputs.

The training objective for Matching Nets is as follows:

$$\underset{\theta}{\operatorname{argmax}}\, E_{L\sim T}\left[E_{S\sim L, B\sim L}\left[\sum_{(x,y)\in B} log P_\theta(y|x,S)\right]\right]$$

Here L is the label set sampled from a task T. B represents a batch. S is the support set.

2.3 Neural Complexity Measures

Neural Complexity Measures(NC) (Lee *et al.* (2020)) takes a slightly different approach than the techniques we have already discussed. Neural Complexity Measure is a technique for estimating the generalization gap between train and test time performance on any task. It involves training a model to learn to predict the generalization gap given training and test observations. The trained NC model can then be added to the standard training loss to regularize any task learner in a standard supervised learning scenario. In their paper the authors demonstrate that a trained NC model consistently prevents overfitting and accelerates training. Also they claim that the knowledge gained by the model is more stable across longer learning trajectories. The architecture of the NC model is described below.

Given a task with data $x \in R^D$. The symbols $X_{tr} \in R^{m\times D}, X_{te} \in R^{m'\times D}, Y_{tr} \in R^{m'\times 1}$ denote the train data, test data and train labels respectively. The learner produces outputs $h(X_{tr}), h(X_{te})$.

The train and test data are first embedded using a Fully Connected network f_{enc}.

$$f_{enc}(X_{tr}) = e_{tr} \in R^{m \times d}, f_{enc}(X_{te}) = e_{te} \in R^{m \times d}$$

The embeddings are fed into a muti-head attention layer(Vaswani *et al.* (2017)).The queries,keys and values are $Q = e_{te}, K = e_{tr}, V = W(e_{tr}, [Y_{tr}, 1, \mathcal{L}(X_{tr})]) \in R^{m' \times d} (W \in R^{d \times d \times c})$. Here c denotes the number of classes. The output of the attention layer is

$$f_{attn}(Q, K, V) = e_{att} \in R^{m' \times d}$$

This is passed through a decoder MLP network and averaged.

$$NC(X_{tr}, X_{te}, Y_{tr}, h(X_{tr}), h(X_{te})) = \frac{1}{m'} \sum_{i=1}^{m'} f_{dec}(e_{att}) \in R$$

The Neural Complexity approach consists of two algorithms namely task learning and meta learning. Also the paper prescribes two different models for task learning and meta learning namely the learner and the NC model respectively. In what follows we shall give a brief informal description of the training and inference procedure.

Algorithm 2 Task Learning

Require: NC model, train and test datasets
  Randomly initialize parameters θ of learner h
  while inner iterations not complete **do**
    sample minibatch X_{tr}, X_{te}, Y_{tr}
    $\mathcal{L}_{reg} \leftarrow \hat{\mathcal{L}}_{T,S}(h) + \lambda \cdot NC(X_{tr}, X_{te}, Y_{tr}, h(X_{tr}), h(X_{te}))$ ▷ NC-regularized task loss
    $\theta \leftarrow \theta - \nabla_\theta L_{reg}$ ▷ Gradient Step
  end while
  $G_{T,S}(h) \leftarrow \mathcal{L}_T(h) - \hat{\mathcal{L}}_{T,S}(h)$ ▷ Compute Gap
  return Snapshot H = $(X_{tr}, X_{te}, Y_{tr}, h(X_{tr}), h(X_{te}), G_{T,S})$

Algorithm 3 Meta Learning

Require: Memory Bank
  Initialize parameters ϕ of NC model
  while not converged **do**
    Sample $X_{tr}, X_{te}, Y_{tr}, h(X_{tr}), h(X_{te}), G_{T,S}(h)$ from memory bank
    $\Delta \leftarrow G_{T,S}(h) - NC(X_{tr}, X_{te}, Y_{tr}, h(X_{tr}), h(X_{te}))$
    $\phi \leftarrow \phi - \nabla_\phi \mathcal{L}_{NC}(\Delta)$ ▷ NC's Loss Function
  end while

During training, the task learning and meta learning algorithms are run alternatively.

During the task learning algorithm, the parameters of the learner are updated. During meta learning the parameters of the NC model are updated. When the task learning algorithm runs, snapshots of the learner's training trajectory are stored in a memory bank. Random samples from the memory bank are used to train the NC model during meta learning.

The NC model is trained using Huber Loss. This is defined below:

$$\mathcal{L}_{NC}(\Delta) = \begin{cases} \frac{1}{2}\Delta^2 & for\ \Delta \leq 1 \\ |\Delta| - \frac{1}{2} & \text{otherwise} \end{cases}$$

where $\Delta = G_{T,S}(h) - NC(h)$

During inference the NC model is used as a regularizer when the learner is finetuned on the meta test train dataset. The final performance of the learner without the NC model can then be evaluated on the meta test test dataset.

CHAPTER 3

METHODOLOGY

3.1 Datasets

This section provides information about the datasets used for meta-learning experiments.

3.1.1 Pancreas

The pancreas data used is made up of five publicly available pancreatic islet datasets, namely Baron (Baron *et al.* (2016)), Muraro (Muraro *et al.* (2016)), Segerstolpe (Segerstolpe *et al.* (2016)), Lawlor (Lawlor *et al.* (2016)) and Grun (Grün *et al.* (2016)). Collectively these datasets contain a total of 15,681 cells. The scARches transfer learning toolkit allows the user to download the annotated version of the above dataset. According to the authors, the five datasets mentioned above were obtained from the Scanorama (Hie *et al.* (2019)) dataset which has already assigned its cell types using batch corrected gene expression using Scanorama (Hie *et al.* (2019)) . The dataset obtained is normalized and log transformed using the scanpy (Wolf *et al.* (2018)) preprocessing library. One thousand highly variable genes were selected for experiments.

3.1.2 Lung

The lung data used in experiments is publicly available and published as part of a (Madissoon *et al.* (2019)) . It contains a total of 57,020 cells. The cell types detected include ciliated, alveolar type 1 and 2 cells, fibroblast, muscle and endothelial cells. The last three were from blood and lymph vessels. From the immune compartment NK,T,B,macrophages,monocytes and dendritic cells were detected.Lung club marker genes were also detected in a small number of cells. The dataset was preprocessed using the scanpy (Wolf *et al.* (2018)) library. Two thousand highly variable genes were selected for experiments.

3.1.3 Mouse and Human

The Mouse brain data was obtained from Campbell *et al.* (2017). The primary accession codes associated with this study are (GSE90806 and GSE93374). It contains a total of 21086 cells.The cells belong to the arcuate-median eminence complex (Arc-ME) of the hypothalamus from adult mice. The profiling protocol was Drop-seq. The data was originally clustered using Seurat (Satija *et al.* (2015)).

The human brain data was obtained from Lake *et al.* (2018). The primary accession code associated with this study is (GSE97942). It contains a total of 3042 cells. The cells are sourced from the human adult visual cortex, frontal cortex, and cerebellum. The authors performed experiments to analyze the transcriptonal heterogeneity which yielded thirty five distinct cellular clusters including exitatory , inhibitory , cerebral granule , Purkinje neurons as well as non neuronal cells like endothelial cells , smooth muscle cells, astrocytes, oligodendrocytes and their precursors and microglia.

Both datasets discussed in this section were preprocessed using the scanpy (Wolf *et al.* (2018)) library. One thousand highly variable genes were selected from both datasets for experiments.

3.2 Preprocessing

The figure below summarizes the preprocessing pipeline. The well known single cell analysis toolkit SCANPY (Wolf *et al.* (2018)) was used to do preprocessing. This versatile toolkit automates the entire single cell preprocessing workflow by providing inbuilt functions for filtering,normalization,quality control,visualization,clustering and more.

The first step is acquiring the raw data. Next, we filter outliers based on counts and number of genes expressed using the `filter_cells` function. The next step is to filter out genes based on number of cells or counts using `filter_genes`. For both preceeding steps the functions were run with default parameters. The processed data is then used for calculation of quality control metrics using `calculate_qc_metrics`.

At this stage are primarily interested in filtering mitochondrial content. Once this

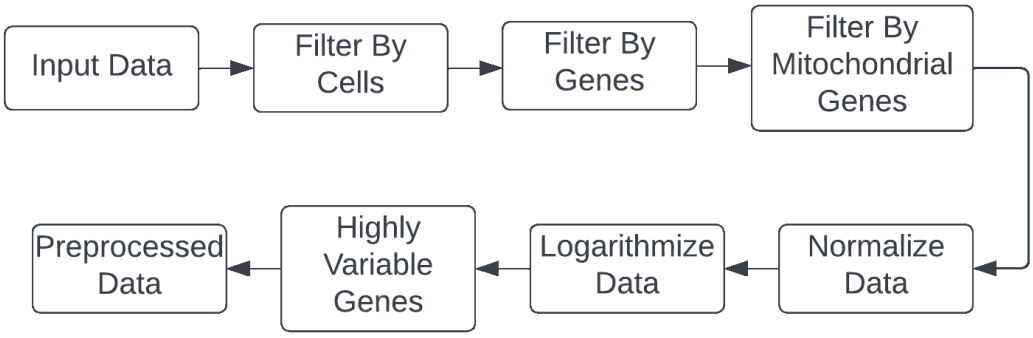

Figure 3.1: Flowchart of preprocessing pipeline common to all experiments

is done, the next step involves normalization,logarithmization and separation of highly variable genes. The corresponding scanpy functions are `normalize_total`,`log1p` and `highly_variable_genes`. This concludes the preprocessing pipeline.

3.3 Experiments

Once the input data has been preprocessed, it can then be used for training and inference. Custom Dataset and Dataloader classes have been written(by author) for this purpose. In contrast with traditional deep learning setup with a train and test dataset, meta learning approaches require that the train and test datasets each be divided into their own train an and test partitions. The train partition of the meta test dataset is really small and contains about 100 data points.

The details of training and inference are different for each meta learning approach.

3.3.1 MAML

During training MAML uses the meta train set and adapts the learner to its train partition which is followed by normal gradient descent based optimization of the learner's parameters on the test partition. During inference the model is first adapted to the train partition of the meta test set followed by standard inference on the test partition.

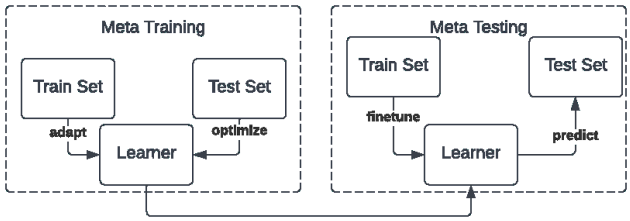

Figure 3.2: Flowchart of Training and Inference for MAML

3.3.2 Matching Network

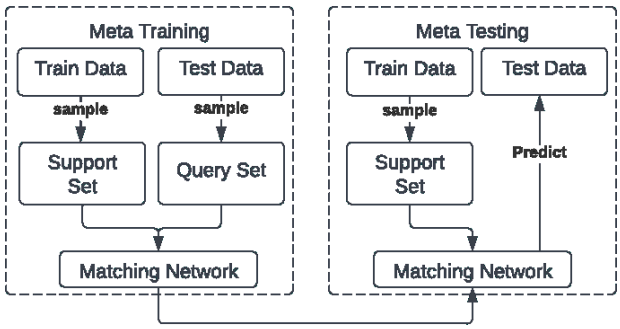

Figure 3.3: Flowchart of Training and Inference for Matching Networks

During training, the support set is drawn from the train partition of the meta train set. The test partition is used as the query set. The model parameters are optimized normally. During inference, the train partition of the meta test set acts as the support set and inference is made on the test partition.

3.3.3 Neural Complexity

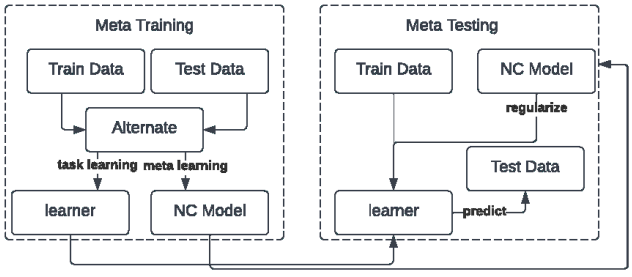

Figure 3.4: Flowchart of Training and Inference for Neural Complexity Measures

During training, the NC model is trained using snapshots of the learner's trajectory.During inference, the train partition of the meta test set is used to finetune the learner. During finetuning, the NC model acts as a regularizer. During inference, the learner is used on its own to make predictions.

3.4 Baselines

In this section, we describe the baselines used to benchmark the performance of the meta learning algorithms.

3.4.1 scANVI

SCANVI (Xu *et al.* (2021)) is a semi supervised variation of SCVI (Lopez *et al.* (2018)). It is designed to leverage existing cell annotations. The authors of SCANVI, on the basis of experiments, state that it compares favourably to state of the art approaches in terms of accuracy and stability.

3.4.2 CORAL

CORAL Sun *et al.* (2015) is a simple and effective method designed to mitigate the effects of domain adaptation. It minimizes domain shift by alignment of second order statistics of the source and target distributions. It is also one of the benchmarks for the SCANVI algorithm.

3.5 Metrics

This section describes the metrics used to assess the performance of all models and baselines used in this work.

3.5.1 Accuracy

Accuracy measures the fraction of correctly classified examples.Accuracy lies in the range of 0 and 1. It is defined by the formula given belowGan *et al.* (2022).

$$Accuracy = \frac{\sum_{i=1}^{N} \delta(true_i, predicted_i)}{N}$$

Here N is the size of the set of true labels. δ is an indicator function which is defined belowGan *et al.* (2022).

$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & otherwise \end{cases}$$

3.5.2 ARI

The Adjusted Rand Index (Rand (1971)) is an adjusted for chance version of the Rand Index. ARI provides a similarity measure between predicted cluster labels and real cluster labels which lies in the range [-1,1].The similarity between the group of true labels L and the group of predicted labels U is captured in a contingency table R. Items in R represent the number of objects shared between L and UGan *et al.* (2022). ARI is defined below.

$$ARI = \frac{\sum_{i,j} \binom{n_{i,j}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/2 - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

Where (.) denotes binomial coefficient, n_{ij} denotes data in contingency table R, a_i is sum of the ith line of R and b_j is the sum of the jth line of R Gan *et al.* (2022).

3.5.3 NMI

Normalized Mutual Information (Strehl and Ghosh (2002)) measures the amount of information obtained about one partition through observing the other partition in a permutation invariant way. The formulation of NMI is provided below Gan *et al.* (2022).

$$NMI = \frac{MI(L,U)}{F(H(L),H(U))}$$

Here L and U are the true and predicted cluster labels.In the formula above, MI calulates the mututal information between L and U.

$$MI = \sum_{i=1}^{N}\sum_{j=1}^{C} p_{i,j} log(\frac{p_{i,j}}{p_i p_j})$$

H represents the entropy which is defined below.

$$H = -\sum_{i=1}^{N} p_i log(p_i)$$

F can be the min,max or mean function.

3.5.4 Confusion Matrix

Confusion Matrices are always square. The (i,j) entry of the matrix indicates that the number of samples with true label being i-th class and predicted label being j-th class.

CHAPTER 4

RESULTS

In this section we discuss results. We will examine how our meta learning models fare against the benchmark methods. The results have been organised by dataset.

4.1 Performance on Pancreas Dataset

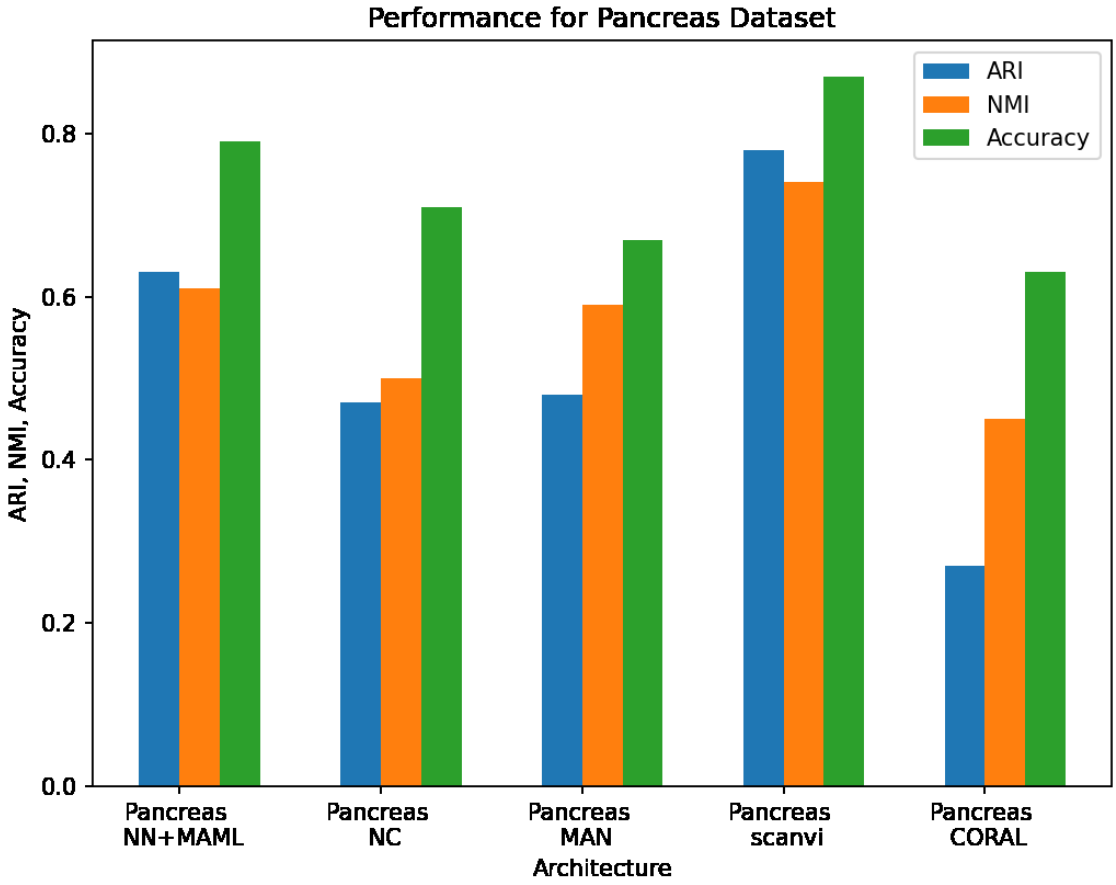

Figure 4.1: Performance in terms of ARI,NMI, and accuracy for various models on the Pancreas dataset.

The results indicate that SCANVI Xu *et al.* (2021) is superior to all other approaches in terms of ARI,NMI and accuracy. All meta learning approaches are able to outperform the second benchmark method i.e. CORAL Sun *et al.* (2015). Among the meta learning

methods, it is surprising to find that the simplest method , the neural net with MAML, is able to perform the best. It outperforms the other meta learning approaches namely Matching Networks (Vinyals *et al.* (2016)) and Neural Complexity Measures (Lee *et al.* (2020)).

4.2 Performance on Lung Dataset

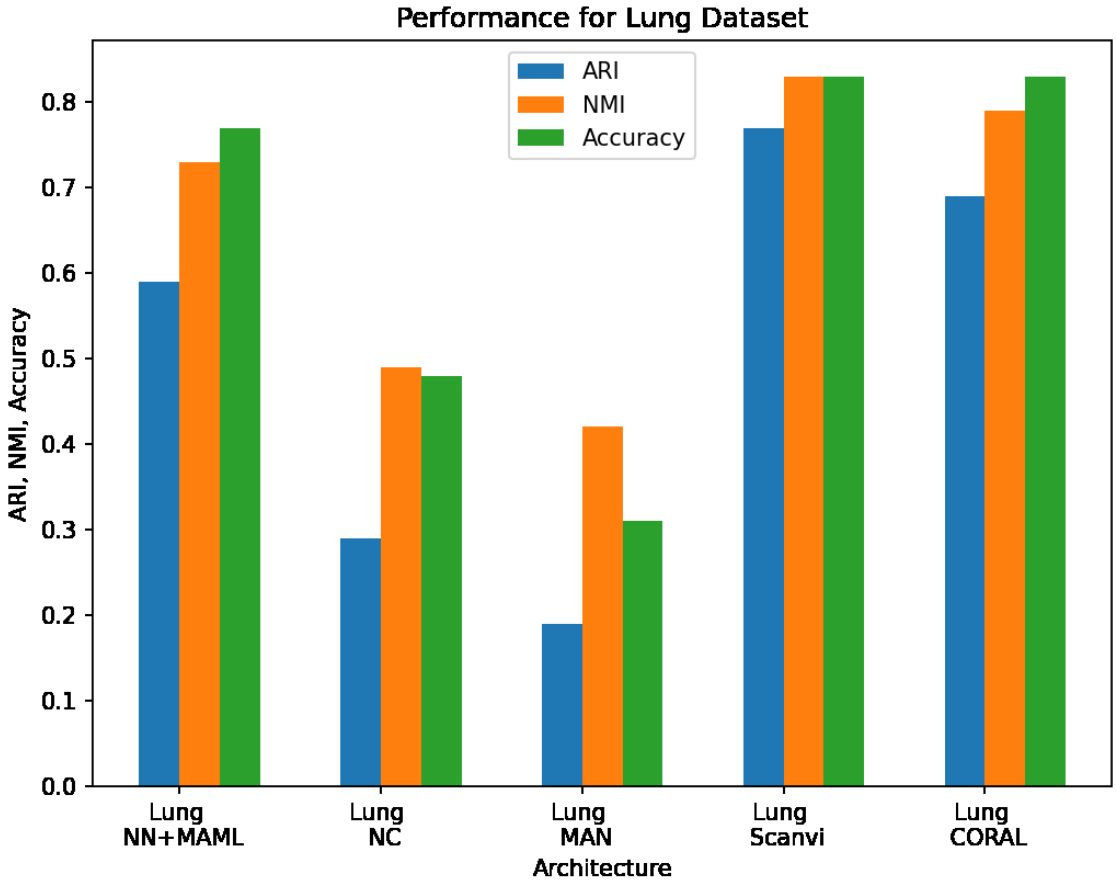

Figure 4.2: Performance in terms of ARI,NMI, and accuracy for various models on the Lung dataset.

The results indicate that for the lung dataset both baselines namely SCANVI (Xu *et al.* (2021)) and CORAL (Sun *et al.* (2015)) outperform all meta learning methods tested. Among the meta learning models the neural net MAML combination provides the best performance. Additionally, it also outperforms other meta learning methods. It can be argued that although it does not outperform either of the benchmarks, it is still very competitive with an accuracy of close to 77 percent compared to 83 percent for CORAL and SCANVI respectively. An unprecedented result of the experiment was that

CORAL which is a simple linear algebra based approach achieved similar performance compared to SCANVI in terms of accuracy although it is inferior to SCANVI in terms of ARI and NMI.

4.3 Mouse and Human

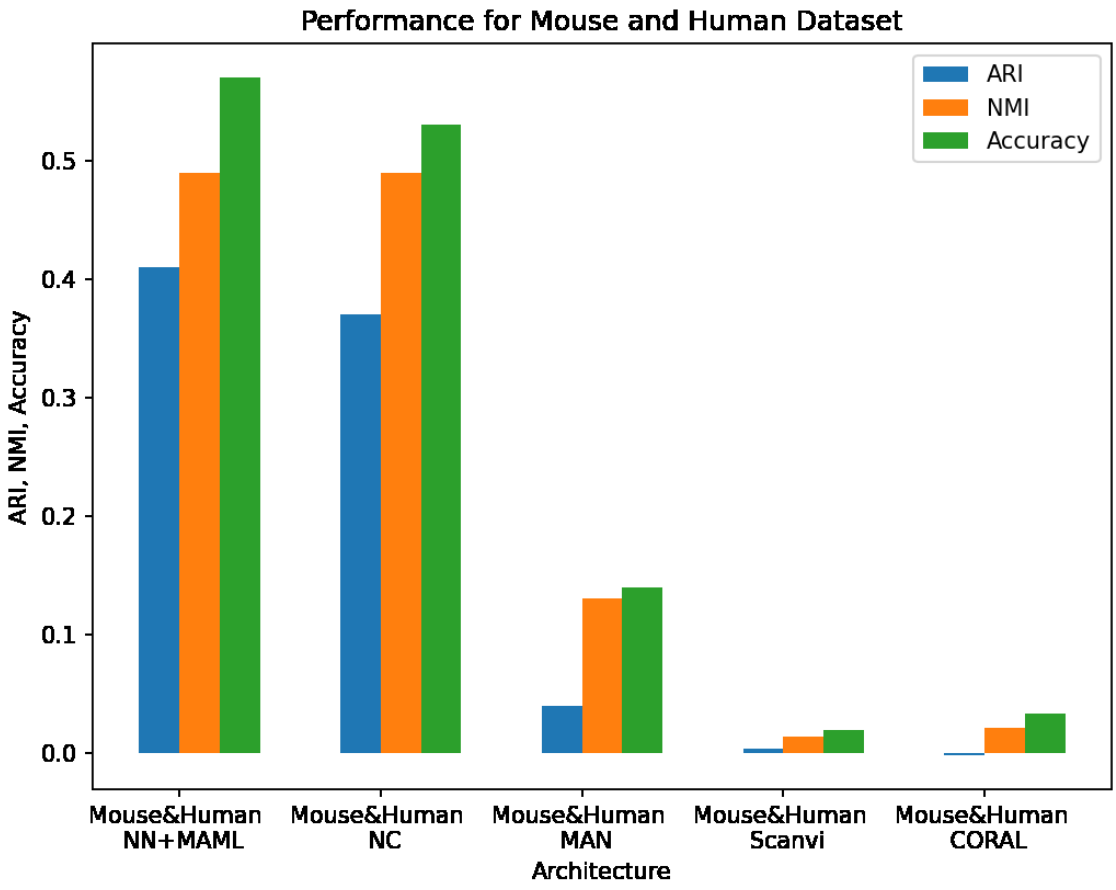

Figure 4.3: Performance in terms of ARI,NMI, and accuracy for various models on the Mouse and Human dataset.

The results in the figure above may be interpreted as a demonstration of the ability of the meta learning methods to adapt to domain shift rather than in terms of their performance relative to other methods. It should be noted that SCANVI Xu *et al.* (2021) is not intended for use in such settings and the results are presented here for the sake of completeness. CORAL Sun *et al.* (2015) is designed to be able to handle domain shift and the results suggest that there is a limit to its generalization ability. Overall, meta learning approaches can generalize quite well even in settings where there is severe domain shift.

CHAPTER 5

TRANSCEND

The importance of single cell analysis for revealing population heterogeneity and biological diversity is well known. Thus tools and techniques that address the challenges of single cell analysis are constantly being developed. Deep learning approaches e.g. SCANVI (Xu *et al.* (2021)) have yielded state of the art results for various single cell analysis tasks like clustering, batch effect removal , visualization etc. But training deep learning methods requires large amounts of data and compute. Thus a need is felt for a resource that provides both pre-trained models with associated metadata as well as examples demonstrating complete end to end use of these models.

In this chapter, we explore TranSCend, a web server dedicated to transfer learning approaches accessible through a user-friendly interface that can be used to explore pre-trained models from different transfer learning pipelines to carry out single cell analysis.

5.1 Tools

We begin by detailing the tools that TranSCend allows us to explore.

5.1.1 Single Cell Variational Inference

scVI (Lopez *et al.* (2018)) is a scalable framework which allows the user to generate a probablistic representation of gene expression data. It uses autoencoders and stochastic optimization to approximate the underlying distribution the the input belongs to. It can be used for a wide array of tasks like clustering, batch correction, visualization and differential expression analysis.

5.1.2 Transformer Variational Autoencoder

The trVAE architecture (Lotfollahi *et al.* (2019)) was developed in part as a response to the shortcomings of Conditional Variational Autoencoders (CVAE). The design of these models does not incentivize learning a compact joint distribution across conditions. The trVAE design overcomes this by matching distributions using maximum mean discrepancy (MMD) in the decoder layer. The result is increased robustness and accuracy as well as improved generalization.

5.1.3 Single Cell Annotation Using Variational Inference

The SCANVI model (Xu *et al.* (2021)) extends the SCVI model discussed above. It is a semi supervised algorithm that is capable of leveraging existing cell annotations. SCANVI is highly flexible and can even work for datasets that are partially annotated.

5.1.4 Total Variational Inference

This model (Gayoso *et al.* (2021)) is a framework for end to end analysis of CITE-seq data. It is useful for analysis tasks such as dimensionality reduction, integration of datasets with different measured proteins, correlation estimation between molecules and testing differential expression .

5.1.5 Single Cell Embedded Topic Model

Compared to bulk RNA-seq, scRNA-seq data is susceptible to batch effects which impact clustering by masking true biological cell signals. Another challenge faced by clustering methods is that partitioning of cell population alone does not have sufficient biological interpretability. The authors introduce scETM(Zhao *et al.* (2021)), a generative topic model which consists of a neural network based encoder and a linear decoder that uses matrix trifactorization. The model simultaneously learns the encoder network parameters and a set of highly interpretable gene embeddings, topic embeddings, and batch-effect linear intercepts from scRNA-seq data. When compared to other methods, scETM enables zero shot knowledge transfer of characteristics learned from a reference

dataset in annotating a target dataset without any further training. It outperforms state of the art methods in cross-tissue cross-species and cross-technology applications

5.1.6 scRNA

The main objective of this work(Mieth *et al.* (2019)) is to demonstrate how information from a large well annotated source dataset can be used to perform clustering on a small sc-RNA dataset. The approach modifies the target dataset while incorporating key information contained in the source/reference dataset of interest, using Non-negative Matrix Factorization (NMF). The modified target dataset is then provided to a clustering algorithm. Thus a transfer of knowledge between reference and target datasets is achieved. However there must be a significant overlap in the cell types of the reference and target datasets.

Owing to the fact that the current work incorporates information from true source labels and uses it to cluster the target dataset, it outperforms baseline methods for all sample sizes of the target dataset when there is a complete overlap in the clustering structures of both datasets. In settings where there is a partial overlap, the method still outperforms baseline methods albeit by a smaller margin. Finally, for setting with no overlap the method when compared to one of the other baselines does not significantly reduce clustering performance compared to de novo clustering of the target data alone.

5.1.7 Learning With Autoencoder

One of the primary shortcomings of commonly used scRNA-seq technologies (e.g. droplet based technologies) is that data generated using these techniques have many zero values. Often, more than 80 percent of measurements across all genes and all cells have a read count of zero which is problematic because of the difficulties it poses for downstream analysis for instance the blurring of differences between subpopulations of cells. Although it is true that some zeros represent no expression, the vast majority are due to the inability to capture the transcript and do not indicate the true expression values. The current work(Badsha *et al.* (2020)) details a set of deep learning algorithms to recover the true gene expression values. The authors claim that their approach outperforms existing imputation methods for sc-RNA data achieving lower Mean Squared

Error in most cases while also being scalable and efficient.

5.2 Datasets

This section provides information about the datasets used to train the models.

5.2.1 Pancreas

The details for this dataset are identical to the one discussed in section 3.1.1.

5.2.2 Lung, Oesophagus and Spleen

The three datasets were introduced by Madissoon *et al.* (2019).

The lung data contains a total of 57,020 cells. The cell types detected include ciliated, alveolar type 1 and 2 cells, fibroblast, muscle and endothelial cells. The last three were from blood and lymph vessels. From the immune compartment NK,T,B,macrophages,monocytes and dendritic cells were detected.Lung club marker genes were also detected in a small number of cells.

The oesophagus data contains 87,947 cells. Over 90 percent of the cells belong to four epithelial cell types upper, stratified, suprabasal, and dividing cells of the suprabasal layer. Additionaly, immune cells like T cells, B cells, monocytes, macrophages, DCs, and mast cells are also present.

The spleen data contained 94,257 cells. All the cells were annotated as immune cells. Annotations include B cells, plasmoblasts, T cells , Natural Killer (NK) cells etc. An interesting aspect of the data, that the authors point out is that the analysis did not detect any stromal cells. This is attributed to the fact that no enzymatic digestion is employed to release them.

5.2.3 Peripheral Blood Mononuclear Cells

This is the well known pbmc68k dataset and was downloaded from 10x genomics(Zheng *et al.* (2017)). The authors use the GemCode single cell technology to perform sequencing. Fresh PBMCs were obtained from a healthy donor. Between 8-9k cells were sampled from each of the 8 channels for a total of 68k cells. The data from multiple runs was merged using the CellRanger pipeline. Clustering analysis was performed using PCA on the top 1000 highly variable genes followed by k-means clustering

5.2.4 Prostate

Dataset was introduced in Henry *et al.* (2018). It contains 98,000 cells from 5 healthy human prostates. The single cell transcriptomes were clustered using a modified version of the Seurat (Satija *et al.* (2015)) pipeline. The study was able to isolate two unknown types of epithelial cells and derive previously unknown markers for these cell types.

5.2.5 Kidney

Dataset was introduced in Stewart *et al.* (2019). It contains 27,203 annotated cells obtained by rigorous quality control. The amjor cell types identified were immune, endothelial, developing nephron epithelium and stromal cell clusters based on canonical marker expression and transcriptional analyses of fetal kidney.

5.3 Contribution

This section describes the author's contributions towards TranSCend. The main contribution was the training of 10 transfer learning models. The details of these models can be found in section 5.1 of this work. Additionally, the author has provided the jupyter notebooks that go with each pretrained model. These notebooks walk the user through the process of setting up and using these models. Finally, the author has provided the model cards for each model. A model card provides dataset specific and model specific information for each model.

CHAPTER 6

Conclusion

In this work we explored the applications of meta learning to single cell clustering. We examined these methods both qualitatively through theory and quantitatively through experiments. Their performance was evaluated against strong baselines and in settings where domain shift was prominent. Finally, we introduced a webserver for transfer learning for single cell data TranSCend and discussed it in some detail.

REFERENCES

1. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.*, **36**(5), 411–420.

2. **Badsha, M. B.**, **R. Li**, **B. Liu**, **Y. I. Li**, **M. Xian**, **N. E. Banovich**, and **A. Q. Fu** (2020). Imputation of single-cell gene expression with an autoencoder neural network. *Quant. Biol.*, **8**(1), 78–94.

3. **Baron, M.**, **A. Veres**, **S. L. Wolock**, **A. L. Faust**, **R. Gaujoux**, **A. Vetere**, **J. H. Ryu**, **B. K. Wagner**, **S. S. Shen-Orr**, **A. M. Klein**, **D. A. Melton**, and **I. Yanai** (2016). A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst.*, **3**(4), 346–360.e4.

4. **Campbell, J. N.**, **E. Z. Macosko**, **H. Fenselau**, **T. H. Pers**, **A. Lyubetskaya**, **D. Tenen**, **M. Goldman**, **A. M. J. Verstegen**, **J. M. Resch**, **S. A. McCarroll**, **E. D. Rosen**, **B. B. Lowell**, and **L. T. Tsai** (2017). A molecular census of arcuate hypothalamus and median eminence cell types. *Nature Neuroscience*, **20**(3), 484–496. ISSN 1546-1726. URL `https://doi.org/10.1038/nn.4495`.

5. **Finn, C.**, **P. Abbeel**, and **S. Levine** (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, **abs/1703.03400**. URL `http://arxiv.org/abs/1703.03400`.

6. **Gan, Y.**, **X. Huang**, **G. Zou**, **S. Zhou**, and **J. Guan** (2022). Deep structural clustering for single-cell RNA-seq data jointly through autoencoder and graph neural network. *Briefings in Bioinformatics*, **23**(2). ISSN 1477-4054. URL `https://doi.org/10.1093/bib/bbac018`. Bbac018.

7. **Gayoso, A.**, **Z. Steier**, **R. Lopez**, **J. Regier**, **K. L. Nazor**, **A. Streets**, and **N. Yosef** (2021). Joint probabilistic modeling of single-cell multi-omic data with totalvi. *Nature Methods*, **18**(3), 272–282. ISSN 1548-7105. URL `https://doi.org/10.1038/s41592-020-01050-x`.

8. **Grün, D.**, **M. J. Muraro**, **J.-C. Boisset**, **K. Wiebrands**, **A. Lyubimova**, **G. Dharmadhikari**, **M. van den Born**, **J. van Es**, **E. Jansen**, **H. Clevers**, **E. J. P. de Koning**, and **A. van Oudenaarden** (2016). De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell*, **19**(2), 266–277.

9. **Henry, G. H.**, **A. Malewska**, **D. B. Joseph**, **V. S. Malladi**, **J. Lee**, **J. Torrealba**, **R. J. Mauck**, **J. C. Gahan**, **G. V. Raj**, **C. G. Roehrborn**, **G. C. Hon**, **M. P. MacConmara**, **J. C. Reese**, **R. C. Hutchinson**, **C. M. Vezina**, and **D. W. Strand** (2018). A cellular anatomy of the normal adult human prostate and prostatic urethra. *Cell Rep.*, **25**(12), 3530–3542.e5.

10. **Hie, B.**, **B. Bryson**, and **B. Berger** (2019). Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nat. Biotechnol.*, **37**(6), 685–691.

11. **Hospedales, T. M.**, **A. Antoniou**, **P. Micaelli**, and **A. J. Storkey** (2020). Meta-learning in neural networks: A survey. *CoRR*, **abs/2004.05439**. URL https://arxiv.org/abs/2004.05439.

12. **Lake, B. B.**, **S. Chen**, **B. C. Sos**, **J. Fan**, **G. E. Kaeser**, **Y. C. Yung**, **T. E. Duong**, **D. Gao**, **J. Chun**, **P. V. Kharchenko**, and **K. Zhang** (2018). Integrative single-cell analysis of transcriptional and epigenetic states in the human adult brain. *Nature Biotechnology*, **36**(1), 70–80. ISSN 1546-1696. URL https://doi.org/10.1038/nbt.4038.

13. **Lawlor, N.**, **J. George**, **M. Bolisetty**, **R. Kursawe**, **L. Sun**, **V. Sivakamasundari**, **I. Kycia**, **P. Robson**, and **M. L. Stitzel** (2016). Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes. *Genome Res*, **27**(2), 208–222.

14. **Lee, Y.**, **J. Lee**, **S. J. Hwang**, **E. Yang**, and **S. Choi** (2020). Neural complexity measures. *CoRR*, **abs/2008.02953**. URL https://arxiv.org/abs/2008.02953.

15. **Levine, J. H.**, **E. F. Simonds**, **S. C. Bendall**, **K. L. Davis**, **E.-A. D. Amir**, **M. D. Tadmor**, **O. Litvin**, **H. G. Fienberg**, **A. Jager**, **E. R. Zunder**, **R. Finck**, **A. L. Gedman**, **I. Radtke**, **J. R. Downing**, **D. Pe'er**, and **G. P. Nolan** (2015). Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell*, **162**(1), 184–197.

16. **Li, X.**, **K. Wang**, **Y. Lyu**, **H. Pan**, **J. Zhang**, **D. Stambolian**, **K. Susztak**, **M. P. Reilly**, **G. Hu**, and **M. Li** (2020). Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nature Communications*, **11**(1), 2338. ISSN 2041-1723. URL https://doi.org/10.1038/s41467-020-15851-3.

17. **Lopez, R.**, **J. Regier**, **M. B. Cole**, **M. I. Jordan**, and **N. Yosef** (2018). Deep generative modeling for single-cell transcriptomics. *Nature Methods*, **15**(12), 1053–1058. ISSN 1548-7105. URL https://doi.org/10.1038/s41592-018-0229-2.

18. **Lotfollahi, M.**, **M. Naghipourfar**, **F. J. Theis**, and **F. A. Wolf** (2019). Conditional out-of-sample generation for unpaired data using trvae. *CoRR*, **abs/1910.01791**. URL http://arxiv.org/abs/1910.01791.

19. **Madissoon, E.**, **A. Wilbrey-Clark**, **R. J. Miragaia**, **K. Saeb-Parsy**, **K. T. Mahbubani**, **N. Georgakopoulos**, **P. Harding**, **K. Polanski**, **N. Huang**, **K. Nowicki-Osuch**, **R. C. Fitzgerald**, **K. W. Loudon**, **J. R. Ferdinand**, **M. R. Clatworthy**, **A. Tsingene**, **S. van Dongen**, **M. Dabrowska**, **M. Patel**, **M. J. T. Stubbington**, **S. A. Teichmann**, **O. Stegle**, and **K. B. Meyer** (2019). scrna-seq assessment of the human lung, spleen, and esophagus tissue stability after cold preservation. *Genome Biology*, **21**(1), 1. ISSN 1474-760X. URL https://doi.org/10.1186/s13059-019-1906-x.

20. **Mieth, B.**, **J. R. F. Hockley**, **N. Görnitz**, **M. M.-C. Vidovic**, **K.-R. Müller**, **A. Gutteridge**, and **D. Ziemek** (2019). Using transfer learning from prior reference knowledge to improve the clustering of single-cell rna-seq data. *Scientific Reports*, **9**(1), 20353. ISSN 2045-2322. URL https://doi.org/10.1038/s41598-019-56911-z.

21. **Muraro, M. J.**, **G. Dharmadhikari**, **D. Grün**, **N. Groen**, **T. Dielen**, **E. Jansen**, **L. van Gurp**, **M. A. Engelse**, **F. Carlotti**, **E. J. P. de Koning**, and **A. van Oudenaarden** (2016). A single-cell transcriptome atlas of the human pancreas. *Cell Syst.*, **3**(4), 385–394.e3.

22. **Park, S.** and **H. Zhao** (2018). Spectral clustering based on learning similarity matrix. *Bioinformatics*, **34**(12), 2069–2076.

23. **Rand, W. M.** (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, **66**(336), 846–850. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356.

24. **Satija, R.**, **J. A. Farrell**, **D. Gennert**, **A. F. Schier**, and **A. Regev** (2015). Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.*, **33**(5), 495–502.

25. **Segerstolpe, Å.**, **A. Palasantza**, **P. Eliasson**, **E.-M. Andersson**, **A.-C. Andréasson**, **X. Sun**, **S. Picelli**, **A. Sabirsh**, **M. Clausen**, **M. K. Bjursell**, **D. M. Smith**, **M. Kasper**, **C. Ämmälä**, and **R. Sandberg** (2016). Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab.*, **24**(4), 593–607.

26. **Stewart, B. J.**, **J. R. Ferdinand**, **M. D. Young**, **T. J. Mitchell**, **K. W. Loudon**, **A. M. Riding**, **N. Richoz**, **G. L. Frazer**, **J. U. L. Staniforth**, **F. A. Vieira Braga**, **R. A. Botting**, **D.-M. Popescu**, **R. Vento-Tormo**, **E. Stephenson**, **A. Cagan**, **S. J. Farndon**, **K. Polanski**, **M. Efremova**, **K. Green**, **M. Del Castillo Velasco-Herrera**, **C. Guzzo**, **G. Collord**, **L. Mamanova**, **T. Aho**, **J. N. Armitage**, **A. C. P. Riddick**, **I. Mushtaq**, **S. Farrell**, **D. Rampling**, **J. Nicholson**, **A. Filby**, **J. Burge**, **S. Lisgo**, **S. Lindsay**, **M. Bajenoff**, **A. Y. Warren**, **G. D. Stewart**, **N. Sebire**, **N. Coleman**, **M. Haniffa**, **S. A. Teichmann**, **S. Behjati**, and **M. R. Clatworthy** (2019). Spatiotemporal immune zonation of the human kidney. *Science*, **365**(6460), 1461–1466.

27. **Strehl, A.** and **J. Ghosh** (2002). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, **3**, 583–617. URL http://dblp.uni-trier.de/db/journals/jmlr/jmlr3.html#StrehlG02.

28. **Sun, B.**, **J. Feng**, and **K. Saenko** (2015). Return of frustratingly easy domain adaptation. *CoRR*, **abs/1511.05547**. URL http://arxiv.org/abs/1511.05547.

29. **van Dijk, D.**, **R. Sharma**, **J. Nainys**, **K. Yim**, **P. Kathail**, **A. J. Carr**, **C. Burdziak**, **K. R. Moon**, **C. L. Chaffer**, **D. Pattabiraman**, **B. Bierie**, **L. Mazutis**, **G. Wolf**, **S. Krishnaswamy**, and **D. Pe'er** (2018). Recovering gene interactions from single-cell data using data diffusion. *Cell*, **174**(3), 716–729.e27.

30. **Vaswani, A.**, **N. Shazeer**, **N. Parmar**, **J. Uszkoreit**, **L. Jones**, **A. N. Gomez**, **L. Kaiser**, and **I. Polosukhin** (2017). Attention is all you need. *CoRR*, **abs/1706.03762**. URL http://arxiv.org/abs/1706.03762.

31. **Vinyals, O.**, **C. Blundell**, **T. P. Lillicrap**, **K. Kavukcuoglu**, and **D. Wierstra** (2016). Matching networks for one shot learning. *CoRR*, **abs/1606.04080**. URL http://arxiv.org/abs/1606.04080.

32. **Wang, J.**, **A. Ma**, **Y. Chang**, **J. Gong**, **Y. Jiang**, **R. Qi**, **C. Wang**, **H. Fu**, **Q. Ma**, and **D. Xu** (2021). scgnn is a novel graph neural network framework for single-cell rna-seq analyses. *Nature Communications*, **12**(1), 1882. ISSN 2041-1723. URL https://doi.org/10.1038/s41467-021-22197-x.

33. **Wolf, F. A.**, **P. Angerer**, and **F. J. Theis** (2018). Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, **19**(1), 15. ISSN 1474-760X. URL https://doi.org/10.1186/s13059-017-1382-0.

34. **Xu, C.**, **R. Lopez**, **E. Mehlman**, **J. Regier**, **M. I. Jordan**, and **N. Yosef** (2021). Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular Systems Biology*, **17**(1), e9620. URL https://www.embopress.org/doi/abs/10.15252/msb.20209620.

35. **Zhao, Y.**, **H. Cai**, **Z. Zhang**, **J. Tang**, and **Y. Li** (2021). Learning interpretable cellular and gene signature embeddings from single-cell transcriptomic data. *Nature Communications*, **12**(1), 5261. ISSN 2041-1723. URL https://doi.org/10.1038/s41467-021-25534-2.

36. **Zheng, G. X. Y.**, **J. M. Terry**, **P. Belgrader**, **P. Ryvkin**, **Z. W. Bent**, **R. Wilson**, **S. B. Ziraldo**, **T. D. Wheeler**, **G. P. McDermott**, **J. Zhu**, **M. T. Gregory**, **J. Shuga**, **L. Montesclaros**, **J. G. Underwood**, **D. A. Masquelier**, **S. Y. Nishimura**, **M. Schnall-Levin**, **P. W. Wyatt**, **C. M. Hindson**, **R. Bharadwaj**, **A. Wong**, **K. D. Ness**, **L. W. Beppu**, **H. J. Deeg**, **C. McFarland**, **K. R. Loeb**, **W. J. Valente**, **N. G. Ericson**, **E. A. Stevens**, **J. P. Radich**, **T. S. Mikkelsen**, **B. J. Hindson**, and **J. H. Bielas** (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, **8**(1), 14049. ISSN 2041-1723. URL https://doi.org/10.1038/ncomms14049.