Improving Cell Type Matching across Species in scRNA-seq Data using Protein Embeddings and Transfer Learning

by

Kirti Biharie

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday July 6, 2022 at 15:00.

Student number:	4722922	
Project duration:	November, 2021 – July, 2022	
Thesis committee:	Prof. Dr. Ir. M. J. T. Reinders,	TU Delft, thesis advisor
	Dr. A. Mahfouz,	TU Delft & LUMC, daily supervisor
	Dr. E. Isufi,	TU Delft

Contents

Abstract

Knowing the relation between cell types is crucial for translating experimental results from mice to humans. Establishing cell type matches, however, is hindered by the biological differences between the species. A substantial amount of evolutionary information between genes that could be used to align the species, is discarded by most of the current methods since they only use one-to-one orthologous genes. Some methods try to retain the information by explicitly including the relation between genes, however, not without caveats. In this work, we present a model to Transfer and Align Cell Types in Cross-Species (TACTiCS). First, TACTiCS uses an natural language processing model to match genes using their protein sequences. Next, TACTiCS employs a neural network to classify cell types within a species. Afterwards, TACTiCS uses transfer learning to propagate cell type labels between species. We applied TACTiCS on scRNA-seq data of the primary motor cortex and the ventral tegmental area. Our model can accurately match and align cell types on these datasets. Moreover, at a high resolution, our model outperforms two state-of-the-art methods, SAMap and CAME. Finally, we show that our gene matching method results in better matches than BLAST, both in our model and SAMap.

1 Introduction

With single-cell RNA sequencing (scRNA-seq) new cell types have been identified in many different domains. Previously, a cell type was characterized only by its morphology, but using scRNA-seq, the expression pattern across ten-thousands genes can be used to describe a cell-type. In particular, scRNA-seq allowed for the identification of new cell types within specific brain regions [1], which results in many cell specific cell types for multiple species. This raises the question: how do these newly identified cell types relate across the species? We are mainly interested in the cell type relations between mice and humans, since the mouse is the most commonly used model for brain research. The relation between cell types is important for translational research to translate experimental results correctly from mouse to human. For example, similar cell types might still differ between the species and these differences can indicate if humans will react differently to a drug compared to mice. The relation between cell types can also be used to study evolution mechanics in depth at the gene expression level, rather than comparing the general morphology of a cell.

In scRNA-seq, we first isolate single cells, sequence RNA fragments, map the fragments to a reference genome and finally retrieve the gene counts. Two challenges arise when comparing the gene counts across the species. The first challenge is that the gene counts contain technical and biological batch effects caused by the different individuals and species. Technical batch effects arise from different sequencing techniques, cell disassociation procedure, sequencing depth, using cells or nuclei, among others. Biological batch effects stem from the fundamental differences between the species, e.g. when the function of a similar cell type differs slightly. The gene counts for this cell type would not be identical for both species, even if the technical batch effects are removed. Several methods, such as Seurat [2] and LIGER [3], can remove the batch effects between datasets, including datasets from different species.

The second challenge is that the genes differ per species due to evolution. Although the gene sets differ, individual genes still relate to each other in some way since the two species share a common ancestor. Due to duplication events, however, multiple genes can map to the same gene in the other species. Additionally, some genes might not relate to genes in the other species at all. Without a gene matching method, the gene counts can not be compared across species.

Current methods that match cell types across species can be divided into two groups based on how they solve the gene matching problem. The first group of methods use only the one-to-one orthologous genes, which are genes with exactly one match in the other species. Methods such as scANVI [4], MetaNeighbour [5] and LAMbDA [6] belong to this group as well as the majority of cross-species analyses [7, 8, 9, 10]. These methods remove a substantial amount of information to characterize a cell type and additionally remove the evolutionary information between the species. The second group of methods try to overcome this loss of information by explicitly modelling the relation between the genes, and this group include SAMap [11], CAME [12], Kmermaid [13] and C3 [14]. In particular, SAMap uses protein sequence similarity to solve the gene matching problem, however as we show later, the performance of SAMap detoriates for higher resolutions and is not optimized for large datasets. CAME constructs a heterogenous graph consisting of cells and genes with edges connecting homologous genes. Although the gene relations are included in the graph, CAME discards the expression values of the homologous genes. Thus even the methods in the second group do not solve the gene matching problem without caveats.

Here we introduce a method to Transfer and Align Cell Types in Cross-Species (TACTiCS), that overcomes the limitations of the current methods by taking the gene relations into account. TACTiCS consists of four steps: 1) matching genes based on the protein sequences, 2) creating a shared feature space by mapping expression values with the gene matches, 3) training within-species cell type classifiers and 4) matching cell types by swapping the classifiers. In this paper we show that we can use TACTiCS to correctly match human and mouse brain cell populations from the primary motor (M1) cortex and from the ventral tegmental area (VTA).

2 Methods

TACTiCS (Figure 1) takes as input two single-cell (sc) or single-nucleus (sn) RNA sequencing datasets, for species A and B. For both datasets we are using, the cells are labeled at multiple resolutions (Section 2.9). We define a the major resolution and a subtype resolution and use these resolutions for the gene selection. We first create embeddings for every gene using ProtBERT and match the genes across the species based on the embedding distances. Using the gene matches we impute the gene expression for genes in the other species to create a shared feature space. We train a classifier for each species on the shared feature space to predict cell types and finally we switch the classifiers to transfer the cell types.

2.1. Creating gene embeddings using ProtBERT

Choosing only the one-to-one orthologous genes discards a lot of information. To overcome this we implemented a gene-matching strategy to retain information as much as possible (Figure 1a). For all genes for both species we retrieved the protein sequences from UniProt [15], filtered to only the curated sequences from Swiss-Prot [16]. We input the protein sequences to ProtBERT [17], a Transformer-based [18] model, to create embeddings. ProtBERT was pretrained to predict the masked amino-acids of a protein sequence, with the objective to model the protein language. Every protein sequence is prepended with a classifier (CLS) token that can be used in downstream tasks. ProtBERT produces an embedding of length 1024 for the CLS token, in addition to an embedding for every position in the sequence. We use the embedding for the CLS token to represent the whole gene. From here on we refer to the protein embeddings created by ProtBERT as gene embeddings. Protein sequences longer than 2500 amino-acids (<2% of all sequences) are truncated to the first 2500, since larger protein sequences do not fit in the VRAM of a GPU.

2.2. Retrieving gene matches

We compare every human gene embedding to every mouse gene embedding with the cosine distance and create a distance matrix with the pair-wise distances (Figure 1a). For 18k human genes and 16k mouse genes, the distance matrix contains 288M distances or possible gene matches. We apply a filter on the distance matrix to generate gene matches. The filter consists of two parts to include both the one-to-one orthologous genes as well as the many-to-many orthologous genes. First, for every gene we retrieve the cross-species gene that is closest in the embedding space. This is done for both species, thus resulting in two sets of gene matches. We take the intersection of the two sets to filter for reciprocal matches between the two species. These reciprocal matches represent the one-to-one gene matches. Next, we filter the original distance matrix using a threshold to retrieve an additional set of matches, where distance ≤ threshold. The default value of the threshold is set to 0.0002, justified in Section 3.2.2. This set of matches can include multiple matches for a single gene and is thus more similar to the many-to-many gene matches. This set does not include all of the one-to-one matches since not all one-to-one matches have distance ≤ threshold. Thus, we take the union of the one-to-one matches and threshold matches as the final gene matches, reducing from 288M gene matches to 15k gene matches.

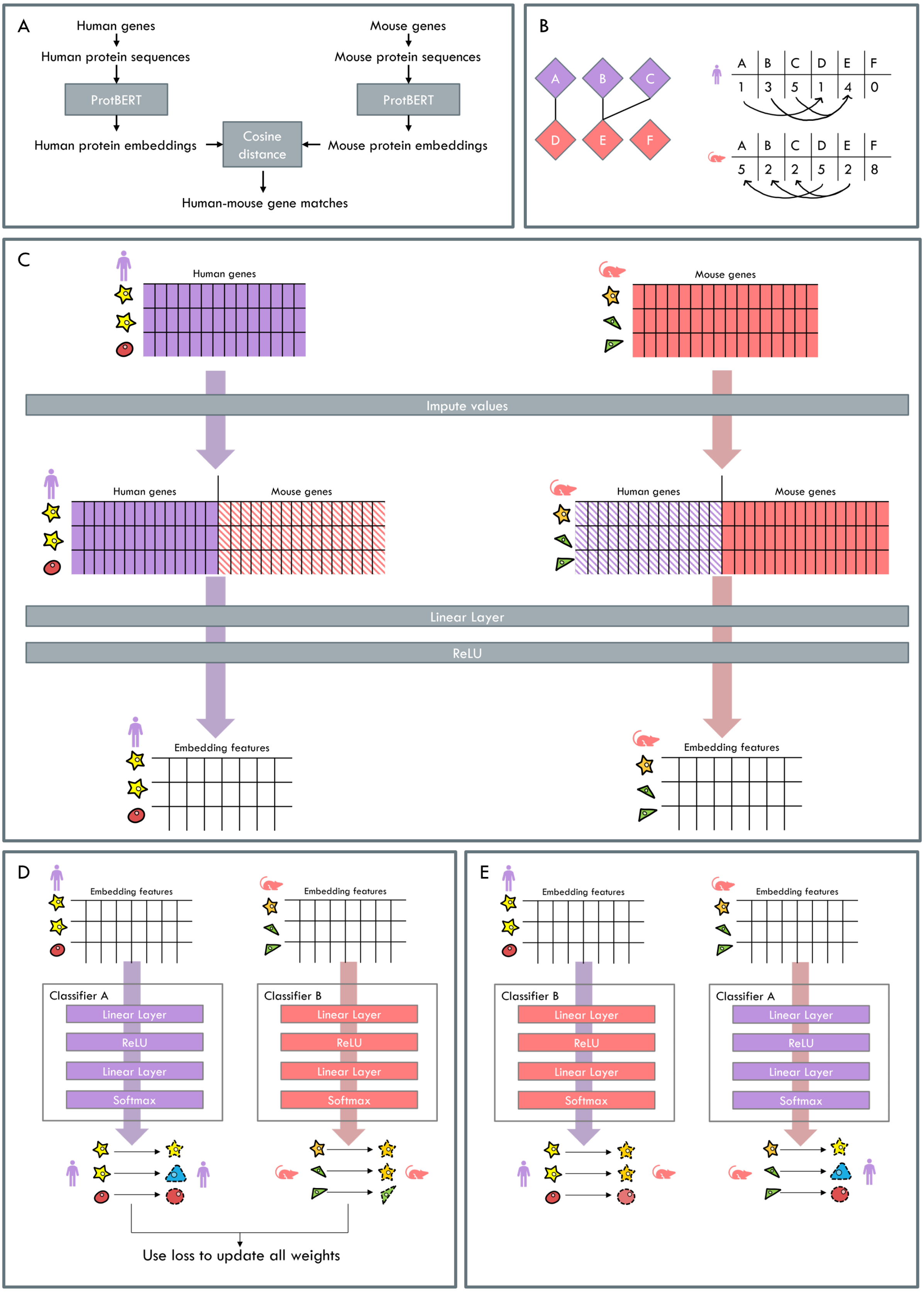
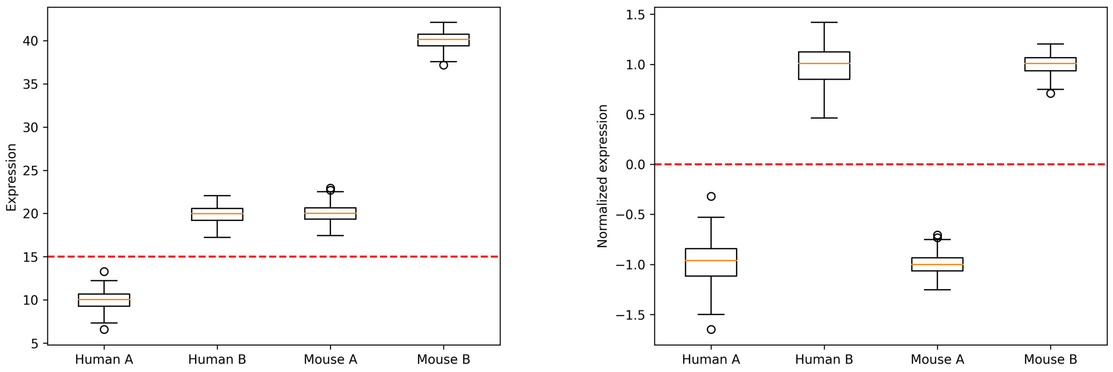


Figure 1: Schematic overview of TACTiCS. (a) Matching of genes using ProtBERT on protein sequences. (b) Bipartite graph of gene matches. Gene expression is imputed by taking weighted average from connected genes in the bipartite graph. (c) Creating cell embeddings using linear layer on shared feature space. (d) Classifying within-species cells. Predictions are are used to update loss in classifier and embedding layer. (e) Classifying cross-species cells using transfer learning. The predictions are used to match cell types.



(a)

(b)

Figure 2: Simulated expression of a human gene for two human cell types and a mouse gene for two mouse cell types. (a) Unnormalized expression. (b) Normalized expression. The expression is normalized per gene by subtracting the mean and dividing by the standard deviation. Decision boundaries for the human cell types are shown in red.

2.3. Gene selection

Single-cell datasets often contain the expression for ten-thousand genes and training a model on all genes is time consuming. Not all genes are equally important and it is more computationally efficient to only use the genes that provide the most valuable information. Informative genes include Highly Variable Genes (HVG), which vary across cells, and Differentially Expressed Genes (DEG), which vary across groups of cells. We restrict our method to the DEG since we are more interested in the variation across cell types than the variation of individual cells. Additionally, we restrict our method to the genes that we retrieved protein sequences for. We calculate the DEG for a species using Scanpy `rank_gene_groups()` [19] with the Wilcoxon rank-sum, which ranks the genes based on how much they differ across two groups. Here, the first group consists of a single cell type and the second group is chosen in two ways. The first way is to identify the top 50 DEG separately for each cell type at the major and subtype resolutions to capture the larger differences. Here, the second group consists of all other cell types at the used resolution. This set of DEG cannot distinguish a cell type from a different cell type that is similar, since the similar cell type is only a small portion of the cell types that were compared to. The second way is to identify the top 30 DEG for each cell type at the subtype resolution (subtype) by comparing the subtypes within each major cell type and thus here the second group consists of the other cell types within the same major cell type. This way the second group of DEG can capture the differences between similar cell types at the subtype resolution by only comparing similar cell types. We define DEG_A as the union of all DEG acquired by both methods for species A.

Although DEG_A can distinguish between the cell types of species A, they are not guaranteed to distinguish cell types of species B, e.g. when species B is clustered at a higher resolution. We hypothesize that if a gene characterizes a cell type of species A, the matching genes in species B characterize the corresponding cell type in species B. To distinguish between all cell types, we keep the genes from species A that match to DEG_B and vice versa. This corresponds to filtering out gene matches that consist of no DEGs. The final set of genes for species A, G_A, consists of the DEG_A with a match in species B and the genes that are a match to DEG_B. Likewise we create DEG_B and G_B. This again brings down the number of gene matches from 15k to 2k.

2.4. Creating a shared feature space

We construct a gene-gene bipartite graph with edges between G_A and G_B (Figure 1b). The weight for an edge between gene $i \in G_A$ and gene $j \in G_B$ is calculated as:

$$e_{ij} = max\left(1 - \frac{cosine(h_i^{\text{ProtBERT}}, h_j^{\text{ProtBERT}})}{\text{max\_cosine\_distance}}, 0\right) \tag{2.1}$$

where h_i^{ProtBERT} is the ProtBERT embedding for the protein sequence of gene i and $cosine()$ calculates the cosine distance between two embeddings. We divide by the maximum cosine distance of the gene matches to scale the weights to the interval [0, 1] and finally invert the weights such that higher weights are assigned to more similar genes. The edge weight for gene pairs that do not match is set to 0, i.e. the gene pairs that do not match the criteria from Section 2.2.

The expression matrices X^A and X^B contain the normalized expression for genes G_A and G_B respectively. The expression matrices were first normalized with the natural logarithm on the count matrices. Next, the matrices were normalized again with the mean and standard deviation for each gene. Normalizing the matrices per gene removes some of the biological differences between the two species and thus makes it easier to compare the two matrices. Figure 2 shows simulated expression for a mouse and human gene. There is a clear relation between cell type A and B in both species, namely the unnormalized expression of cell type B is double the expression of cell type A. However, the decision boundary for the human cell types classifies all mouse cells as cell type B. With the normalized values the human decision boundary works correctly for the mouse cell types. Thus normalizing the expression allows for relative comparison of gene expression in this case. However, we cannot compare X^A and X^B directly since the gene sets G_A and G_B are different. To this end we construct the augmented expression matrices $\overline{X^A}$ and $\overline{X^B}$ that both contain gene expression for $G_A \cup G_B$ according to SAMap [11]:

$$\overline{X^A_{ij}} = \begin{cases} X^A_{ij}, & \text{if } j \in G_A \\ \frac{1}{\sum_{m \in G_A} e_{jm}} \sum_{k \in G_A} e_{jk} X^A_{ik}, & \text{if } j \in G_B \end{cases} \tag{2.2}$$

where $\overline{X^A_{ij}}$ is the expression of cell i from species A for gene j. The expression of within species genes does not change, thus X^A is a direct subset of $\overline{X^A}$. For a cross-species gene, i.e. gene $j \in G_B$, we impute the expression using the expression of similar within-species genes. Specifically, the expression is calculated as the weighted average of the matching genes' expression, where the weight is based on the ProtBERT embedding distance. To normalize the edge weights, we divide by the sum of all edges gene j is connected to. The resulting matrices $\overline{X^A}$ and $\overline{X^B}$ span the same feature space, i.e. $G_A \cup G_B$, and can thus be compared directly.

2.5. Cell embeddings

We create cell embeddings with a linear layer followed by a ReLU (Figure 1c). The linear layer has $|G_A| + |G_B|$ input features and creates an embedding of length 32:

$$h_i^{emb} = \sigma(W^{emb}\ \overline{x_i^A} + b^{emb}) \tag{2.3}$$

where h_i is the embedding of cell i and $\overline{x_i^A}$ is the augmented expression vector. The weight matrix W^{emb} and bias b^{emb} are shared for both species.

2.6. Classifiers

We train a classifier for each species consisting of two linear layers followed by a softmax activation function (Figure 1d). For the classifier of species A, the class probabilities can be calculated as:

$$h_i^{A,0} = \sigma(W^{A,0}\ h_i^{emb} + b^{A,0}) \tag{2.4}$$

$$h_i^{A,out} = \text{softmax}(W^{A,out}\ h_i^{A,0} + b^{A,out}) \tag{2.5}$$

where $h_i^{A,0}$ are the hidden features with length 16 and $h_i^{A,out}$ are the class probabilities for cell i. $h_i^{A,out}$ only includes class probabilities for cell types from species A and likewise for $h_i^{B,out}$. Unlike the embedding layer, the weight matrices of the classifiers are not shared. While training, the cells are only input to the classifier belonging to that species. Thus we calculate $h_i^{A,out}$ for cells of species A and $h_i^{B,out}$ for cells of species B.

2.7. Training

2.7.1. Loss

We update the weights in the network using a classification loss and embedding distance loss. The classification loss is calculated separately for each classifier and only takes cells into account that belong to the corresponding species. We use the weighted cross-entropy loss between the predictions and targets as below:

$$L_{cls_A} = \sum_{i=1}^{N_A} \sum_{t=1}^{T_A} w_t Y_{it}^{LS} \ln(h_{it}^{A,out}) \tag{2.6}$$

where L_{cls_A} is the classification loss for species A, N_A and T_A are the number of cells and cell types in species A. w_t is the weight for cell type t, explained further below. $h_{it}^{A,out}$ is the probability that cell i belongs to cell type t according to the classifier. The one-hot encoded targets Y are modified with label smoothing to prevent overfitting and improve stability:

$$Y_{it}^{LS} = \begin{cases} 1-\epsilon, & \text{if } Y_i = t \\ \frac{\epsilon}{T-1}, & \text{otherwise} \end{cases} \tag{2.7}$$

where ϵ (= 0.1) controls the smoothness. To prevent overfitting on some classes, the weight w_t for every cell type is updated every epoch based on the accuracy of that class:

$$w_t = (1 - \text{acc}_t) * \alpha + 1 \tag{2.8}$$

where acc_t is the accuracy of class t in the current epoch. α is a hyperparameter that controls the influence of the accuracy. In our experiments, $\alpha = 9$ such that the weights are in the interval [1, 10]. If we do not add 1 to the final weight, a low accuracy class can be infinitely more important than a high accuracy class, which is undesirable behaviour and leads to instability. As a result of the updated cross-entropy loss, classes with a low accuracy will have a higher weight in the next epoch and thus a larger influence on the loss. In that sense, the cross-entropy loss is similar to the focal loss [20], which gives a higher weight to individual samples with a bad prediction, rather than classes.

Because of the biological differences between the species, a cell will usually be more similar to cells from the same species, than cross-species cells. To ensure that the cell embeddings are aligned across species, we introduce the embedding distance loss:

$$L_{emb} = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{j \in N_i^{\text{cross}}} \text{euclidean}(h_i^{emb}, h_j^{emb})}{\sum_{j \in N_i^{\text{within}}} \text{euclidean}(h_i^{emb}, h_j^{emb})} \tag{2.9}$$

where N_i^{cross} and N_i^{within} are the 20 nearest cross-species and within neighbours for cell i. $\texttt{euclidean}(i, j)$ is the Euclidean distance between the cell embeddings for cells i and j. The embedding distance loss aims to even out the distance from a cell to within-species cells and the distance to cross-species cells. If the embedding distance loss is minimized, the embedding space is well mixed between both species.

The final loss is a combination of the two classifier losses and the embedding distance loss:

$$L = L_{cls_A} + L_{cls_B} + \beta L_{emb} + \gamma \|\theta\|_2^2 \tag{2.10}$$

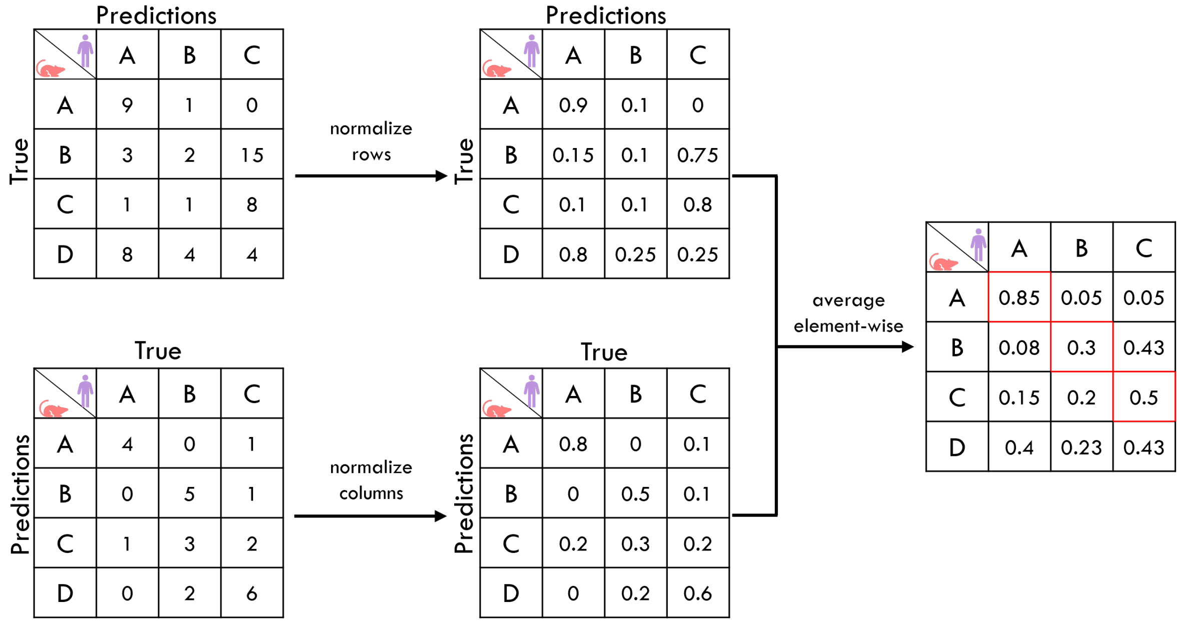
where β is the weight of the embedding distance loss, with a default value of 0.01. A higher value for β increases the integration of the species, but removes the ability to distinguish cell types. Vice versa a lower value increases the distinction between cell types but separates the species. θ consists of all parameters in the model, and is used for the L2 regularization to prevent overfitting. γ is the weight of the L2 norm, which is set to 0.01.

2.7.2. Mini-batches

Since scRNA-seq datasets can be rather large, the neural network is trained in batches. A batch size of 5000 is used to speed up training while still having enough cells per cell type. Cells are sampled from the whole dataset for every batch and smaller cell types are oversampled to deal with class imbalance. More specifically, every cell is assigned a probability N_A/n_t or N_B/n_t where N_A is the number of cells for species A and n_t is the number of cells for cell type t. These probabilities are then used to sample individual cells for each batch.

Table 1: Hyperparameters with used values.

Name	Value	Description
ProtBERT embedding threshold	0.0002	Threshold for embedding distance to select gene matches
Cell embedding length	32	Length of cell embeddings created by linear layer
ϵ	0.1	Smoothness factor for label smoothing
α	9	Scaling factor for classes with low accuracy
β	0.01	Weight of embedding distance loss
γ	0.01	Weight of L2 penalty
Learning rate	0.001	Learning rate in Adam optimizer



	A	B	C
A	9	1	0
B	3	2	15
C	1	1	8
D	8	4	4

	A	B	C
A	0.9	0.1	0
B	0.15	0.1	0.75
C	0.1	0.1	0.8
D	0.8	0.25	0.25

	A	B	C
A	4	0	1
B	0	5	1
C	1	3	2
D	0	2	6

	A	B	C
A	0.8	0	0.1
B	0	0.5	0.1
C	0.2	0.3	0.2
D	0	0.2	0.6

	A	B	C
A	0.85	0.05	0.05
B	0.08	0.3	0.43
C	0.15	0.2	0.5
D	0.4	0.23	0.43

Figure 3: Combined confusion matrix for three human cell types and four mouse cell types. The confusion matrices with the raw counts are normalized such that the true cell type slices sum up to 1. The normalized confusion matrices are combined by taking the element-wise average. The diagonal entries (red) are used to calculate the ADS and recall.

Smaller cell types will have a higher probability to be included and overall every cell type will occur equally often in the batch. Oversampling small cell types is also a prequisite for the embedding distance loss. Without oversampling, the embedding distance loss clusters cross-species cell types together based on the number of cells, rather than the cell type characteristics.

The dataset is split into train/validation/test sets as 0.7/0.15/0.15. Only the cell types in the training set are oversampled and thus the distribution of cell types in the training set differs from the validation and test sets. TACTiCS is trained for 200 epochs on the training subset. The Adam optimizer is used to update the weights with a learning rate of 0.001. A single epoch takes around 10 seconds for 30 batches with a AMD Ryzen 7 5800X CPU, GeForce GTX 1070 and 16GB of memory.

2.7.3. Hyperparameters

The individual steps of TACTiCS all contain multiple hyperparameters, thus resulting in a large set of hyperparameters for the whole model. The hyperparameters and their chosen values are summarized in Table 1.

2.8. Transferring labels

After the neural network is trained, the cell types are transferred by using the classifiers on the species they were not trained on (Figure 1e). That is, we calculate $h_i^{B,out}$ for cells of species A and $h_i^{A,out}$ for cells of species B. This results in a predicted cell type for every cell. To aggregate the information to cell types, we calculate the fraction of cells that are predicted to be each cross-species cell type, which is summarized with a confusion matrix. We perform this step for both species, resulting in two confusion matrices. To aggregate the information of both directions we average the two confusion matrices element-wise to create a combined matrix (Figure 3). An entry in this matrix describes how much the cell types were predicted in either direction. Thus high values indicate reciprocal matches while medium values may indicate one-sided matches.

2.9. Datasets

2.9.1. M1 data

TACTiCS is most thoroughly tested on the Allen Brain Data [8], consisting of snRNA-seq data taken from the primary motor cortex (M1) of human and mouse. These datasets consist of 76k human cells and 159k mouse cells. Every cell is labeled at four different resolutions: Class, Subclass, Cross-species and RNA-cluster. The Class resolution distinguishes between GABAergic, Glutamatergic and Non-neuronal cells. Only 5% of the human cells are labeled as non-neuronal while the mouse non-neuronal cells make up 36% of the mouse dataset. The distribution of the labels varies significantly at the lowest resolution and this effect increases at higher resolutions. The Subclass resolution contains more specific labels, i.e. 20 human cell types and 23 mouse cell types. At this higher resolution, the Astro cells make up <1% of the human cells and 11% of the mouse cells. Additionally, the cell types Meis2, Peri and SMC are present only in the mouse dataset. The largest cell types at this resolution are L2/3 IT for human and L5 IT for mouse making up 32% and 21% of the species' cells respectively. The Cross-species resolution is harmonized across both species and thus all labels occur in both species. Cells labeled with species-specific cell types at the Subclass resolution are labeled as NA at the Cross-species resolution. We use the harmonized labels as a reference when evaluating the cell type matches of TACTiCS. The Cross-species resolution consists of 46 cell types for both species. The highest resolution, RNA-cluster, consists of 128 human labels and 116 mouse labels which are not harmonized between the species. Therefore, we do not use the RNA-cluster resolution to evaluate TACTiCS. For the M1 data, the Subclass resolution serves as the major resolution and the Cross-species resolution is used as the subtype resolution.

2.9.2. VTA data

We were provided with unpublished snRNA-seq datasets from the UMC Utrecht (Basak's lab). These datasets consist of 40k human cells and 20k mouse cells taken from the Ventral Tegmental Area (VTA). These datasets are labeled across two resolutions which are not harmonized across the species. The major resolution consists of 8 human and 14 mouse cell types with no straightforward matching between them. For example, both datasets contain the cell type DA but the human dataset also contains the cell type DA/Glut/GABA while the mouse datasets contains DA/Glut and DA/GABA. Additionally, the mouse Ependyma and Glutamergic cell types do not occur in the human data, and some mouse cell types are marked as unknown, i.e. Astrocyte (unknown) and neuron (unknown). The largest cell type for both species is Oligodendrocyte, making up 70% of the human cells and 29% of the mouse cells. The subtype resolution consists of 34 human and 46 mouse cell types such as Oligodendrocyte 1-7. Since these cell types are not harmonized and there is no clear matching, we do not use the subtype resolution to evaluate TACTiCS.

2.10. Evaluation

The combined matrix cannot be evaluated using standard metrics for confusion matrices, such as precision or F1 score, since we cannot distinguish between false positives and false negatives after they are combined. Instead, we compare the scores on the diagonal of the combined matrix. This excludes the species-specific cell types. We define the Average Diagonal Score (ADS) as the average score of the diagonal entries. A high ADS indicates that many cell types are correctly matched and with high confidence. A lower ADS indicates more incorrect matches, or less confident matches, e.g. when a cell type matches to multiple cell types in the other species. However, the ADS does not give an indication on how many cell types are correctly matched. To this end, we define the recall as the fraction of diagonal entries where the score is the highest for both its row and column. The recall indicates the fraction of common cell types that are correctly matched, but does not indicate the confidence of those matches.

Next we consider the following example of how the ADS and recall is calculated. The example in Figure 3 constructs the combined matrix for three human cell types and four mouse cell types. The ADS is calculated for the common cell types only, i.e. cell types A, B and C. The ADS is $(0.85+0.3+0.8)/3 = 0.55$. The diagonal entries for cell types A and C, 0.85 and 0.5 respectively, are the highest for their respective rows and columns. Indeed the medium value 0.5 denotes a one-way match, namely mouse cell type C is predicted to be human cell type C, while human cell type C is predicted to be mouse cell type D. For this example the recall becomes $2/3 = 0.67$, indicating that 67% of the matches are correctly retrieved. We use the ADS and recall to compare TACTiCS to two methods, SAMap and CAME.

The embedding loss should promote the mixing of species in the embedding space. We calculate the Local Inverse Simpson's Index (LISI) [21] to evaluate this quantitatively. The LISI describes whether clusters of cells are well-mixed across a categorical variable, in this case the species. First, we compute the UMAP for the embeddings of the test set. We annotate the UMAP coordinates with the label of the species and use the annotated UMAP coordinates as input for `compute_lisi()` with a default perplexity of 30. This outputs a score for every cell in the interval $[1, 2]$, denoting how many species are in the neighbourhood of that cell. We take the average of all cells to summarize the LISI. A mean LISI near 2 indicates that the cells are well-mixed across the species, while a score near 1 indicates that the UMAP consists of species-specific clusters.

2.10.1. SAMap

The shared feature space of TACTiCS is based on SAMap. SAMap constructs a bipartite graph with the genes of the species, where the edges are weighted by the protein sequence similarity. The gene similarity is calculated with BLAST, which we substituted with the ProtBERT embedding distance in TACTiCS. The bipartite graph is used to impute the expression of genes in the other species. Next, SAMap updates the weights of the bipartite graph to increase the correlation of connected genes across clusters of cells. Once the correlation converges, the final weights of the bipartite graph are used to match cross-species clusters with similar expression in the shared feature space.

2.10.2. CAME

CAME transfers cell types across species using a heterogenous graph consisting of cells and genes. For every node in the graph an embedding is created and this embedding is updated using graph convolution. CAME creates embeddings for cells and genes, while TACTiCS only constructs embeddings for cells. The implementation for CAME is not public. To compare to CAME, we implemented an architecture based on the architecture described in [12]. We evaluated several variants and the architecture of the best variant can be found in Section 5.1.

2.11. Implementation

TACTiCS is implemented in Python 3.7. Pytorch was used for the model architecture. The scRNA-seq data is stored in the Anndata [22] objects, containing the gene expression and annotation for the cells and genes. The implementation of TACTiCS is available at https://github.com/kbiharie/TACTiCS.

3 Results

We evaluated the different parts of TACTiCS separately. First we evaluated the general performance of the model on the M1 data. Second, we compared our gene matching method using ProtBERT to BLAST. Next, we investigated the influence of the updated cross-entropy loss and embedding distance loss on the training performance. We also compared TACTiCS to existing methods and finally we applied TACTiCS on the VTA data. The ADS and recall of all experiments are summarized in Table 4 for the Subclass resolution and Table 4 for the Cross-species resolution on the M1 data.

3.1. Performance of TACTiCS

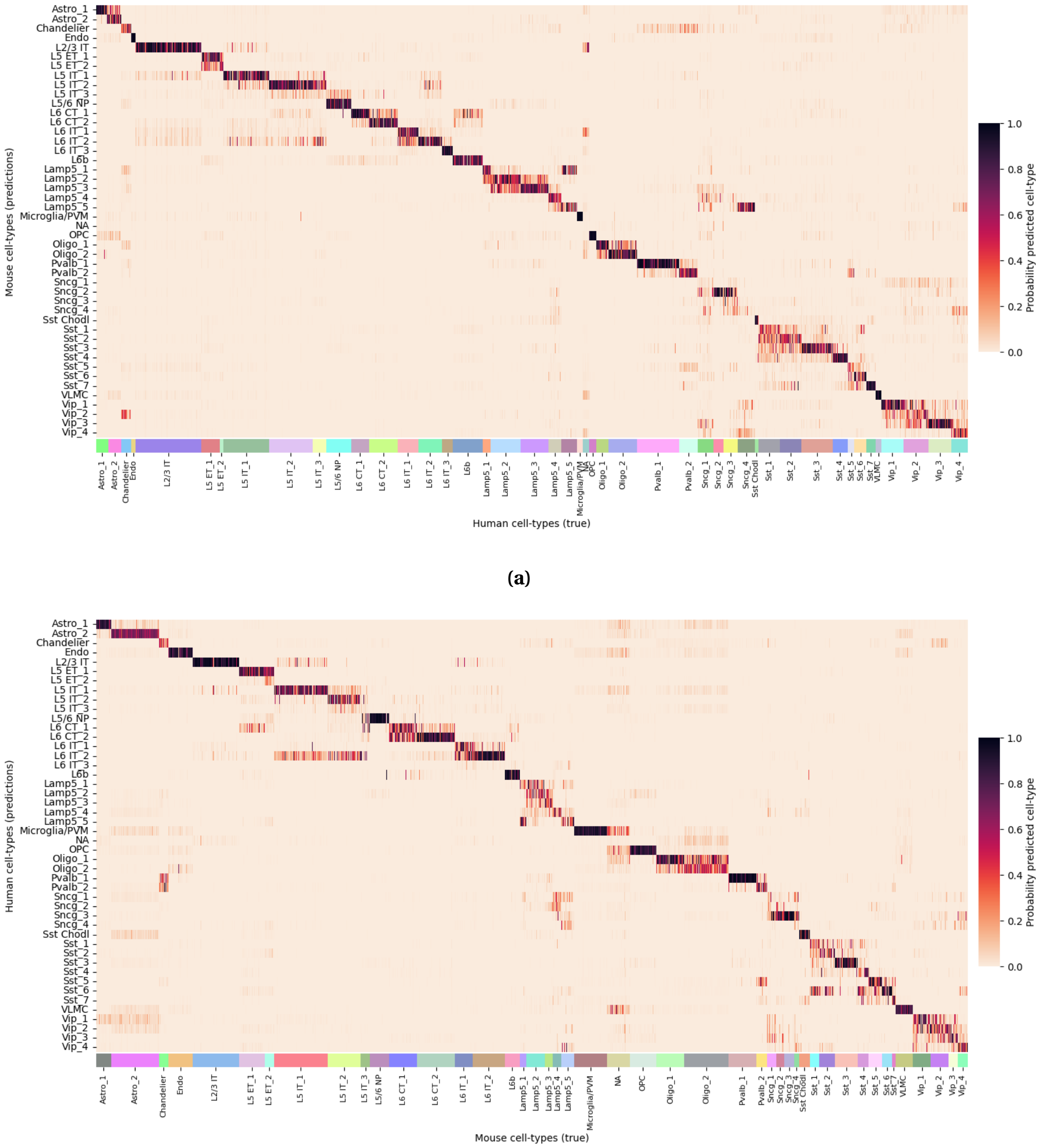
3.1.1. Better predictions for larger and distinct cell types

We investigated the cell type predictions at the cell level using the M1 data since it includes the harmonized labels defined by the authors (Figure 4). For most cells there is one confident prediction, that is a single cell type with a high probability, such as the Astro 1 and Astro 2 mouse cells. Other cells have less confident predictions, such as the subtypes from Vip, Lamp5 and Sst. These cell types are relatively small with a high number of subtypes, in this case Vip, Lamp5 and Sst and Vip have four, five and seven subtypes respectively. The predictions are spread across subtypes within the same major type for these subtypes. For instance, the probabilities for mouse Lamp5 2 are spread across all human Lamp5 subtypes, but with a bias towards human Lamp5 1 and Lamp5 2. Sncg also has a relatively high number of subtypes, namely four, but the probabilities are not spread across multiple cell types. Instead, the human Sncg cells are predicted to be Sncg 2, with the exception of Sncg 4 which is predicted as Lamp5 5. Mouse Scng 2 is not the largest mouse Sncg subtype. The mouse Sncg cells are predicted to be human Sncg 3, which again is not the largest human Sncg subtype. Other small cell types are correctly and confidently matched, such as Endo, Microglia/PVM, OPC and Sst. These cell types have in common that they are the only subtype within their respective major type, indicating that they are quite distinct from all other subtypes. Thus TACTiCS works better for larger, more distinct cell types and has less confident predictions for small and very similar cell types, even if oversampling is used.

3.1.2. The shared feature space aligns similar cell types

We tested how good the shared feature space, so without training the network, can already align similar cell types for the M1 data. We created the augmented expression matrices according to according to Section 2.4. For every cell type per species, we calculated the mean augmented expression across the cells. Next, for every cross-species cell type pair we calculated the cosine distance of the mean augmented expression. We evaluated this for the Cross-species and Subclass resolutions (Figure 5). For cell types in the Subclass resolution there is usually a high similarity with one cross-species cell type and a no similarity with the other cross-species cell types. Additionally, the cell type with the highest similarity is the correct cell type in most cases. This does not hold for human and mouse Sncg. The human Sncg is similar to mouse Sncg, Lamp5 and Vip, and likewise the mouse Sncg is similar to human Sncg, Lamp5, and Vip.

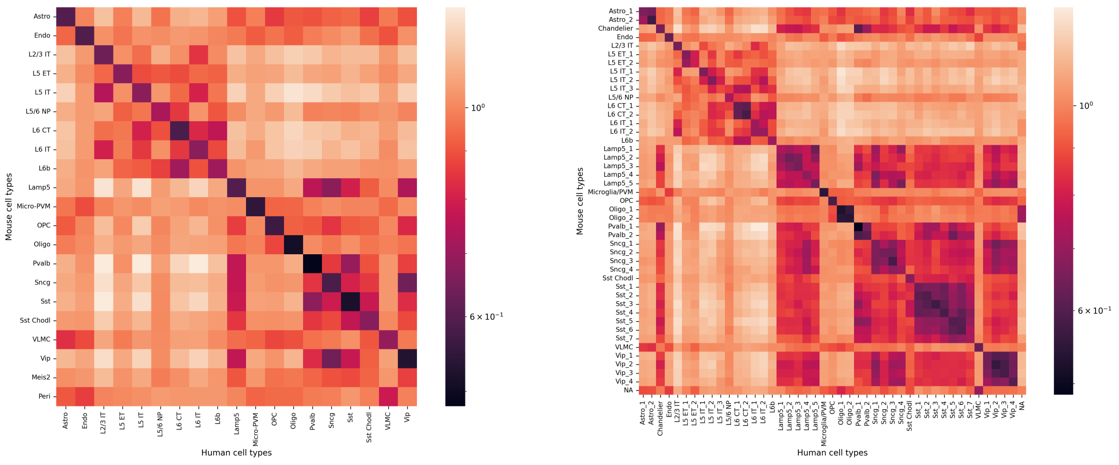
For the higher resolution, Cross-species, there are clusters of cell types with a high similarity, rather than cell type pairs. We cannot directly establish cell type matches between the subtypes for these clusters, since there is no clear peak visible. These clusters, i.e. Vip, Lamp5 and Sst, directly relate the the major cell types without a confident prediction from Section 3.1.1. Even though these clusters are still visible in the



(a)

(b)

Figure 4: Cell type probabilities for M1 data at Cross-species resolution for (a) human cells and (b) mouse cells. After training both classifiers, the cells were input to the classifier of the other species to transfer the cell types and generate the probabilities.



(a) **(b)**

Figure 5: Shared feature space distance for resolution Subclass (a) and Cross-species (b) on the M1 data. The distance is calculated as the cosine distance of the mean shared feature space. Lower values (darker colours) denote a higher similarity between the cell types. The distance is log normalized to make smaller differences more pronounced.

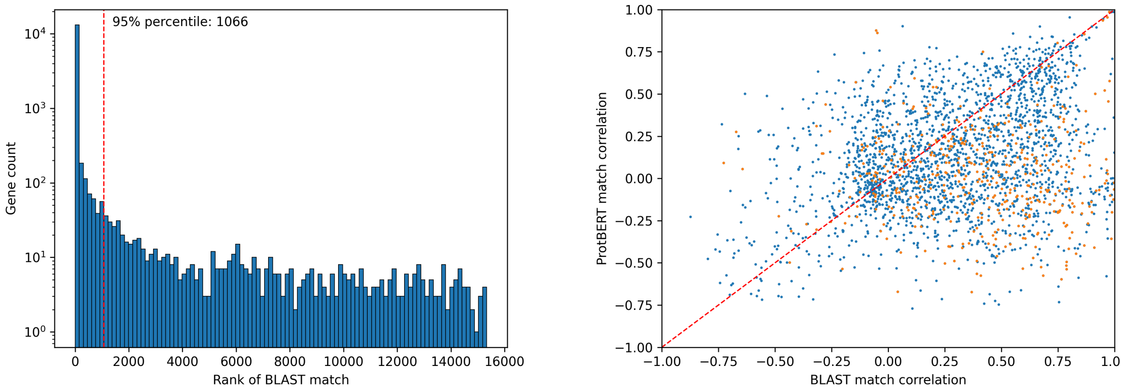
predictions after the model is trained, the clusters are less pronounced (Figure 4). Vip 1, for example, has no clear match in the shared feature space (Figure 5b), but shows a clear bias towards Vip 1 in both species after the network was trained. Thus the shared feature space can clearly distinguish cell types at a lower resolution, but can only capture the larger differences at a higher resolution. Training of the network is needed to clearly distinguish between the subtypes as well.

3.2. ProtBERT compared to BLAST

3.2.1. ProtBERT embeddings are in general similar to BLAST

We compared gene matches from the ProtBERT embeddings to the reference gene matches from the M1 data which were created using BLAST. We retrieved the protein sequences of 18478 human genes and 15969 mouse genes from Uniprot. For 14374 of these human a matching mouse gene is defined in the reference gene matches. For each of the latter human genes we retrieved the mouse gene with the closest embedding distance (ProtBERT match) and the mouse gene from the reference gene matches (BLAST match). The best ProtBert match is identical to the BLAST match for 11543 of the 14374 human genes, or 80.3%. Thus the ProtBERT embeddings behave similar to BLAST for the majority of the genes. We also ranked all mouse genes for every human gene according to the embedding distance. The ProtBERT match is always at rank 0 since per definition the ProtBERT match is the mouse gene closest in the embedding space. BLAST matches that differ from the ProtBERT match have a rank > 0. We looked up the rank of the BLAST match for every human gene (Figure 6a) and over 90% of the BLAST matches have a rank less than 10. Thus the BLAST match is identical to the ProtBERT match for the majority of genes and the other BLAST matches have a relatively high rank.

Next, we evaluated how well the gene matches align the cell types of the two species. We first aggregate the expression in each cell type for each gene by taking the average. This results in average expression profiles for each gene across the cell types and we create these average expression profiles for both species. Then, for every human gene we calculated the Pearson correlation with the ProtBERT match and BLAST match across the expression profiles (Figure 6b). Out of the 2831 human genes where the ProtBERT match and BLAST match differ, 891 human genes have a higher correlation with the ProtBERT match than the BLAST match. For example, the human gene SLC6A20 is matched to the mouse gene SLC6A20A according to the ProtBERT embedding distance and matched to the mouse gene SLC6A20B according to the BLAST distance. Human SLC6A20 has a low expression for most cell types and shows a peak for VLMC. Likewise, mouse gene SLC6A20A shows a peak for VLMC, while mouse SLC6A20B has a higher expression overall and no peak for VLMC. In this case the correlation with the ProtBERT match is thus higher than with the BLAST match. Overall the BLAST matches correlate better with the cell types, but this is to be expected since the cell types were defined based on those matches.



(a) Rank of BLAST Match

(b) Pearson correlation across cell types

Figure 6: Comparison of ProtBERT embeddings to Allen reference matches. (a) Rank of BLAST match according to ProtBERT embedding distances. Rank 0 indicates that the retrieved ProtBERT match and BLAST match are the same. The 95% percentile is portrayed with a dashed red line. (b) Scatterplot of correlation with BLAST match and ProtBERT match. Correlation is calculated as the Pearson correlation across the average expression profiles of the Cross-species cell types of the M1 data. We omit human genes where the BLAST match and ProtBERT match are equal. Dashed line indicates equal correlation. Gene matches where either the human gene, ProtBERT match or BLAST match is a DEG are marked orange.

Table 2: Performance of value-selection strategies on the M1 data for Cross-species resolution.

Top-k	Threshold	ADS	Recall	Train accuracy	Validation accuracy
1	∞	0.6317	0.6818	0.9454	0.9380
5	∞	0.6431	0.7273	0.9514	0.9430
1	0.0001	0.6814	0.7045	0.9423	0.9329
1	0.0002	**0.7084**	**0.8182**	0.9697	0.9439
1	0.0005	0.6259	0.6818	0.9595	0.9428
1	0.001	0.6358	0.6590	**0.9751**	**0.9450**

3.2.2. Thresholding the embedding distance increases the training efficiency and improves the quality of matches

To generate gene-matches from the distance matrix we evaluated several threshold and top-k strategies. The top-k strategy filters entries in the distance matrix where both genes are in the top-k for each other. We assume that the top-1 strategy retrieves the one-to-one orthologous genes. The threshold strategy includes entries with a distance less than the threshold in addition to the top-1 gene matches. We perform each strategy on the distance matrix to filter entries that correspond to gene matches. We afterwards use the gene selection from Section 2.3 to only include matches in which at least one gene is differentially expressed.

We use the filtered gene matches to train TACTiCS on the M1 data for the Cross-species resolution. The ADS, recall and accuracies for each strategy are found in Table 2. The tested thresholds were chosen to result in 1k to 100k matches. Each strategy has a similar training and validation accuracy. This is not guaranteed, since the distribution of cell types differs in the training compared to the validation set. Thus no strategy overfits on the distribution of the training data. The top-1 with a threshold of 0.0002 achieved the highest ADS and recall, while the top-1 with a threshold of 0.001 achieved the highest train and validation accuracy. The highest accuracy has been achieved with the highest threshold, which reinforces that including more gene matches, and thus more information, can increase the performance of the classifier. However, the strategy with the highest threshold did not result in the best cell type matches, and thus species-specific performance does not always relate to the quality of the matches. We use the top-1 strategy with a threshold of 0.0002 for the other experiments.

Table 3: Performance of different models on the M1 data at the Subclass resolution. Gene distances are calculated using ProtBERT embeddings or BLAST. Gene selection “All” indicates that all genes with available protein sequences were used, while “DEG” indicates that the genes were selected according to Section 2.3. The rows are grouped by the section their results are presented in.

Model	Gene matching	Gene selection	ADS	Recall
TACTiCS	ProtBERT	DEG	0.9544	0.9474
TACTiCS, no embedding loss	ProtBERT	DEG	0.8980	0.8947
TACTiCS, no updated loss	ProtBERT	DEG	0.9799	1.0000
SAMap	BLAST	All	0.7471	1.0000

Table 4: Performance of different models on the M1 data at the Cross-species resolution. Gene distances are calculated using ProtBERT embeddings or BLAST. Gene selection “All” indicates that all genes with available protein sequences were used, while “DEG” indicates that the genes were selected according to Section 2.3. “One-to-one” filters the one-to-one gene matches, as described in Section 3.2.4. The rows are grouped by the section their results are presented in.

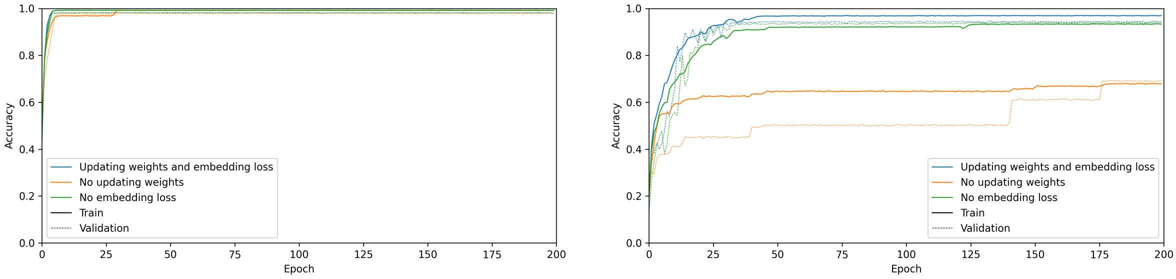
Model	Gene matching	Gene selection	ADS	Recall
TACTiCS	ProtBERT	DEG	0.7084	0.8182
TACTiCS	ProtBERT	One-to-one	0.6927	0.7955
TACTiCS	BLAST	DEG	0.5720	0.6136
TACTiCS	BLAST	All	0.6153	0.6818
TACTiCS, no embedding loss	ProtBERT	DEG	0.6017	0.6364
TACTiCS, no updated loss	ProtBERT	DEG	0.5402	0.3864
SAMap	BLAST	All	0.4521	0.5227
SAMap	ProtBERT	DEG	0.4797	0.5909
SAMap	ProtBERT	All	0.4648	0.5682
Graph convolution (Section 5.1)	ProtBERT	DEG	0.3527	0.2843

3.2.3. ProtBERT embeddings outperform BLAST in TACTiCS

Next we investigated the influence of the ProtBERT embeddings in TACTiCS. We substituted the embedding distance in our model by the BLAST distance. We executed BLAST on the protein sequences and filtered out matches with an E-value > 1e-6. BLAST distances are directional, while TACTiCS only uses undirected edges between the genes. We combined the BLAST distances from mouse to human and human to mouse, by filtering out the non-reciprocal edges followed by averaging the bitscores for the remaining edges. First we used our default gene-selection, which resulted in an ADS of 0.57 and recall of 0.61 on the M1 data at the Cross-species resolution. Without a gene-selection, the ADS increased to 0.62 and the recall to 0.68, but the runtime was twice as long. Thus if the BLAST distances are used in TACTiCS, it is beneficial to keep all genes rather than the DEG. If TACTiCS uses the ProtBERT distances instead, the ADS is 0.71 and the recall is 0.82. Thus the ProtBERT embeddings are beneficial for TACTiCS, since the model with BLAST distances performs worse and results in a longer runtime.

3.2.4. Many-to-many gene matches do not impact performance

Finally, we investigated the influence of the many-to-many gene matches on the final cell type matches. We generated gene matches for the M1 data using default settings, but afterwards filtered the list with a top-1 strategy. In contrast to the aforementioned top-1 strategy, we did not filter the genes again based on the gene matches. Thus, every gene has zero or one gene match. For instance, human genes LRRC8A and LRRC8C both match only with mouse gene LRRC8C. The mouse gene is more similar to human LRRC8C. In this case we remove the match between human LRRC8A and mouse LRRC8C, but we do not discard the human gene itself. Thus in the shared feature space the mouse cells will have a zero expression for the human LRRC8A gene, while the human cells keep the original expression. This removed ~ 600 gene matches from the original 1900 matches. We trained our model for the Cross-species resolution which resulted in a slightly lower ADS of 0.69 and recall of 0.80 in comparison to our default gene matching method. From this we can conclude that in this case the many-to-many matches do not influence the cell types matches much.



(a) **(b)**

Figure 7: Training and validation accuracies across epochs on the M1 data for resolution (a) Subclass and (b) Cross-species.

3.3. Influence of loss functions on classifier accuracy and alignment of species

3.3.1. Updating cell type weights improves classifier accuracy

While designing TACTiCS we noticed that the classifier is sensitive to the cell type size. The classifier predicted only large cell types during training. We introduced oversampling in the training phase to solve this, such that all cell types are seen equally by the classifier. Although this somewhat improved the training accuracy, the accuracy plateaued. To solve this we introduced the updated cross-entropy loss. With this loss the weight of a cell type is increased when the accuracy of was low in the last epoch. The influence of the updated cross-entropy loss was evaluated on the M1 data for the Subclass and Cross-species resolutions. The training and validation accuracies are plotted in Figure 7.

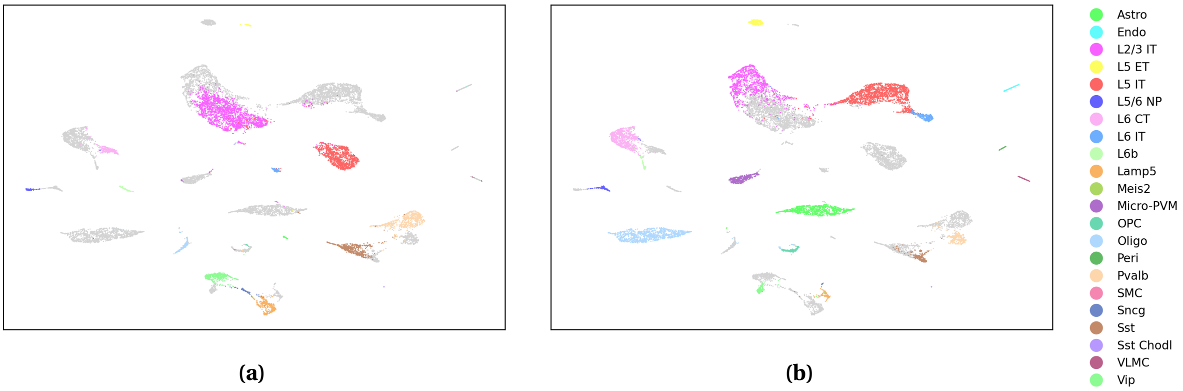
For the Subclass resolution (Figure 7a) the updated cross-entropy loss has no effect on the training accuracy, namely the training accuracy is 99.2% with and without the updated cross-entropy loss. However, the recall actually decreased with the updated cross-entropy loss. For the Cross-species resolution the difference in accuracies becomes more apparent (Figure 7b). The training accuracy is 97.0% with the updated cross-entropy loss and only 67.9% with the standard cross-entropy loss. Training the model with the standard cross-entropy loss for more epochs does not close this gap. The ADS and recall increased with the updated cross-entropy loss, in contrast to the Subclass resolution. Thus using the updated cross-entropy loss improves the final training and validation accuracy of the classifiers significantly at higher resolutions.

3.3.2. Embedding loss improves alignment of cross-species cell embeddings

Next we examined the influence of the embedding loss on the results. The embeddings created by the networks can be used to visualize the alignment of the species. The cell embeddings were visualized using Uniform Manifold Approximation and Projection (UMAP) plots (Figure 8 and 9). The embedding loss was not used for the plot in Figure 8. Although the species are not aligned everywhere in the plot, corresponding cell types are mostly near each other. Human and mouse L2/3 IT cells, for example, are part of the same cluster although the cell types are separable within that cluster. Some cell types are aligned better such as human and mouse Sst, while other cell types are far apart such as human and mouse L5 IT and L5 ET.

We introduced the embedding loss to improve alignment. Figure 9 shows the UMAP of the cell embeddings after the network was trained with the embedding loss. The majority of the clusters contain cells from both species. Additionally, the clusters contain cells belonging to the same cell type. While in Figure 8 the human and mouse L2/3 IT were separable, in Figure 9 they form one cluster and the cells are mostly mixed across the species. The smaller cell types such as L6b and Sncg are clustered correctly, as well as cell types that vary in number of cells across the species, such as Micro-PVM and Astro. The influence of the embedding loss on the cell embeddings can be seen in Figure S2. As shown in the UMAP plots, without the embedding loss, the cross-species nearest neighbour is much further away than the within-species nearest neighbour. Both distances become more similar when the embedding loss is used. In most cases the nearest neighbour is still from the same species, but the distance between the species has been reduced. The species-specific cell types, i.e. mouse Meis2, Peri and SMC, are quite small and do not form separate clusters. Instead, mouse Peri and SMC cluster with VLMC.

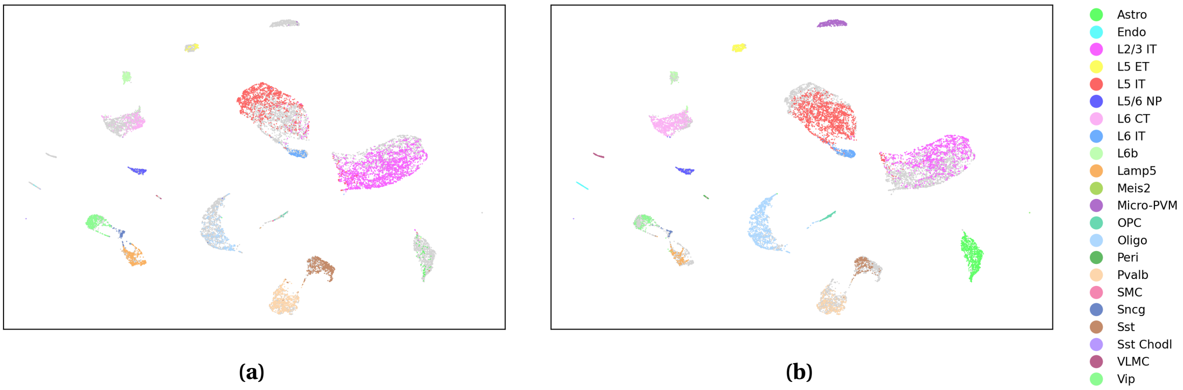
We use the LISI to quantify the mixing of species within clusters. The LISI increased from 1.05 to 1.41 for



(a)

(b)

Figure 8: UMAP plots for the M1 data for (a) human cells and (b) mouse cells for the Subclass resolution. The UMAP is calculated for the cell embeddings after the model is fully trained. No embedding loss was used.



(a)

(b)

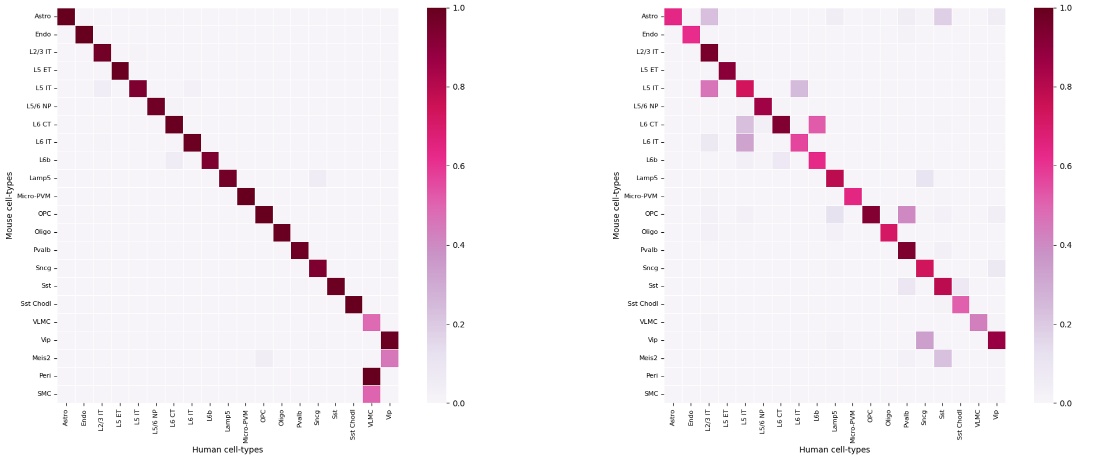
Figure 9: UMAP plots for the M1 data for (a) human cells and (b) mouse cells for the Subclass resolution. The UMAP is calculated for the cell embeddings after the model is fully trained.

the Subclass resolution and from 1.01 to 1.06 for the Cross-species resolution with the embedding distance loss. Thus in both resolutions the embedding distance loss increased the mixing across the species, but more significantly for the Subclass resolution. However, even for the Subclass resolution the LISI is not nearing 2, which means that cells still show a heavy bias towards cells from the same species. The mixing of species is also not same throughout the UMAP plots. The mouse Sst cells, for example, only cluster with the left half of the human Sst cells. Likewise the human L6 CT cells only cluster with the right half of the mouse L6 CT cells. This might be caused by the biological differences between the species, and thus it is acceptable if the cell types do not overlap completely across the species.

3.4. Comparison to CAME and SAMap

3.4.1. Simple architecture outperforms graph convolution

Next we compared our architecture to an alternative, more complicated architecture (Section 5.1) based on CAME. Since CAME is not yet available, we implemented a CAME-like architecture instead. We evaluated different variants of the architecture, however, none of the variants achieved the performance of TACTiCS. The best variant reached an ADS of 0.35 and recall of only 0.28. The alternative architecture used graph convolution to propagate information across the species in order to align them. TACTiCS already aligns the species with the shared feature space and the embedding loss, and thus it we believe it is unnecessary to also perform graph convolution. The graph convolution architecture used an attention mechanism with the genes to update the embeddings. This has the advantage that we can theoretically retrieve the genes that are used to characterize a cell type. However, this is a trade-off with a worse performance, longer run-time and a larger model which takes longer to converge. Additionally there is no guarantee that the attention scores relate to the importance of a gene to a cell type. Due to these reasons we conclude that our simple network



(a) TACTiCS

(b) SAMap

Figure 10: Combined matrix for TACTiCS and results from SAMap on the M1 data for the Subclass resolution. Higher values (darker colours) indicate that the cell types match.

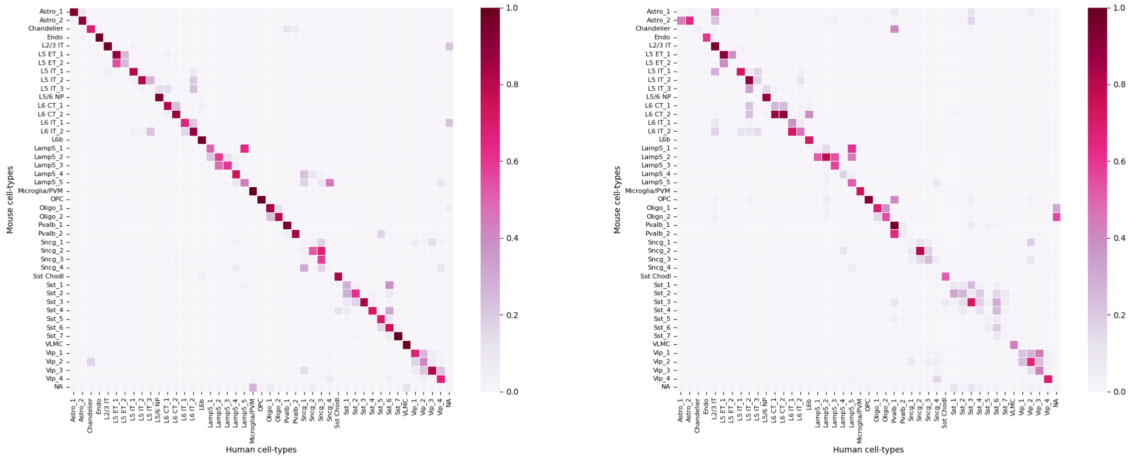
architecture outperforms the graph convolution architecture.

3.4.2. TACTiCS outperforms SAMap at a higher resolution

To compare the results of TACTiCS to SAMap, we matched the cell types of the M1 data using SAMap at the Subclass and Cross-species resolutions. SAMap needs two labeled datasets and initial gene distances as input. For the initial gene distances we used BLAST to calculate the protein sequence similarity between the mouse and human genes. We did not manage to run SAMap on the entire M1 data as SAMap continued running indefinitely. We instead ran SAMap on a subset of 30k mouse cells and 30k human cells. We compared the combined matrix of TACTiCS to the results of SAMap (Figure 10 and 11).

At the Subclass resolution the resulting cell type matches from SAMap are very similar to TACTiCS (Figure 10). The ADS of TACTiCS (0.95) is higher than of SAMap (0.75). TACTiCS matches only one cell type wrong, namely the human VLMC is matched to mouse Peri, SMC and VLMC. SAMap correctly assigns a low score to the species-specific cell types, i.e. mouse Meis2, Peri and SMC, while our method assigns high scores. The performance of species-specific cell types can be explained by two steps in TACTiCS, namely the embedding distance loss and the transfer of classifiers. The embedding loss promotes clusters with cells from both species. Cells which belong to the same cell type cluster together in the embedding space and thus also share the same cross-species neighbours. This results in the clustering of cross-species cell types, even if the cell type is species-specific, such as mouse Peri and SMC in Figure 9. The initial confusion matrices were generated using the mouse classifier on human cells and vice versa. All cells are this way assigned a transferred cell type, including cells from to species-specific cell types. Thus TACTiCS outperforms SAMap at the Subclass resolution for the common cell types, but SAMap works better for the species-specific cell types.

We see more prominent differences in the cell type matches at the Cross-species resolution. TACTiCS again has a higher ADS (0.71) than SAMap (0.45), although both methods perform worse than at the Subclass resolution. The recall for TACTiCS (0.82) is higher as well compared to SAMap (0.52). In particular, SAMap does not find a match for most of the Sncg and Sst cell types. TACTiCS works quite well for all Sst subtypes except for Sst 1. The results for the Sncg subtypes look more comparable to SAMap and interestingly both methods do not match human Sncg 4 to mouse Sncg 4, but instead show some similarity between human Sncg 4 and mouse Lamp5 5. SAMap maps the subtypes of Astro, L6 CT, L6 IT and Pvalb to only one subtype in the other species, while TACTiCS matches them to the correct subtype. SAMap matches both human Astro 1 and 2 to mouse Astro 2 for instance, while TACTiCS correctly matches human Astro 1 to mouse Astro 1 and human Astro 2 to mouse Astro 2. SAMap seems unable to distinguish between some subtypes at this resolution, however, this might be a consequence of the downsampling of the dataset.



(a) TACTiCS **(b)** SAMap

Figure 11: Combined matrix for TACTiCS and results from SAMap on the M1 data for the Cross-species resolution. Higher values (darker colours) indicate that the cell types match.

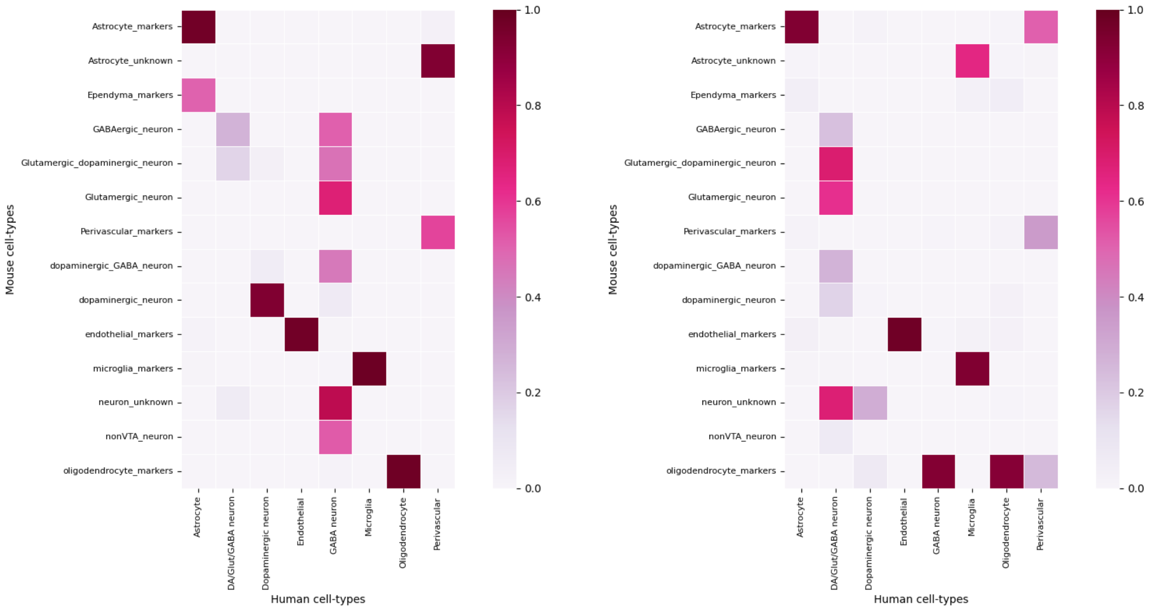
3.4.3. ProtBERT embeddings outperform BLAST in SAMap

SAMap differs from TACTiCS in both the gene-matching and cell type matching. For gene-matching, SAMap uses the BLAST distances while TACTiCS uses the ProtBERT embedding distance. To compare the ProtBERT embeddings to BLAST, we substituted the BLAST distances in SAMap by the ProtBERT embedding distances. We used this modified version of SAMap on the M1 data at the Cross-species resolution, resulting in a higher ADS (0.48) and recall (0.59) compared to BLAST. We used the default strategy to generate the gene-matches, which includes filtering matches to include at least one DEG. Since SAMap does not perform any gene selection, the increase in performance might be caused solely by the gene-selection, rather than by the ProtBERT embeddings. To rule this out, we repeated the experiment without the gene-selection, thus keeping all gene-matches regardless whether they consist of a DEG. This increased the number of filtered genes from ~1.5k to ~11k per species. The resulting ADS (0.47) and recall (0.57) are slightly worse than with the gene-selection, but still better than the SAMap with BLAST distances. We conclude that the ProtBERT embedding distance has the biggest impact on the performance and that the gene-selection strategy can improve the performance slightly. The gene selection is still beneficial for SAMap since it resulted in a 5X speed up, thus increasing the performance while decreasing the runtime. SAMap did not reach the performance of TACTiCS despite the improvement in performance.

3.5. TACTiCS correctly matches cell types on the VTA data

Finally, we compared TACTiCS to SAMap on the VTA Data (unpublished data from Basak Lab at UMC Utrecht) at the major resolution (Figure 12). We calculated the protein sequence similarity with BLAST, and used this as input to SAMap, similar to the M1 data. We do not calculate the ADS and recall, since there is no established matching for the VTA data. The main difference between the cell type matches of the methods is that TACTiCS matches the majority of the mouse neurons to human GABA neuron, while SAMap matches them to human DA/Glut/GABA neuron. Which match is better is debatable. SAMap correctly marks the species-specific cell types as such by assigning a low score, while TACTiCS matches mouse Ependyma to human Astrocyte and mouse nonVTA to human GABA. Mouse Ependyma is still matched to a similar cell type, even though the match is incorrect. Ependymal glia and astrocytes are both macroglia, and within macroglia these cells are more similar than to oligodendrocytes [23]. Other cell types matches can be assessed more objectively. For example, both methods match Astrocyte, Perivascular and Endothelial correctly across the species. TACTiCS matches the mouse dopaminergic neurons to human dopaminergic neurons, while SAMap does not have a clear match for either. SAMap also incorrectly matches mouse Oligodendrocyte to human GABA and Perivascular, while TACTiCS has a reciprocal match between human and mouse Oligodendrocyte. Thus TACTiCS returns better matches for the common cell types in both datasets.

The UMAP plots of the cell embeddings created by TACTiCS are shown in Figure 13. The biggest clusters, i.e. Astrocyte, Microglia and Oligodendrocyte, are all aligned and mixed across the species. Both the Astrocyte



(a) TACTiCS

(b) SAMap

Figure 12: Combined matrix for TACTiCS and results from SAMap on the VTA data for the major resolution. Higher values (darker colours) indicate that the cell types match.

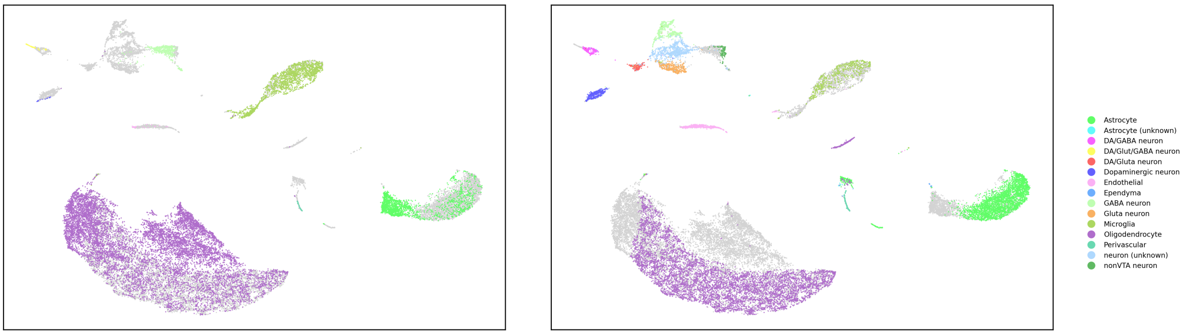


Figure 13: UMAP plots for VTA Data for human cells (left) and mouse cells (right) for the major resolution. The UMAP is calculated for the cell embeddings after the model is fully trained.

and Oligodendrocyte clusters consist of human-specific parts. Interestingly, only the GABA neurons do not overlap at all across the species, even though the cells are relatively common in both species. The number of cells varies heavily across the species for the other cell types. Despite this, the Dopaminergic neurons and Endothelial cells are clustered together correctly across the species. Thus TACTiCS correctly aligned the VTA cell types in the embedding space.

4
Discussion

scRNA-seq has identified many new cell types in the human and mouse brain. This created the need for methods that can accurately find the relation between cell types in a cross-species setting. The cell type matches are necessary for translational research and can be used to study the evolutionary mechanisms of cell specializations. We introduced TACTiCS, a model to match cell types across species using scRNA-seq data. TACTiCS consists of two improvements compared to current methods: matching sequences using ProtBERT instead of BLAST and a simple neural network architecture to create aligned cell embeddings. We show that TACTiCS outperforms SAMap and an architecture based on CAME.

We evaluated TACTiCS on the M1 data and showed that TACTiCS can accurately match cell types at the Subclass and Cross-species resolutions. The cell type matches are in general better for larger cell types and for cell types that are very distinct. The performance deteriorates for smaller, more similar cell types and the same trend is more evident in SAMap. This hints to a fundamental similarity between the methods that does not work for very similar cell types. scRNA-seq contains the expression for many genes, however, only one or two genes might characterize the differences between two very similar cell types. Current methods narrow the search for these genes by using the DEG, but there is no guarantee that these include the characterizing genes. The TH gene, for example, defines a Dopaminergic neuron, but this gene was not retrieved as the most differential expressed gene for that cell type. Thus to distinguish between very similar cell types, methods might need to use another way to select characterizing genes for a cell type.

We also compared TACTiCS to an architecture based on CAME. The CAME architecture uses graph convolution to aggregate information across cells and species, but despite this it performs worse than TACTiCS. In the CAME architecture, information is shared between the species in two ways. First, graph convolution updates embeddings using the embeddings of neighbours. After some layers, the embeddings will have propagated between the species. Second, cell embeddings are created for both species using shared weights. The embeddings are created in the same manner, which also improves the alignment. TACTiCS only shares information with the second method, but also results in a correct alignment. This might indicate that the first method is unnecessary to align the species. The first method, however, is useful to explicitly use the genes to classify cells. As a result CAME can backtrack the genes that were used to classify a cell type and thus highlight important genes this way. TACTiCS is not capable of indicating which genes are important for each cell type. Thus, even though the graph convolution does not improve the performance, it is useful to characterize a cell type more in depth.

We found that the ProtBERT embedding distances are similar to BLAST, since the majority of top-1 gene matches are identical for ProtBERT and BLAST. We found, however, that using ProtBERT improved the performance of TACTiCS drastically. Likewise, ProtBERT improved the performance of SAMap. Thus ProtBERT embeddings provide a viable alternative to BLAST, including in existing methods. In general, this indicates that embedding distances are similar to the sequence similarity, but that the embeddings contain more functional information about the genes that is beneficial to the model.

The embedding loss and updated cross-entropy loss both increased the training accuracy of TACTiCS. The updated cross-entropy loss can be easily used in other methods with many classes. The embedding loss correctly aligned the cell types in the embedding space. The aligned cell embeddings can be used in further downstream tasks, for instance to visualize the alignment, but also to compare the cross-species cells belonging to the same cell type. For example, although Oligodendrocytes match for human and mouse, there are also subclusters within the Oligodendrocyte consisting of only human or mouse cells. These subclusters of one species can indicate differences between the species for the same cell type. The within-cluster nuances between the subclusters are only visible because of the aligned cell embeddings.

We investigated wehther including many-to-many orthologous genes increases the quality of the cell type matches. We found that our gene matching method improved the cell type matches, however, we did not find evidence that the relation of many-to-many orthologous genes is important for the cell type matches. This might mean that the genes are more important than the relation between them, or that the relation between many-to-many orthologous genes is not important for comparisons between human and mouse. These relations might be more important for more distant comparisons with less one-to-one orthologous genes available.

4.1. Limitations

TACTiCS does not include a rejection option and thus we currently cannot classify a cell type as species-specific. It is not straightforward to add a rejection option to our model. We could for example add an extra class to the cell type classifier that represents species-specific cells from the other species. Another option is to add a hard boundary to every existing class. Cells that meet none of the boundaries are then classified to be species-specific. Both approaches, however, need positive examples of cross-species cells that are species-specific which are not available in the training phase. Additionally, there is no guarantee that the boundaries for within-species cells also work for cross-species cells. This is also supported by Figure S2, where even with the embedding loss, a cell is closer to cells from the same species, than cross-species cells. For these two reasons it is not straightforward to add a rejection option, and thus our model is best used in datasets without species-specific cell types.

We evaluated TACTiCS on scRNA-seq data from mouse and human, whose genomes are considered close. As a result, these species will have relatively more one-to-one orthologous genes compared to more distant species. For more distant comparisons TACTiCS can prove more useful since it includes the many-to-many relations. We were not able to evaluate TACTiCS on scRNA-seq data of more distant species. More species-specific cell types are present for more distant species, however, which makes the rejection option more important. We evaluated TACTiCS by comparing to the harmonized labels, assuming that these are correct. The labels were again harmonized by matching genes with BLAST. We thus evaluate TACTiCS on data with a bias for BLAST and when we compare ProtBERT to BLAST, this leads to an unfair advantage for BLAST.

TACTiCS uses the protein sequences to match the genes and can thus not be used for non-coding regions. TACTiCS can, however, be extended to match genes with DNA sequences by using DNABERT [24] instead of ProtBERT. Additionally, TACTiCS can be extended to include more information about each cell. The added information should span the same features across the species (e.g. identical genes) or should relate somehow across the species (e.g. matching genes). Without this requirement, the added information can not be compared across species and the cell embeddings will not be aligned correctly. For example, TACTiCS can be expanded to use isoform expression by matching the individual isoforms across species rather the the protein sequences.

TACTiCS consists of many steps and each step introduces multiple hyperparameters. We noticed that some of the hyperparameters have a large influence on the performance. The embedding loss weight, for instance, can cause all cell types to be clustered together or all in separate clusters. Likewise, the L2 normalization weight can influence whether the training accuracy increases at all. We empirically tuned the hyperparameters, however, we did not exhaustively test the combination of hyperparameters. Other hyperparameters might result in better quality gene matches. A cyclic learning rate [25] can improve the training accuracy of TACTiCS, for instance, rather than the current constant learning rate. Tuning the hyperparameters might prove difficult, since the number of cells per cell type is quite small for most cell types, even for large datasets. The resulting hyperparameters might thus not be transferable for other datasets or cell types.

4.2. Conclusion

Currently, the majority of the methods discard the relation between genes and match cell types using only the BLAST gene matches. With TACTiCS, we showed that using protein embeddings to match genes is a viable alternative to BLAST. In the future our gene matching method can be used as part of other methods for cross-species comparisons, as we showed with SAMap. Additionally, we showed that a complicated matching algorithm, such as CAME, does not necessarily improve the quality of the cell type matches and that a more simple architecture can have a better performance. The flexible structure of TACTiCS allows the individual parts to be extended or integrated in other methods.

5 Supplementary

5.1. Graph convolution architecture

In addition to the network architecture described in Sections 2.5, we experimented with different architectures based on CAME [12]. In this section we describe the best architecture variant. A schematic overview is shown in Figure S1.

5.1.1. Heterogenous graph of cells and genes

For every batch we construct a graph consisting of all cells in the batch. We perform PCA on the gene expression for every cell and store the first 20 PCs. For every cell we use the PCs to calculate the 5 nearest within-species neighbours and add an edge to those neighbours in the graph. Note that the cells are not connected across the species. Next we add all genes in the graph, G_A and G_B. We connect a cell to all genes with an expression > 0 in the augmented matrix, thus a cell from species A can be connected to genes in G_A and G_B.

5.1.2. Initial embeddings

We create embeddings for both cells and genes. The cell embeddings are created as described in Section 2.5. The embedding of a gene is based on the gene expression of cells that it is connected to. For every cell we create an additional embedding in the same way, but with different weights. The weighted average of these cell embeddings are used as the gene embedding. A gene embedding is thus based on cells from both species and from multiple cell types. We weigh every cell embedding of species A with $\frac{1}{n_t N_A}$ to remove the influence of the number of cells on the gene embedding. Similarly we weigh the cell embeddings of species B.

5.1.3. Updating embeddings using graph convolution

We update all embeddings in the graph twice using graph convolution to propagate information through the graph. For every cell embedding we create three new embeddings with three linear layers with the purpose to update its own embedding, embeddings of neighbours and embeddings of connected genes. For every gene embedding we create two new embeddings, one to update its own embedding and the other to update embeddings of connected cells. This results in five weight matrices: W^c, W^{cc}, W^{cg}, W^g and W^{gc}.

The updated gene embedding is based on the old gene embedding and embeddings from connected cells. This is calculated as the sum of 1) its own cell embedding with weight W^g and 2) the average of embeddings of connected cells with W^{cg}. This sum and thus the updated gene embedding now contains information of connected cells. Likewise the updated cell embedding is based on the old cell embedding, embeddings from connected cells and embeddings from connected genes. To update a cell embedding we sum 1) its own cell embedding with weight W^c, 2) the average of embeddings of neighbours with W^{cc} and 3) the average of embeddings of connected genes with W^{gc}. The updated cell embedding thus contains information from its neighbours.

5.1.4. Updating cell embeddings using attention

Next we update the cell embeddings again using the attention mechanism described [26]. We use 2 heads, thus for every cell and gene we create two new embeddings. For every cell-gene pair we calculate an attention score per head with a linear layer. The attention score expresses how important the gene is to classify that cells. We weigh the new gene embeddings with the attention scores to create the updated cell embedding. Thus, the cell embedding consists solely of gene embeddings and does not contain its own old embedding. Finally, we average the cell embeddings of the two heads to create a final embedding for each cell.

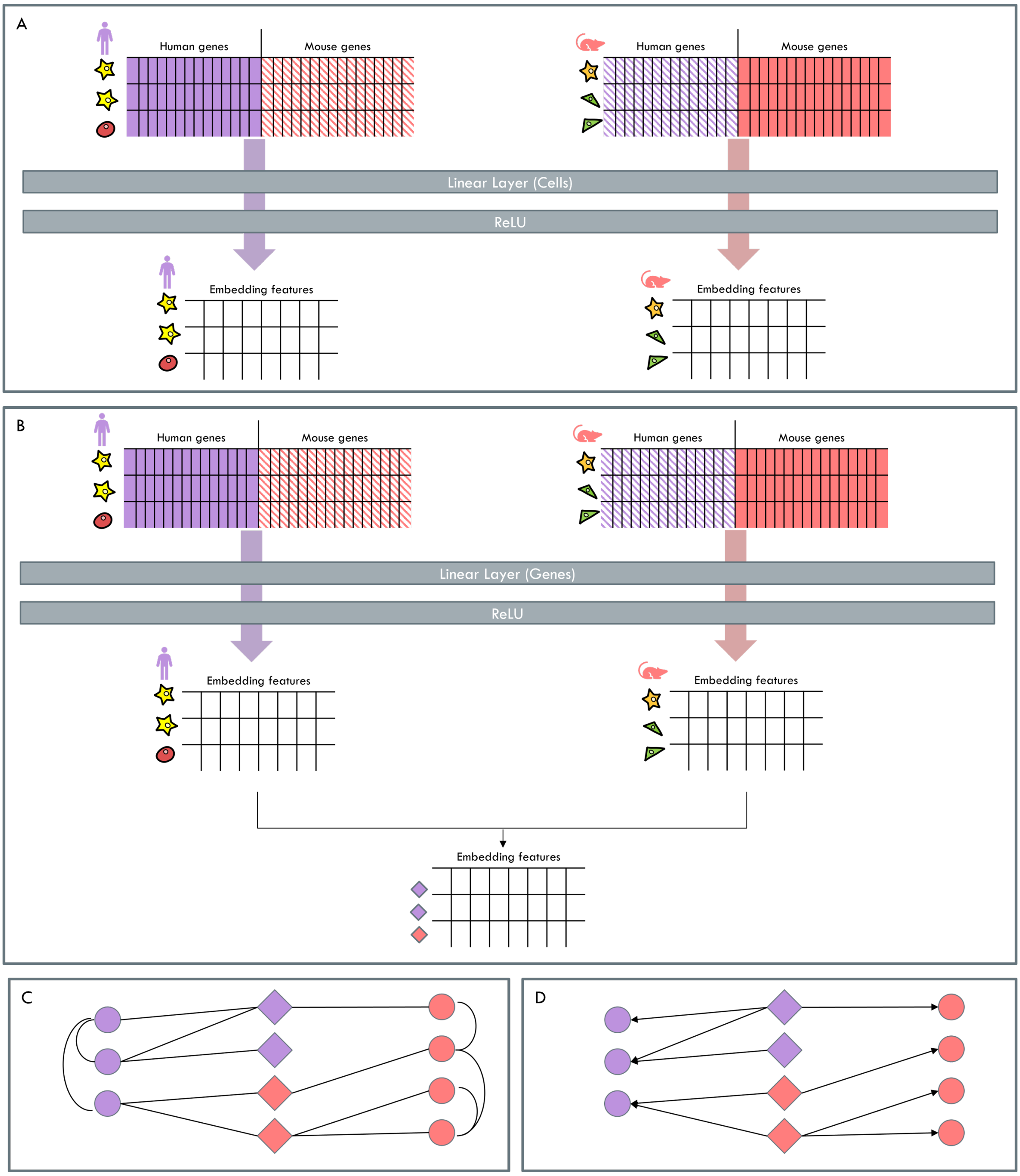
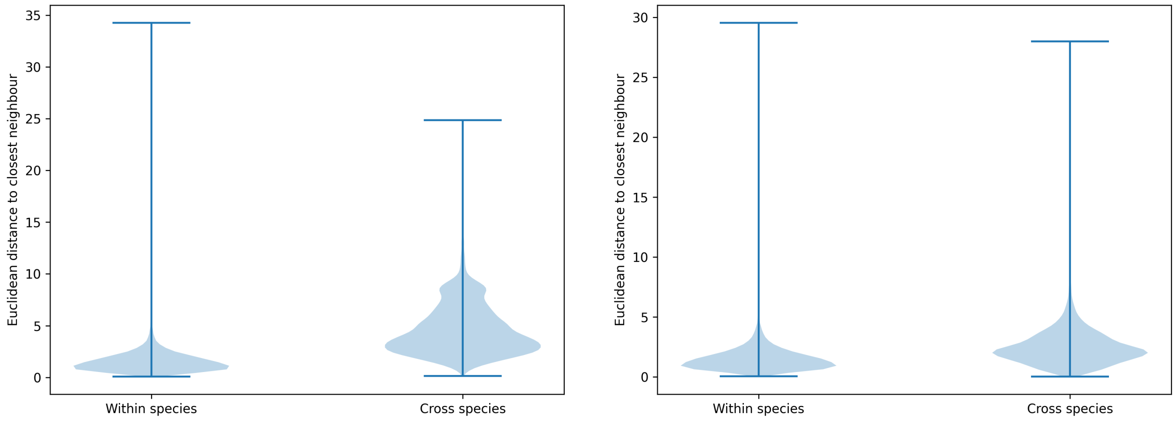


Figure S1: Schematic overview of graph convolution architecture. (a) Creating initial cell embeddings. (b) Creating initial gene embeddings. (c) Graph consisting of cells and genes on which graph convolution is applied. (d) Updating cell embeddings with attention.



(a) Without embedding loss

(b) With embedding loss

Figure S2: Euclidian distance to nearest within-species and cross-species neighbour (a) without embedding loss and (b) with embedding loss.

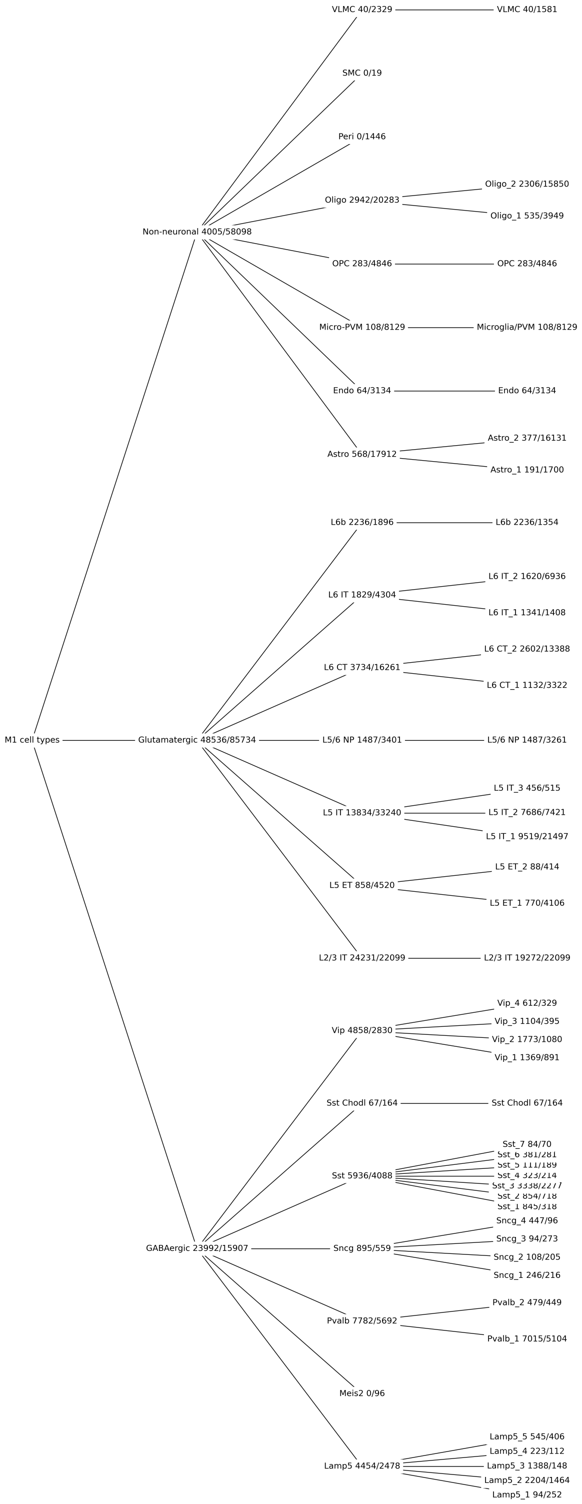
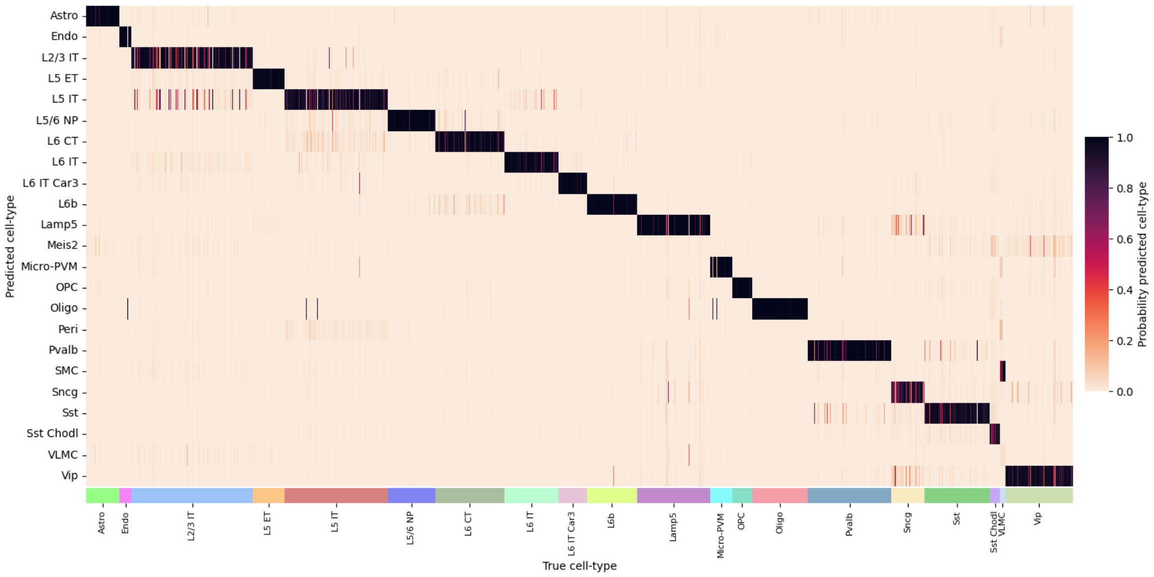
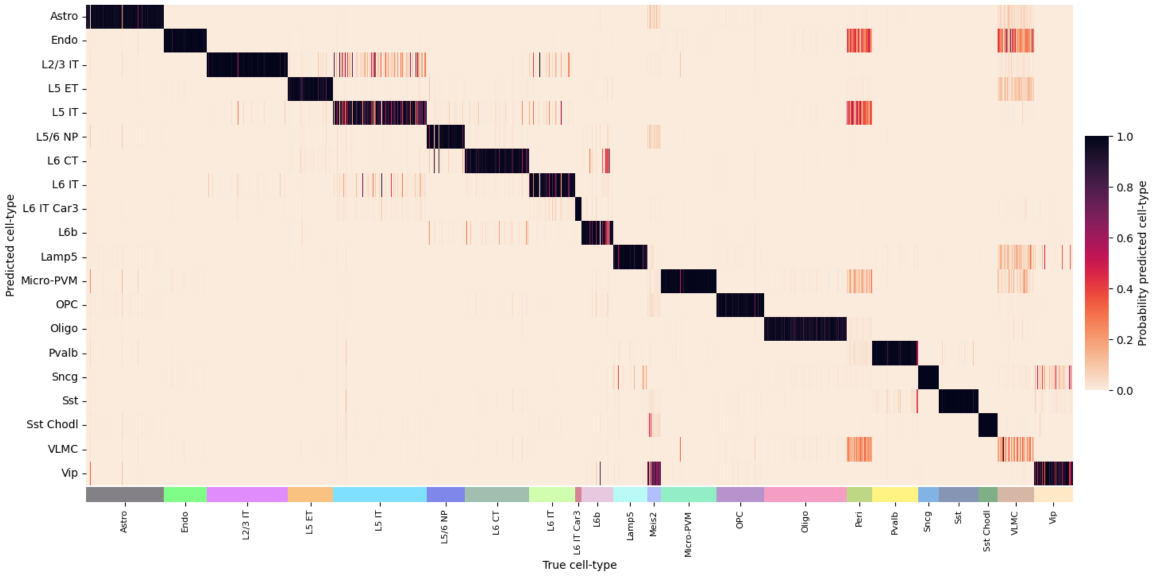


Figure S3: Cell types in the M1 data for the Class, Subclass and Cross-species resolutions, annotated with the number of human cells/mouse cells.



(a)



(b)

Figure S4: Cell type probabilities for M1 data at Subclass resolution for (a) human cells and (b) mouse cells. After training both classifiers, the cells were input to the classifier of the other species to transfer the cell types and generate the probabilities.

Bibliography

[1] Raquel Cuevas-Diaz Duran, Haichao Wei, and Jia Qian Wu. Single-cell rna-sequencing of the brain. *Clinical and translational medicine*, 6(1):1–14, 2017.

[2] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.

[3] Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

[4] Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular systems biology*, 17(1):e9620, 2021.

[5] Megan Crow, Anirban Paul, Sara Ballouz, Z Josh Huang, and Jesse Gillis. Characterizing the replicability of cell types defined by single cell rna-sequencing data using metaneighbor. *Nature communications*, 9 (1):1–12, 2018.

[6] Travis S Johnson, Tongxin Wang, Zhi Huang, Christina Y Yu, Yi Wu, Yatong Han, Yan Zhang, Kun Huang, and Jie Zhang. Lambda: label ambiguous domain adaptation dataset integration reduces batch effects and improves subtype detection. *Bioinformatics*, 35(22):4696–4706, 2019.

[7] Rebecca D Hodge, Trygve E Bakken, Jeremy A Miller, Kimberly A Smith, Eliza R Barkan, Lucas T Graybuck, Jennie L Close, Brian Long, Nelson Johansen, Osnat Penn, et al. Conserved cell types with divergent features in human versus mouse cortex. *Nature*, 573(7772):61–68, 2019.

[8] Trygve E Bakken, Nikolas L Jorstad, Qiwen Hu, Blue B Lake, Wei Tian, Brian E Kalmbach, Megan Crow, Rebecca D Hodge, Fenna M Krienen, Staci A Sorensen, et al. Evolution of cellular diversity in primary motor cortex of human, marmoset monkey, and mouse. *BioRxiv*, 2020.

[9] Sonia M Leach, Sophie L Gibbings, Anita D Tewari, Shaikh M Atif, Brian Vestal, Thomas Danhorn, William J Janssen, Tor D Wager, and Claudia V Jakubzick. Human and mouse transcriptome profiling identifies cross-species homology in pulmonary and lymph node mononuclear phagocytes. *Cell reports*, 33(5):108337, 2020.

[10] Maxwell ER Shafer, Ahilya N Sawh, and Alexander F Schier. Gene family evolution underlies cell type diversification in the hypothalamus of teleosts. *BioRxiv*, 2020.

[11] Alexander J Tarashansky, Jacob M Musser, Margarita Khariton, Pengyang Li, Detlev Arendt, Stephen R Quake, and Bo Wang. Mapping single-cell atlases throughout metazoa unravels cell type evolution. *Elife*, 10, 2021.

[12] Xingyan Liu, Qunlun Shen, and Shihua Zhang. Cross-species cell-type assignment of single-cell rna-seq by a heterogeneous graph neural network. *bioRxiv*, 2021.

[13] Olga Borisovna Botvinnik, Venkata Naga Pranathi Vemuri, N Tessa Pierce, Phoenix Aja Logan, Saba Nafees, Lekha Karanam, Kyle Joseph Travaglini, Camille Sophie Ezran, Lili Ren, Yanyi Juang, et al. Single-cell transcriptomics for the 99.9% of species without reference genomes. *bioRxiv*, 2021.

[14] Md Humayun Kabir, Djordje Djordjevic, Michael D O'Connor, and Joshua WK Ho. C3: An r package for cross-species compendium-based cell-type identification. *Computational biology and chemistry*, 77: 187–192, 2018.

[15] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100. URL `https://doi.org/10.1093/nar/gkaa1100`.

[16] Amos Bairoch and Rolf Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28(1):45–48, 01 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.45. URL `https://doi.org/10.1093/nar/28.1.45`.

[17] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of life's code through self-supervised learning. *bioRxiv*, 2021. doi: 10.1101/2020.07.12.199554. URL `https://www.biorxiv.org/content/early/2021/05/04/2020.07.12.199554`.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL `https://arxiv.org/abs/1706.03762`.

[19] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):1–5, 2018.

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[21] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

[22] Isaac Virshup, Sergei Rybakov, Fabian J Theis, Philipp Angerer, and F Alexander Wolf. anndata: Annotated data. *bioRxiv*, 2021.

[23] Helmut Kettenmann, Helmut Kettenmann, and Bruce R. Ransom. *Neuroglia*. Oxford University Press, 2013.

[24] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 02 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab083. URL `https://doi.org/10.1093/bioinformatics/btab083`.

[25] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.