Research Article

Enhancement of Network Architecture Alignment in Comparative Single-Cell Studies

Clemens Schächter[1], Martin Treppner[1], Maren Hackenberg[1], Hanne Raum[2], Joschka Boedecker[2,3], Harald Binder[1,4,5]

1. Institute of Medical Biometry and Statistics (IMBI), Faculty of Medicine and Medical Center, University of Freiburg, Germany; 2. Neurobotics Lab, Dept. of Computer Science, University of Freiburg, Germany; 3. BrainLinks-BrainTools CRIION - Collaborative Research Institute Intelligent Oncology, University of Freiburg, Germany; 4. Freiburg Center for Data Analysis and Modelling, University of Freiburg, Germany; 5. CIBSS, Centre for Integrative Biological Signalling Studies, University of Freiburg, Germany

Animal data can provide meaningful context for human gene expression at the single-cell level. This can improve cell-type detection and clarify how well animal models represent human biology. To achieve this, we propose a deep learning approach that identifies a unified latent space to map complex patterns between datasets. The proposed method is tested to facilitate information transfer in liver, adipose tissue, and glioblastoma datasets from various animal models. Our results are robust for small datasets and large differences in the observed gene sets. Thus, we reliably uncover and exploit similarities between species to provide context for human single-cell data.

Corresponding authors: Clemens Schächter, clemens.schaechter@uniklinik-freiburg.de; Harald Binder, harald.binder@uniklinik-freiburg.de

1. Background

Model organisms are crucial in advancing biomedical research by offering advantages such as easy genetic manipulation and access to datasets from a variety of experimental contexts[1]. As a popular choice, mouse models have significantly contributed to the study of human diseases[2], including diabetes[3], glioblastoma[4], and non-alcoholic fatty liver disease[5]. However, translating experimental findings to humans is challenging owing to biological differences between species. Efforts to bridge this evolutionary gap include engineered mouse models that replicate human biology more closely[6]. The

emergence of single-cell RNA sequencing (scRNA-seq) has also opened up opportunities for deep learning approaches to compare experimental findings across species.

Transfer learning techniques have established themselves as powerful tools for sharing information between scRNA-seq datasets. These approaches often use encoder-decoder architectures to compress datasets into a low-dimensional manifold. Examples include Cell BLAST[7] and ItClust[8], which annotate and cluster cells based on knowledge transfer from reference datasets.

Architecture surgery techniques adjust network architectures according to the characteristics of different datasets. After pretraining, additional neurons are inserted into the encoder and decoder input layers. These neurons correct for unseen batch effects in the new data, while all other weights remain fixed during subsequent training. This approach, pioneered by scArches[9], now spans a diverse set of models[10][11][12]. Despite the method's success, two primary challenges remain unaddressed for datasets of different species (Figure 4).

First, some genes lack orthologs in other genomes, which requires different interpretations of certain input nodes in their neural network architectures. For example, 20% of human protein-coding genes and a significant percentage of small and long noncoding RNAs lack one-to-one mouse orthologs[13]. To enable training, architecture surgery-based approaches restrict datasets to orthologous genes or zero-fill missing values. Outside of architecture surgery, some models like SATURN[14] and TACTiCS[15] match genes via protein sequences with transformer-based language models.

The second challenge is that biological similarities between cells do not always translate into similar gene expression patterns, which can vary significantly between species[13]. Therefore, neural networks may struggle to recognize similar cells.

To account for differences between gene sets and expression levels, we introduce scSpecies. Our approach pretrains a conditional variational autoencoder-based model[16] and fully reinitializes the encoder input layers and the decoder network during fine-tuning. Architecture alignment is guided by a nearest neighbor search performed on homologous genes, which estimates the similarity between cells in both datasets. This incentivizes our model to map biologically related cells into similar regions of the latent space. The neighbor search requires only a small subset of observed genes to be homologs, while all remaining genes can have no relationship at all. Moreover, scSpecies enables nuanced comparisons of gene expression profiles by generating gene expression values for both species from a single latent variable.

We tested our method on data from various species and organs, including liver cells[17], white adipose tissue cells[18], and glioblastoma immune response cells[19]. Our results demonstrate that scSpecies effectively aligns network architectures and latent representations. We improve upon cell label transfer from the initial nearest neighbor search and existing architecture surgery approaches when measured in terms of accuracy and multiple clustering metrics.

2. Results

Figure 1. Graphical representation of the scSpecies workflow. Step 1: The encoder and decoder neural networks are trained on the dataset of the context species. The weights of the last encoder layers are incorporated into the encoder model for the target species. Step 2: A nearest neighbor search is performed on the shared genes of the context and target dataset. This identifies a set of k context neighbors for every target cell. Step 3: The cells of the target dataset are encoded into the latent space. For cells with high agreement among the cell labels of their neighbors, we retrieve the latent variables of their neighbors. Step 4: The latent values of their k neighbors are passed to the decoder together with the human batch label. Step 5: The optimal candidate among the k neighbors is chosen as the cell with the highest log-likelihood. Step 6: The distance between the optimal candidate and the intermediate representation of its target cell is minimized. Step 7: After training, normalized gene expression profiles can be compared by decoding latent variables with both decoder networks. Additionally, labels can be transferred via the aligned latent representation.

We present scSpecies, a tool for researchers who wish to use one scRNA-seq dataset as a context for another from a different species. In the following, the dataset of the model organism is referred to as the 'context dataset', and the dataset of the target organism is referred to as the 'target dataset'. scSpecies aligns context scRNA-seq datasets with human target data, enabling the analysis of similarities and differences between the datasets.

In addition to the context and target datasets, the model requires a sequence containing indices of homologous genes, indicator variables for batch effects, and cell type labels for the context dataset.

The proposed workflow (Figure 1) aligns the network architectures of two single-cell variational inference (scVI)[20] models in a pretraining strategy. In scVI, encoder neural networks map gene expression vectors into a compressed latent space separating cells by biological features. Conversely, a

decoder maps from this low-dimensional representation onto parameters of a negative binomial distribution to (re-)generate gene expression data.

First, our proposed approach pretrains a scVI model on the context dataset. Afterwards, the last encoder layers are transferred into a second scVI model for the target species. The aim of this architecture transfer is to share learned information within the network weights between datasets and species. During subsequent fine-tuning, the shared weights remain frozen while all other weights are optimized.

Unlike existing architecture surgery approaches, we align the architectures in a reduced intermediate feature space instead of at the data level. This approach is inspired by the notion of midlevel features from computer vision[21][22]. These represent abstractions of the input image learned by neural networks in their intermediate layers. Midlevel features combine individual elements into more general structures, such as contours, specific shapes, or parts of objects. Transfer learning approaches then retrain the last layers to transition these intermediate representations into task-specific network outputs for different datasets[23].

Unlike images, scRNA-seq datasets lack ordered patterns as gene expression vectors can be permuted without changing their information content. Nevertheless, the first encoder layers translate dataset-specific features, such as influences of experimental batches or interactions between observed genes, into a higher abstraction level (Figure 5). The resulting representation may correspond to more fundamental cell properties that are less perceptible to noise and systematic differences between species.

To connect the new encoder layers with the pretrained structure, we identify sets of similar cells through a nearest neighbor search performed on homologous genes. Afterward, scSpecies minimizes the distance between a target cell's midlevel representation and a suitable candidate from its set of neighbors. The model determines the most suitable context cell as the candidate whose decoded latent representation yields the highest log-density value at the location of the target cell within the decoder's distribution. To counter misclassifications, we align midlevel features for only those target cells whose context neighbors have high agreement in their cell labels.

During model fitting, we thus encode similarity information both at the original data level and at the level of learned features. The aligned latent space then captures cross-species similarity relationships based on the fitted model, which facilitates information transfer across species.

2.1. scSpecies aligns architectures across species

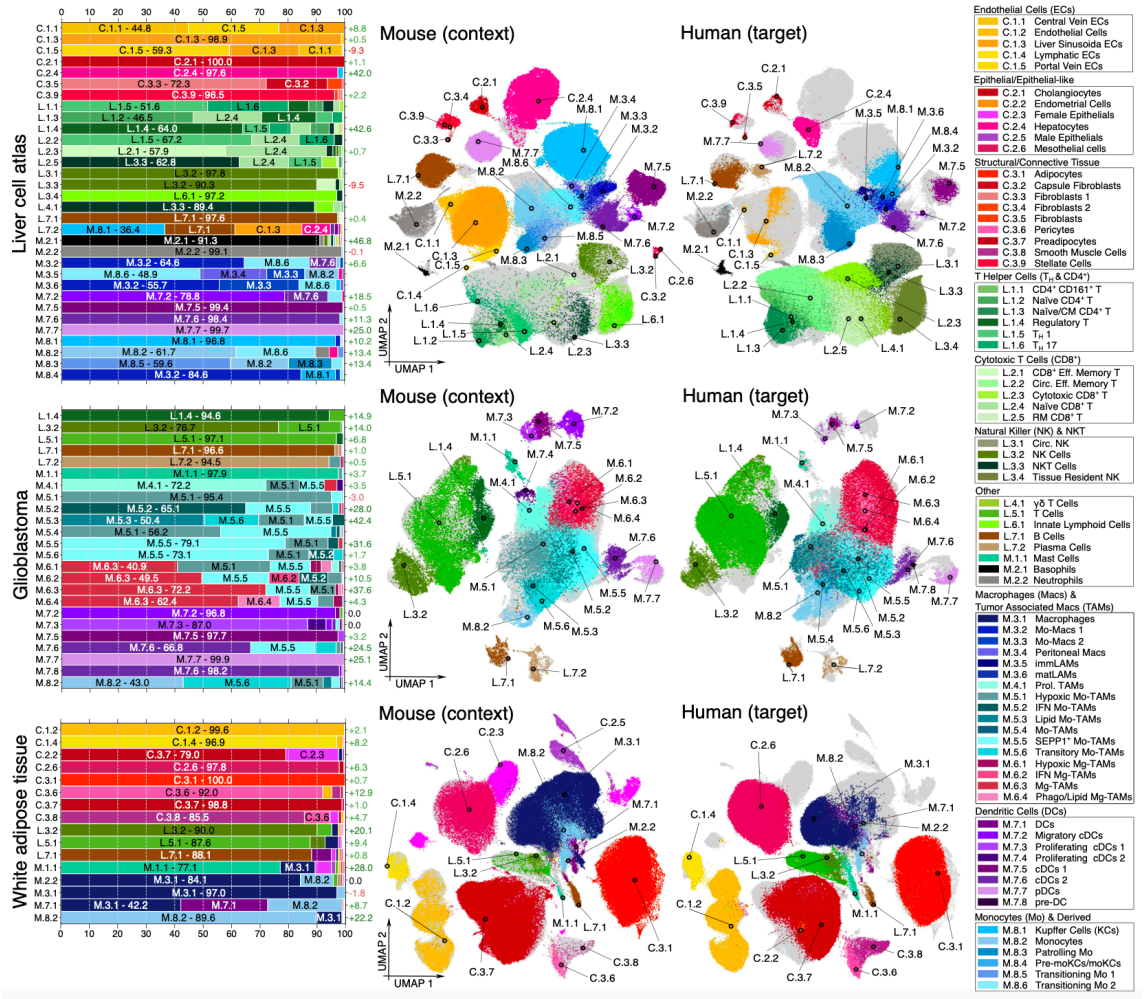

Figure 2. Visualization of the aligned representations for three dataset pairs obtained by training scSpecies with a set of 25 neighbors. We color cells by fine cell type labels for the liver and glioblastoma datasets, and by coarse cell labels for the adipose tissue dataset. On the left, the bar plots indicate the accuracy of cell label transfer through a nearest neighbor search in the aligned latent space. The left y-axis labels indicate cell type codes corresponding to human cell labels. These codes are referenced in the legend. The bars contain the frequency of assigned mouse cell labels. The results are averaged over five random seeds. The left y-axis labels indicate improvement in accuracy for shared cell types over the data-level nearest neighbor search. In addition to the bar plots, the UMAP coordinates of the aligned latent representations are visualized. The lymphoid cell types are colored in green and brown; the myeloid cell types are colored blue and purple; and the CD45$^-$ cell types are colored red, pink and yellow. The cells from the other dataset are indicated in a light gray.

doi.org/10.32388/D37AFF

We applied the scSpecies workflow to three mouse-human dataset pairs containing liver cells, white adipose tissue cells, and immune response cells to glioblastoma.

We visually examined alignment through UMAP coordinates[24] of the combined latent variables of dataset pairs (Figure 2). The 2D representation showed biologically meaningful alignment of the cells. Cell types without context counterparts aligned with related cell types or formed distinct clusters.

To facilitate label and information transfer for target cells, we conducted a second nearest neighbor search on the shared latent representation of both datasets. Afterwards, we inferred target cell labels from their set of latent context neighbors via majority voting. For labels at the subcell type resolution, the accuracy was 73% for liver, 49% for adipose tissue, and 69% for glioblastoma datasets. Misclassifications mostly occurred within biologically related cells belonging to the same overarching cell type. For broader cell type labels, accuracy increased to 92% for the liver, 82% for the adipose tissue, and 80% for the glioblastoma dataset. These values represent significant improvements upon the data-level nearest neighbor search and existing architecture surgery approaches (Table 3). We also calculated the adjusted Rand index and adjusted mutual information and observed improvements in these metrics.

We observed a greater increase in label transfer accuracy for cell types with noisy data-level nearest neighbor search but clear separation in their pretrained latent space. For example, the initial neighbor search matched less than half of all human liver basophils (cluster M.2.1) with mouse counterparts. This value improved to over 90% through our method. However, in the adipose tissue datasets, neither the context scVI model nor the nearest neighbor search separated dendritic cells, monocytes, and macrophages. Thus, scSpecies could not separate these cell types either.

The results were consistent over architecture variations and averaged over five random seeds; however, for cell types with noisy neighbor search results, like hepatocytes or portal vein endothelial cells, misclassifications of the whole cell type occurred in one random seed.

We also tested scSpecies in a scenario where the target dataset was small but equally diverse in terms of cell types and batch effects. Specifically, we randomly sampled 5000 cells from the human liver dataset and trained the model to align with the full mouse context dataset. We repeated sampling and training ten times and obtained accuracy scores of 88% and 68% for coarse and fine cell labels, respectively, which still indicates reasonable performance.

2.2. The nearest neighbor search is an important component of scSpecies

We explored the importance of incorporating the nearest neighbor search into scSpecies. (Table 3) Without this component, we observed misaligned latent representations and significantly reduced label transfer accuracy. Initializing the inner encoder layers with random, frozen weights yielded similar results to using the pretrained structure. This implies that without an explicit neighbor alignment component, transferred layers were treated like random nuisances.

Training with one neighbor forced the model to align some cells with mismatched counterparts as the approach could not choose from a set of suitable options. We observed meaningful alignment but with reduced performance.

Training with 25 neighbors improved the results noticeably on all datasets. To investigate the preferred candidate choice, we tracked the cell prototypes during alignment. We created context and target prototype cells consisting of empirical median gene expression values within a cell type. For each target prototype, we included all context prototypes within its set of candidates and tracked their log-likelihoods during alignment (Figure 10). At onset, the likelihoods for all prototypes were nearly equal. This resulted in alignment driven by chance favoring cell candidates of the most occurring cell label. For cell types with a noisy neighbor set, corrections during later training stages eventually aligned them with appropriate prototypes. We observed this with hepatocytes, migratory cDCs, and basophils, which had nearest neighbor search accuracies of 56%, 61%, and 45%, respectively. The cell types where the neighbor search yielded predominantly incorrect results did not align correctly, such as killer T cells and cytotoxic $CD8^+$ cells, which had initial accuracies of only 11% and 1%, respectively.

Finally, alignment with a large neighbor set caused neglect of rare cell types, resulting in lower corresponding accuracy scores. Metrics such as the adjusted Rand index and adjusted mutual information were comparable or improved, as they do not reflect different cell type label sizes.

2.3. scSpecies can help to better separate latent cell clusters

To investigate the intermediate representations, we compared the clustering quality of intermediate representations in unaligned and aligned scVI architectures. We found that clustering based on experimental batches became increasingly mixed as the data progressed toward the latent space. In the unaligned architectures, the Davies-Bouldin index (DBI) increased from 10 to 21.9 in the mouse context, and from 15.8 to 33.5 in the human liver dataset. Conversely, cell type clusters showed increasingly better

separation, resulting in a DBI reduction from 4.6 to 1.6 and from 4.9 to 2.4 for the mouse and human datasets, respectively (Figures 5,6,7).

This phenomenon is caused by the design of scVI, which removes batch influences to enforce a normal distribution in the latent space. Batch patterns are added by the decoder through their provided labels. However, scVI must separate cell types to reconstruct cell characteristics from the latent representation.

Yet, certain cell types in the human liver dataset, such as hepatocytes, stellate cells, and fibroblasts, are predominantly associated with a single batch label. Consequently, the model inferred cell type information from batch labels, removing biological characteristics from their latent variables. However, these cell types were still separated in the intermediate spaces which are not regularized to follow a normal distribution.

Alignment adjusted the target encoder architecture to the well-separated latent mouse context representation. This improved latent cell cluster separation, as measured by a decrease in DBI from 2.4 to 1.8. For white adipose tissue and glioblastoma dataset pairs, clustering improvement was marginal, with a decrease in DBI from 1.7 to 1.6 and from 2.2 to 2, respectively.

We also studied the effectiveness of directly aligning latent representations. Direct latent alignment does not require access to the context model weights. However, we observed a decline in performance metrics across all datasets. This underlines the potential of better alignment within the more information-rich midlevel feature spaces.

2.4. scSpecies can align datasets of multiple species

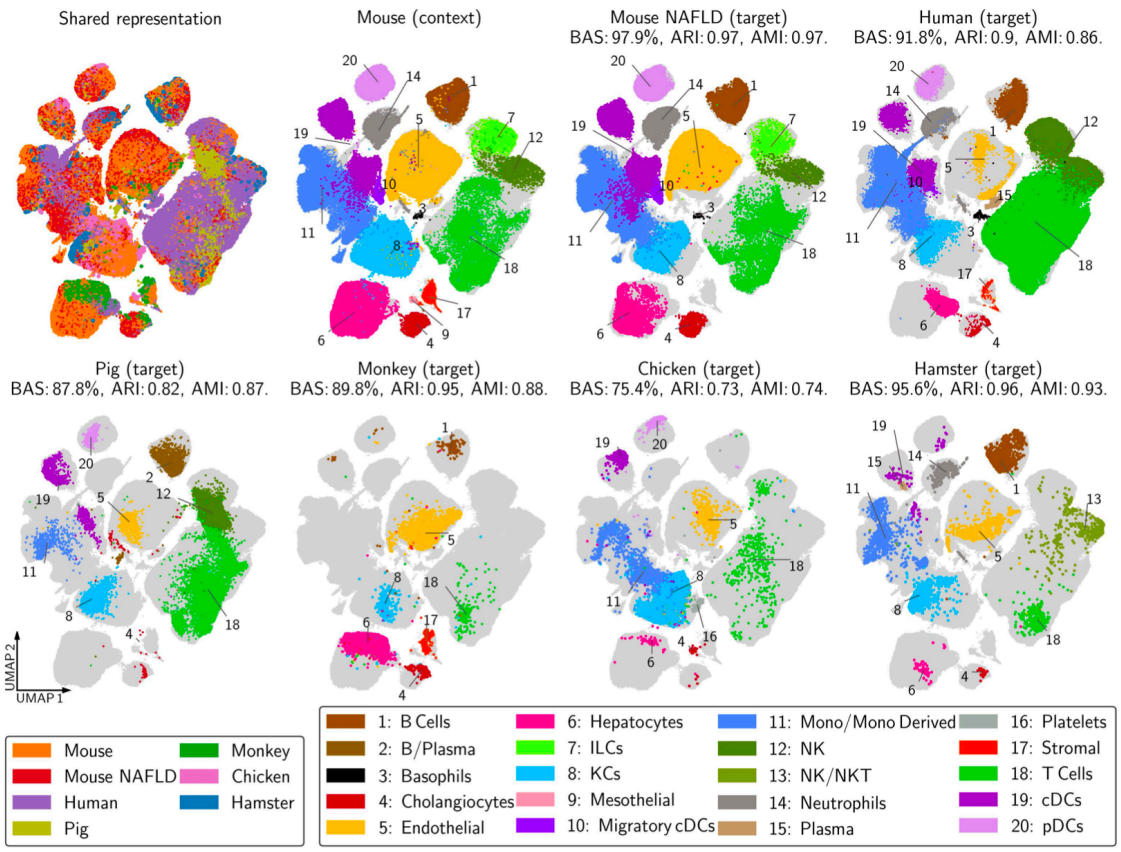

Figure 3. We utilized scSpecies to obtain an aligned liver cell landscape that spans multiple species. The mouse dataset serves as a context for each species.

We employed scSpecies to simultaneously align liver cells from mice with fatty liver disease, humans, pigs, monkeys, chickens, and hamsters, using a context dataset of healthy mice (Figure 3).

We successfully obtained aligned latent representations across species, despite fewer than half of the genes having mouse orthologs in some datasets.

An intriguing application of scSpecies is the potential to align datasets with very limited gene coverage, or even when there is no overlap in the observed gene set. This can be achieved by aligning each dataset to a comprehensive context dataset that shares a common gene set with both.

However, a limitation of this approach is its inability to align cell types not present in the context dataset. For example, plasma cells, which were absent from the mouse dataset, were not aligned across the human, pig, and hamster datasets.

2.5. scSpecies offers insights into the genetic manifestations of cells across species

To better understand the similarities and differences between context and target datasets, e.g., to clarify in what aspects an animal might be a good model of human biological processes, we extended our analysis from the latent space to the data level. Here, we compared the reconstructed gene expression profiles and assigned relevance scores to the input genes.

We decoded latent representations using both decoder models to obtain normalized gene expression vectors for each species. These vectors allow us to compare and analyze the gene expression profiles of cells that have similar underlying biological properties. This analysis benefits from the correspondence between latent representations of both species, which is difficult to establish at the data level.

For our investigation, we focused on cell types present in both the mouse and human liver datasets. We assessed Log2Fold changes (LFCs) in normalized gene expression vectors, which indicate differences in gene expression levels between species. We also calculated the probability of observing genes as differentially expressed when sampling from the latent distribution of a cell type (Figure 8). Averaging across cell types revealed that 56% of the genes exhibited an LFC value above one. Among these, 15% of mouse genes were upregulated and 21% were downregulated compared with their human counterparts in over 90% of decoded cells. With an LFC threshold of two, 24% of genes had an LFC outside this boundary. With an LFC value of 0.4, a substantial 82% of genes showed an LFC outside this boundary. These results agree in magnitude with[25], who found an LFC value of greater than 0.4 in 78% of genes comparing humans with non-alcoholic liver disease and mice on a high-fat diet.

For white adipose tissue datasets, 50%, and for glioblastoma datasets, 47% of genes exhibited an LFC value greater than one.

We compared this with training on context-target dataset pairs of healthy mice and mice with liver disease. Here, only 22% of genes had an LFC value above one. Of those differentially expressed genes, 4% and 5% were upregulated and downregulated in more than 90% of samples. Only 6% of genes had an LFC over two, while 55% of genes showed LFC values above 0.4.

We extended our study by calculating relevance scores via Layer-wise relevance propagation (LRP) [26] (Figure 9). These scores measure each gene's contribution to a cell's latent value, offering insights into the learned significance of specific genes across different cell types and species. LRP was recently used to explain neural network predictions on scRNA-seq data[27].

First, we found no significant difference in relevance scores between non-homologous and shared genes, suggesting that training networks on a reduced gene set omits informative parts of the data.

Second, we found that the relevance scores were correlated with the gene expression levels. For the mice and human liver datasets, we found a Spearman's ρ between the expression level of genes and their relevance scores of 0.67 and 0.69 and a Pearson correlation coefficient of 0.63 and 0.71. This suggests that differences in gene expression translate into relevant features for the neural networks. A gene with high relevance scores across most cell types was *MALAT1*, which is highly conserved across mammals[28].

3. Discussion

We introduced scSpecies, a novel deep learning approach designed to align neural network architectures across different species. Aligning such architectures has been a challenging task due to differences in genomes between species and variations in gene expression levels, even among homologous genes. Key features of scSpecies include the retraining of the first encoder layers and integrating a nearest neighbor search within the model. By focusing on the alignment of intermediate neural network layers rather than the input layers, scSpecies captures more abstract biological properties that are less affected by noise and species-specific variations. Additionally, the integration of a nearest neighbor search based on homologous genes leverages model-based similarity information to guide the alignment process, ensuring that biologically similar cells are mapped closely in the latent space.

Our results demonstrate that scSpecies effectively aligns scRNA-seq data from diverse species, including mouse, human, pig, monkey, chicken, and hamster, across various tissues such as liver, white adipose tissue, and glioblastoma cells. The method shows robust performance even when the datasets have a limited number of shared genes or when the target dataset is small but diverse.

However, one limitation of the presented method is that cell types unique to the target dataset tend to be aligned with biologically close cell types in the context dataset instead of being identified as new clusters by the model. This could lead to misinterpretation of species-specific cell populations. Additionally, when creating a collection of multiple species, cell types not present in the context dataset will not align across species that exhibit them. To avoid misalignment, the context dataset should therefore encompass all suspected cell types of the reference datasets.

There remain multiple potential directions for further development of our approach. While we initially tested scSpecies with a scVI base model, the method could be easily adapted to other CVAE-based models in the future. Furthermore, scSpecies could be extended to handle multimodal datasets, such as those

integrating scRNA-seq with protein expression data (CITE-seq). Our method would also benefit from a direct metric that identifies cell types unique to the target datasets and detects cells that may be misclassified due to noisy nearest neighbor search results.

4. Conclusions

We have introduced scSpecies, a novel deep learning approach that extends architecture surgery techniques to align scRNA-seq datasets across species. By retraining the first encoder layers, our method overcomes challenges posed by non-orthologous genes and divergent gene expression patterns, enabling more accurate cross-species comparisons. By aligning datasets from multiple species — even with minimal gene overlap — scSpecies provides a framework to better understand and compare the cellular and molecular similarities and differences of scRNA-seq datasets across species. Therefore, we envision that our method could lead to more effective translation of experimental findings from model organisms to humans, ultimately advancing our understanding of human biology.

5. Methods

In the following, we represent multidimensional vectors using bold italics and scalar values in regular italics. Dataset elements are indicated with superscript indices, and vector positions with subscript indices. The context dataset is indicated by the subscript C and the target dataset by the subscript T. Superscripts and subscripts are omitted when they are exchangeable. Random variables are expressed in a sans-serif mathematical font, as in $\mathsf{X}, \mathsf{Z}, \mathsf{L}$. We represent distributions of random variables with uppercase letters, such as P_{Z}, and their probability density functions with lowercase letters, like $\mathsf{p}_{\mathsf{Z}}(\boldsymbol{z})$. Conditional distributions are denoted as $\mathsf{P}_{\mathsf{X}|\boldsymbol{s}} := \mathsf{P}_{\mathsf{X}|\mathsf{S}=\boldsymbol{s}}$. In the following, we briefly describe the scVI model, which we subsequently use as a core of our proposed approach.

5.1. Single cell variational inference

Consider a dataset $\mathbb{D} = \left\{\left(\boldsymbol{x}^{(i)}, \boldsymbol{s}^{(i)}\right)\right\}_{i=1}^{M}$ obtained through a single-cell RNA sequencing experiment. The mathematical model behind scVI[20] assumes that gene expression count vectors $\boldsymbol{x}$, and batch indicator variables $\boldsymbol{s}$, correspond to observations of random variables X and S. The gene expression data distribution $\mathsf{P}_{\mathsf{X}|\boldsymbol{s}}$ is conditioned on its batch effect $\mathsf{S} = \boldsymbol{s}$. This accounts for technical artifacts during data collection. Within an experimental batch, gene expression vectors are independent and identically distributed samples from $\mathsf{P}_{\mathsf{X}|\boldsymbol{s}}$.

scVI models the data distribution within a parametric family. Building on conditional variational autoencoders[16], a latent variable model is introduced. The random variable Z, corresponding to the representation of a cell in the latent space $\mathbb{R}^d$, is employed to capture biological variability among cells in the dataset. The one-dimensional random variable L with latent space $\mathbb{R}_{>0}$ accounts for technical variability due to different library sizes. Within the model, data is generated by drawing samples for Z and L from a prior distribution $\mathsf{P}_{\mathsf{Z},\mathsf{L}|\boldsymbol{s}}$. Then, gene expression data is generated by drawing from the sampling distribution $\mathsf{P}_{\mathsf{X}|\boldsymbol{z},l,\boldsymbol{s}}$.

The data p.d.f. $\mathsf{p}_{\mathsf{X}|\boldsymbol{s}}$ can be expressed by integrating the joint probability across the latent spaces and then applying the general product rule of probability,

$$\mathsf{p}_{\mathsf{X}|\boldsymbol{s}}(\boldsymbol{x}) = \int_{\boldsymbol{z}}\int_{l} \mathsf{p}_{\mathsf{X}|\boldsymbol{z},l,\boldsymbol{s}}(\boldsymbol{x})\mathsf{p}_{\mathsf{Z},\mathsf{L}|\boldsymbol{s}}(\boldsymbol{z},l)\,\mathrm{d}\boldsymbol{z}\mathrm{d}l. \tag{1}$$

To approximate this integral, scVI performs variational inference on the intractable posterior distribution $\mathsf{P}_{\mathsf{Z},\mathsf{L}|\boldsymbol{x},\boldsymbol{s}}$. Therefore, the posterior probability is approximated by a variational distribution, denoted as $\mathsf{Q}_{\mathsf{Z},\mathsf{L}|\boldsymbol{x},\boldsymbol{s}} \approx \mathsf{P}_{\mathsf{Z},\mathsf{L}|\boldsymbol{x},\boldsymbol{s}}$. Further, scVI applies a mean field approximation, where p.d.fs of both variational and prior distribution are factorized,

$$\mathsf{q}_{\mathsf{Z},\mathsf{L}|\boldsymbol{x},\boldsymbol{s}}(\boldsymbol{z},l) = \mathsf{q}_{\mathsf{Z}|\boldsymbol{x},\boldsymbol{s}}(\boldsymbol{z})\mathsf{q}_{\mathsf{L}|\boldsymbol{x},\boldsymbol{s}}(l),\ \mathsf{p}_{\mathsf{Z},\mathsf{L}|\boldsymbol{s}}(\boldsymbol{z},l) = \mathsf{p}_{\mathsf{Z}}(\boldsymbol{z})\mathsf{p}_{\mathsf{L}|\boldsymbol{s}}(l). \tag{2}$$

The prior P_{Z} is assumed to be independent of S and fixed as standard normal distribution $\mathsf{P}_{\mathsf{Z}} = \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. The prior $\mathsf{P}_{\mathsf{L}|\boldsymbol{s}}$ is set as a log-normal distribution $\mathsf{P}_{\mathsf{L}|\boldsymbol{s}} = \mathrm{LogNormal}(\boldsymbol{l}_{\mu}^{\top}\boldsymbol{s}, \boldsymbol{l}_{\sigma^2}^{\top}\boldsymbol{s})$. The prior parameters are derived from empirical batch means and variances of the observed log-library sizes. The variational distribution $\mathsf{Q}_{\mathsf{Z}|\boldsymbol{x},\boldsymbol{s}}$ is chosen as a normal distribution $\mathcal{N}(\boldsymbol{\mu}_{\mathsf{Z}}, \boldsymbol{\sigma}_{\mathsf{Z}}^2\boldsymbol{I}_d)$, and $\mathsf{Q}_{\mathsf{L}|\boldsymbol{x},\boldsymbol{s}}$ is set as a log-normal distribution $\mathrm{LogNormal}(\mu_{\mathsf{L}}, \sigma_{\mathsf{L}}^2)$.

The parameters for these distributions are determined by two encoder neural networks,

$$f_{\mathsf{enc}\,\mathsf{Z}}(\boldsymbol{x},\boldsymbol{s}) = (\boldsymbol{\mu}_{\mathsf{Z}}, \boldsymbol{\sigma}_{\mathsf{Z}}) \text{ and } f_{\mathsf{enc}\,\mathsf{L}}(\boldsymbol{x},\boldsymbol{s}) = (\mu_{\mathsf{L}}, \sigma_{\mathsf{L}}). \tag{3}$$

scVI obtains latent variables by sampling from the variational distributions through the reparametrization trick[29].

The sampling distribution $\mathsf{P}_{\mathsf{X}|\boldsymbol{z},l,\boldsymbol{s}}$ for generating gene-expression data from a given latent variable is assumed to follow a Gamma-Poisson mixture, resulting in a negative binomial distribution. The corresponding decoder network outputs a denoised gene expression vector that sums to one.

$$f_{\mathsf{dec}}(\boldsymbol{z},\boldsymbol{s}) = \boldsymbol{\rho},\ \sum_{g=1}^{N}\rho_g = 1. \tag{4}$$

The value ρ_g provides an estimate of the percentage of transcripts in a cell that originate from gene g. Gene expression values x_g can be drawn from a negative binomial distribution $\mathrm{NB}(l\rho_g, \theta_{g,\boldsymbol{s}})$ parameterized by mean $l\rho_g$ and dispersion $\theta_{g,\boldsymbol{s}}$. The dispersion parameter is constant for every gene across cells of batch $\boldsymbol{s}$. To address the potential issue of dropout, a zero-inflated negative binomial distribution can be used to model count data. The dropout probability parameter $\boldsymbol{\pi}$ is also obtained from the decoder network. The weights of the three neural networks and the parameters $\theta_{g,\boldsymbol{s}}$ are optimized simultaneously by empirically estimating and maximizing the ELBO function

$$\begin{aligned}\mathrm{ELBO}\big(\boldsymbol{x}, \boldsymbol{s}, \beta\big) = \mathsf{E}_{\mathsf{q}_{\mathsf{Z},\mathsf{L}|\boldsymbol{x},\boldsymbol{s}}}\left[\log \mathsf{p}_{\mathsf{X}|\boldsymbol{z},l,\boldsymbol{s}}(\boldsymbol{x})\right] - \\ \beta\left(D_{\mathsf{KL}}\left[\mathsf{Q}_{\mathsf{Z}|\boldsymbol{x},\boldsymbol{s}}\mathsf{P}_{\mathsf{Z}}\right] + D_{\mathsf{KL}}\left[\mathsf{Q}_{\mathsf{L}|\boldsymbol{x},\boldsymbol{s}}\mathsf{P}_{\mathsf{L}|\boldsymbol{s}}\right]\right)\end{aligned} \tag{5}$$

on mini batches $\mathbb{M} \subset \mathbb{D}$.

5.2. The scSpecies approach

We consider a scenario involving two scRNA-seq datasets,

$$\mathbb{D}_C = \left\{\left(\boldsymbol{x}_C^{(i)}, \boldsymbol{s}_C^{(i)}, c_C^{(i)}\right)\right\}_{i=1}^{M_C} \text{ and } \mathbb{D}_T = \left\{\left(\boldsymbol{x}_T^{(j)}, \boldsymbol{s}_T^{(j)}\right)\right\}_{j=1}^{M_T}. \tag{6}$$

Their data points consist of gene expression measurements $\boldsymbol{x}$ and batch indicator variables $\boldsymbol{s}$ from a context species C and a target species T. Furthermore, context count vectors are clustered into distinct groups based on cell type labels c_C, whereas target labels c_T are unknown.

The count vectors from both datasets share a gene subset $\boldsymbol{h}$ comprising count values from homologous genes,

$$\boldsymbol{x} = (\underbrace{x_1, \ldots, x_H}_{\boldsymbol{h}\text{ homologous}}, \underbrace{x_{H+1}, \ldots, x_N}_{\text{non-homologous}})^\top. \tag{7}$$

The number of non-homologous genes can differ in both datasets, either because a gene has no ortholog in the genome of the other species or because it is not observed within the dataset. Therefore, gene expression vectors can be of different dimension, $N_C \neq N_T$.

To map both datasets into a unified latent space, we define separate scVI models for each dataset,

$$\mathrm{scVI}^C = \left(f^C_{\mathsf{encZ}}, f^C_{\mathsf{encL}}, f^C_{\mathsf{dec}}\right),\ \mathrm{scVI}^T = \left(f^T_{\mathsf{encZ}}, f^T_{\mathsf{encL}}, f^T_{\mathsf{dec}}\right). \tag{8}$$

We divide the training procedure for scSpecies into three steps: Training of the context scVI model, followed by an initial data-level nearest neighbor search, and alignment of context and target latent representations.

5.2.1. Pretraining on the context dataset

First, the model scVI^C is trained on the context dataset by minimizing its negative ELBO function. Following training, the architecture of the encoder network for the latent variable Z is split up into two parts:

$$f^C_{\mathsf{enc\,Z}} = f^C_{\mathsf{outer}} \circ f^C_{\mathsf{inner}}. \tag{9}$$

The outer part f^C_{outer} consists of the first L layer functions and maps data from the input space $\mathcal{X}_C$ to an intermediate feature space $\mathcal{T}$. The inner part, f^C_{inner}, consists of the last M layers. It encodes an intermediate representation onto the variational parameters with subsequent reparametrization into the latent space $\mathcal{Z}$. We incorporate this inner encoder part into the encoder architecture of scVI^T,

$$f^T_{\mathsf{enc\,Z}} = f^C_{\mathsf{outer}} \circ f^T_{\mathsf{inner}}. \tag{10}$$

5.2.2. Nearest neighbor search

When the first layers are initialized randomly, the target model scVI^T cannot leverage the learned structure in its subsequent encoder layers. To leverage the learned weights, we incentivize alignment of intermediate target representations with intermediate features of similar context cells. This leads to an aligned latent space as layer weights mapping from the intermediate space to the latent space are not updated. To quantify similarity and establish a direct correspondence between cells of context and target dataset, we perform a nearest neighbor search on the shared homologous gene subset $\boldsymbol{h}$. The nearest neighbors serve as a set of candidates for every target cell from which the model can choose a best fit to align their intermediate representations during the last training phase.

The nearest neighbor search identifies an index set $\mathbb{I}_k\left(\boldsymbol{x}_T^{(j)}\right) \subset \mathbb{I}_C$ of k nearest neighbors for every target gene count vector $\boldsymbol{x}_T^{(j)}$. That is, for every context cell with index $i \in \mathbb{I}_k\left(\boldsymbol{x}_T^{(j)}\right)$, the chosen measure of association[1] between the homologous gene counts $\boldsymbol{h}_C^{(i)}$ and $\boldsymbol{h}_T^{(j)}$ is lower than for cells outside the set:

$$d\big(\boldsymbol{h}_C^{(i)}, \boldsymbol{h}_T^{(j)}\big) \leq d\big(\boldsymbol{h}_C^{(l)}, \boldsymbol{h}_T^{(j)}\big) \text{ for all } l \in \mathbb{I}_C \backslash \mathbb{I}_k\left(\boldsymbol{x}_T^{(j)}\right). \tag{11}$$

Common metrics or distance functions can be used as a measure of association d to compare count values of single-cell data. Some popular choices have been investigated in[30]. We utilize cosine similarity, measuring the cosine of the angle between log1p-transformed count vectors, as it is fast to calculate even on datasets containing numerous samples:

$$d\big(\boldsymbol{h}_C^{(i)}, \boldsymbol{h}_T^{(j)}\big) = 1 - \frac{\Big\langle \log\big(\boldsymbol{h}_C^{(i)} + 1\big), \log\big(\boldsymbol{h}_T^{(j)} + 1\big) \Big\rangle}{\Big\| \log\big(\boldsymbol{h}_C^{(i)} + 1\big) \Big\|_2 \Big\| \log\big(\boldsymbol{h}_T^{(j)} + 1\big) \Big\|_2}. \tag{12}$$

The data-level nearest neighbor search can also be used to assign preliminary labels. We count the multiplicity of cell labels for all context neighbors and assign, as a preliminary label prediction, the most occurring label,

$$\hat{c}_T^{(j)} = \mathrm{mode}\Big[c_C^{(i)} : i \in \mathbb{I}_k\big(\boldsymbol{x}_T^{(j)}\big)\Big]. \tag{13}$$

As the data-level nearest neighbor search is noisy, we additionally assign agreement scores based on the occurrence of a cell label prediction $\hat{c}_T^{(j)}$.

$$P\big(\hat{c}_T^{(j)}\big) = \frac{\big|\big\{i : c_C^{(i)} = \hat{c}_T^{(j)} \text{ and } i \in \mathbb{I}_k\big(\boldsymbol{x}_T^{(j)}\big)\big\}\big|}{k} \tag{14}$$

A higher agreement score indicates lower noise, as there is high agreement among cell labels of the context neighbors. During the following alignment, only target cells exhibiting high agreement scores are considered for alignment in the intermediate space. For this, we collect all agreement scores for target cells predicted to have label $\hat{c}_T^{(j)}$ and compute the quantile at level p over this set $\big\{P\big(\hat{c}\big) : \hat{c} = \hat{c}_T^{(j)}\big\}$. Finally, we collect the indices of all target cells whose agreement scores of their predicted cell label are higher than the quantile Q at level p,

$$\mathbb{J}(p) = \Big\{j : P\big(\hat{c}_T^{(j)}\big) > Q\Big(p, \big\{P\big(\hat{c}\big) : \hat{c} = \hat{c}_T^{(j)}\big\}\Big)\Big\}. \tag{15}$$

5.2.3. Aligning the intermediate and latent representations

During alignment, the weights of the pretrained encoder part f_{inner}^C are not updated. To guide the model towards leveraging the learned structure, scSpecies aligns intermediate representations with high accuracy scores

$$\boldsymbol{t}_T^{(j)} = f_{\mathsf{outer}}^T\big(\boldsymbol{x}_T^{(j)}, \boldsymbol{s}_T^{(j)}\big), j \in \mathbb{J}(p) \tag{16}$$

with a representation of a suitable context neighbor representation

$$\boldsymbol{t}_C^{(i^*)} = f_{\mathsf{outer}}^C\big(\boldsymbol{x}_C^{(i^*)}, \boldsymbol{s}_C^{(i^*)}\big), i^* \in \mathbb{I}_k\big(\boldsymbol{x}_T^{(j)}\big). \tag{17}$$

This is facilitated by minimizing the squared Euclidean distance.

$$\text{minimize} \Big\| \boldsymbol{t}_T^{(j)} - \boldsymbol{t}_C^{(i^*)} \Big\|_2^2, \text{ if } j \in \mathbb{J}(p). \tag{18}$$

The optimal choice $i^* \in \mathbb{I}_k$ for minimization among the k candidates is dynamically determined during the alignment phase: First, we obtain a set of latent context neighbor variables for the target cells considered during alignment,

$$\mathbb{L}_k\left(\boldsymbol{x}_T^{(j)}\right) = \left\{\boldsymbol{z}_C^{(i)} : i \in \mathbb{I}_k\left(\boldsymbol{x}_T^{(j)}\right)\right\}. \tag{19}$$

These latent variables $\boldsymbol{z}_C^{(i)}$ are then decoded with the batch indicator variable $\boldsymbol{s}_T^{(j)}$ of their target cell. The decoder output and target library size $l_T^{(j)}$ parameterize a sampling distribution $\mathsf{P}_{\mathsf{X}|\boldsymbol{z}_C^{(i)}, l_T^{(j)}, \boldsymbol{s}_T^{(j)}}$, which is used to calculate log density values for every candidate. The cell i^* whose latent representation results in the highest log density value at $\boldsymbol{x}_T^{(j)}$ is chosen as optimal neighbor candidate:

$$\boldsymbol{z}_C^{(i^*)} = \underset{\boldsymbol{z}_C^{(i)} \in \mathbb{L}_k\left(\boldsymbol{x}_T^{(j)}\right)}{\operatorname{argmax}} \log\left(\mathsf{p}_{\mathsf{X}|\boldsymbol{z}_C^{(i)}, l_T^{(j)}, \boldsymbol{s}_T^{(j)}}\left(\boldsymbol{x}_T^{(j)}\right)\right). \tag{20}$$

Using this procedure, it is possible to assign a context neighbor with a fitting cell type if at least one candidate with this cell type is found in this set. The training criterion for the model scVI^T on the target dataset for a data point is

$$-\mathrm{ELBO}\left(\boldsymbol{x}_T^{(j)}, \boldsymbol{s}_T^{(j)}, \beta\right) + \gamma\left\|\boldsymbol{t}_T^{(j)} - \boldsymbol{t}_C^{(i^*)}\right\|_2^2 \left[j \in \mathbb{J}(p)\right], \tag{21}$$

where $[j \in \mathbb{J}(p)]$ is the Iverson Bracket that takes value 1 when an index of a target cell j is in $\mathbb{J}(p)$, and 0 otherwise. This holds true for cells that exhibited a high degree of agreement during the data-level nearest neighbor search. As minimization in the intermediate space is only incentivized for cells with these indices, the remaining cells within a mini-batch are grouped around them in a way that minimizes the nELBO of the scVI model.

The scalars $\gamma, \beta \geq 0$ weighing different parts of the loss function, the quantile niveau $p \in [0, 1]$ and number of nearest neighbors $k \in \mathbb{N}$ are hyperparameters.

5.2.4. Transferring cell states and cell types

The aligned latent representations $\mathbb{L}_C = \left\{\boldsymbol{z}_C^{(i)}\right\}_{j=1}^{M_C}$ and $\mathbb{L}_T = \left\{\boldsymbol{z}_T^{(j)}\right\}_{i=1}^{M_T}$ can be analyzed for similarities and differences. For example, their dimensionality can be further reduced into two dimensions using a dimension reduction algorithm like UMAP[24]. To remove the random influence of the latent sampling process, we calculate UMAP coordinates using the variational mean parameters $\boldsymbol{\mu}$.

We can transfer cell labels or cell states from the context to target species by performing a second neighbor search on aligned latent representations. A suitable measure of association is the learned log-

density, as it considers the learned manifold of the latent space:

$$d\left(\boldsymbol{z}_C^{(i)}, \boldsymbol{z}_T^{(j)}\right) = -\log\left(\mathsf{p}_{\mathsf{X}|\boldsymbol{z}_C^{(i)}, l_T^{(j)}, \boldsymbol{s}_T^{(j)}}\left(\boldsymbol{x}_T^{(j)}\right)\right) \tag{22}$$

We transfer the most common cell type among the top k candidates to the target cell.

5.2.5. Comparison of gene profiles

To perform a comparison of gene expression profiles between cells of context and target dataset, we tailor the methods outlined in[31] and[32] to scSpecies. For a latent variable $\boldsymbol{z}$, we obtain normalized gene expression profiles by decoding it with both decoder networks and averaging over all possible batches $\mathbb{S}$:

$$\boldsymbol{\rho}_C = \frac{1}{|\mathbb{S}_C|}\sum_{\boldsymbol{s}_C \in \mathbb{S}_C} f_{\mathsf{dec}}^C(\boldsymbol{z}, \boldsymbol{s}_C),\ \boldsymbol{\rho}_T = \frac{1}{|\mathbb{S}_T|}\sum_{\boldsymbol{s}_T \in \mathbb{S}_T} f_{\mathsf{dec}}^T(\boldsymbol{z}, \boldsymbol{s}_T) \tag{23}$$

Differences in gene expression profiles can be analyzed for homologous genes, for example, by calculating the log2-fold change (LFC)

$$r_{C,T}^g = \log_2\left(\frac{\rho_{C,g} + \varepsilon}{\rho_{T,g} + \varepsilon}\right) \tag{24}$$

For genes g with low expression levels in both species but still high differences, the offset ε ensures the associated LFC maintains a low order of magnitude. We modify the decoder output layers to avoid artifacts from the softmax function. These artifacts can arise due to highly expressed non-homologous genes or due to different data dimensions. We apply the softmax function to homologous and non-homologous genes separately to obtain

$$\boldsymbol{\rho}_{\mathsf{hom}} = \mathrm{softmax}(\rho_1, \ldots, \rho_H),\ \boldsymbol{\rho}_{\mathsf{nhom}} = \mathrm{softmax}(\rho_{H+1}, \ldots, \rho_N), \tag{25}$$

where N is the dimensionality of the gene expression vector and H the number of homologous genes. Afterwards, both vectors are scaled so that they sum to one,

$$\boldsymbol{\rho} = \left(\frac{H}{N}\boldsymbol{\rho}_{\mathsf{hom}}^\top, \frac{N-H}{N}\boldsymbol{\rho}_{\mathsf{nhom}}^\top\right)^\top. \tag{26}$$

Following[32], for a cell type $\mathsf{C} = c_C$ we calculate a mixture distribution of latent states.

$$\mathsf{p}_{\mathsf{C}}\left(\boldsymbol{z}_C\right) = \frac{1}{|\mathbb{C}_C(c_C)|}\sum_{\boldsymbol{x}_C^{(i)} \in \mathbb{C}_C(c_C)} \mathsf{q}_{\mathsf{Z}|\boldsymbol{x}_C^{(i)}, \boldsymbol{s}_C^{(i)}}(\boldsymbol{z}_C) \tag{27}$$

The set $\mathbb{C}_C(c_C)$ is the set of cells with label c_C with removed outliers. These outliers are identified by estimating the covariance matrix from variational mean samples $\boldsymbol{\mu}_C$. Cells whose variational mean falls

outside the 90%-confidence ellipse described by the covariance estimate are removed. An LFC distribution of homologous genes for cell types present in both datasets can be estimated by sampling latent variables from P_C and computing the corresponding LFC values $r^g_{C,T}$. We calculate the median of the empirical LFC distribution as well as the probability $P(|r^g_{C,T}| > \delta)$ of observing an LFC in gene g higher than level $\delta > 0$.

5.3. Layer-wise relevance propagation

In the following, we briefly describe Layer-wise Relevance Propagation (LRP)[26]. LRP explains the output $f(\boldsymbol{x})$ of a neural network f by decomposing it into local contributions of input nodes x_i, called relevance scores $R_i(x_i)$[26]. These relevance scores serve as a measure of each input's influence on the network's output: positive scores ($R_i > 0$) signify a positive influence, whereas negative scores ($R_i < 0$) indicate a negative effect. LRP structurally decomposes the function learned by neural networks into a set of smaller, simpler sub-functions of adjacent layers, while ensuring the conservation of relevance scores across the network. This applies locally, where the sum of the relevance score R_i is conserved across two successive layers of the neural network, and globally between the resulting relevance score for each input node x_i and the output $f(\mathbf{x})$ of the model[26].

Considering a neural network with ReLU activation function, the output a_k of a neuron is given by the input $\hat{a}_j$ of the previous layer and their connected weights w_{jk} of the neurons by

$$a_k = \max\left(0, \sum_j \hat{a}_j w_{jk}\right), \tag{28}$$

including the bias with $\hat{a}_0 = 1$. The relevance scores R_k describe the contribution of each neuron activation $\hat{a}_j$ to a_k. They can be computed by the LRP-γ rule through

$$R_j = \sum_k \frac{\hat{a}_j(w_{jk} + \gamma w^+_{jk})}{\sum_l \hat{a}_l(w_{lk} + \gamma w^+_{lk})} R_k. \tag{29}$$

Here, w^+_{jk} are the positive weights, while γ controls how much these positive contributions are emphasized[33]. LRP methodology aligns with the principles of Deep Taylor Decomposition, which breaks down and redistributes the network's output function $f(\boldsymbol{x})$ layer by layer through Taylor series expansions. This decomposition allows for the derivation of various LRP rules tailored to the network architecture and the specific function being analyzed[34]. To compute relevance scores for context and target gene expression vectors $\boldsymbol{x}_C, \boldsymbol{x}_T$ we propagated the relevance of their latent variational mean parameters $\boldsymbol{\mu}_C, \boldsymbol{\mu}_T$ through the corresponding encoder network. We aggregate relevance scores through

averaging over latent dimensions and data points of a cell type. A direct comparison of scores between species is complicated by the influence of non-homologous genes and batch-effects on the relevance scores of homologous genes through the conservation property. Rather, ranked lists of genes by scores can be compared across species.

5.4. Metrics

We evaluated label transfer and clustering performance using four key metrics:

- **BAS:** The balanced accuracy score calculates the proportion of cells correctly labeled in both context and target datasets, averaging over all shared cell types and adjusting for the occurrence of smaller cell labels by weighing them equally.
- **ARI:** The adjusted Rand index[35] measures the similarity between predicted and true cell labels, correcting for chance. It considers both correct pairings and misclassifications.
- **AMI:** The adjusted mutual information[35] quantifies how much information the predicted labels share with the true labels, adjusting for random label assignments.
- **DBI:** The Davies-Bouldin index[36] evaluates clustering quality by comparing the compactness of clusters to the separation between them. Lower values indicate better clustering.

These metrics collectively assess the accuracy of cell type label transfer and the quality of cell clustering in the aligned latent space. Details regarding their calculation are found in the documentation of the package skikit learn[37] which we used to calculate these metrics.

5.5. Hyperparameters

Model	Layer	In	Architecture	Out
f_{outer}	1	$N+S$	$\xrightarrow{\text{Linear, LN, ReLU, Dropout}}$	300
f_{inner}	1	300	$\xrightarrow{\text{Linear, LN, ReLU, Dropout}}$	200
	2	200	$\xrightarrow{\text{Linear}} 2 \cdot 10 \xrightarrow{\text{Rep. trick}}$	10
$f_{\text{enc L}}$	1	$N+S$	$\xrightarrow{\text{Linear, LN, ReLU, Dropout}}$	200
	2	200	$\xrightarrow{\text{Linear}} 2 \cdot 1 \xrightarrow{\text{Rep. trick}}$	1
f_{dec}	1	$10+S$	$\xrightarrow{\text{Linear, LN, ReLU, Dropout}}$	200
	2	200	$\xrightarrow{\text{Linear, LN, ReLU, Dropout}}$	300
	3	300	$\xrightarrow{\text{Linear, (Softmax, Sigmoid)}}$	$2N$
	$\theta_{g,s}$	S	$\xrightarrow{\text{Matrix multiplication}}$	N

Table 1. The network architecture used for all models. N denotes the gene expression data dimension, and S the number of batch effects. Layer functions contain an affine linear transformation, followed by layer normalization (LN), ReLU activation functions which are clipped to the interval $[0, 6]$, and dropout layers with a dropout rate of $p = 0.1$. Latent representations are obtained from the variational mean and scale encoder model output via the reparametrization trick.

All models were trained with the same network architecture. Gene expression was modeled using a zero-inflated negative binomial distribution with constant dispersion for genes within an experimental batch. We chose a 10-dimensional latent space and a 300-dimensional intermediate space and mapped to and from these spaces with network architectures listed in Table 1. We trained models for 30 epochs on datasets with more than 10,000 cells and 60 epochs on datasets with less observed samples. Network parameters were updated with the ADAM optimizer[38] using standard hyperparameters and a batch size of $M = 128$.

We chose to weigh the KL-Divergence terms with $\beta = 0.1$ at epoch 1, incrementally increasing their influence to $\beta = 1$ over 10 epochs. Similarly, the alignment term started with a weight of $\eta = 10$, which was raised to $\eta = 25$. The number of nearest neighbors was set to $k = 25$ and the quantile cut-off for alignment was set to $p = 0.8$ across datasets exceeding 10,000 samples. For smaller datasets, we lowered the threshold to $p = 0.6$ to avoid discrimination against scarce cell types. In the latent nearest neighbor search, we pre-computed for each target cell a set of 200 nearest neighbors using the Euclidean distance between the variational mean vectors. Among the 25 cells that resulted in the highest likelihood values, we transferred the most occurring cell label. For differential gene expression analysis, we sampled 10,000 times from the plugin estimator and set the offset variable to $\varepsilon = 10^{-6}$.

To compute layer-wise relevance scores we retrained the networks with unbounded ReLU activation functions and without layer normalization, as it is difficult for LRP to handle normalization layers. To counteract exploding intermediate values caused by high gene expression values, we trained the model on log1p-transformed values. Omitting layer normalization lead to a slight performance drop of around 2.5% across all performance metrics. We calculated relevance scores using the LRP-γ rule with $\gamma = 0.15$.
We trained both scArches and scPoli on a scVI base model using the scArches package implementation. These models were trained with the same network architecture as scSpecies. We trained both models on homologous genes, as the scArches publication states that zero-filling only produces reliable results when less than 25% of genes are affected[9][See feature overlap between reference and query]. scPoli received training with 10-dimensional batch representations. All other hyperparameters were left at default values.

5.6. Pre-processing of the datasets

Dataset	Organism	Shared genes	Cells	Batches	Number of cell types	
		H	M	S	Coarse	Fine
Liver	C Mouse	4 000	165 680	34	15 (15)	36 (36)
	T Mouse NAFLD	2 860	91 787	22	14 (14)	28 (22)
	T Human	1 808	146 839	30	15 (14)	32 (20)
	T Human small	1 808	5 000	30	15 (14)	32 (20)
	T Pig	1 694	21 907	2	9 (8)	unknown
	T Monkey	1 293	8 483	2	7 (7)	unknown
	T Chicken	1 197	7 456	2	9 (7)	unknown
	T Hamster	1 662	5 955	2	11 (9)	unknown
White fat	C Mouse	4 000	192 470	26	17 (17)	47 (47)
	T Human	1 937	137 306	24	16 (15)	44 (37)
Glioblastoma	C Mouse	4 000	46 321	6	14 (14)	23 (23)
	T Human	1 823	58 560	12	14 (14)	24 (22)

Table 2. The datasets employed for evaluating scSpecies use mice as context species C. The number H of homologous genes of context and target dataset are listed in the third column. Furthermore, all datasets are annotated with cell type labels, both at coarse and fine levels. The amount of distinct labels are detailed in the 'Number of cell labels' columns. Additionally, the amount of shared cell labels with the context dataset, are indicated in parentheses.

Our model underwent testing on publicly available datasets. (Table 2)

The 'Liver Cell Atlas'[17][39] contains a diverse collection of liver cells from multiple species, including mice (both with and without non-alcoholic fatty liver disease), humans, pigs, monkeys, chickens, and hamsters. We utilized all cells acquired through the scRNA-seq and CITE-seq pipelines.

The 'Single-Cell Atlas of Human and Mouse White Adipose Tissue'[18][40] contains gene expression data from human and murine white fat cells. We selected cell samples obtained via single-nucleus sequencing.

The 'Brain Immune Atlas' profiles immune response to a grade IV glioma. For humans we selected cells obtained via scRNA-seq of newly diagnosed and recurrent glioblastoma. For mice we selected cells from the immune response to transplanted glioblastoma[19][41].

We applied a uniform pre-processing pipeline across all datasets. Initially, the dimension of gene expression vectors was reduced to 4000 most highly variable genes[42]. Then we excluded cells with less than 2% nonzero genes or belonging to extremely scarce batch and cell labels with less than 20 samples. To obtain a consistent nomenclature between the datasets some cell labels were renamed. In the liver and glioblastoma datasets, some cells have inconsistent cell type labels. For example, some human liver cells are labeled as neutrophils in the fine and monocytes in the coarse cell label category. We excluded all cells with such a labeling conflict.

6. Extended Data

Model	**scArches**		**scPoli**		**kNN classifier**					
Neighbors	-		-		$k = 1$		$k = 25$		$k = 250$	
Cell labels	coarse	fine	coarse	fine	coarse	fine	coarse	fine	coarse	fine
Balanced label transfer accuracy score in % (BAS)										
Liver - human	64.05	48.05	80.74	55.72	80.72	59.46	79.70	62.04	75.16	57.25
Liver - mouse	97.62	78.50	**98.67**	**81.44**	97.69	79.40	98.03	80.03	97.45	76.08
White fat	65.79	37.50	65.45	37.41	74.37	40.20	73.80	41.17	67.64	37.86
Glioblastoma	51.96	46.60	80.92	59.94	75.59	54.47	76.37	56.65	71.70	54.51
Adjusted Rand index (ARI)										
Liver - human	0.725	0.248	0.841	0.263	0.740	0.194	0.824	0.253	0.859	0.290
Liver - mouse	0.983	0.837	0.984	0.825	0.983	0.822	0.985	0.839	0.982	**0.844**
White fat	0.773	0.414	0.846	0.443	0.868	0.371	0.884	0.438	0.877	0.469
Glioblastoma	0.458	0.401	0.583	0.581	0.481	0.384	0.537	0.455	0.525	0.470
Adjusted mutual information (AMI)										
Liver - human	0.685	0.516	0.794	0.538	0.711	0.487	0.781	0.554	0.809	0.575
Liver - mouse	0.976	0.871	**0.983**	0.869	0.977	0.860	0.981	0.875	0.977	0.870
White fat	0.768	0.607	0.831	0.657	0.839	0.599	0.861	0.654	0.848	0.659
Glioblastoma	0.576	0.500	0.656	0.598	0.610	0.507	0.679	0.568	0.672	0.568
scSpecies	lat. alignment		intermediate alignment							
Neighbors	$k = 25$		$k = 0$		$k = 1$		$k = 25$		$k = 250$	
Cell labels	coarse	fine	coarse	fine	coarse	fine	coarse	fine	coarse	fine
Balanced label transfer accuracy score in % (BAS)										
Liver - human	90.35	71.12	5.01	2.81	86.35	66.74	**92.08**	**73.29**	91.54	71.62
Liver - small	86.57	65.67	7.91	4.52	79.45	59.59	**87.76**	**67.78**	81.19	62.66

Model	scArches		scPoli		kNN classifier					
Liver - mouse	97.56	80.40	5.36	1.83	97.99	81.06	98.11	81.24	97.82	79.51
White fat	79.31	48.81	5.79	2.27	78.14	47.02	82.02	**49.15**	**83.17**	48.42
Glioblastoma	88.41	67.54	9.61	6.26	84.69	63.87	**88.88**	**68.87**	84.07	64.90
Adjusted Rand index (ARI)										
Liver - human	0.865	0.456	0.204	0.163	0.872	0.406	**0.888**	0.509	0.887	**0.593**
Liver - small	0.841	0.451	0.237	0.181	0.747	0.275	**0.863**	0.481	0.849	**0.545**
Liver - mouse	0.975	0.832	0.182	0.192	0.985	0.834	**0.987**	0.837	0.984	0.834
White fat	0.944	0.519	0.142	0.137	0.880	0.487	0.959	0.528	**0.963**	**0.540**
Glioblastoma	0.717	0.648	0.144	0.216	0.633	0.551	**0.753**	**0.684**	0.734	0.666
Adjusted mutual information (AMI)										
Liver - human	0.824	0.703	0.351	0.408	0.827	0.673	0.855	0.731	**0.864**	**0.760**
Liver - small	0.805	0.676	0.334	0.354	0.697	0.540	0.825	0.696	**0.830**	**0.727**
Liver - mouse	0.971	0.870	0.380	0.455	0.980	0.875	0.981	**0.878**	0.978	0.876
White fat	0.912	0.711	0.268	0.352	0.867	0.690	0.929	0.725	**0.934**	**0.734**
Glioblastoma	0.782	0.698	0.246	0.401	0.745	0.628	**0.799**	**0.683**	0.783	0.675

Table 3. Comparison of model performance on four different datasets. The results are averaged over five random seeds and the best results highlighted by bold font. The results for each dataset are listed for the coarse - fine cell label categories. The upper table contains the results obtained by scArches and scPoli. The kNN columns refer to the results of a data-level k nearest neighbor classifier trained on shared homologous genes. The results from scSpecies are listed in the bottom table. The first column corresponds to the results of a scSpecies model where latent representations instead of the intermediate representations are aligned. The column with zero neighbors corresponds to completely omitting the nearest neighbor integration within the model. The column with one neighbor corresponds to omitting learning a suitable neighbor candidate, as the choice is fixed.

Figure 4. Alignment performance of the architecture surgery-based approaches scArches and scPoli. The four left-hand plots were generated by aligning two mouse liver cell datasets. One dataset contains cell samples from healthy organisms, while the other contains cells from mice with non-alcoholic fatty liver disease. Despite the difference in disease conditions the latent representations are well aligned. The four plots on the right side were obtained by aligning human liver cells with those of healthy mice. Here, both approaches encounter difficulties with cross-species alignment.

Figure 5. Intermediate spaces of a scVI model applied to the mouse liver context dataset. It details the layer transformations from data space to latent space. Subplot 1 represents the UMAP coordinates of the original dataset, while subplot 8 shows the variational mean vectors in the latent space. Subplots 2–7 depict the UMAP coordinates of the intermediate dataset representation obtained by applying the corresponding layer transformation. Each subplot presents two scatter plots: the upper one showing clusters based on cell labels and the lower one depicting experimental batches. Additionally, the Davies-Bouldin index is used to assess the clustering quality for each subplot.

Figure 6. Intermediate spaces of a scVI model applied to the unaligned human liver target dataset. For an explanation of the subplots, see Figure 5.

Figure 7. Intermediate spaces of a scSpecies model applied to the mouse-human liver dataset pair. Each subplot presents two scatter plots: the upper one showing context cell label clusters and the lower one depicting the human target cell clusters. Additionally, the Davies-Bouldin index is used to asses clustering quality for each subplot. Alignment of the two datasets is encouraged in subplot 4.

Figure 8. A comparative analysis of gene expression profiles between humans and mice using scSpecies. We computed the median of the empirical log2 fold change distribution, displayed along the x-axis. The y-axis illustrates the likelihood of a gene being differentially expressed in mice versus humans with an LFC exceeding one. The compared cells are decoded from a randomly selected latent value within a latent cell type distribution. The figure highlights the top seven genes in mice that are significantly up-regulated (indicated in red) and the top seven that are notably down-regulated (blue) in comparison to their human equivalents.

Figure 9. Plots of human and mouse gene LRP scores against each other. Each dot represents a homologous gene. For every cell, Spearman's ρ and Person's R between human and mice LRP values are given in the axis label. Coloring corresponds to combined products of human and mice gene expression, with values of 0 are colored in dark tones and high values in bright colors.

Figure 10. Illustration of the alignment process of scSpecies with $k = 25$ neighbors. On the y-axis, we plot the negative log-density values derived from reconstructing human liver cell prototypes using their candidate set of mouse latent variables. The x-axis shows a log-scale trajectory of these values, averaged over the last $\lceil \min(10, 0.05 \times \text{steps}) \rceil$ iterations.

Statements and Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

The datasets can be accessed via the URLs[39][40][41]. Our model is implemented in Python 3.11.5 with PyTorch 2.1. The preprocessing scripts to obtain the datasets and the code to reproduce our results can be accessed at https://github.com/cschaech/scSpecies. We recommend to use a device equipped with an NVIDIA GPU.

Competing interests

The authors declare that they have no competing interests.

Funding

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 499552394 – SFB 1597 Small Data.

Authors' contributions

H.B. conceived and coordinated the project. H.B., C.S., and M.T. jointly developed the approach for aligning network architectures across species. C.S. implemented the corresponding code. H.R. and J.B. designed the methodology for extending the analysis from the latent space to the data level, with H.R. handling the implementation. C.S., H.B., M.H., and H.R. contributed to the writing of the manuscript. All authors reviewed and approved the final version of the manuscript.

Footnotes

[1] Lower values indicate higher association.

References

1. ^*Leonelli S, Ankeny RA. (2013). "What makes a model organism?" Endeavour. 37 (4): 209–212. doi:10.1016/j.endeavour.2013.06.001.*
2. ^*Cesar P. Canales, Katherina Walz. Chapter 6 - the mouse, a model organism for biomedical research. In: Katherina Walz, Juan I. Youngeditors. Cellular and animal models in human genomics research.: Academic Press 2019. pp. 119–140. (Translational and applied genomics). doi:10.1016/B978-0-12-816573-7.00006-7. ISBN 978-0-12-816573-7*

3. ^ McMurray F, Moir L, Cox RD. (2012). "From mice to humans". Current Diabetes Reports. 12. doi:10.1007/s11892-012-0323-2.
4. ^ Haddad AF, Young JS, Amara D, Berger MS, Raleigh DR, et al. (2021). "Mouse models of glioblastoma for the evaluation of novel therapeutic strategies". Neuro-Oncology Advances. 3 (1): vdab100. doi:10.1093/noajnl/vdab100.
5. ^ Lau JK, Zhang X, Yu J. (2017). "Animal models of non-alcoholic fatty liver disease: Current perspectives and recent advances". J Pathol. 241 (1): 36–44.
6. ^ Stripecke R, Münz C, Schuringa JJ, Bissig KD, Soper B, et al. (2020). "Innovations, challenges, and minimal information for standardization of humanized mice". EMBO Mol Med. 12 (7): e8662.
7. ^ Cao ZJ, Wei L, Lu S, Yang DC, Gao G. (2020). "Searching large-scale scRNA-seq databases via unbiased cell embedding with cell BLAST". Nature Communications. 11 (1). doi:10.1038/s41467-020-17281-7.
8. ^ Hu J, Li X, Hu G, Lyu Y, Susztak K, et al. (2020). "Iterative transfer learning with neural network for clustering and cell type classification in single-cell RNA-seq analysis". Nature Machine Intelligence. 2 (10): 607–618. doi:10.1038/s42256-020-00233-7.
9. ^ a, b Lotfollahi M, Naghipourfar M, Luecken MD, Khajavi M, Büttner M, et al. (2021). "Mapping single-cell data to reference atlases by transfer learning". Nature Biotechnology. :1–10.
10. ^ De Donno C, Hediyeh-Zadeh S, Moinfar AA, Wagenstetter M, Zappia L, et al. (2023). "Population-level integration of single-cell datasets enables multi-scale analysis across samples". Nature Methods. 20 (11): 1683–1692. doi:10.1038/s41592-023-02035-2.
11. ^ Lotfollahi M, Rybakov S, Hrovatin K, Hediyeh-zadeh S, Talavera-López C, et al. (2023). "Biologically informed deep learning to query gene programs in single-cell atlases". Nature Cell Biology. 25 (2): 337–350. doi:10.1038/s41556-022-01072-x.
12. ^ Michielsen L, Lotfollahi M, Strobl D, Sikkema L, Reinders MJT, et al. (2023). "Single-cell reference mapping to construct and extend cell-type hierarchies". NAR Genomics and Bioinformatics. 5 (3): lqad070. doi:10.1093/nargab/lqad070.
13. ^ a, b Breschi A, Gingeras TR, Guigó R. (2017). "Comparative transcriptomics in human and mouse". Nature Reviews Genetics. 18 (7): 425–440. doi:10.1038/nrg.2017.19.
14. ^ Rosen Y, Brbić M, Roohani Y, Swanson K, Li Z, et al. (2024). "Toward universal cell embeddings: Integrating single-cell RNA-seq datasets across species with SATURN". Nature Methods. 21 (8): 1492–1500. doi:10.1038/s41592-024-02191-z.

15. ^Biharie K, Michielsen L, Reinders MJT, Mahfouz A. (2023). "Cell type matching across species using protein embeddings and transfer learning". Bioinformatics. 39 (Supplement_1): i404–i412. doi:10.1093/bioinformatics/btad248.

16. a, bSohn K, Yan X, Lee H. (2015). "Learning structured output representation using deep conditional generative models". In: Proceedings of the 28th international conference on neural information processing systems - volume 2.: Cambridge, MA, USA: MIT Press pp. 3483–3491. (NIPS'15).

17. a, bGuilliams M, Bonnardel J, Haest B, Vanderborght B, Wagner C, et al. (2022). "Spatial proteogenomics reveals distinct and evolutionarily conserved hepatic macrophage niches". Cell. 185 (2): 379–396.

18. a, bEmont MP, Jacobs C, Essene AL, Pant D, Tenen D, et al. (2022). "A single-cell atlas of human and mouse white adipose tissue". Nature. 603 (7903): 926–933. doi:10.1038/s41586-022-04518-2.

19. a, bPombo Antunes AR, Scheyltjens I, Lodi F, Messiaen J, Antoranz A, et al. (2021). "Single-cell profiling of myeloid cells in glioblastoma across species and disease stage reveals macrophage competition and specialization". Nature Neuroscience. 24 (4): 595–610. doi:10.1038/s41593-020-00789-y.

20. a, bLopez R, Regier J, Cole M, Jordan MI, Yosef N. (2018). "Deep generative modeling for single-cell transcriptomics". Nature methods. 15: 1053–1058. Available from: https://api.semanticscholar.org/CorpusID:53643161

21. ^Fernando B, Fromont E, Tuytelaars T. (2014). "Mining mid-level features for image classification". International Journal of Computer Vision. 108 (3): 186–203. doi:10.1007/s11263-014-0700-1.

22. ^Boureau Y-L, Bach F, LeCun Y, Ponce J. (2010). "Learning mid-level features for recognition". In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 2559–2566. doi:10.1109/CVPR.2010.5539963.

23. ^Yosinski J, Clune J, Bengio Y, Lipson H. (2014). "How transferable are features in deep neural networks?" In: Proceedings of the 27th international conference on neural information processing systems - volume 2.: Cambridge, MA, USA: MIT Press pp. 3320–3328. (NIPS'14).

24. a, bMcInnes L, Healy J, Melville J. UMAP: Uniform manifold approximation and projection for dimension reduction. 2020. Available from: https://arxiv.org/abs/1802.03426

25. ^Jiang C, Li P, Ruan X, Ma Y, Kawai K, et al. (2020). "Comparative transcriptomics analyses in livers of mice, humans, and humanized mice define human-specific gene networks". Cells. 9 (12): 2566.

26. a, b, c, dBach S, Binder A, Montavon G, Klauschen F, Müller K-R, et al. (2015). "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". PLOS ONE. :46.

27. ^Keyl P, Bischoff P, Dernbach G, Bockmayr M, Fritz R, et al. (2023). "Single-cell gene regulatory network prediction by explainable AI". Nucleic Acids Res. 51 (4): e20.

28. ^Ma X-Y, Wang J-H, Wang J-L, Ma CX, Wang X-C, et al. (2015). "Malat1 as an evolutionarily conserved lncRNA, plays a positive role in regulating proliferation and maintaining undifferentiated status of early-stage hematopoietic cells". BMC Genomics. 16 (1): 676.

29. ^Kingma DP, Welling M. Auto-encoding variational bayes. 2022. Available from: https://arxiv.org/abs/1312.6114

30. ^Skinnider MA, Squair JW, Foster LJ. (2019). "Evaluating measures of association for single-cell transcriptomics". Nature methods. 16 (5): 381—386. doi:10.1038/s41592-019-0372-4.

31. ^Boyeau P, Lopez R, Regier J, Gayoso A, Jordan MI, Yosef N (2019). "Deep generative models for detecting differential expression in single cells". bioRxiv. doi:10.1101/794289.

32. a, bBoyeau P, Regier J, Gayoso A, Jordan MI, Lopez R, Yosef N (2023). "An empirical bayes method for differential expression analysis of single cells with deep generative models". Proceedings of the National Academy of Sciences. 120 (21): e2209124120. doi:10.1073/pnas.2209124120.

33. ^Montavon G, Binder A, Lapuschkin S, Samek W, Müller KR. "Layer-wise relevance propagation: An overview". In: Samek W, Montavon G, Vedaldi A, Hansen LK, Müller KR, editors. Explainable AI: Interpreting, explaining and visualizing deep learning. Cham: Springer International Publishing; 2019. pp. 193–209. doi:10.1007/978-3-030-28954-6_10.

34. ^Montavon G, Bach S, Binder A, Samek W, Müller KR (2017). "Explaining NonLinear classification decisions with deep taylor decomposition". Pattern Recognition. 65: 211–222. doi:10.1016/j.patcog.2016.11.008.

35. a, bNguyen Xuan Vinh, Julien Epps, James Bailey. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. J Mach Learn Res. 11:2837–2854.

36. ^Davies DL, Bouldin DW (1979). "A cluster separation measure". IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-1 (2): 224–227. doi:10.1109/TPAMI.1979.4766909.

37. ^Userguide to skikit learn. 2024. Available from: https://scikit-learn.org/stable/modules/model_evaluation.html.

38. ^Kingma DP, Ba J. Adam: A method for stochastic optimization. 2017. Available from: https://arxiv.org/abs/1412.6980.

39. a, bBrain immune atlas. 2022. Available from: https://www.livercellatlas.org/.

40. a, bSingle-cell atlas of human and mouse white adipose tissue. Available from: https://singlecell.broadinstitute.org/single_cell/study/SCP1376.

41. a, bBrain immune atlas. 2021. Available from: https://www.brainimmuneatlas.org/.

42. ^ *Satija R, Farrell JA, Gennert D, Schier AF, Regev A (2015). "Spatial reconstruction of single-cell gene express ion data". Nature Biotechnology. 33 (5): 495–502. doi:10.1038/nbt.3192.*

Declarations

Funding: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 499552394 – SFB 1597 Small Data.

Potential competing interests: No potential competing interests to declare.