Enhancing interpretability of VAEs using biologically informed decoder for single-cell transcriptome analysis

Masters Thesis

Presented to the Faculty of Biosciences
of the Ruprecht-Karls-Universität Heidelberg

Qian-Wu Liao

2022

This Thesis was written at **Health Data Science Unit (HDSU) of Medical Faculty Heidelberg** at the Ruprecht-Karls-Universität Heidelberg in the period from **11.11.2021** to **25.05.2022** under the supervision of **PD Dr. Carl Herrmann** and **Daria Doncevic**.

1th Examiner: **Prof. Dr. Ursula Kummer**
Institute: **Centre for Organismal Studies (COS) Heidelberg**

2th Examiner: **Prof. Dr. Julio Saez-Rodriguez**
Institute: **Institute for Computational Biomedicine of Universitätsklinikum Heidelberg**

I herewith declare that I wrote this Masters Thesis independently, under supervision, and that I used no other sources and aids than those indicated throughout the thesis.

Date _____ Signature _____

Abstract

Single-cell RNA sequencing (scRNA-seq) has revolutionized the analysis of transcriptomes for essentially any cell types and variational autoencoders (VAEs) have been proved to be a powerful tool for analyzing noisy single-cell transcriptomic data. However, typical VAE models provide little to no interpretability in their internal networks which are valuable for understanding biological mechanisms. To this end, this work aims at investigating the ability and various properties of two kinds of interpretable VAEs that either use the hard-coded decoder or use the regularized decoder where prior gene set annotations can be incorporated to guide the decoder wiring, which ends up modeling latent variables as the activity of distinctive gene modules. The interpretable latent space can then be used for many downstream studies such as cell type identities, cell states and differential activity analysis of different cell populations. We performed the benchmark studies using the published human adrenal medulla dataset to demonstrate the ability of these two types of interpretable models and indicate that they can be a complementary method to each other due to their unique attributes. Moreover, we also investigated some properties of these interpretable VAEs, such as the model reproducibility and the tolerance of the models to incorrect prior information, to provide more knowledge for further development and improvements.

Acknowledgements

Firstly, I would like to express my deepest appreciation to my supervisors, Dr. Carl Herrmann and Daria Doncevic. Thank you for patiently imparting your knowledge to me and proactively guiding me through the journey of my thesis work. I enjoyed the time working with you very much. Secondly, I want to thank Prof. Dr. Ursula Kummer and Prof. Dr. Julio Saez-Rodriguez for being my thesis examiners. Thanks also to the people in Health Data Science Unit, Dr. Andres Quintero, Dr. Carlos Ramirez, Ana Luisa Costa, Youcheng Zhang and Lin Yang for any help and happy-happy moments during my thesis time. Thirdly, I want to thank Prof. Dr. Ju-Chien Cheng (鄭如茜教授) and Dr. Jimmy Kuo (郭傑民博士) for generously helping and guiding me towards my goal back in 2018 that was such a turning point of my life. Last but not least, I would like to thank my lovely family for always believing in and supporting me. Enriched by your energy, I will keep moving forward with faith in my life. (謝謝我最親愛的家人總是相信與支持我做的任何決定，在未來的路上我會帶著你們的能量與信念繼續朝我的目標前進。)

Contents

List of abbreviations

AE	Autoencoder
AUC	Area under the curve
BF	Bayes factor
ChIP-seq	Chromatin immunoprecipitation sequencing
GMV	Gene module variable
GRN	Gene regulatory network
IQR	Interquartile range
KLD	Kullback-Leibler divergence
PBMCs	Peripheral blood mononuclear cells
PCC	Pearson correlation coefficient
scRNA-seq	Single-cell RNA sequencing
SCENIC	Single-cell regulatory network inference and clustering
SCPs	Schwann cell precursors
siRNA	Small interfering RNA
TF	Transcription factor
UMAP	Uniform Manifold Approximation and Projection
VAE	Variational autoencoder
VEGA	VAE enhanced by gene annotations

Chapter 1

Introduction

1.1 Single-cell RNA sequencing

Recent advances in single-cell RNA sequencing (scRNA-seq) technologies[1] have boosted many biological and clinical discoveries at an unprecedented resolution. Previously, sequencing technologies, such as bulk RNA-seq[2], can only provide a gene expression profile of an entire sample containing thousands to millions of cells, revealing an overall state and biological traits of a whole organ or tissue. However, knowledge from a population of cells cannot represent that from an individual cell since biological signals of a minority of cells within a cell pool or slight differences in two cells of the same cell type may fail to be effectively detected[3], for example, malignant tumor cells within a tumor mass[4] and individual T cells with highly diverse T cell receptors[5]. To address this issue, scRNA-seq can be used to disentangle single cells from pooled cells, providing cellular heterogeneity that is valuable for many research focuses such as cellular traits, cellular responses to different scenarios and so on. In addition, data generated from scRNA-seq is also beneficial to gene regulatory networks (GRN) inferences in a data-driven way, providing context-specific GRNs which enables us to decipher functional cellular heterogeneity[3]. Nevertheless, although scRNA-seq fulfills high-resolution transcriptomic analysis, it still has some challenges and limitations, such as drop-out events due to extremely low gene expression[6], batch effects between datasets from different measurements[7] and so on, which complicates scRNA-seq data analysis[8]. To extract meaningful biological information from high-dimensional noisy scRNA-seq data (i.e., dimensionality reduction),

the selected computational method is critical. There are a variety of developed or developing computational methods used for data dimensionality reduction, e.g., for linear algorithms, principal component analysis[9] (PCA) and non-negative matrix factorization[10] (NMF). Compared to linear algorithms, nonlinear algorithms such as deep learning-based models have been proved to be a more advanced technique to learn more complex patterns from noisy transcriptomic data. Specifically, we focused on autoencoder-based models in our work.

1.2 Autoencoder-based approaches

Autoencoders[11] (AEs) have emerged as potent tools to tackle many different biological tasks such as dimensionality reduction of data[12], cell clustering[13] and data denoising[14]. Compared to those methods that are limited by their linear nature (e.g., PCA), AEs are capable of learning more complex nonlinear patterns in single-cell transcriptomes. Generally, AEs consist of two parts: An encoder which can learn a nonlinear transformation projecting data from the high-dimensional input space to the lower-dimensional latent space (i.e., the representations of the input data) and a decoder which also learns a nonlinear transformation projecting the representations from the low-dimensional latent space back to the original high-dimensional input space[15] (Fig.1.1). To optimize the quality of the information that the representations hold, AEs are trained end-to-end by minimizing the difference between the input data and the reconstructed data. Through this way, we can also turn dimensionality reduction tasks which are usually handled by unsupervised learning methods into supervised learning problems, which can potentially improve the accuracy of representations of input data. However, as previously mentioned, since single-cell transcriptomic data is usually very noisy and complex, it will be problematic if we want to use AEs as generative models due to their discrete latent space, leading to the poor generalizability. To be more concrete, the representations of all data points in the input data are vectors which are disjoint and non-continuous[15], so nuances in input transcriptomic data may bring about unsatisfying reconstructed gene expression profiles. To this end, variational autoencoders[16] (VAEs) were developed. Instead of discrete latent vectors, VAEs map high-dimensional data to a probability distribution (e.g., a multivariate normal distribution) from which the lower-dimensional representations can be sampled (Fig.1.2), which makes the latent space more continuous and complete[15,17]. VAEs have been demonstrated to work well for the probabilistic modeling of transcriptomic data in previous single-

cell transcriptome studies such as scVI[18] and scGen[19]. Even though the latent space of VAEs is modeled as a fairly complete and informative distribution, which enables VAEs to have the generative capacity and make accurate predictions, yet, it provides little to no interpretability that is crucial for understanding biological mechanisms. To gain more interpretability of internal networks, prior biological knowledge can be integrated into network structures. DCell[20], a deep interpretable neural network, has successfully embedded the hierarchies of molecular subsystems about cellular processes in the network architecture, which models not only functional outcomes but also the mechanisms resulting in these outcomes. To this end, the question is how VAEs can incorporate prior knowledge with network architectures to improve the model interpretability.

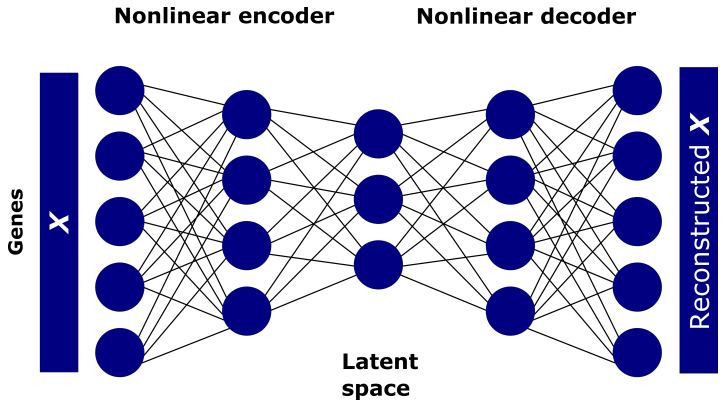

Figure 1.1: **Architecture of typical AE**

1.3 Interpretable variational autoencoders

Inspired by f-scLVM[21] that uses prior gene set annotations to guide factor analysis and VAEs which use a factor model as a decoder[22], a novel network architecture called VEGA (VAE enhanced by gene annotations) was proposed by Seninge et al. (2021), which consists of a two-layer nonlinear encoder and a single-layer masked linear decoder (Fig.1.3). Owing to the single-layer decoder where the latent variables (the representation of a single-cell transcriptome) directly connect to the output variables (the gene features), predefined gene set annotations either taken from databases (e.g., Reactome[23] and MSigDB[24]) or inferred using computational methods (e.g., SCENIC[25] and ARACNE[26]) can be used to guide the decoder wiring through a binary mask, which models the la-

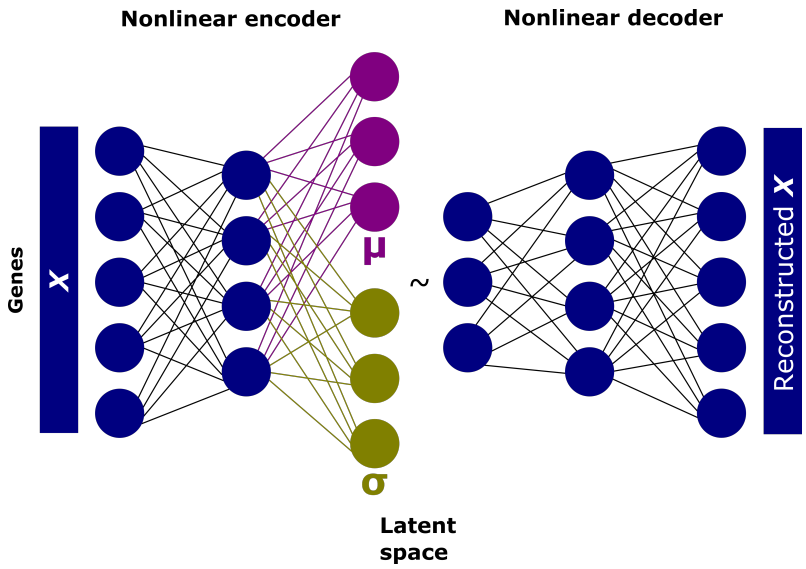

Figure 1.2: **Architecture of typical VAE**

tent variables as biologically meaningful gene modules referred to as gene module variables (GMVs, see Methods 4.1). To be more concrete, the values of gene expression of each cell are input to the encoder and encoded to the certain number of latent variables depending on the number of predefined gene modules provided. With the single-layer decoder, the connections between each latent variable and the genes in the output layer can be initiated by the prior gene set annotations, forcing the latent variables to be meaningful and interpretable by limiting them to certain corresponding sets of genes. Since any biological processes can be related to genes, it makes VEGA extremely flexible in terms of the specification of the connectivity of GMVs, which enables us to interpret single-cell transcriptomes from many other different viewpoints. Furthermore, since the decoder was designed as a linear factor model, the weights of a certain GMV to the gene reconstructions can be directly interpreted as the relationships between the GMV and the genes. Last, because different single-cell sequencing measurements always have technical noise and bias, VEGA can also incorporate batch information into the encoder and the latent space through one-hot encoding to combat batch effects, which was introduced in Lopez et al. (2018). Nevertheless, VEGA's hard-coded linear decoder leaves no room for further correcting or expanding existing prior knowledge which is usually incomplete or non-context-specific[27]. To have the decoder use prior knowledge more flexibly, instead of hard coding the decoder wiring, the L1 regularization technique[28] can be

employed on weights in the decoder to penalize those unannotated connections, which was introduced in Rybakov et al.(2020). Oppositely to the description above, predefined gene modules are used to pinpoint those GMV-gene relationships which are not included in the prior and the weights of those decoder connections will be penalized by gradually shrinking them to zero (see Methods 4.2). Using the regularized decoder enables the VAE to not only form the interpretable latent space but also potentially recover missing relationships between GMVs and genes in a data-driven fashion.

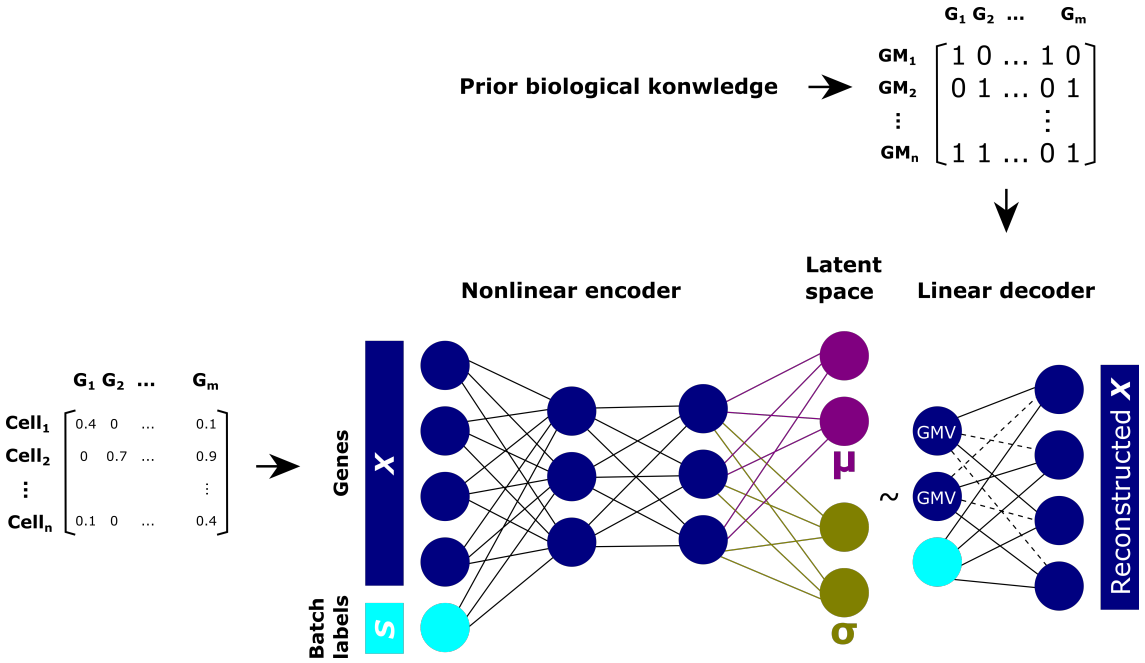

Figure 1.3: **Architecture of interpretable VAE** — The encoder is a two-layer nonlinear neural network which encodes high-dimensional input data to lower-dimensional representations and the representations are then modeled as a multivariate normal distributions in the latent space. The decoder is a single-layer factor model which attempts to reconstruct the representations sampled from the latent normal distribution to the original high-dimensional data. The connections between the latent variables (a representation of input data) and the output variables (input features) can be guided by prior knowledge through a binary mask. For scRNA-seq data analysis, the transcriptome of individual cells is the input to the model and predefined gene modules can be used to guide the decoder wiring. The solid lines in the decoder indicate there are relationships between GMVs and genes according to the prior used and the dashed lines indicate the opposite. There are two circumstances: (1) for the hard-coded decoder, the dashed connections always have zero-valued weights and (2) for the regularized decoder, the weights of the dashed connections are penalized through gradually shrinking them to zero. Moreover, batch information can also be incorporated into the encoder and the latent space through one-hot encoding to alleviate batch effects if needed. G above the matrices stands for gene and GM beside the binary matrix stands for gene module. GMVs represent gene module variables in the latent space.

1.4 Outline of our work

In this work, we aimed at investigating the ability and various properties of interpretable VAEs that use the hard-coded decoder[27] and use the regularized decoder[29]. Our work can be roughly split into three parts; firstly, to verify the availability of VEGA and better understand how exactly the model works, we reproduced the analyses of the well-studied PBMCs (peripheral blood mononuclear cells) dataset[30] in the VEGA paper[27] using the Reactome pathways[31] as prior knowledge. Moreover, we sought to improve the model reproducibility which is important to the VEGA's main characteristic, interpretability, so we explored potential ways to boost the model reproducibility using the same dataset and prior knowledge. Secondly, we performed the benchmark studies on VEGA using the published human adrenal medulla dataset[32] and the SCENIC regulons[32] inferred from the same dataset as the prior. Note that we took advantage of the inferred TF activities of each adrenal medulla cell type group described in Jansky et al. (2021) as references for our benchmarks. Furthermore, we also employed the non-context-specific DoRothEA regulons[33] as prior knowledge to observe the changes in the behavior of VEGA. Inspired by the results from the analyses of the adrenal medulla dataset[32] using the DoRothEA regulons as the prior, we investigated the tolerance of VEGA to incorrect information in prior knowledge by randomizing prior gene set annotations to different degrees. The last task we conducted in this section was testing the batch correction function[18] that is incorporated in VEGA. To do so, we took advantage of the human neuroblastoma dataset[32] which consists of 22 neuroblastoma samples and suffers from severe batch effects and the SCENIC regulons inferred from this tumor data as prior knowledge to see if VEGA is capable of screening out technical differences. In the last part of our work, we investigated the behavior and the ability of the interpretable VAE model using the regularized decoder by artificially modifying prior gene set annotations (i.e., artificial gene removal and addition) to see whether the regularized decoder manages to recover missing target genes or exclude non-biologically meaningful genes. Last, we studied the inference capacity of the regularized decoder by using the non-context-specific DoRothEA regulons as the prior, that is, we were interested in if the regularized decoder can infer more dataset-specific GRNs from general GRNs.

Chapter 2

Results

2.1 VEGA reproducibility

2.1.1 Reproducing results in VEGA paper

Firstly, to verify the ability and availability of VEGA [27], we reproduced the analyses of the published PBMCs (peripheral blood mononuclear cells) dataset [30] which consists of two groups of blood cells: Control cells and cells stimulated with interferon-β (see Methods 4.4). We wanted to see whether VEGA could recapitulate biological information in its interpretable latent space using the Reactome collection of pathways and processes [31] as prior knowledge to initiate the wiring of the decoder part of VEGA (see Methods 4.5). The hyperparameters used in this section are recorded in Appendix A. After the model training, we ran UMAP [34] on the VEGA embedding (see Methods 4.9). The results show that VEGA was capable of clustering cells into cell types and treatment conditions (Fig.2.1A;Appx.B.1). Moreover, when specifically looking into GMV activities, we observed that the GMV representing the interferon-α/β signaling pathway separated stimulated and control cells, which is consistent with our result and previous studies [35] (Fig.2.1B).

Next, we wanted to see if the differential GMV activities between two treatment conditions are statistically significant. For this, we performed differential activity analysis proposed by Seninge et al. (2021). Since each GMV is modeled into the posterior distribution, two mutually exclusive hypotheses can be formulated (i.e., a given GMV activity is higher in the stimulated group compared to the control group or the other way around) and the posterior probabilities of these hypotheses

can be approximated through Monte Carlo sampling from the GMV distributions. Finally, a Bayes factor[36,37] (BF), the ratio of the hypothesis probabilities, can be used to determine the significantly differential GMV activities between two treatment conditions (see Methods 4.3). The volcano plot indicates that the activities of interest (interferon-α/β signaling and interferon signaling pathways) were significantly more active in the stimulated cells compared to the control cells ($|\log_e(\mathrm{BF})| > 3$, Fig.2.1C). Together, these reproduced results confirm the interpretability of VEGA, enabling us to interpret gene expression data from the view of biological pathways and processes.

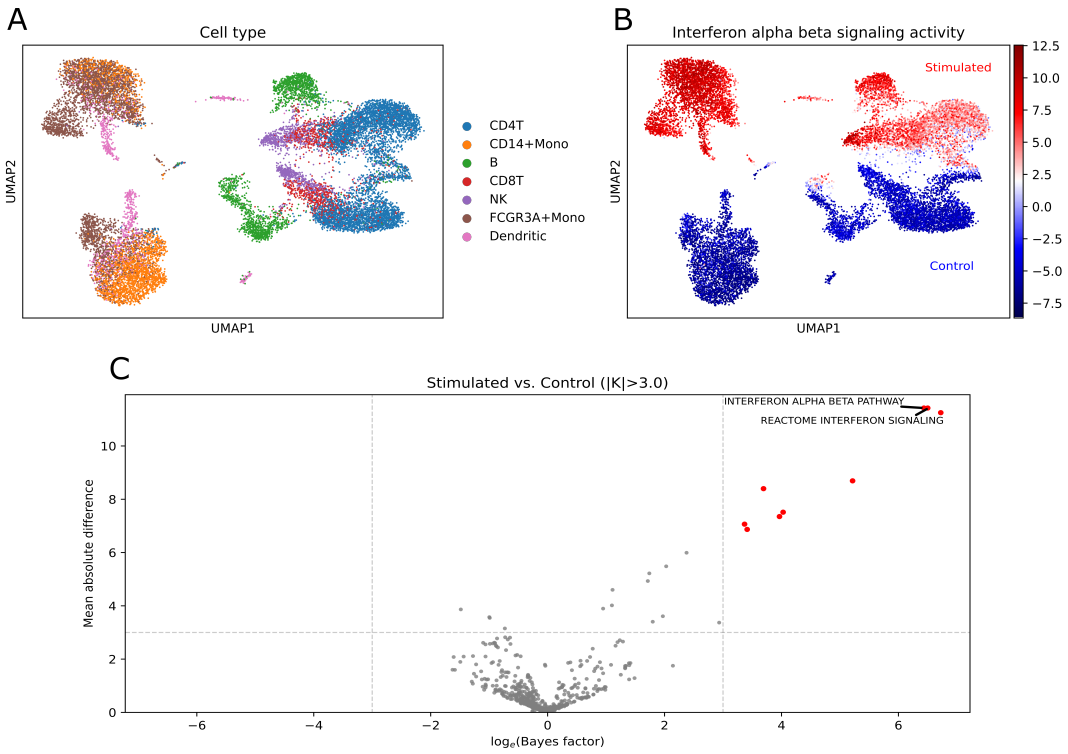

Figure 2.1: **Reproduction of results in VEGA paper** — We reproduced the PBMCs analysis using the Reactome pathways as the prior. **(A)** The UMAP plot shows the model could cluster cells into cell types. **(B)** Since the dataset consists of control cells and cells stimulated by interferon-β, the interferon-α/β signaling pathway was of interest. The UMAP plot colored by activity levels of the GMV representing the interferon-α/β signaling pathway shows that stimulated and control cells were nicely separated and the stimulated cell group had a high level of the activity. **(C)** The x-axis and the y-axis indicate the significance level and the mean absolute difference of activity comparisons between two cell types. The volcano plot shows that interferon-α/β signaling and interferon signaling pathways were significantly more active in stimulated cells. We considered pathways to be significantly differentially activated if $|\log_e(\mathrm{BF})| > 3$.

2.1.2 Using dropout layer in latent space stabilizes model reproducibility

At the beginning of this work, we performed benchmarks for VEGA in an attempt to check the interpretability of the model and understand how exactly the model works. Specifically, we were interested in the stability of VEGA training since we found that some of the GMV activities in the latent space could be unrelated or even anticorrelated between two individual trained models on the same dataset (see Methods 4.6). Take the PBMCs dataset [30] as an example using the same hyperparameters specified above (see Appendix A). We observed that the median of the correlations between two matching GMVs was 0.75 and some of them had negative correlations (Fig.2.2A). We wondered if there are hyperparameters in control of the reproducibility of the GMV activities and then first dissected this question by investigating two novel hyperparameters introduced in VEGA: A dropout layer and additional fully connected nodes in the latent space. Note that the model used as the control was with $z\_dropout = 0.5$ and $add\_nodes = 1$ and we tuned one of these two hyperparameters each time to see the changes in the reproducibility. We found that, without employing a dropout layer in the latent space ($z\_dropout = 0$), the median of the correlations dropped to 0.42 and the anticorrelated events became even worse (up to -0.8, Fig.2.2A). There was no notable difference in the model reproducibility using a dropout rate of 0.3 and 0.5. On the other hand, we observed that with or without using additional nodes in the latent space did not significantly affect the reproducibility, where the median of the correlations slightly decreased from 0.72 to around 0.65 (Fig.2.2A). Collectively, we suggest that the implementation of a dropout layer in the latent space have the effect of stabilizing the model reproducibility.

We next questioned whether the correlation coefficients of matching GMVs have any relationships with the numbers of genes in the output layer they connect to. We calculated the number of genes connected to each GMV and vice versa according to the Reactome prior knowledge and the considered highly variable genes (Appx.B.2). We display the results of using $z\_dropout = 0.5$ and 0 (both with $add\_nodes = 1$) for this task because we observed that employing a dropout layer in the latent space is a way to improve the model training stability. The results do not show any evident correlations between the reproducibility of GMVs and the numbers of genes which GMVs connect to (Fig.2.2B,C). However, interestingly, we observed that additional nodes in the latent space always had a high correlation coefficient (0.99) when there was only one additional node employed, but had varied correlation coefficients ranging from -0.76 to 0.99 when more than one additional nodes were used (Fig.2.2D). We infer that these additional nodes randomly shared learnable information

with each other in different individual trained models since they all fully connected to genes in the output layer, leading to the poor characterization. Therefore, when we computed correlations, two corresponding latent variables might hold different information, leading to a low correlation coefficient. To this end, we propose the possible further work could be the investigation of whether gene programs having highly similar gene sets in prior information used to initiate the decoder wiring blur the reproducibility and the interpretability of GMVs because VEGA fails to model certain latent variables to specific biologically meaningful gene programs.

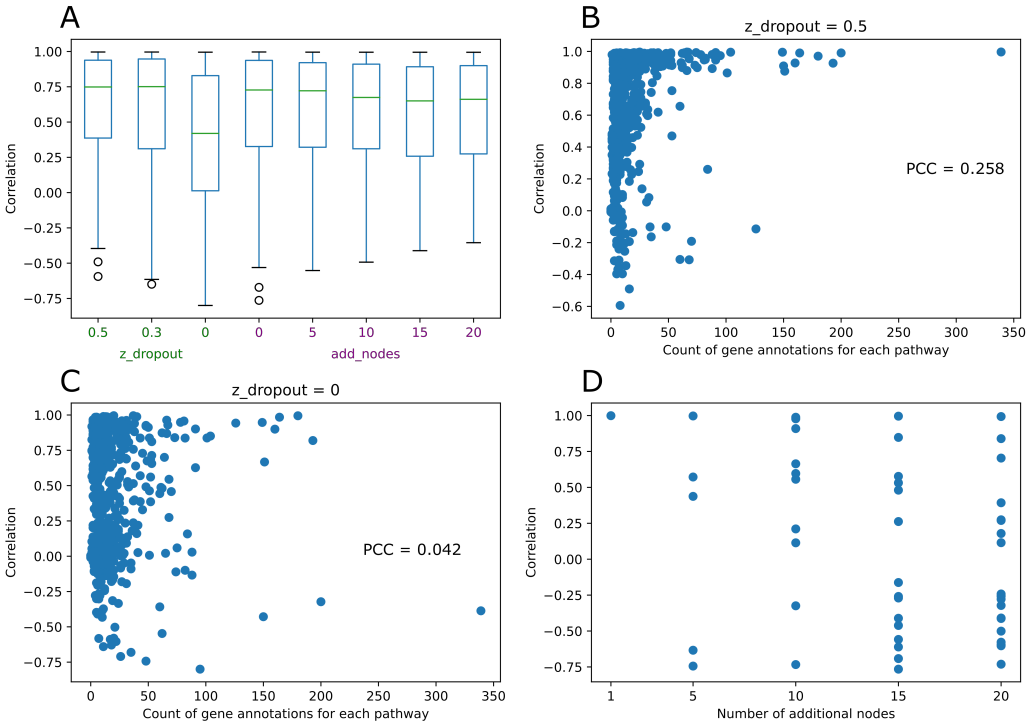

Figure 2.2: The caption is on the next page.

Figure 2.2: **Investigation of model training stability** — The method we used to measure the model training stability was computing Pearson correlation coefficients between two matching GMVs from two individual trained models using the same set of hyperparameters. We tuned two hyperparameters used in the latent space, one at a time: (1) *z_dropout* (the dropout rate) and *add_nodes* (the number of additional fully connected nodes) to investigate the changes in the training stability. Note that the correlations of the additional fully connected nodes were excluded from plot A, B and C. **(A)** The x-axis indicates the value of a hyperparameter and the y-axis indicates the correlation coefficient. The first boxplot ($z\_dropout = 0.5$ and $add\_nodes = 1$) was used as the control. The boxplots show that using a dropout layer in the latent space improved the stability of model training (green) and the number of additional nodes used in the latent space did not have influence on the training stability (purple). The box extends from the Q1 (25^{th} percentile) to the Q3 (75^{th} percentile) of data with the horizontal line inside denoting the median of data (Q2, 50^{th} percentile). The whiskers extend from the box show the range of data (Q0, 0^{th} percentile and Q4, 100^{th} percentile) if all data points are within 1.5 * IQR (IQR = Q3 - Q1). Otherwise, the data points beyond the range are displayed as separate circles. IQR stands for interquartile range. **(B,C)** The x-axis indicates the number of genes each GMV connects to. The scatterplots show there were no evident correlations between the reproducibility of GMVs and the numbers of genes which GMVs connect to. PCC stands for Pearson correlation coefficient. **(D)** The x-axis indicates the number of additional fully connected nodes used in the latent space. The plot shows the reproducibility of some additional nodes could be very unstable when the number of additional nodes used was more than 1. The reason may be that the additional nodes randomly shared the learned information due to the same decoder wiring.

2.2 Performing benchmarks for VEGA

2.2.1 TF activity analysis recapitulates adrenal medulla development

Secondly, we applied VEGA[27] to the published human adrenal medulla dataset[32] which consists of Schwann cell precursors (SCPs), chromaffin cells, neuroblasts and the other transient cells in the adrenal medulla from various stages of embryonic and fetal development (see Methods 4.4). The UMAP[34] embedding of the gene expression space of the adrenal medulla cells shows the nice cell clustering and developmental trajectories where SCPs gave rise to chromaffin cells and neuroblasts through bridge and connecting progenitor cells[32] (Fig.2.3A, see Methods 4.9). Moreover, the UMAP plot colored by samples reveals that the cell clustering was subject to biological differences between the cell types rather than technical differences between the samples (Appx.B.3A). As mentioned previously, owing to the flexibility of the GMV specification, we employed the SCENIC[25] regulons inferred from this human adrenal medulla dataset[32] as prior biological knowledge to guide the decoder wiring (see Methods 4.5). We wanted to see how capable VEGA is to capture TF activities at the single-cell level in its latent space, which can be used to conduct cell clustering and differential activity analysis. We first split the task into three parts where VEGA models were trained on (1) the dataset with the top 2000 highly variable genes, (2) the dataset with the whole gene features and

(3) the dataset with only those genes connected to the GMVs according to the SCENIC regulons. The hyperparameters used in this task are recorded in Appendix A. After the VEGA training, we ran UMAP on the VEGA embeddings. Among these three trained models, all of them could nicely cluster cells into cell types (Fig.2.3B;Appx.B.3B,C). Even so, the model trained on the dataset with the top 2000 highly variable genes best preserved the adrenal medulla development, which was thereby mainly used for the further work.

To investigate whether these GMVs truly represented certain SCENIC regulons, we performed differential activity analysis to statistically compare GMV activities between different cell types (see Methods 4.3). Note that the inferred TF activities of each adrenal medulla cell type group described in Jansky et al. (2021) were used as references for our benchmarks (Fig.2.3C). The volcano plots show that VEGA precisely modeled the latent variables as the corresponding TFs in the prior because nearly all of the GMV (TF) activities of interest, except HES6, were averagely active in correct cell type groups (Fig.2.3D) according to the references in Fig.2.3C. For example, GATA3 and TFAP2B were found to have a high level of activities in neuroblasts and many members of the JUN/FOS TF family are associated with the chromaffin differentiation [32]. In the light of the ability of VEGA, those TFs which had significantly differential activities between different cell types but were not captured in Jansky et al. (2021), such as SREBF2 and E2F1 in neuroblasts compared to SCPs (Appx.B.3D), are extremely worth further investigation. However, we did not find any supporting biological evidence from previous studies for them in the context of healthy cells. Therefore, further work on verifying whether those significantly differential TF activities are biologically meaningful or just false positives is needed. Last but not least, the one additional fully connected node we used in the latent space was significantly active in neuroblasts and in chromaffin cells compared to SCPs, which indicates employing additional nodes helps capture extra information that is not explained in the prior (UNANNOTATED_0 in Appx.B.3D).

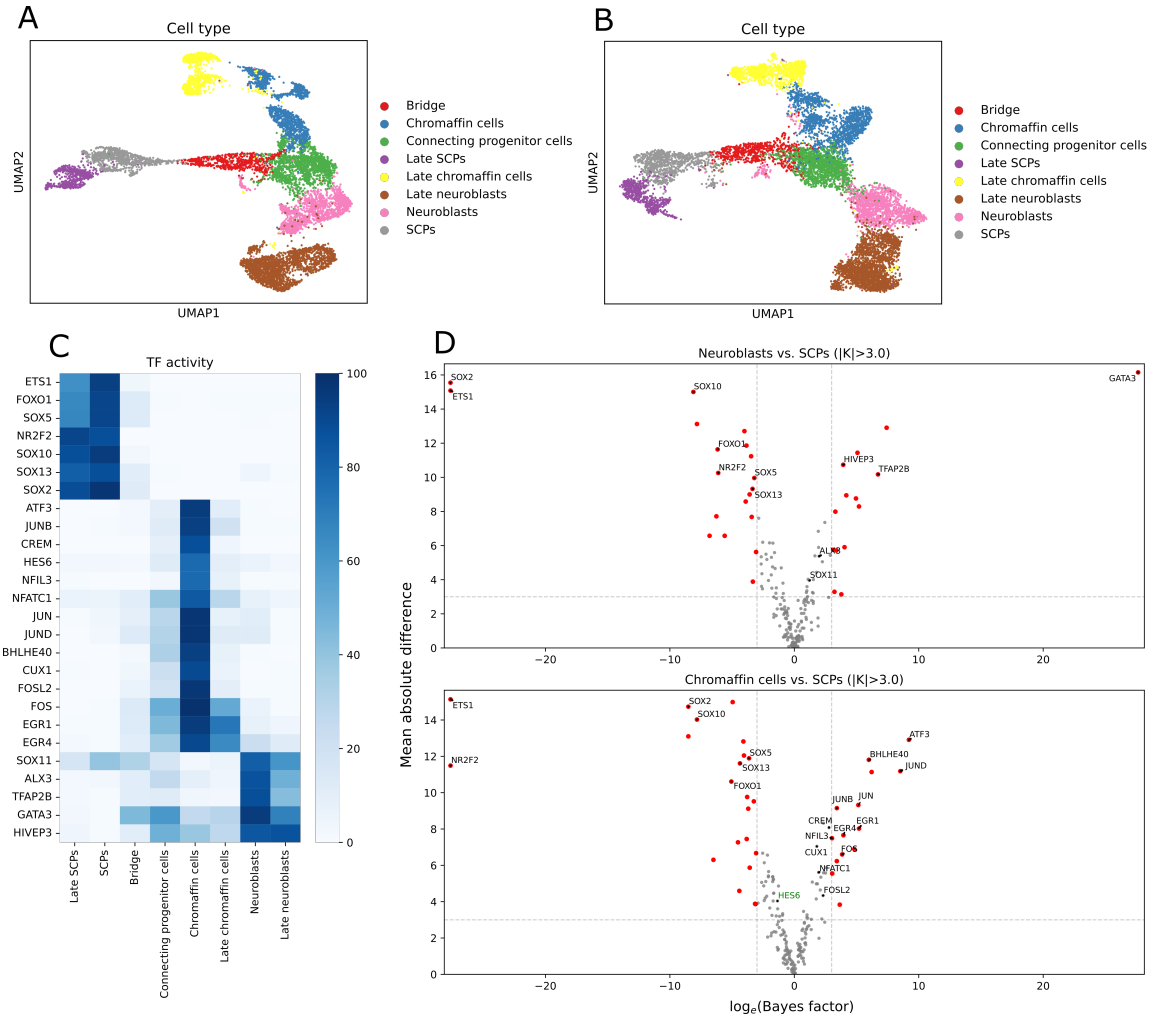

Figure 2.3: **Benchmarks for VEGA** — We used the SCENIC regulons inferred from the human adrenal medulla dataset as prior knowledge to analyze TF activities of human adrenal medullary cells. **(A)** The UMAP embedding of the gene expression space shows the clear cell clustering and the developmental trajectories of the adrenal medulla. **(B)** The UMAP embedding of the VEGA latent space shows that the model could cluster cells into cell types and preserve the developmental trajectories. Note that the model were trained on the dataset with the top 2000 highly variable genes and we removed 23 SCENIC regulons from the prior, whose target genes did not overlap with any considered gene features, i.e., there would be no connections between these 23 GMVs and the genes in the output layer. **(C)** The inferred TF activities of each cell type group from Jansky et al. (2021) were used as references for our TF differential activity analysis. Firstly, the TF activities in individual cells were inferred using SCENIC and for each TF, the activities across cells were discretized into two levels (active or inactive) using k-means clustering. Finally, the TF activities of each cell type group were determined by computing the fraction of cells whose TF states were active in each cell type. **(D)** The x-axis and the y-axis indicate the significance level and the mean absolute difference of activity comparisons between two cell types. The volcano plots show that nearly all of the TF activities of interest, except HES6 (colored in green), were averagely active in correct cell type groups. We considered TFs to be significantly differentially activated if $|\log_e(\mathrm{BF})| > 3$.

2.2.2 Hard-coded decoder leaves little freedom for further inferences

Next, instead of using the dataset-specific prior to initiate the decoder wiring, we employed the DoRothEA [33] regulons which is not context-specific to investigate the behavior of VEGA (see Methods 4.5). Of note, the hyperparameters used in this task are the same as Section 2.2.1, which is recorded in Appendix A. After the model training, we ran UMAP on the VEGA embedding. The result shows that VEGA was still able to cluster cells into cell types and also exhibit the developmental trajectories of the human adrenal medulla (Fig.2.4A). However, when we looked into the differential activity analysis results, we observed that VEGA using the general prior knowledge could not faithfully reflect the correct activations of TF activities in certain cell types anymore (Fig.2.4B). The differential activity analysis results took Fig.2.3C as references. Although nearly all of the significantly differential TF activities, except CUX1, were correctly presented in the corresponding cell types, a majority of TF activities could not be captured or were even falsely predicted by VEGA (e.g., HIVEP3, JUNB, JUND and so on). We then checked the overlaps of the target genes between the DoRothEA and the SCENIC regulons and found that all target gene sets hardly overlap with each other (Table 2.1). Collectively, these results indicate that VEGA has limited freedom to further make inferences beyond the prior and fails to provide the interpretability in the latent space when prior knowledge used is not context-specific. Given that we still obtained such the clear clustering of the adrenal medullary cells, we reasoned that the compressed information that the VEGA latent variables hold can be extracted from the expression data with any decoder settings, but just the learned information will be randomly distributed in the latent space if the decoder wiring is biologically meaningless, which means the model loses its interpretability. This hypothesis will be corroborated in the next section.

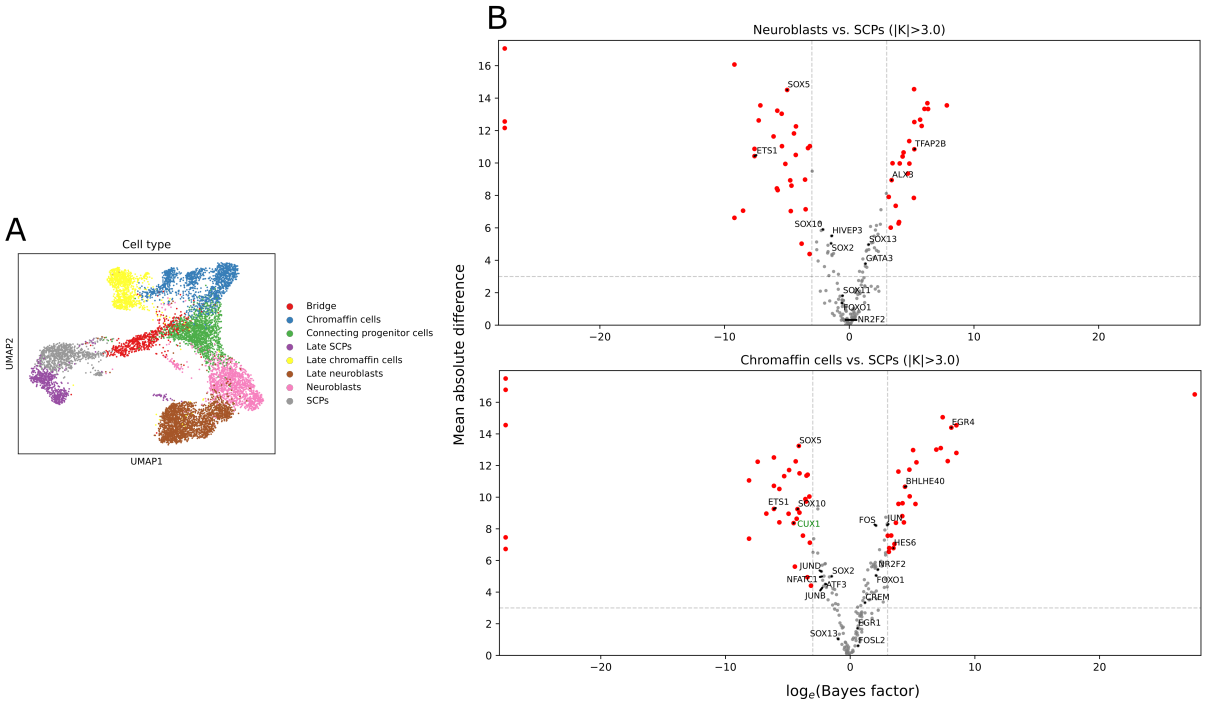

Figure 2.4: **Observations of VEGA using general regulons as prior knowledge** — We used the DoRothEA regulons which is not context-specific as the prior to investigate the changes in the model interpretability. **(A)** The UMAP plot shows the model could still cluster cells into cell types and preserve the developmental trajectories of the adrenal medulla. Note that we removed 19 DoRothEA regulons from the prior, whose target genes did not overlap with any considered gene features. **(B)** The x-axis and the y-axis indicate the significance level and the mean absolute difference of activity comparisons between two cell types. Note that the differential activity analysis results took Fig.2.3C as references. The volcano plots show that even though nearly all the significantly differential TF activities, except CUX1 (colored in green), were correctly predicted, a majority of TF activities could not be captured or were even wrongly predicted (e.g., HIVEP3, JUNB and JUND) by the model. We considered TFs to be significantly differentially activated if $|\log_e(\text{BF})| > 3$.

2.2.3 Randomizing predefined gene sets breaks model interpretability

To scrutinize the importance of prior biological knowledge in terms of the interpretability of the VEGA latent space, we randomized the aforementioned SCENIC regulons inferred from the human adrenal medulla dataset[32] on varied levels (from 10% to 90%) and trained VEGA using them as the prior to see the changes in the interpretability (see Methods 4.7). Of note, the set of hyperparameters used in this task is the same as Section 2.2.1, which is recorded in Appendix A. The UMAP embeddings of the latent spaces show that the models could cluster cells into cell types and

Table 2.1: **Outline of DoRothEA and SCENIC regulons of interest** — Confidence levels are of DoRothEA, ranging from A (most confident) to E (least confident) depending on the amount of supporting biological evidence.

TF	Confidence level	Gene set size -DoRothEA-	-SCENIC-	Overlapping of target genes
ALX3	E	385	101	4
ATF3	C	50	970	8
BHLHE40	C	58	633	5
CREM	C	52	1715	24
CUX1	C	41	617	5
EGR1	A	124	299	5
EGR4	E	476	62	4
ETS1	A	145	868	6
FOS	A	90	1071	10
FOSL2	A	10	1557	1
FOXO1	A	43	396	4
GATA3	A	73	543	1
HES6	E	199	202	4
HIVEP3	E	192	13	0
JUN	A	121	1975	19
JUNB	C	52	673	3
JUND	A	19	3470	3
NFATC1	C	35	120	0
NFIL3	D	11	252	0
NR2F2	A	18	287	0
SOX10	A	16	285	3
SIX11	C	29	161	1
SOX13	C	42	14	0
SOX2	A	10	168	0
SOX5	E	875	160	29
TFAP2B	E	796	984	55

retain the developmental trajectories no matter how randomized the *a priori* defined gene sets were (Fig.2.5A). This result is theoretically expected because the compressed information that the VEGA latent variables hold mainly depends on the encoder part. The latent variables should in principle be able to learn the learnable variation in the gene expression data with any decoder settings. However, randomizing the predefined gene sets affected the VEGA interpretability to different degrees depending on the level of randomization. The models could preserve the interpretability when slightly randomized gene sets (up to 30%) were used as the prior, which indicates little

incorrect information in prior knowledge is tolerable (Fig.2.6;Appx.B.4). From 40% randomization, the models started making some obvious mistakes. For example, SOX11 has been found having high TF activities in neuroblasts, many members of the JUN TF family are associated with the chromaffin differentiation and FOXO1 has higher TF activities in SCPs compared to neuroblasts[32] (Fig.2.6;Appx.B.4). These previous findings could not be presented by the corresponding GMVs for differential activity tests when the prior knowledge was greatly disorganized. Finally, when 90% of target genes in the prior were shuffled, most of the GMVs could not faithfully reflect differential TF activities in different cell types anymore (Fig.2.6). Of note, the differential activity analysis results took Fig.2.3C as references.

Furthermore, we looked into the learning curves of VEGA which used these different levels of randomized gene sets as prior knowledge. We observed that the efficiency of the model training was highly related to the correctness of the prior knowledge, where the training loss dropped more rapidly when less randomized predefined gene sets were provided (Fig.2.5B). To be more concretely, simulating the decoder wiring as real gene regulatory networks (GRNs) helps VEGA discern relationships between TFs and genes, resulting in more efficient learning. For predictive performance, the validation loss reveals that the models trained using the original and up to 20% randomized gene sets outperformed the other models trained using more than 20% randomized gene sets and there was no apparent difference among the original prior, 10% and 20% randomized gene sets, which implies VEGA should be able to tolerate a bit of wrong information in the prior knowledge (Fig.2.5C). This finding is in line with our earlier differential activity test results, showing the models' interpretability could be preserved when the gene sets were mildly randomized (Fig.2.6). To sum up, correct prior knowledge is important for VEGA to keep the interpretability and generate accurate inferences. Slightly wrong prior knowledge is acceptable but when the prior is incorrect to the certain degree, VEGA starts losing its interpretability since the hard-coded linear decoder leaves no room for correcting prior knowledge[27].

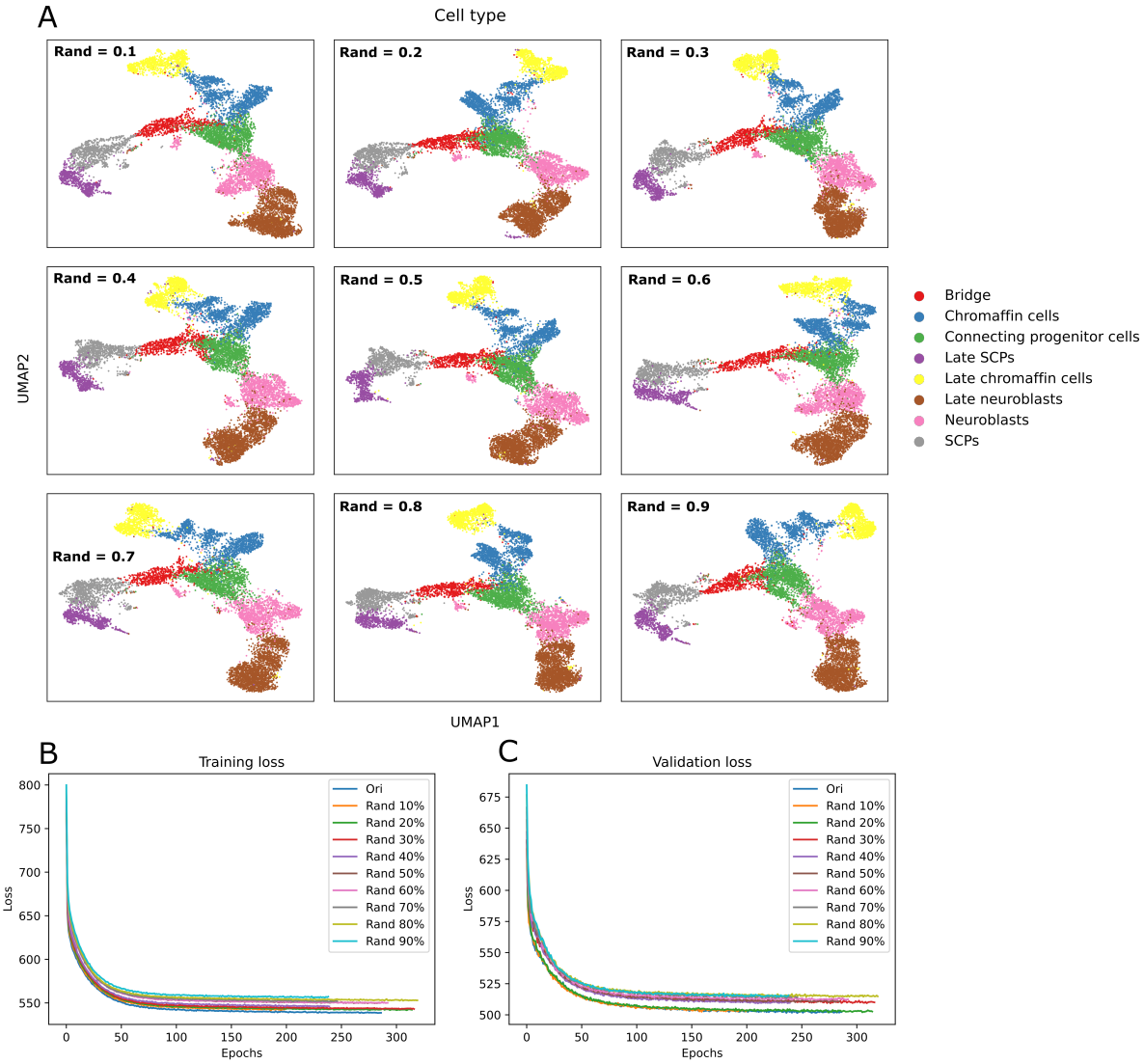

Figure 2.5: **Investigation of model behavior by using disorganized prior** — We randomized the SCENIC regulons inferred from the human adrenal medulla data to different degrees (10% - 90%) to investigate the changes in the model behavior. **(A)** The UMAP plots show that the models could cluster cells into cell types and preserve the adrenal medulla developmental trajectories no matter how randomized the prior was, which corroborates VEGA is able to learn variation in gene expression data with any decoder settings. Note that we removed the same 23 SCENIC regulons from the prior, whose target genes did not overlap with any considered gene features from all versions of the prior knowledge. Rand indicates the degree of randomization of the prior. **(B,C)** The x-axis indicates the epoch of model training and the y-axis indicates the training and the validation loss. The loss curves show that the training efficiency was highly related to the correctness of the prior (see plot B) and the model could tolerate little wrong information in the prior (up to 20% randomization, see plot C). Ori represents the original prior and Rand represents the randomized prior to different degrees.

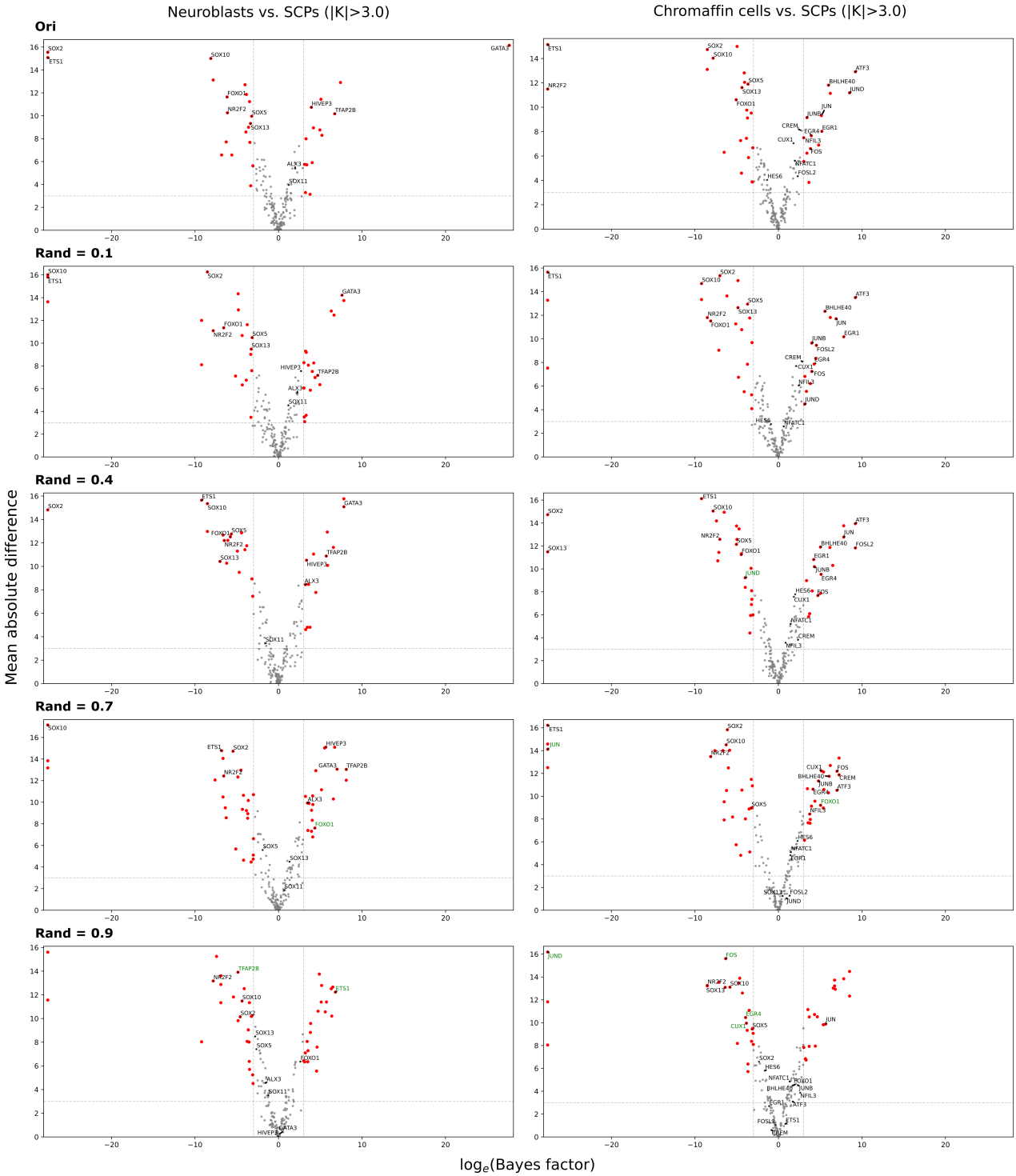

Figure 2.6: The caption is on the next page.

Figure 2.6: **Changes in model interpretability when randomizing prior to different degrees** — The x-axis and the y-axis indicate the significance level and the mean absolute difference of GMV activity comparisons between two cell types. Note that the differential activity analysis results took Fig.2.3C as references. The volcano plots show that the model kept the interpretability when the prior was 10% randomized, made some obvious mistakes (TFs colored in green) when the prior was 40% and 70% randomized and failed to capture most of the TFs of interest when the prior was 90% randomized. Rand indicates the degree of randomization of the prior. We considered TFs to be significantly differentially activated if $|\log_e(\mathrm{BF})| > 3$.

2.2.4 Incorporating batch annotations in model combats batch effects

Deep generative models taking batch information into account have been demonstrated to have the ability to combat batch effects which frequently occur in single-cell transcriptome measurements [18]. VEGA, as an interpretable VAE, can take advantage of this technique and incorporate batch annotations in the encoder and the latent space (see Methods 4.1). To test the function of batch correction of VEGA, we used the human neuroblastoma dataset [32] which consists of 22 neuroblastoma samples and suffers from severe batch effects even though the data integration was performed (see Methods 4.4). The UMAP embedding of the gene expression space clearly shows that cells were clustered mainly due to technical differences between the samples rather than biological differences between the cell types (Fig.2.7A). To remove technical confounders, we incorporated the batch annotations of the considered dataset during model training. Of note, the SCENIC regulons inferred from this human neuroblastoma dataset [32] were used as prior knowledge and the hyperparameters used in this task can be found in Appendix A. After the model training, we ran UMAP on the VEGA embedding. The results show that cells from the different samples were nicely blended and also clustered into cell types, which indicates the model was able to distinguish biological variation from technical variation when the batch information was provided (Fig.2.7B). We also trained VEGA with all the same settings except incorporating the batch information as the contrast. The results did not differ much from the UMAP plot of the gene expression space (Fig.2.7A), which still had strong batch effects (Appx.B.5). Collectively, VEGA incorporating batch information in the encoder and the latent space can effectively alleviate batch effects and provide biologically meaningful representations of combined data. This is an exciting application because we can employ the same model to carry out not only data analysis but also data integration if needed, making the whole analysis procedure more consistent.

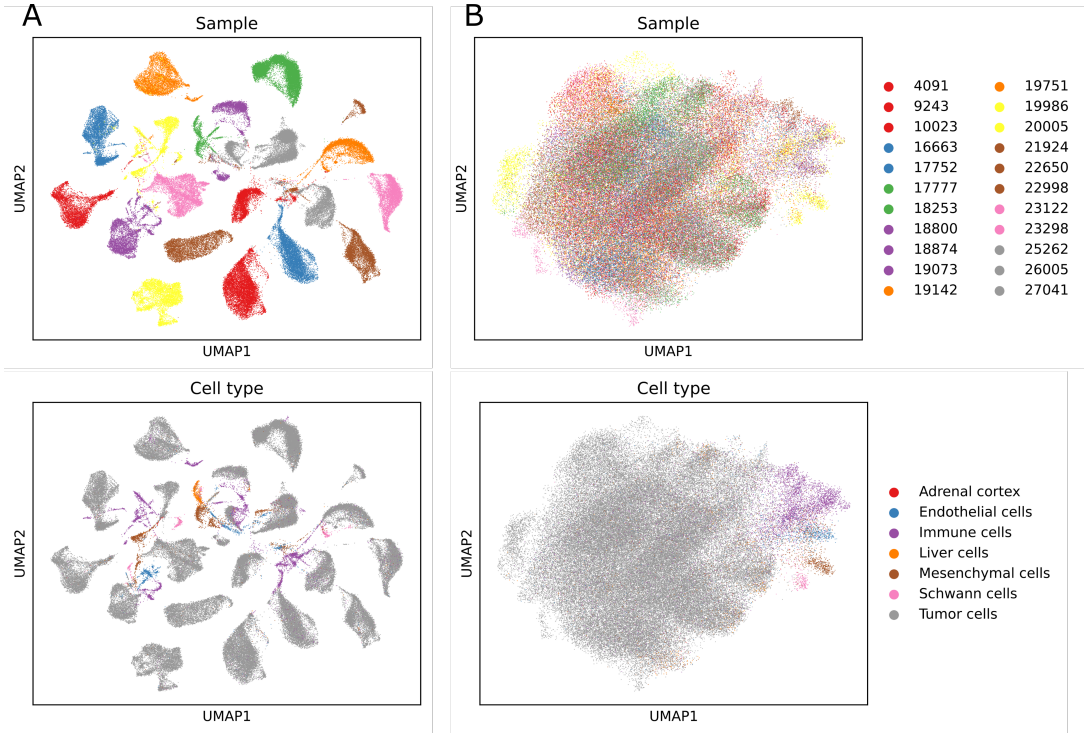

Figure 2.7: **Testing of batch correction function of VEGA** — We tested the function of batch correction of the model by the integration of 22 samples in the neuroblastoma dataset which suffers from severe batch effects. **(A)** The UMAP embedding of the gene expression space shows that cells were clustered mainly based on technical differences between the samples rather than biological differences between the cell types. **(B)** The UMAP embedding of the VEGA latent space shows that cells from the different samples were blended and clustered into cell types when the batch annotations were provided for the model training.

2.3 Model using L1 regularized linear decoder

In the previous work, we performed the benchmark studies on VEGA[27] and investigated some of its properties. The results demonstrated that VEGA is capable of learning biologically meaningful representations of gene expression data when correct prior knowledge was provided. However, one obvious drawback of VEGA lies in its hard-coded linear decoder. Hard-coded connections in the decoder leaves no room for further correcting or expanding existing prior biological knowledge which is usually incomplete or not context-specific[27]. To enable a model to use prior knowledge more flexibly, we employed the L1 regularization technique[28] on decoder weights of unannotated relationships between GMVs and genes in the output layer, introduced in Rybakov et al. (2020).

During model training,

$$w^{(t+1)} = w^t - G \qquad \rightarrow \text{for updating weights of annotated connections} \tag{1}$$

$$w^{(t+1)} = w^t - G - \lambda\alpha|w^t| \qquad \rightarrow \text{for updating weights of unannotated connections} \tag{2}$$

where $w^{(t+1)}$ denotes an updated weight, w^t denotes a weight before updated and G represents the standard gradient descent, which describes the updating of weights of annotated relationships between GMVs and feature genes in Eq.(1). For those weights of unannotated relationships, the weights are additionally penalized by the proximal gradient descent $\lambda\alpha|w^t|$ where λ and α designate the regularization hyperparameter and the learning rate respectively, described in Eq.(2). The product of λ and α in the proximal gradient descent procedure[38] can be intuitively seen as the degree of imposing L1 regularization on weights in the decoder (see Methods 4.2 for more details).

2.3.1 Model recovers artificially removed target genes

Firstly, to gain more knowledge of how a model using the L1 regularized linear decoder works, we systematically trained a number of individual models using different values of $\lambda\alpha$ and artificially modified prior knowledge. We attempted to understand how the regularized decoder behaves when different values of $\lambda\alpha$ are employed and check whether the model is able to correct artificially modified prior knowledge. To this end, we artificially removed target genes from the aforementioned SCENIC[25] regulons inferred from the human adrenal medulla dataset[32]. For instance, we removed three target genes, *EML5*, *STMN2* and *ALK*, from the SCENIC GATA3 regulon according to the top rankings of the GATA3 weights in the hard-coded linear decoder of VEGA (Appx.B.6A). To be clearer, the reason that we picked these three high-ranking genes of GATA3 (i.e., the weights of the connections between GATA3 and these genes are relatively high) was to first check if the model is capable of recovering the missing genes that have strong relationships with GATA3. We made a recovery plot on the GATA3 weights for each of the trained models and computed the area under the curve (AUC) to evaluate the degree of the prior used by the model (see Methods 4.8). The hyperparameters used in this section can be found in Appendix A. The results show that, when $\lambda\alpha = 10^{-5}$ and 10^{-4}, the rankings of the GATA3 annotated genes were evenly distributed over the feature list (the scaled AUC scores were both 0.51), which indicates the prior was not really used by the models (Fig.2.8A). This finding was verified by the recovery plot on the GATA3 weights

from the model using the fully connected decoder (AUC of 0.48, Appx.B.6B). From $\lambda\alpha = 10^{-3}$ to $\lambda\alpha = 0.1$, we observed the higher AUC scores, indicating that the models used the prior more evidently (Fig.2.8A). When $\lambda\alpha = 1$, the model obtained the highest AUC score (0.91) but lost its capacity to recover the removed genes (Fig.2.8A). Moreover, for the count of non-zero links between GATA3 and genes in the output layer (i.e., the putative GATA3 regulon), we observed that the number decreased when the model obviously started using the prior (from $\lambda\alpha = 10^{-3}$) and had a drastic drop when $\lambda\alpha$ was reaching 1 (Appx.B.6C), which is consistent with the information given by the recovery plots. Collectively, we can roughly split the behavior of the regularized decoder into three modes: The fully connected mode when $\lambda\alpha < 10^{-3}$, the regularized mode when $10^{-3} \leq \lambda\alpha < 1$ and the hard-coded mode when $\lambda\alpha = 1$. These findings hold true for all of our experiments (artificially modifying predefined gene sets), that is, the overall decoder behavior is not influenced by slightly modified prior knowledge.

Next, to investigate how the model deciphers the relationships between GATA3 and the feature list and check whether it can recover those three genes artificially removed from the SCENIC GATA3 regulon, we ranked the genes in order of their tuned weights. The results show that, when $\lambda\alpha = 10^{-5}$ and 10^{-4}, the top 20 high-ranking genes of GATA3 were mostly not in the SCENIC GATA3 regulon (Fig.2.8B), and from $\lambda\alpha = 10^{-3}$ to $\lambda\alpha = 1$, the top 20 high-ranking genes were gradually filled with GATA3 target genes (Fig.2.8B), which corroborates our earlier conclusion of the model behavior in terms of different $\lambda\alpha$ values used. Furthermore, when looking into the rankings of those three removed genes, we found the model using $\lambda\alpha = 10^{-3}$ could best recover the removed genes and the models using the other $\lambda\alpha$ values diverging from $\lambda\alpha = 10^{-3}$ (either to the fully connected mode or to the hard-coded mode) had worse recovering performance (Table 2.2). This result made those genes which were high-ranking but not from the SCENIC GATA3 regulon in the model using $\lambda\alpha = 10^{-3}$ extremely intriguing. After checking research papers and existing databases, we found, for example, *LINGO2* and *CTNND2* are significantly upregulated in CD4+ T cells when GATA3 is knocked down by siRNA[39] and *AGBL4* is bound by GATA3 according to TF binding site profiles measured by ChIP-seq from the ENCODE Project[40,41,42]. These pieces of biological evidence indicates the inferred target genes are associated with GATA3, yet, they are not in the context of adrenal medullary cells. Consequently, further work on verifying whether those inferred genes are truly regulated by GATA3 in adrenal medullary cells is needed. Together, the model using $\lambda\alpha = 10^{-3}$ can best recover the missing genes artificially removed from the prior and

potentially infer some other possible target genes.

Apart from GATA3, we performed the same analyses on another TF, JUN, to double check if the behavior of the regularized linear decoder is as our earlier findings. We artificially removed three target genes, *DDC*, *ROBO1* and *GRK5*, from the SCENIC JUN regulon in the same manner as described above. We observed the very similar model behavior that the decoder behaved in a more fully connected way when $\lambda\alpha < 10^{-3}$, in a regularized way when $10^{-3} \leq \lambda\alpha < 1$ and in a hard-coded way when $\lambda\alpha = 1$ (Appx.B.6D,E). Interestingly, the model also best recovered the removed genes when $\lambda\alpha = 10^{-3}$, which is in line with the experiment on GATA3 (Table 2.2). For those genes which were high-ranking but not from the SCENIC JUN regulon in the model using $\lambda\alpha = 10^{-3}$ (Appx.B.6E), we found that *SLC7A5* and *HSP90AA1* are both bound by JUN according to the previous studies[40,41,42,43] and *SLC7A5* is significantly upregulated in BT-549 cells from the mammary gland when JUN is knocked down by siRNA[39]. Again, since the evidence supporting the relationships between JUN and inferred target genes is not context-specific, further work on confirming the inferences is needed. In conclusion, the model using the L1 regularized linear decoder can successfully recover the artificially removed target genes and potentially expand the existing TF target gene sets.

Figure 2.8: **Investigation of regularized decoder behavior** — We artificially removed three genes from the SCENIC regulons inferred from the human adrenal medulla data to investigate the general behavior and the ability to recover the missing target genes of the regularized decoder. Collectively, the results show that the decoder behaved in a fully connected way when $\lambda\alpha < 10^{-3}$, in a regularized way when $10^{-3} \leq \lambda\alpha < 1$ and in a hard-coded way when $\lambda\alpha = 1$. **(A)** The x-axis indicates the ranking of the weights of a certain GMV (i.e.,GATA3) to the gene reconstructions in the decoder and if the corresponding gene was annotated in the SCENIC GATA3 regulon, the frequency increased 1 in the y-axis. The recovery plots display the increasing AUC scores when the values of $\lambda\alpha$ increased. The red bar on a curve indicates the number of non-zero weights of GATA3 (i.e., the putative GATA3 regulon). Note that the zero-valued weights were randomly ranked. **(B)** The x-axis indicates the top 20 high-ranking genes (highest weights) of GATA3 and the y-axis indicates the weight magnitude. The plots show that most of the high-ranking genes were not annotated in the SCENIC GATA3 regulon (colored in black) when $\lambda\alpha < 10^{-3}$ and with increasing $\lambda\alpha$ vlues, the top 20 high-ranking genes were gradually filled with genes annotated in the SCENIC GATA3 regulon (colored in red). Gene names prefixed by double asterisks indicate the artificially removed genes.

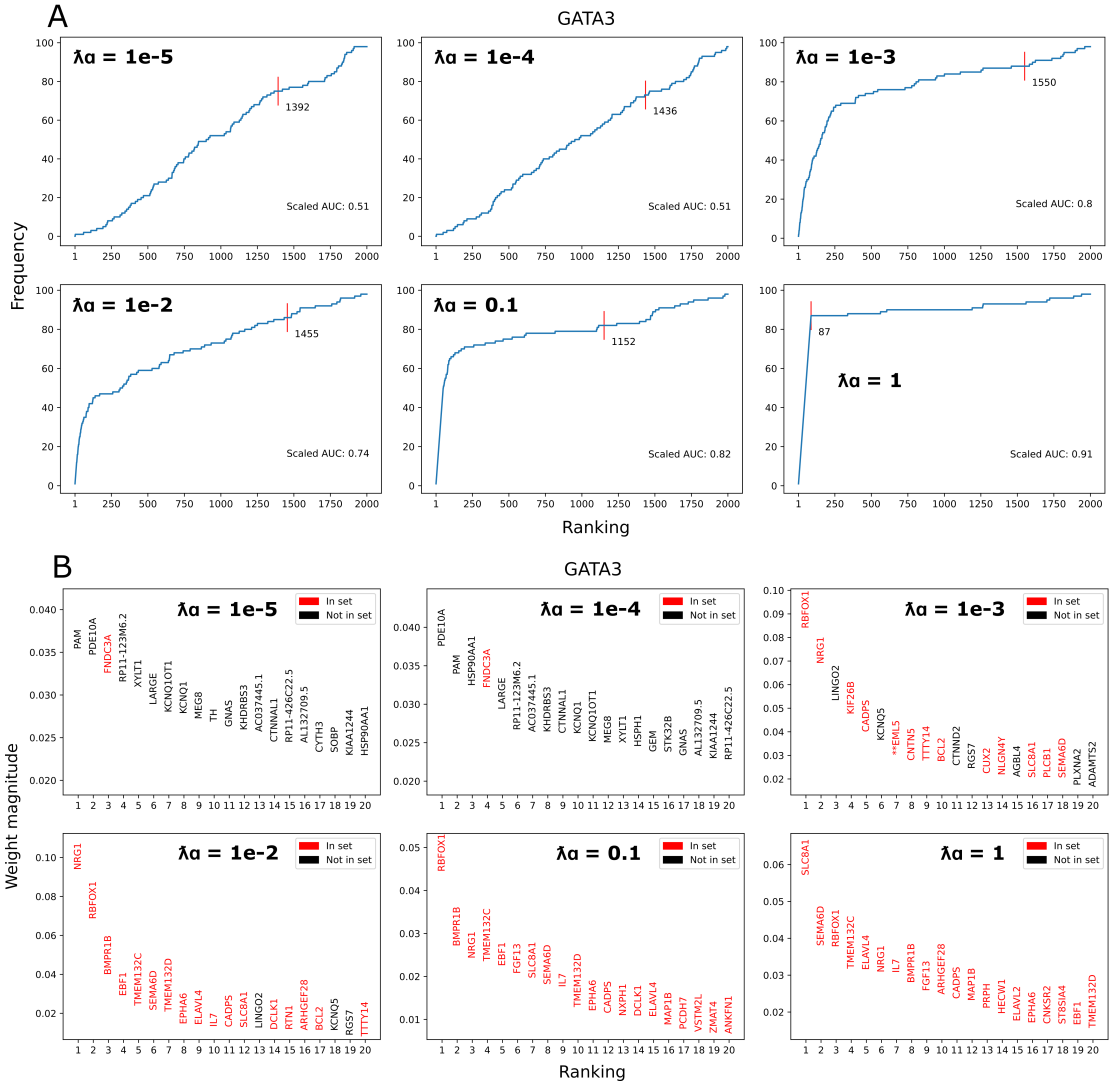

Figure 2.8: The caption is on the previous page.

Table 2.2: **Rankings of artificially removed genes** — *EML5*, *STMN2* and *ALK* were artificially removed from the SCENIC GATA3 regulon and *DDC*, *ROBO1* and *GRK5* were artificially removed from the SCENIC JUN regulon. NA indicates that the weight of GATA3 or JUN to the certain gene reconstruction was zero, which means the artificially removed gene was not recovered.

	GATA3			JUN		
	EML5	***STMN2***	***ALK***	***DDC***	***ROBO1***	***GRK5***
$\boldsymbol{\lambda\alpha = 10^{-5}}$	838	NA	110	22	773	6
$\boldsymbol{\lambda\alpha = 10^{-4}}$	1430	NA	121	293	NA	138
$\boldsymbol{\lambda\alpha = 10^{-3}}$	**7**	**29**	**32**	**114**	**102**	**105**
$\boldsymbol{\lambda\alpha = 10^{-2}}$	77	55	59	118	738	126
$\boldsymbol{\lambda\alpha = 0.1}$	81	64	80	155	NA	169
$\boldsymbol{\lambda\alpha = 1}$	NA	NA	NA	NA	NA	NA

2.3.2 Model excludes artificially added genes

Secondly, we investigated how the model using the L1 regularized decoder treats artificially added genes in prior knowledge using the same human adrenal medulla dataset [32] and the same strategy described in Section 2.3.1. Since we observed that the regularized decoder tends to capture genes whose expression levels are averagely high for regulon inferences, i.e., a large proportion of high-ranking putative target genes of a TF have high average expression levels in the single-cell transcriptome data, e.g., GATA3 (Fig.2.8B), we were inspired to study the behavior of the regularized decoder in two aspects: Artificially adding genes with either high average expression levels or low average expression levels. The way we selected added genes was computing the mean of expression values of each gene across all cells, ranking the genes based on the average expression levels and avoiding picked genes being biologically meaningful to a certain regulon by looked over a couple of existing databases, such as DoRothEA [33], Harmonizome [42], KnockTF [39], etc. For instance, we artificially added three genes with high expression levels: *GNAS*, *MEG8* and *NRXN1* and three genes with low expression levels: *PRDM16*, *MEGF6* and *AJAP1* into the SCENIC GATA3 regulons in the duplicate files containing the prior to create two versions of modified prior knowledge. We then systematically trained several individual models using different $\lambda\alpha$ values and these two versions of artificially modified prior knowledge and checked the rankings of weights of the artificially added non-biologically meaningful genes. The hyperparameters used in this task can be found in Appendix A. Notice that the behavior of the regularized decoder using different $\lambda\alpha$ values is very

similar to our previous findings that the regularized decoder behaves in a more fully-connected way when $\lambda\alpha < 10^{-3}$, in a regularized way when $10^{-3} \leq \lambda\alpha < 1$ and in a hard-coded way when $\lambda\alpha = 1$ (Appx.B.7). The results show that, when $\lambda\alpha = 10^{-5}$ and 10^{-4}, the artificially added genes with high expression levels had the higher average rankings than the added genes with low expression levels, which is in line with our previous observations that the decoder tends to capture genes with high average expression levels (Table 2.3). Theoretically, the reason might be that genes with high average expression levels usually indicates that genes also have high variances, which might need larger weights to take care of more variable gene reconstructions in the decoder. When $10^{-3} \leq \lambda\alpha < 1$ where the regularized decoder uses the prior evidently, some of the artificially added genes could be excluded, yet the changes in the rankings across different $\lambda\alpha$ values did not have a clear pattern (Table 2.3). Roughly, the regularized decoder could best exclude the artificially added genes with high and low expression levels when $\lambda\alpha = 0.1$. Interestingly, when $\lambda\alpha = 1$, all of the artificially added genes with low expression levels were excluded while the added genes with high expression levels were kept, which corroborates again that genes with high expression levels (high variances) are more likely to be captured by the decoder (Table 2.3).

Similarly, apart from GATA3, we performed the same analyses on another TF, JUN. We artificially added three genes with high expression levels: *RBFOX1*, *MEG8* and *DLGAP1* and three genes with low expression levels: *PRDM16*, *TSHR* and *PTPRT* into the SCENIC JUN regulons in the duplicate files containing the prior information. At first glance, the ranking results of these added genes from JUN are not very similar to those from GATA3. When $\lambda\alpha = 10^{-5}$ and 10^{-4}, there was no obvious contrast between the rankings of the artificially added genes with high expression levels and those with low expression levels and, interestingly, *RBFOX1* whose expression level is averagely high was not captured by the decoder, which is not in line with our earlier hypothesis and finding in the example of GATA3 (Table 2.4). When $10^{-3} \leq \lambda\alpha < 1$, in general, the artificially added genes acquired higher rankings when $\lambda\alpha$ increased and the regularized decoder seemed to be able to exclude the added genes with low expression levels when $\lambda\alpha$ was set to 10^{-2} (Table 2.4). When $\lambda\alpha = 1$, all of the artificially added genes were kept except *PRDM16* and the added genes with high expression levels had the higher average ranking than those with low expression levels (Table 2.4). Collectively, the regularized decoder has the potential to exclude artificially added non-biologically meaningful genes, yet there is no specific pattern on how the regularized decoder behaves with different values of $\lambda\alpha$. Therefore, further work on how to generalize this technique to

all genes is needed.

Table 2.3: **Rankings of artificially added genes in SCENIC GATA3 regulon** — We split the task into two parts: artificially adding the genes with high average expression levels (*GNAS*, *MEG8* and *NRXN1*) and the genes with low average expression levels (*PRDM16*, *MEGF6* and *AJAP1*) into the SCENIC GATA3 regulons in the duplicate files containing the prior. The rankings marked in bold represent relatively better outcomes of our testing. NA indicates that the weight of GATA3 to the certain gene reconstruction was zero, which means the artificially added gene was not included in the putative GATA3 regulon.

	HIGH			LOW		
	GNAS	***MEG8***	***NRXN1***	***PRDM16***	***MEGF6***	***AJAP1***
$\boldsymbol{\lambda\alpha = 10^{-5}}$	17	12	724	1158	NA	196
$\boldsymbol{\lambda\alpha = 10^{-4}}$	21	2	503	1024	303	NA
$\boldsymbol{\lambda\alpha = 10^{-3}}$	71	NA	242	NA	348	440
$\boldsymbol{\lambda\alpha = 10^{-2}}$	936	NA	NA	NA	143	191
$\boldsymbol{\lambda\alpha = 0.1}$	**603**	**NA**	**NA**	**951**	**NA**	**166**
$\boldsymbol{\lambda\alpha = 1}$	35	14	94	NA	NA	NA

Table 2.4: **Rankings of artificially added genes in SCENIC JUN regulon** — We split the task into two parts: artificially adding the genes with high average expression levels (*RBFOX1*, *MEG8* and *DLGAP1*) and the genes with low average expression levels (*PRDM16*, *TSHR* and *PTPRT*) into the SCENIC JUN regulons in the duplicate files containing the prior. The rankings marked in bold represent relatively better outcomes of our testing. NA indicates that the weight of JUN to the certain gene reconstruction was zero, which means the artificially added gene was not included in the putative JUN regulon.

	HIGH			LOW		
	RBFOX1	***MEG8***	***DLGAP1***	***PRDM16***	***TSHR***	***PTPRT***
$\boldsymbol{\lambda\alpha = 10^{-5}}$	NA	325	893	628	794	NA
$\boldsymbol{\lambda\alpha = 10^{-4}}$	NA	663	590	708	1038	1000
$\boldsymbol{\lambda\alpha = 10^{-3}}$	894	51	NA	967	385	NA
$\boldsymbol{\lambda\alpha = 10^{-2}}$	13	17	201	**NA**	**NA**	**379**
$\boldsymbol{\lambda\alpha = 0.1}$	4	8	110	544	114	115
$\boldsymbol{\lambda\alpha = 1}$	6	1	53	NA	172	169

2.3.3 Model infers dataset-specific GRNs using general regulons as prior

In a real-world scenario, existing resources where we require prior biological knowledge are usually not context-specific, but some of those unannotated genes might play an important role in certain biological processes in certain tissues or cells[27]. In the previous tasks, we systematically studied the

behavior of the model using the regularized linear decoder when the provided SCENIC regulons were artificially modified. We observed that the model was able to recover the artificially removed target genes and exclude the artificially added non-biologically meaningful genes with certain $\lambda\alpha$ values. We further asked whether the model can infer dataset-specific GRNs in the decoder using general regulons as prior knowledge and provide the biologically meaningful latent space. Similarly to the previous strategy and sticking to the human adrenal medulla dataset [32], we systematically trained several individual models using different $\lambda\alpha$ values and the aforementioned DoRothEA [33] regulons as the prior. The hyperparameters used in this task can be found in Appendix A. Since the decoder behaves in a more fully-connected way when $\lambda\alpha < 10^{-3}$ according to our previous observation, we focused on the values of $\lambda\alpha$ between 10^{-3} and 1, which makes the decoder in the regularized and hard-coded modes (Fig.2.8A). The reason that we included the model whose decoder has no freedom to make further inferences about dataset-specific GRNs (when $\lambda\alpha = 1$) was to provide the control group, the contrast to the regularized decoder. We evaluated the inference capacity of the regularized decoder by making recovery plots on the weights of six TFs of interest (GATA3, ETS1, TFAP2B, JUN, FOS and SOX11) based on the corresponding SCENIC regulons and computing the AUC scores of them (see Methods 4.8). The regularized decoder is considered having the better inference capacity when the AUC scores of those recovery plots are higher, which indicates the greater number of the high-ranking putative target genes are annotated in the SCENIC regulons. Besides, since the DoRothEA and the SCENIC regulons barely overlap with each other (Table 2.1), we also included the recovery curves based on the DoRothEA regulons to investigate how the regularized decoder treats those target genes from the DoRothEA prior. The results show that, in general, the inference performance of the regularized decoder was gradually ameliorated when the values of $\lambda\alpha$ increased (increasing AUC scores), except the inference of the GATA3 regulon, and had the best inference capacity when $\lambda\alpha = 0.9$ (Fig.2.9A, blue curves). As expected, when $\lambda\alpha = 1$, the regularized decoder lost its inference capacity because of a hard-coded nature of the decoder (Fig.2.9A, blue curves). For the gene regulatory information provided by the DoRothEA regulons, as the regular behavior of the regularized decoder, the prior was used more evidently when the $\lambda\alpha$ values increased, which indicates the model preserved the target genes from the DoRothEA regulons while making the inferences about the SCENIC regulons (Fig.2.9A, green curves). Furthermore, to check the interpretability in the latent space, we performed UMAP and differential activity analysis on the embedding of the model using $\lambda\alpha = 0.9$, which had the best

inference performance. The UMAP plot shows that the model could well cluster cells into cell types and preserve the developmental trajectories of the adrenal medulla (Fig.2.9B). The volcano plots reveal that the model could capture more cell type-specific differential TF activities compared to the results from VEGA using the DoRothEA regulons as the prior (Fig.2.4B), yet some of those significantly differential TF activities were falsely predicted, e.g., CUX1, NFATC1 and NR2F2 (Fig.2.9C). Note that the differential activity analysis results took Fig.2.3C as references. Together, the model has the potential to infer dataset-specific GRNs in the regularized decoder using general regulons as prior knowledge and provide the interpretable context-specific latent space.

Next, to verify the more data-specific regulons inferred by the model were truly based on the DoRothEA regulons, we randomized the DoRothEA gene sets to different degrees (10%, 50% and 90%) to investigate whether the regularized decoder can still keep the inference capacity (see Methods 4.7). We expected that the regularized decoder should lose its inference capacity when the DoRothEA gene sets are greatly randomized, which indicates the reasonable general regulons as the prior are critical for making inferences about more specific GRNs. Similarly to the description above, the recovery curves based on the SCENIC and the DoRothEA regulons were both included, but we will mainly focus on the observations of the changes in the inference performance of the regularized decoder (blue curves). We first looked into the model with $\lambda\alpha = 0.9$, which had the best inference performance. We observed that, when 10% and 50% of the DoRothEA target genes were shuffled, the changes in the inference performance (AUC scores) did not have a clear pattern (Fig.2.9D). In some cases, the regularized decoder had the better inference performance when using the randomized DoRothEA gene sets than using the original ones, e.g., GATA3 and in other cases, the better inferences happened when using the 50% randomized DoRothEA gene sets compared to using the 30% randomized ones, e.g., TFAP2B. The reason for these results may be that, owing to the low degree of overlapping between the DoRothEA and the SCENIC regulons, some unannotated genes which play an important role in the GRN reconstructions were put into corresponding regulons and some unimportant target genes were removed from regulons during the randomization procedure. Even so, when the DoRothEA target genes were 90% randomized, nearly all of the GRN inferences, except GATA3, had the low AUC scores which are very similar to the scores obtained from the regularized decoder using the original DoRothEA regulons as prior knowledge when $\lambda\alpha = 1$ (Fig.2.9A), which indicates the decoder could not faithfully infer the SCENIC regulons (Fig.2.9D). Considering that using $\lambda\alpha = 0.9$ did not give the regularized decoder much freedom to make the

inferences, we also looked into the model with $\lambda\alpha = 0.5$ as the sanity check, which had the poorer inference performance than the model with $\lambda\alpha = 0.9$ but still had a tendency to infer the SCENIC regulons. Similarly to the model with $\lambda\alpha = 0.9$, when 10% and 50% of the DoRothEA target genes were randomized, the changes in the inference performance did not have an obvious pattern, but when the DoRothEA target genes were 90% randomized, most of the GRN inferences, except GATA3 and TFAP2B, were lost (Appx.B.8). Together, using the reasonable general regulons as prior knowledge is important for the model to make further inferences about more data-specific GRNs in the decoder.

Figure 2.9: **Context-specific GRN inferences from general prior GRNs** — We trained the model with the regularized decoder on the human adrenal medulla dataset and using the non-context-specific DoRothEA regulons as prior knowledge in an attempt to infer more dataset-specific GRNs. **(A)** The x-axis indicates the ranking of weights of a certain TF and if the corresponding gene was annotated in the SCENIC regulon (blue curves) or in the DoRothEA regulon (green curves) of the TF, the frequency increased 1 in the y-axis. Note that we will focus on blue curves for investigating the changes in the inference performance of the regularized decoder across different $\lambda\alpha$ values. The columns and the rows of the panel A show the different values of $\lambda\alpha$ and the different TFs. The recovery plots show that the regularized decoder had the best inference performance when $\lambda\alpha = 0.9$ and lost its inference capacity when $\lambda\alpha = 1$ (equivalent to the hard-coded decoder). The red bar on a curve indicates the number of non-zero weights of a certain TF (i.e., the putative regulon of the TF). Note that the zero-valued weights were randomly ranked. **(B)** The UMAP embedding of the latent space from the model with $\lambda\alpha = 0.9$ shows the clear cell clustering and the developmental trajectories of the adrenal medulla. **(C)** The x-axis and the y-axis indicate the significance level and the mean absolute difference of GMV activity comparisons between two cell types. The volcano plots reveal that the model using the regularized decoder with $\lambda\alpha = 0.9$ could capture more cell type-specific differential TF activities than using the hard-coded decoder (Fig.2.4B), yet there were also slightly more wrongly predicted significantly differential TF activities (colored in green). We considered TFs to be significantly differentially activated when $|\log_e(\mathrm{BF})| > 3$. **(D)** To verify the more dataset-specific regulons were truly inferred from the DoRothEA regulons, we trained the model with $\lambda\alpha = 0.9$ using the prior which was randomized to different levels to investigate the changes in the performance of decoder inferences. The columns and the rows of the panel D show the different TFs and the different degrees of randomization of the prior. The results show that there was no evident pattern of the changes in the inference performance when the prior was 10% and 50% randomized and nearly all of the GRN inferences, except GATA3, were lost when the prior was 90% randomized. Ori indicates the original prior and Rand indicates the degree of randomization of the prior.

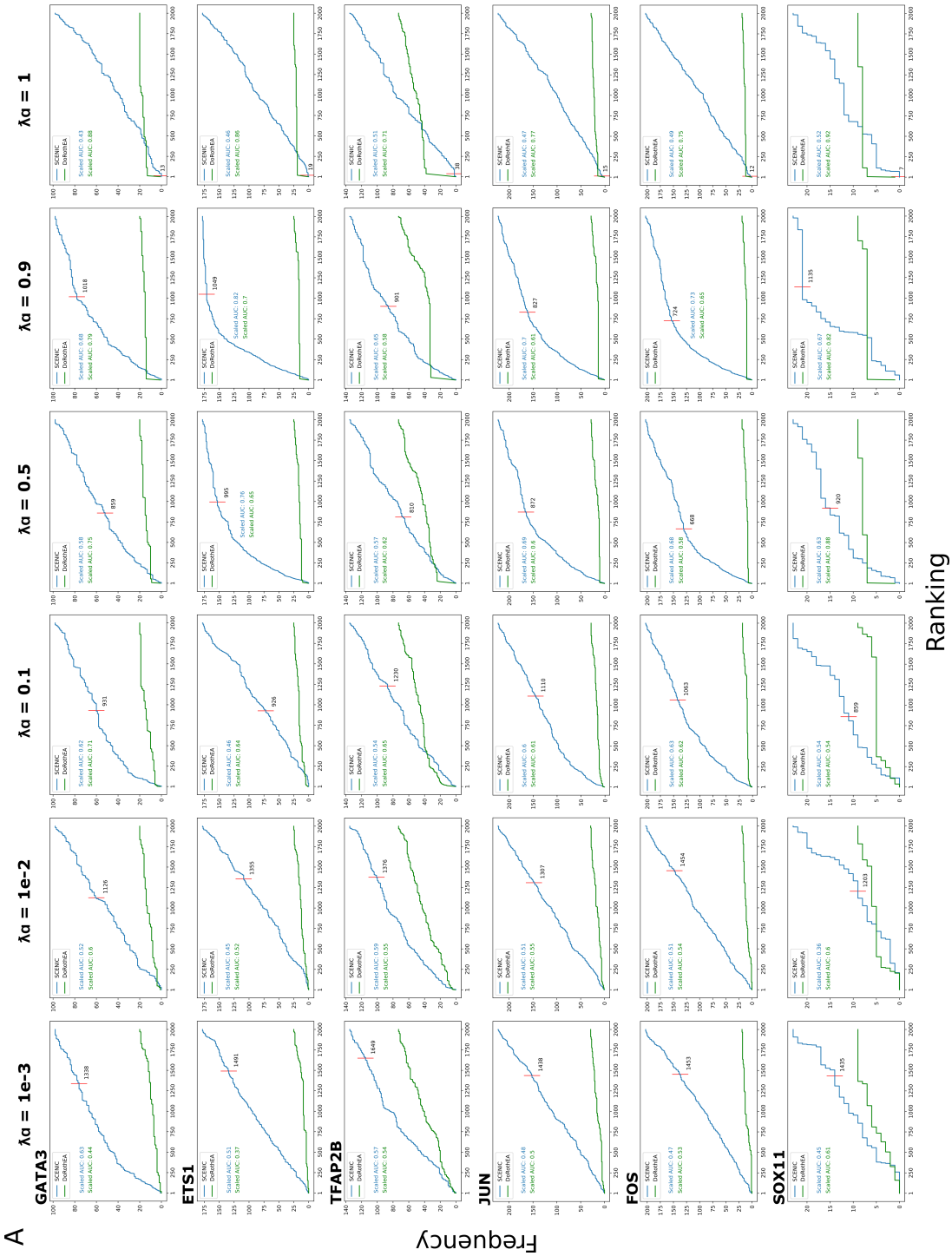

Figure 2.9: The caption is on the previous page.

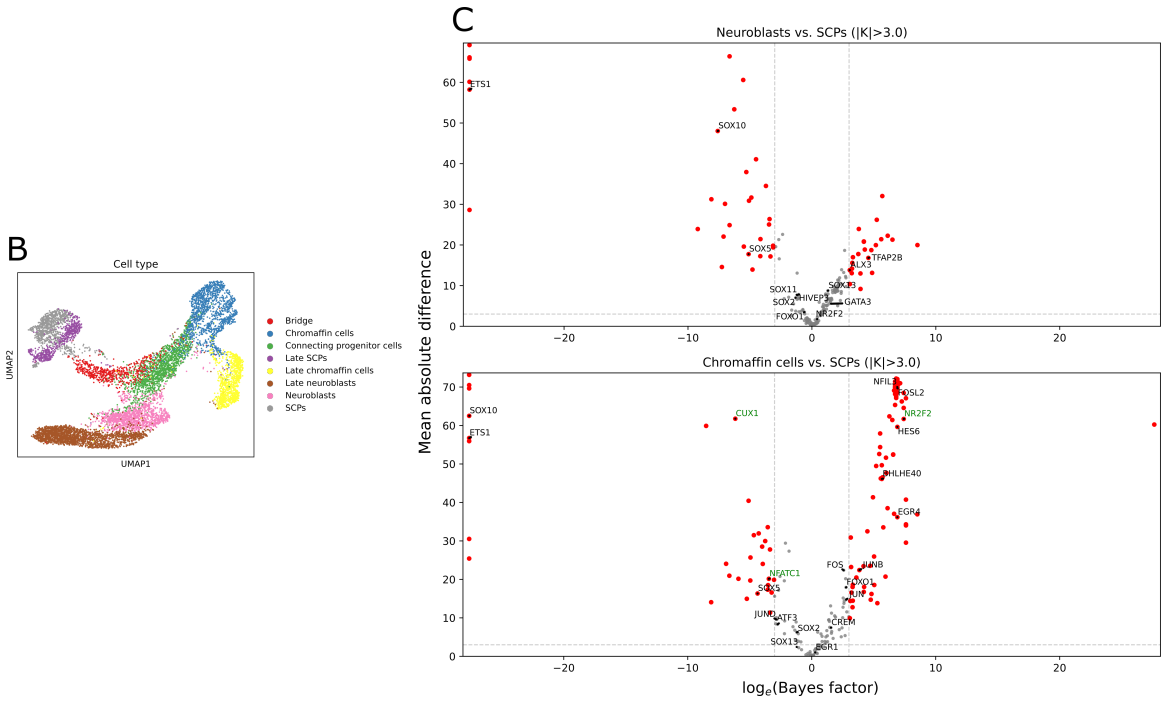

Figure 2.9: This figure is continued from the previous page.

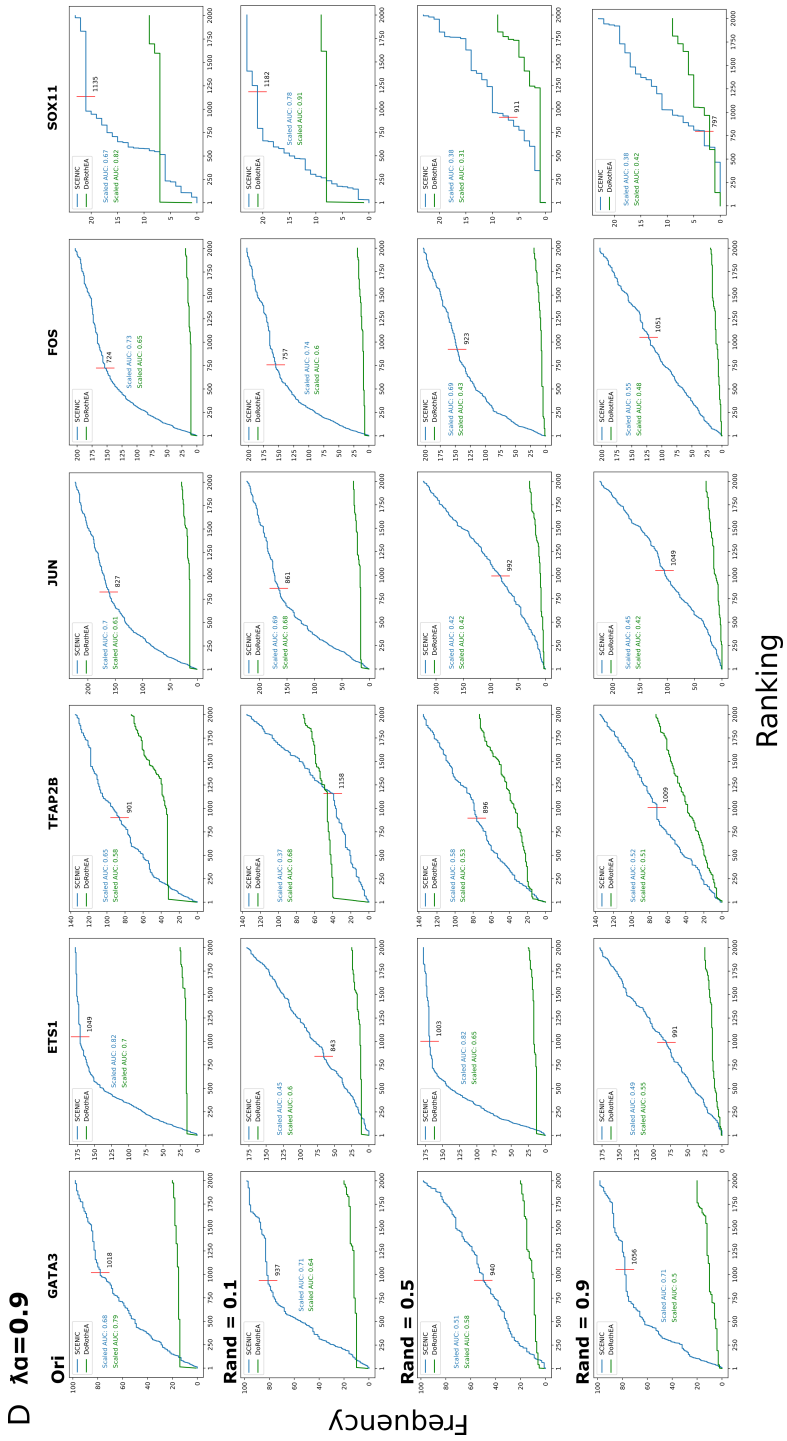

Figure 2.9: This figure is continued from the previous page.

Chapter 3

Discussion

In this work, we investigated the ability and various properties of two kinds of interpretable VAEs which are used to analyze single-cell transcriptomes, introduced in Seninge et al. (2021) and Rybakov et al. (2020) respectively. Firstly, VEGA [27] hard codes the wiring of its single-layer linear decoder through a binary mask that is subject to prior biological knowledge. By limiting the connection of latent variables to certain sets of genes in the output layer, the latent variables can be modeled as a list of gene modules depending on provided prior knowledge. This operation makes VEGA extremely flexible to take advantage of diverse sorts of known biology as prior knowledge, such as pathways, GRNs and cell type marker genes, as long as the prior consists of biologically meaningful gene sets, which enables us to interpret single-cell transcriptomes from many other different viewpoints. From another perspective, VEGA provides an efficient way of inferring the activity of gene modules from transcriptomic data at the single-cell level. To be more concrete, in our work, we analyzed the human adrenal medulla dataset [32] using the SCENIC regulons [32] inferred from the same dataset as the prior to infer the TF activities of individual adrenal medullary cells. Note that VEGA can be seen as an alternative method to the AUCell algorithm in the SCENIC workflow [25], which aims at scoring *a priori* gene sets to identify active TF activities in every single cell. Next, the interpretable VEGA embedding containing inferred gene module activities of all cells can be used for many downstream studies, such as cell type identities, cellular states of different cell populations and so on. Since each GMV is encoded into the posterior distribution, it also provides a natural way of conducting various types of hypothesis testing, e.g., GMV differential activity

analysis[27] used in this work. Our analysis results demonstrate the potential of VEGA in terms of inferring reasonable GMV activities from single-cell transcriptomic data when prior knowledge is provided to guide the decoder wiring (see Section 2.2.1 in Results). Apart from the TFs of interest based on the inferred TF activity result obtained using the AUCell algorithm (Fig.2.3C), VEGA also captured some other significantly differential TF activities between different cell types (Appx.B.3D), which is worth further investigation. However, we did not find any supporting biological evidence for them from previous studies. Therefore, whether these significantly differential TF activities are biologically meaningful or false positives remains unclear.

For the model reproducibility which is important to the core value of VEGA, interpretability, we observed that using a dropout layer in the latent space improves the overall model training stability (see Section 2.1.2 in Results). Besides, using a dropout layer in the latent space also has been proved to be an effective way of preserving the redundancy between similar GMVs (i.e., prior gene modules whose gene sets are overlapped with each other)[27]. However, there were still a number of unrelated or even anticorrelated GMVs between two individual trained models, and we found that a subgroup of GMVs whose decoder wiring is highly similar tend to have low correlations. Note that the PBMCs dataset[30] and the Reactome pathways[31] are discussed in this part. For example, we observed that the GMVs representing the Notch signaling pathways (NOTCH2, NOTCH3 and NOTCH4) whose gene sets are 92% overlapped with each other had the poor reproducibility despite a dropout layer used in the latent space (Table 3.1). The reason may be that the learned GMV-specific information is randomly distributed over these highly similar GMVs due to the overlapping decoder connections, so the information these GMVs hold will be more variable from training to training, which also implies the loss of the GMV interpretability. Yet, highly overlapping prior gene sets is not always the cause because we also observed subgroups of highly similar GMVs had the favorable reproducibility. Therefore, further work on discerning causes of poor reproducibility and finding a way to adress this issue for generalizing the model to possibly all gene modules is needed.

Secondly, one big limitation of VEGA lies in its hard-coded decoder which leaves no room for further correcting or expanding existing biological knowledge bases[27]. To this end, instead of hard coding, the L1 regularization technique can be selectively employed on weights in the single-layer linear decoder through a binary mask based on the prior. By penalizing the weight of decoder connections that are not included in the prior (i.e., gradually shrinking the weight of unannotated relationships between GMVs and genes to zero), the model has a chance to preserve

Table 3.1: **Example of poor reproducibility of highly overlapping gene modules** — *z_dropout* indicates the dropout rate of a dropout layer used in the latent space. The numeric values indicate the PCC of a certain gene module between two individual trained models.

	***z_dropout* = 0**	***z_dropout* = 0.3**	***z_dropout* = 0.5**
NOTCH2	0.006	0.308	0.238
NOTCH3	0.013	0.320	0.189
NOTCH4	0.018	0.333	0.219

potentially important unannotated genes for a certain gene module and in the meantime, can exclude computationally unassociated genes. Note that the human adrenal medulla dataset[32] and the SCENIC adrenal medulla regulons[32] are discussed in this part. Our experimental results indicate that the model using the regularized decoder had the ability to recover the missing target genes and expand the SCENIC adrenal medulla regulons (see Section 2.3.1 in Results). However, for non-biologically meaningful genes, the model could exclude them in certain conditions but there was no clear pattern of how the regularized decoder handle those artificially added genes (see Section 2.3.2 in Results). Besides, we observed that genes with a high average expression level are more likely to have high weight values. Theoretically, the reason might be that genes having a high expression level usually indicate they also have high variance, so they need higher weights to take care of the reconstruction of more variable gene expression levels. Generally, we regard the weight of decoder connections as the strength of relationships between GMVs and genes, yet whether the decoder weights are biologically meaningful or just computationally meaningful needs further investigation.

Moreover, we used the non-context-specific DoRothEA regulons[33] as the prior in an attempt to infer more dataset-specific regulons (see Section 2.3.3). Our experimental results show that the model using the regularized decoder can potentially infer more dataset-specific GRNs from the DoRothEA regulons. However, we also observed that the inference of dataset-specific regulons cannot be generalized to a majority of TFs (Appx. B.9). According to our current knowledge, apart from the value of $\lambda\alpha$, there are many other factors in control of the behavior of the regularized decoder, such as the combination of target gene sets of each TF (i.e., the association of each gene with GMVs). Hence, further work on deciphering the decoder behavior in more detail is needed so that the generalization of the regularized decoder can be improved. To this end, it

is an interesting question whether the model using the regularized decoder can be an alternative method to SCENIC because, in terms of the runtime, the inference of putative regulons and TF activities in individual cells using the model with the regularized decoder is much faster than SCENIC. Even so, as frequently mentioned above, putative regulons inferred by the model using the regularized decoder may contain many false positives (i.e., non-biologically meaningful target genes) due to a data-driven fashion. Functionally, the model using the regularized decoder can be roughly seen as the SCENIC workflow without the second step, motif enrichment analysis, which enables inferred regulons to only keep direct-binding target genes. Therefore, further work can be either proving putative regulons inferred by the model using the regularized decoder are mostly biologically meaningful or developing the model that can incorporate more prior information (e.g., motifs) to exclude false positives.

To sum up, VAEs either using the hard-coded decoder or using the regularized decoder effectively enhance the interpretability of the latent space, which can present the inferred activity of diverse sorts of gene modules from scRNA-seq data. These two different methods to boost the model interpretability can be complementary to each other due to their unique attributes. The VAEs using the hard-coded decoder can faithfully interpret the learned variation between individual cells from different points of view according to provided prior knowledge but are lack of flexibility in terms of exploring unknown or context-specific biological knowledge. Hence, it is a more desirable method when the prior is well-studied or fully context-specific. On the other hand, the VAEs using the regularized decoder enable the model to potentially correct or expand existing biological knowledge but the new-found may be false positives owing to a data-driven fashion.

Chapter 4

Methods

4.1 Architecture of VEGA

For simplicity, we follow the same notation as Seninge et al. (2021). VEGA is an interpretable VAE consisting of a two-layer nonlinear encoder (inference part) and a single-layer linear decoder (generative part), which attempts to maximize the likelihood of a single-cell dataset X under a generative process[27,16,19], formulated as:

$$p(X \mid \Theta) = \int p(X \mid Z, \Theta) p(Z \mid \Theta) dZ \tag{3}$$

where Θ denotes the learnable parameters of the model and Z represents the latent variables. Since the decoder is single-layer and linear, predefined gene modules (gene sets), such as pathways, GRNs and so on, can be used to initiate the decoder wiring, which can then model a set of latent variables Z as certain biological entities. To be more concrete, the connections between a latent variable $z^{(j)}$ and the gene features in the output layer can be specified using a binary mask M where $M_{i,j} = 1$ (true) if a gene i from the feature list is annotated in the gene module j and $M_{i,j} = 0$ (false) otherwise and those connections which are masked off ($M_{i,j} = 0$) will always have zero-valued weights during model training, constraining each latent variable to connect to a certain subset of genes in the decoder (a column of the mask matrix M). Of note, each latent variable $z^{(j)}$ can be referred to as a gene module variable (GMV) because each represents the corresponding gene

modules j, which enables us to interpret a single-cell dataset X from another viewpoint. Besides, the weights in the decoder are constrained to be non-negative to maintain the interpretability of the latent space. Since the gene reconstructions in the decoder are linear transformations and the GMVs can be both positive and negative, predictions in the latent space can be totally opposite if the weights are not constrained to be non-negative.

To have the model generative, the GMVs are modeled as the posterior distribution given single-cell data X with a variational distribution through an inference process in the encoder, which makes the latent space more continuous and complete[17]. However, since the posterior distribution $p(Z \mid X)$ is usually intractable, another simpler and more tractable distribution $q(Z \mid X)$ is used to simplify the problem. Modeling the latent space as a multivariate normal distribution is a common choice and has been demonstrated to work well in previous single-cell transcriptome studies[27,18,19], which is formulated as:

$$q(Z \mid X, \phi) = \mathcal{N}(\mu_\phi(X),\, \Sigma_\phi(X)) \tag{4}$$

where ϕ denotes the learnable parameters of the encoder. As a standard VAE implementation[16], the objective is to maximize the evidence lower bound (ELBO) during model training, described as:

$$\mathcal{L}(X) = E_{q(Z \mid X, \phi)}[\log p(X \mid Z, \Theta)] \;-\; \mathrm{KLD}(q(Z \mid X, \phi) \,||\, p(Z \mid \Theta)) \tag{5}$$

where the expected value of the variational distribution can be approximated using Monte Carlo integration[44] and the Kullback-Leibler divergence[45] (KLD) term has a closed-form solution because the prior $p(Z \mid \Theta)$ is set to $\mathcal{N}(0, 1)$. Since the GMVs are sampled from the latent normal distribution, the gradients fail to flow through fully stochastic nodes during backpropagation[46]. For this end, the reparameterization trick[16] is used to enable effective model training.

There are two novel hyperparameters introduced in VEGA: A dropout layer and additional fully connected nodes in the latent space. A nature of each node in a deep learning model is to learn most important information in data as diverse as possible and the learned information is randomly distributed over the nodes. To this end, if there are two gene modules in prior knowledge highly overlapping with each other, the wiring of these two GMVs in the decoder will be very similar, so instead of modeling both the latent variables as certain highly correlated gene modules, the model may be forced to learn only one arbitrary gene module or randomly share the learned information between the latent variables, leading to loss of information and interpretability. To address this

issue, a dropout layer is employed in the latent space, which has been demonstrated to help the model preserve highly correlated gene modules[27].

Since predefined gene modules which we use as prior knowledge cannot always cover all gene features in the output layer, i.e., there are no connections between the GMVs and the genes which are not annotated in any predefined gene modules, the reconstructions of those unannotated genes during model training will be problematic. Besides, some of those unannotated genes might provide undiscovered meaningful biological information, which can make the interpretable latent space even more informative. To this end, additional fully connected nodes are employed in the latent space to (1) enable the model to reconstruct those unannotated genes during training, which can boost predictive performance on gene expressioin values, and (2) help the model capture possible additional important information that is unexplained in the prior. However, notice that the number of additional fully connected nodes used in the latent space is a trade-off between the model predictive performance and the loss of information in the GMVs. In principle, increasing additional fully connected nodes results in better predictive performance but less informative GMVs.

Last but not least, incorporating batch information through one-hot encoding in the encoder and the latent space has been proved to be able to alleviate batch effects[27,18]. For doing so, categorical covariates are directly concatenated to input data (additional nodes in the input layer) and representations of the input data (additional nodes in the latent space).

4.2 Implementation of L1 regularization technique

The L1 regularization technique[28] adds a penalty term to the loss function, which encourages the sum of the absolute values of the model parameters to be as small as possible during model training. This is effective in preventing overfitting because the weights of less important features will be gradually shrunk to zero, resulting in sparse feature vectors. The loss function with the L1 regularization term is formulated as:

$$L + \lambda \sum_{j=1}^{n} |\theta_j| \tag{6}$$

where L represents the loss function of a model, λ denotes the regularization parameter and θ_j denotes a model parameter. When λ is set to a very large value, it will make more weights become zero during model training, leading to underfitting. On the other hand, when λ is too small, the

effect of model regularization will be unnoticeable.

To enable the single-layer linear decoder of the an interpretable VAE to use prior knowledge more flexibly, instead of hard coding the decoder wiring[27], the L1 regularization technique can be selectively employed on weights in the decoder through a binary mask, introduced in Rybakov et al. (2020). Following the same notation used in Methods 4.1, prior gene modules are converted into a binary mask M where $M_{i,j} = 1$ if a gene i from the feature list is not annotated in a gene module j and $M_{i,j} = 0$ otherwise. Note that a binary mask based on the prior used in the regularized decoder is opposite to that used in the hard-coded decoder, which pinpoints the unannotated relationships between GMVs and genes, where the L1 regularization will be imposed on. The loss function of the model is formulated as:

$$L + \lambda \sum_{j=1}^{n} \|W_{:,j} \circ M_{:,j}\|_1 \tag{7}$$

where $W_{:,j} \circ M_{:,j}$ represents the element-wise product between the j-th column of the decoder weight matrix W and the j-th column of the binary mask M, which means that $\|W_{:,j} \circ M_{:,j}\|_1$ is the sum of the absolute values of weights for feature genes that are not annotated in a gene module j. The weights of those decoder connections which are not annotated in prior knowledge will be penalized by gradually shrinking them to zero during model training rather than straight zeroed out, giving the model some freedom to recover potentially important missing gene module-gene relationships.

Since the loss function with the L1 regularization term makes it a non-differentiable function, the proximal gradient algorithm[38] is employed to enable the model to be optimized during model training. The proximal gradient descent was used after the standard gradient descent to update decoder weights[27], which is formulated as:

$$w^{(t+1)} = w^t - G \qquad \rightarrow \text{for updating weights of annotated connections} \tag{8}$$

$$w^{(t+1)} = w^t - G - \lambda\alpha|w^t| \qquad \rightarrow \text{for updating weights of unannotated connections} \tag{9}$$

where $w^{(t+1)}$ denotes an updated weight, w^t denotes a weight before updated and G represents the standard gradient descent, which describes the updating of weights of annotated relationships between GMVs and feature genes in Eq.(8). For those weights of unannotated relationships, besides the standard gradient descent, the weights are penalized by the proximal gradient descent $\lambda\alpha|w^t|$ where α is the learning rate, described in Eq.(9). The product of λ and α in the proximal gradient

descent procedure is the hyperparameter to control the behavior of the regularized decoder, which can be intuitively seen as the degree of imposing L1 regularization on weights in the decoder. Basically, α is usually a fixed number, so λ is the hyperparameter of interest.

4.3 Bayesian differential activity analysis

The differences in activities of TFs or pathways between two groups of cells can always provide valuable biological insights. To this end, we employed the differential GMV activity analysis procedure proposed by Seninge et al. (2021), which is inspired by the Bayesian differential gene expression procedure introduced in Lopez el al. (2018). For simplicity, we follow the same notation as Seninge et al. (2021). For each GMV k and pair of cells (x_a, x_b) with inferred GMV activities (z_a, z_b) and their group IDs (s_a, s_b) (e.g., two different cell types or cells treated under two different conditions), the two mutually exclusive hypotheses can be formulated as:

$$\mathcal{H}_0^k := E_s[z_a^k] > E_s[z_b^k] \;\text{ vs. }\; \mathcal{H}_1^k := E_s[z_a^k] \leq E_s[z_b^k] \tag{10}$$

where the expectation E represents the empirical frequency. The hypotheses can be simply interpreted as whether a cell has a higher mean of a certain GMV activity than another. Finally, the more probable hypothesis can be determined by a single numeric, a log-Bayes factor[36,37] (BF), defined as:

$$K = \log_e \frac{p(\mathcal{H}_0^k \mid x_a, x_b)}{p(\mathcal{H}_1^k \mid x_a, x_b)} \tag{11}$$

The sign of K indicates which hypothesis is more likely and the magnitude of K reveals the significance level. Since the posterior distribution over each GMV can be approximated via the variational distribution (i.e., $q(Z \mid X)$, the encoding part of the model), the probability of each hypothesis can then be approximated by

$$p(\mathcal{H}_0^k \mid x_a, x_b) \approx \sum_s p(s) \int_{z_a} \int_{z_b} p(z_a^k > z_b^k)\, dq(z_a^k \mid x_a)\, dq(z_b^k \mid x_b) \tag{12}$$

where $p(s)$ denotes the relative abundance of cells in a group s and the integrals can be computed using naive Monte Carlo[47] due to the low dimensionality of all measures.

Sticking to the same assumption made in Lopez el al. (2018) and Seninge et al. (2021) that all

cells are independent, we can compute the average Bayes factor across a large set of cell pairs where cells in each cell pair are randomly sampled and can be repeatedly taken from the corresponding cell groups (i.e., permutations), which brings about more comprehensive cell comparisons. The average Bayes factor can tell us if a GMV is more active at a higher frequency in one group or the other. In this work, we also considered GMVs to be significantly differentially activated when the absolute value of K is greater than 3 (equivalent to a BF ≈ 20)[27,36,18].

4.4 Datasets

4.4.1 Kang et al. dataset

The PBMCs dataset[30] consists of two groups of blood cells: Control cells and cells stimulated with interferon-β. The cell type annotation and the data preprocessing was conducted using Scanpy package[48] in Python and described in the VEGA paper[27]. The final dataset includes 16893 cells (8007 control cells and 8886 stimulated cells) with the top 6998 highly variable genes, which is downloadable at the GitHub repository[a] provided by VEGA authors. The PBMCs dataset was used in this work to reproduce the results in the VEGA paper and investigate the stability of model training. The reproducibility code can also be downloaded at the same GitHub repository[a].

4.4.2 Jansky et al. datasets

Two datasets from Jansky et al. (2021) were used in this work: (1) the human adrenal medulla dataset and (2) the human neuroblastoma dataset. The cell type annotation and the data preprocessing was performed using Seurat R package[49,50] and described in the paper[32]. The adrenal medulla dataset consists of human healthy cells spanning several different embryonic and fetal developmental time points, including SCPs, chromaffin cells, neuroblasts and the other transient populations which are termed bridge and connecting progenitor cells. The final adrenal medulla dataset includes 9387 cells with the whole 28422 genes or the top 2000 highly variable genes, which can be downloaded at the link[b]. The human neuroblastoma dataset consists of mostly tumor cells and a small number of normal cells from 22 neuroblastoma samples, containing 104881 cells with 28312 genes. Even though the data integration was performed using Seurat R package, the

[a] `https://github.com/LucasESBS/vega-reproducibility`
[b] `https://adrenal.kitz-heidelberg.de/developmental_programs_NB_viz/`

combined neuroblastoma data still suffers from severe batch effects (without using Harmony[51]). Therefore, this dataset is favorable in our work for testing the function of batch correction of the model. Note that the designed model takes AnnData (annotated data) as input based on anndata Python package[52], so sceasy R package[53] was used to convert a Seurat object to AnnData which is in h5ad format.

4.5 Prior biological abstractions for guiding decoder wiring

Prior biological abstractions which is composed of gene modules (gene sets) can be used to guide the connections of the linear decoder, which models the GMVs in the latent space as the certain biological entities[27]. In this work, the Reactome collection of pathways and processes[31] (674 gene sets) was employed as prior knowledge for reproducing the analyses of the PBMCs dataset[30] from the VEGA paper[27], which can be downloaded at the GitHub repository[c]. The SCENIC[25] regulons inferred from the human adrenal medulla dataset (215 gene sets) and the human neuroblastoma dataset (67 gene sets) were taken from Jansky et al. (2021) for investigating the ability and the properties of VEGA and the behavior of the model using the L1 regularized decoder. The non-context-specific DoRothEA[33] regulons (1333 gene sets) were used for investigating the inference capacity of the regularized decoder, which can be accessed through dorothea R package. Of note, since the tasks where the DoRothEA regulons were used were closely associated with the adrenal medulla SCENIC regulons, the DoRothEA regulons were filtered based on the adrenal medulla SCENIC regulons, resulting in the final 211 gene sets.

4.6 Examination of model reproducibility

The core value of VEGA[27] is its interpretable latent space that can provide meaningful biological insights at the single-cell level. As a result, the reproducibility of the latent space is fairly important. The strategy we used to measure the VEGA reproducibility was training two individual VEGA models using the same set of hyperparameters on the same dataset and computing a Pearson correlation coefficient of each GMV in the latent space between these two trained models. The Pearson correlation coefficient was computed using SciPy package[54] in Python (`scipy.stats.pearsonr`).

[c] `https://github.com/LucasESBS/vega-reproducibility`

4.7 Randomization of *a priori* defined gene sets

It has been validated that using prior biological knowledge to guide the decoder wiring can model the latent variables as the interpretable GMVs[27,29,55]. To study the importance of correct prior knowledge concerning the model interpretability, we randomized predefined gene sets on varied levels. According to the degree of randomization, e.g., 50%, half genes from every predefined gene set were extracted and put into a gene space that contained all extracted genes. Next, the certain number of genes which did not overlap with the earlier extracted genes were taken from the gene space to refill each of the gene sets. Of note, the number of extracted genes was determined by the size of a specific gene set multiplied by the randomization level and rounded, so gene sets with small sizes on the low randomization level may acquire zero gene to be extracted. In this scenario, the number of extracted gene was set to 1.

4.8 Recovery plot for studying regularized decoder

To study the behavior of the L1 regularized linear decoder, we made a recovery plot[10] on the weights of a specific GMV to each gene reconstruction in the decoder. The weights were ranked from high to low and represented by the rankings in the x-axis and the frequency increased 1 in the y-axis when the corresponding reconstructed gene is annotated in the prior gene module. The area under the curve (AUC) was then computed using SciPy package[54] in Python (`scipy.integrate.simps`) and scaled by the reciprocal of the maximum square area (the number of genes in the feature list multiplied by the number of the intersection of the certain gene module and the feature list) to evaluate the degree of the prior used by the model. In general, the weights of the GMV to the gene reconstructions where genes are annotated in the prior gene module should be relatively high (i.e., the reconstructed genes annotated in the prior gene module averagely have the higher ranks), resulting in the higher scaled AUC score. Of note, zero-valued weights were randomly ranked.

4.9 Implementation of UMAP for visualization

To visualize datasets and model embeddings, the UMAP algorithm[34] was used via Scanpy Python package[48] where `scanpy.pp.neighbors` was first employed to compute a neighborhood graph of data and then `scanpy.tl.umap` was run on the neighborhood graph for dimensionality reduction.

All parameters were set as the default. For the processed data from Jansky et al. (2021), the UMAP embeddings of gene expression spaces had been computed via Seurat R package[49,50]. Hence, we directly used `scanpy.pl.umap` to visualize the embeddings.

Appendix A

Hyperparameters

The model architecture is described in Introduction 1.3. Unless mentioned otherwise in certain tasks, the hyperparameters used for model training are provided in the tables below in each section.

VEGA reproducibility

Table A.1 shows the hyperparameters for VEGA trained on the PBMCs dataset [30] in Results 2.1. In this reproducibility section, we used the Reactome pathways [31] as prior knowledge, so the dimension of the latent space is the number of gene modules included in the Reactome prior information (674) plus the number of additional fully connected nodes (1).

Table A.1: Hyperparameters for VEGA trained on PBMCs dataset

add_nodes	**min_genes**	**max_genes**	**dropout**	**z_dropout**	**beta**
1	0	1000	0.5	0.5	$5 \cdot 10^{-5}$
train_size	**batch_size**	**optimizer**	**learning_rate**	**num_epochs**	**train_patience**
1	64	Adam	10^{-4}	300	100

add_nodes indicates the number of additional fully connected nodes used in the latent space.
min_genes and *max_genes* specify the minimum and maximum numbers of gene annotations per accepted gene module.
dropout indicates the dropout rate of a dropout layer used in the encoder part.
z_dropout indicates the dropout rate of a dropout layer used in the latent space.

beta indicates the weight for KL divergence in the VEGA loss function.
train_size indicates the proportion of training data. Note that the number should be set between 0 and 1. The validation data will be $1 - train\_size$.
batch_size, *learning_rate* and *optimizer* indicates the number of data points taken by VEGA using the certain learning rate with the certain optimizer during the training section for a run.
num_epochs indicates the maximum number of epochs for model training.
train_patience indicates the number of epochs for early stopping if VEGA training loss is not being improved anymore.

Performing benchmarks for VEGA

Table A.2 shows the hyperparameters used for model training in Results 2.2. We employed two datasets for different tasks in this section: The human adrenal medulla dataset [32] which was used to benchmark VEGA and investigate some properties of the model and the human neuroblastoma dataset [32] which was used to test the function of batch correction of the model. For prior knowledge, we used two collections of the SCENIC [25] regulons inferred from these two datasets respectively and the non-context-specific DoRothEA [33] regulons, so the dimension of the latent space is the number of the adrenal medulla SCENIC regulons (215), the neuroblastoma SCENIC regulons (67) and the DoRothEA regulons (211) plus the number of additional fully connected nodes (1).

Table A.2: Hyperparameters for VEGA benchmark studies

add_nodes	**min_genes**	**max_genes**	**dropout**	**z_dropout**	**beta**
1	0	5000	0.5	0.5	$5 \cdot 10^{-5}$
train_size	**batch_size**	**learning_rate**	**num_epochs**	**train_patience**	**valid_patience**
0.8	64	$4 \cdot 10^{-4}$	500	40	40

The explanatory notes of the hyperparameters are given above except *valid_patience* that indicates the number of epochs for early stopping if VEGA validation loss is not being improved anymore. Note that the *optimizer* used in the benchmark studies is also the Adam optimization algorithm.

Model using L1 regularized linear decoder

In Results 2.3, we tuned the value of *lambda* (λ) to investigate the behavior and the ability of the regularized decoder, and the other hyperparameters used for model training can be found in Table A.3. The human adrenal medulla dataset[32] was used in this section. For prior knowledge, the adrenal medulla SCENIC regulons and the DoRothEA regulons were used, so the dimension of the latent space is the number of the adrenal medulla SCENIC regulons (215) and the DoRothEA regulons (211). Notice that we did not employ additional fully connected nodes in the latent space.

Table A.3: Hyperparameters for model using regularized linear decoder

add_nodes	**min_genes**	**max_genes**	**dropout**	**z_dropout**	**beta**
0	0	5000	0.5	0.5	$5 \cdot 10^{-5}$
train_size	**batch_size**	**learning_rate**	**num_epochs**	**valid_patience**	**lambda**
0.8	64	$4 \cdot 10^{-4}$	500	40	0.025

The explanatory notes of the hyperparameters are given above except *lambda* that indicates the degree of imposing L1 regularization on weights in the decoder. Note that the *optimizer* used in this section is also the Adam optimization algorithm and the *train_patience* is 40.

Appendix B

Supplementary material

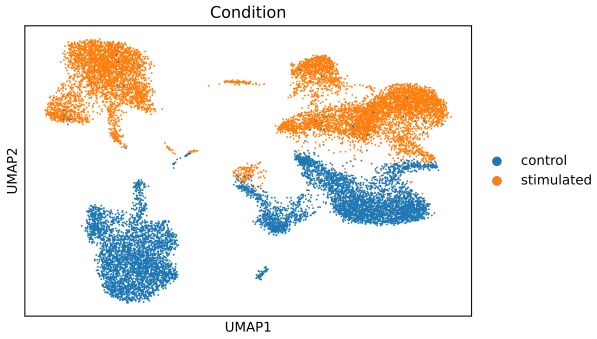

Figure B.1: **Reproduction of results in VEGA paper** — We reproduced the PBMCs analysis using VEGA and the Reactome pathways as the prior. The UMAP plot shows cells were nicely separated based on their treatment conditions.

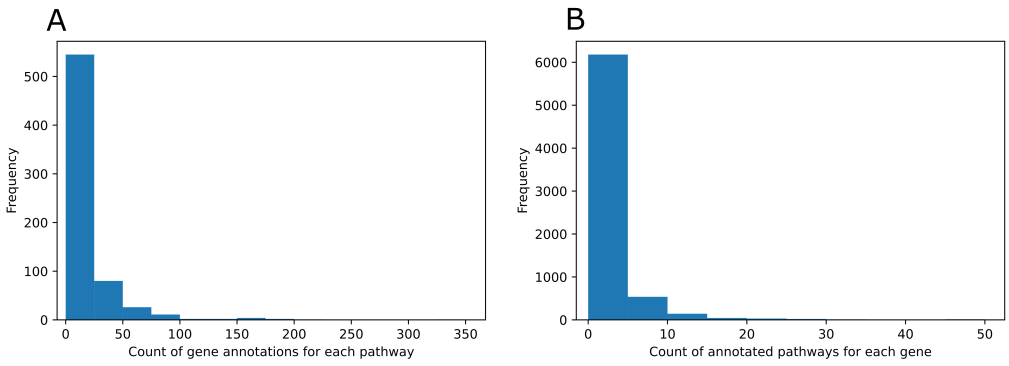

Figure B.2: The caption is on the next page page.

Figure B.2: **Outline of decoder wiring using PBMCs dataset** — **(A)** The x-axis indicates the number of genes each GMV connects to and the y-axis indicates the frequency of each case. We found that, unexpectedly, a high proportion of genes (5086 out of 6998) were not annotated in any of the Reactome pathways, which means the reconstructions of these 5086 genes were dependent only on one additional fully connected node in the latent space. **(B)** The x-axis indicates the number of GMVs each gene in the output layer connects to.

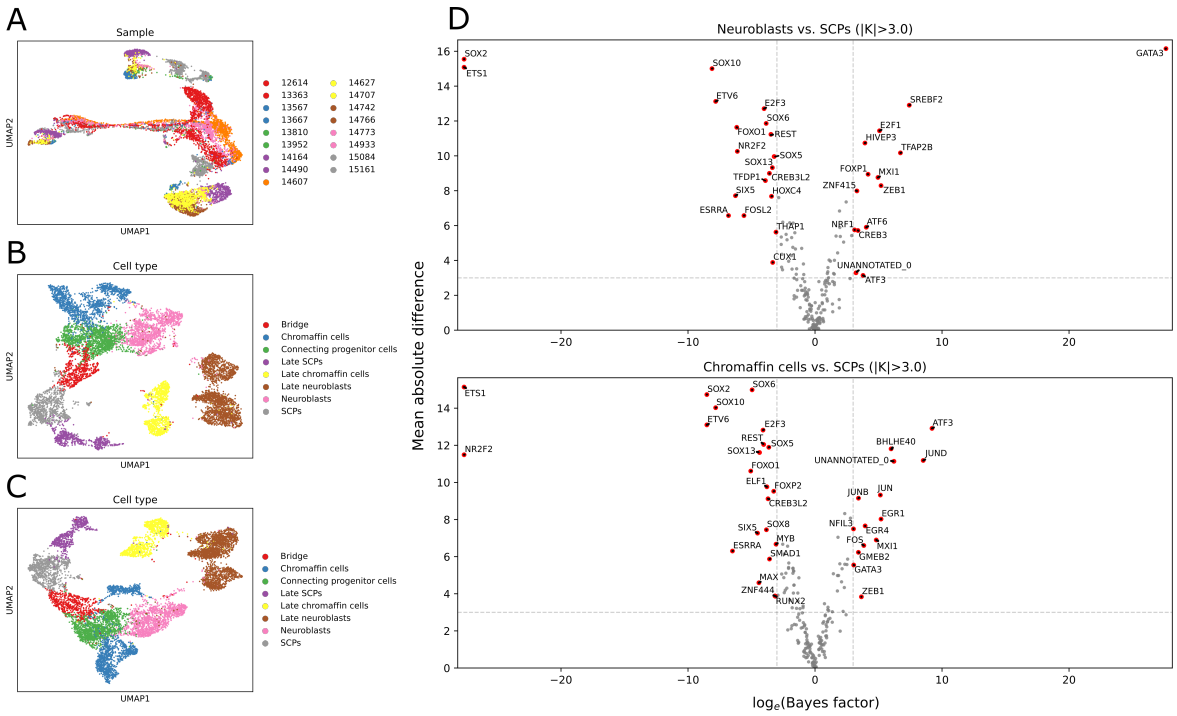

Figure B.3: **Benchmarks for VEGA** — We used the SCENIC regulons inferred from the human adrenal medulla dataset as the prior to analyze TF activities of human adrenal medullary cells. **(A)** The UMAP embedding of the gene expression space of the adrenal medulla dataset colored by samples supports that the cell clustering was based on biological differences between the cell types rather than technical differences between the samples. **(B,C)** The UMAP embeddings of the VEGA latent spaces from the models trained on the datasets with the whole gene features and with only those genes connected to the GMVs according to the prior regulons. The UMAP plots show that the model could cluster cells into cell types but the developmental trajectories of the adrenal medulla were not preserved as well as using the dataset with the top 2000 highly variable genes. **(D)** The x-axis and the y-axis indicate the significance level and the mean absolute difference of GMV activity comparisons between two cell types. These volcano plots annotate all significantly differential TF activities for Fig.2.3D for future use. We considered TFs to be significantly differentially activated when $|\log_e(\mathrm{BF})| > 3$.

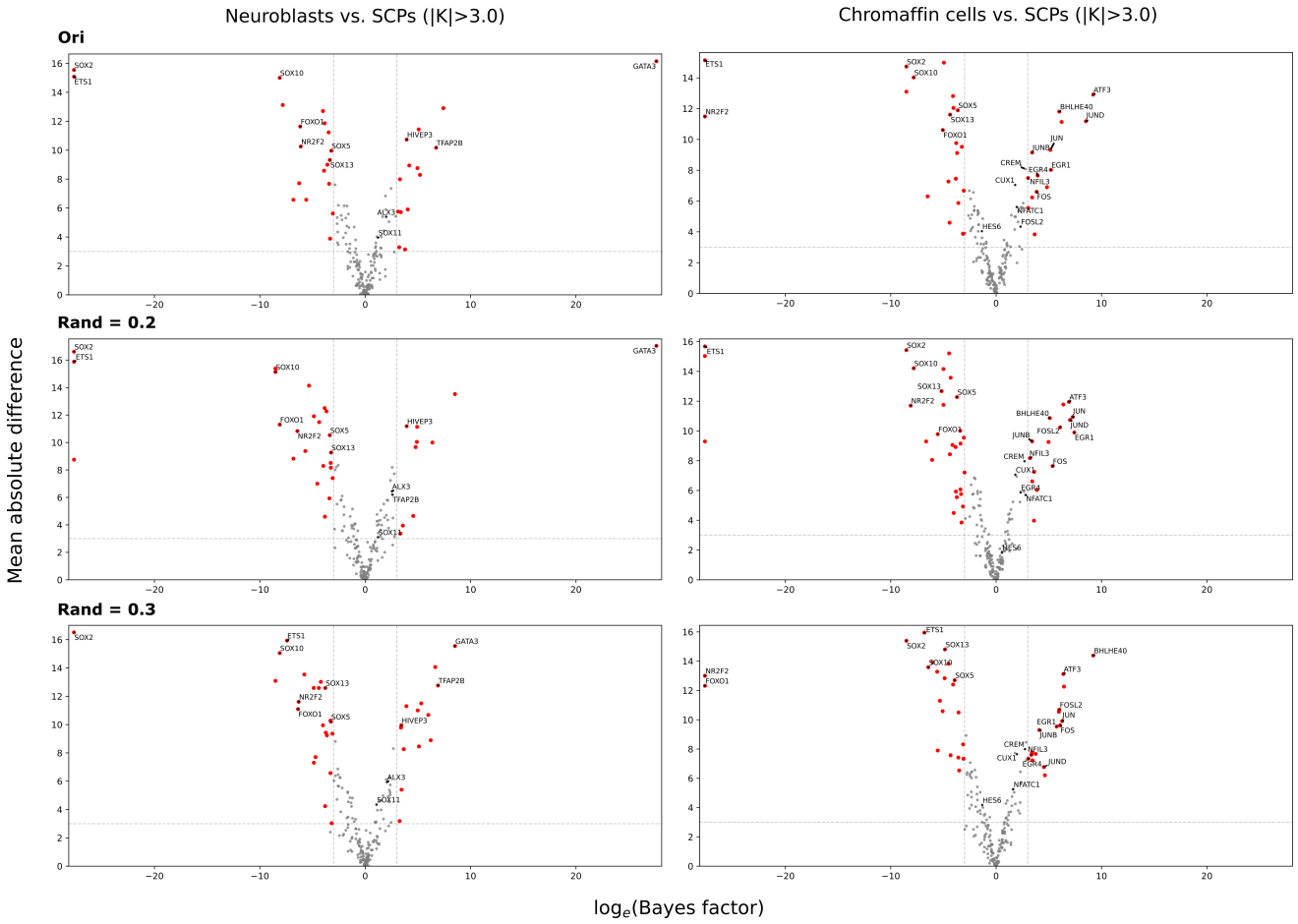

Figure B.4: **Changes in model interpretability when randomizing prior to different degrees** — The x-axis and the y-axis indicate the significance level and the mean absolute difference of GMV activity comparisons between two cell types. This figure is to complete the results of differential activity analysis on the VEGA embeddings using the prior which was randomized on different levels for Fig.2.6. Rand indicates the degree of randomization of the prior. We considered TFs to be significantly differentially activated when $|\log_e(\mathrm{BF})| > 3$. The rest of the results is on the next page.

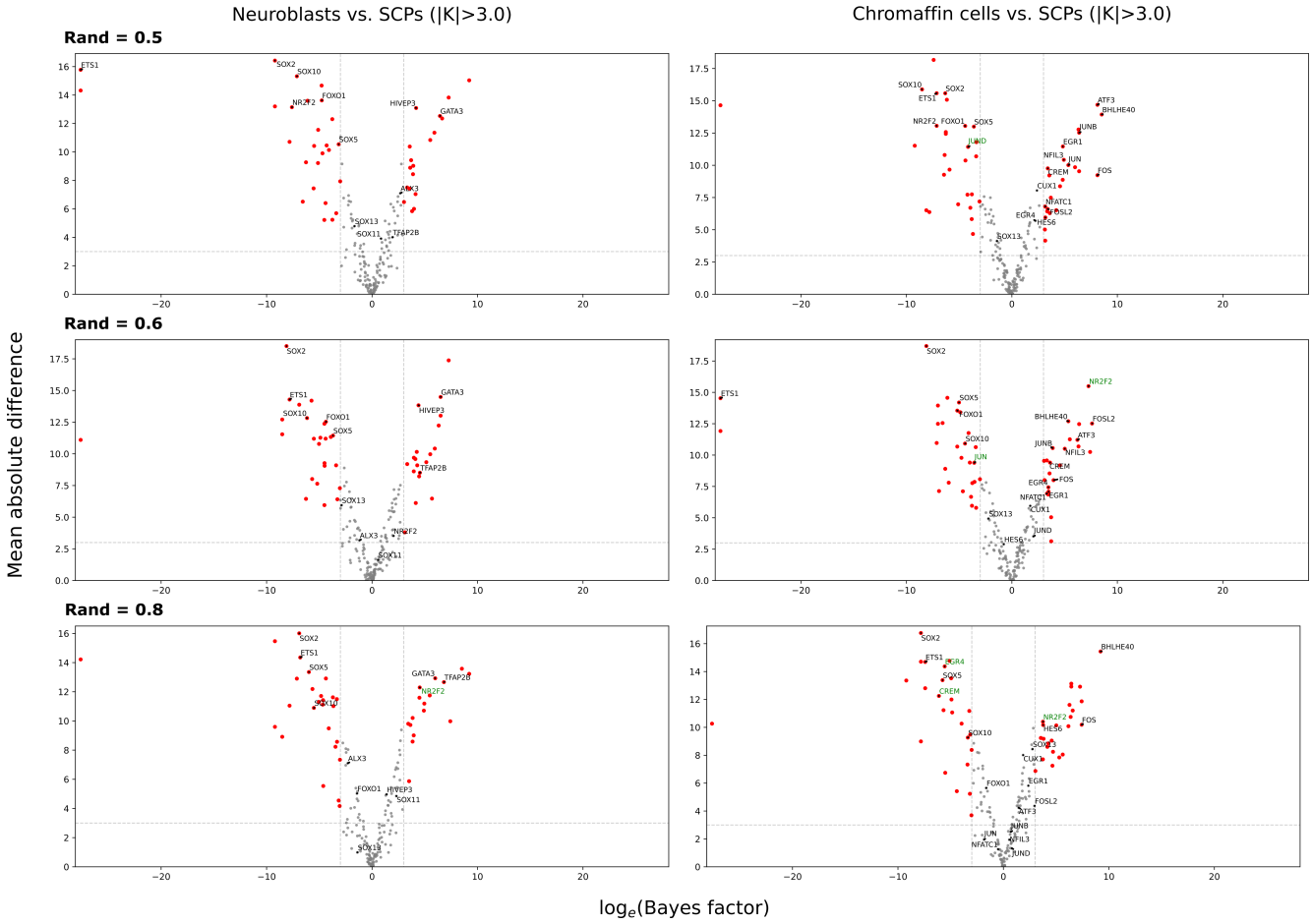

Figure B.4: This figure is continued from the previous page.

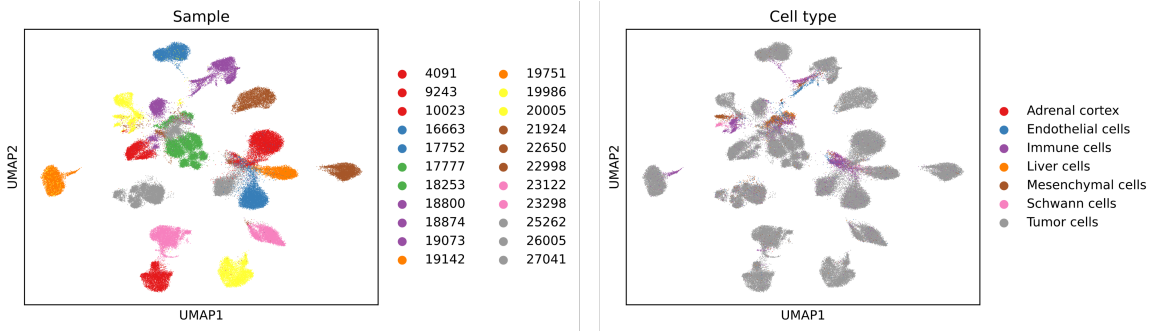

Figure B.5: **Testing of batch correction function of VEGA** — To verify that incorporating batch information in the model training is effective in alleviating batch effects, we trained the model with all the same settings as Fig.2.7B except incorporating the batch annotations. Without providing the batch annotations for the model training, the UMAP embedding of the VEGA latent space shows that cells were clustered mainly due to technical differences between the samples rather than biological differences between the cell types.

Figure B.6: **Investigation of regularized decoder behavior** — **(A)** The x-axis indicates the top 20 high-ranking genes (highest weights) of GATA3 from the hard-coded decoder of VEGA and the y-axis indicates the weight magnitude. This plot is to exhibit how we selected three artificially removed genes (prefixed by double asterisks). **(B)** The x-axis indicates the ranking of the weights of GATA3 and if the corresponding gene was annotated in the SCENIC GATA3 regulon, the frequency increased 1 in the y-axis. The recovery plot shows how the decoder behaved without using any prior knowledge (i.e., the decoder was fully connected). This is to support the finding that the regularized decoder behaves in a fully connected way when $\lambda\alpha$ is small. The red bar on a curve indicates the number of non-zero weights of GATA3 (i.e., the putative GATA3 regulon). Note that the zero-valued weights were randomly ranked. **(C)** The x-axis indicates the value of $\lambda\alpha$ and the y-axis indicates the number of non-zero weights of GATA3 (i.e., the putative GATA3 regulon). The plot shows that the number of putative target genes of GATA3 decreased when the model started using the prior evidently (from $\lambda\alpha = 10^{-3}$) and had a drastic drop when $\lambda\alpha$ was reaching 1. **(D,E)** Apart from GATA3, we also performed the same analyses on JUN to support the findings based on GATA3. Altogether, the results from JUN are very similar to those from GATA3, described in the caption of Fig.2.8.

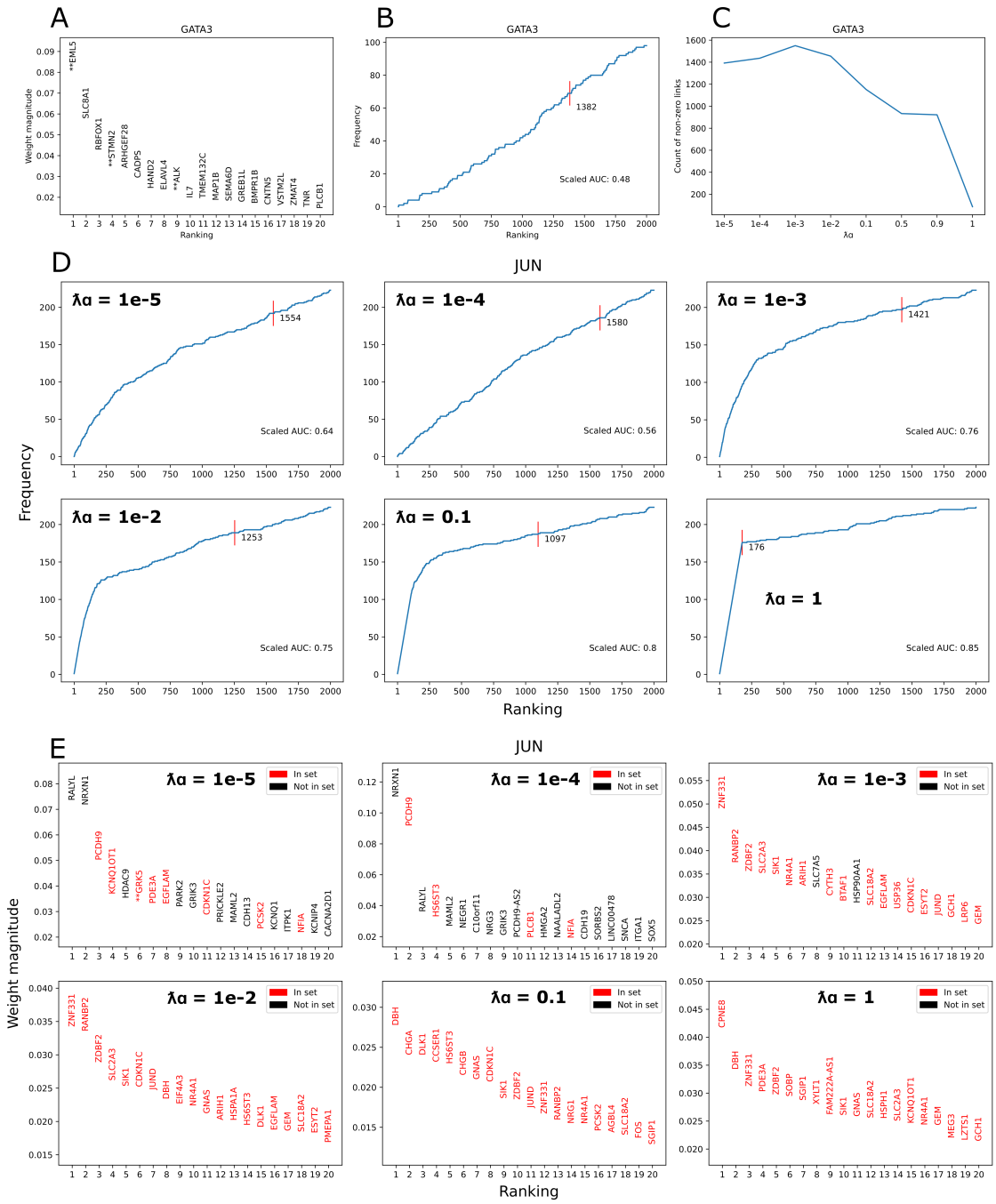

Figure B.6: The caption is on the previous page.

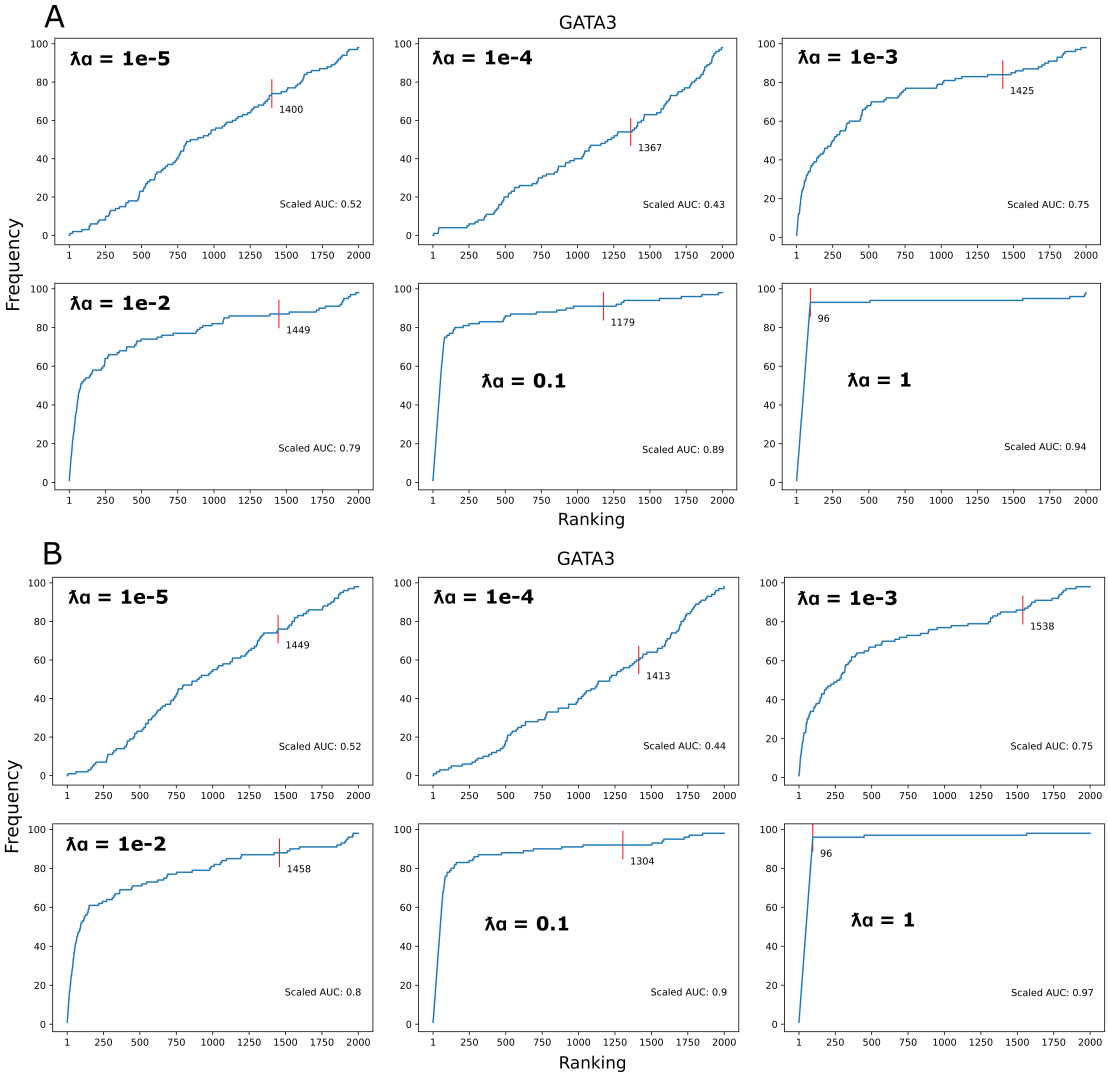

Figure B.7: **Results for supporting our findings on regularized decoder behavior** — The x-axis indicates the ranking of the weights of GATA3 and if the corresponding gene was annotated in the SCENIC GATA3 regulon, the frequency increased 1 in the y-axis. These recovery plots are to provide more evidence for our findings that the decoder behaves in a fully-connected way when $\lambda\alpha < 10^{-3}$, in a regularized way when $10^{-3} \leq \lambda\alpha < 1$, in a hard-coded way when $\lambda\alpha = 1$ (Fig.2.8A). **(A,B)** The difference between panel A and B lies in the SCENIC regulons we used as the prior where GATA3 was added with three genes with high average expression levels in panel A and with low average expression levels in panel B. The overall decoder behavior is not influenced by slightly artificially modified prior knowledge.

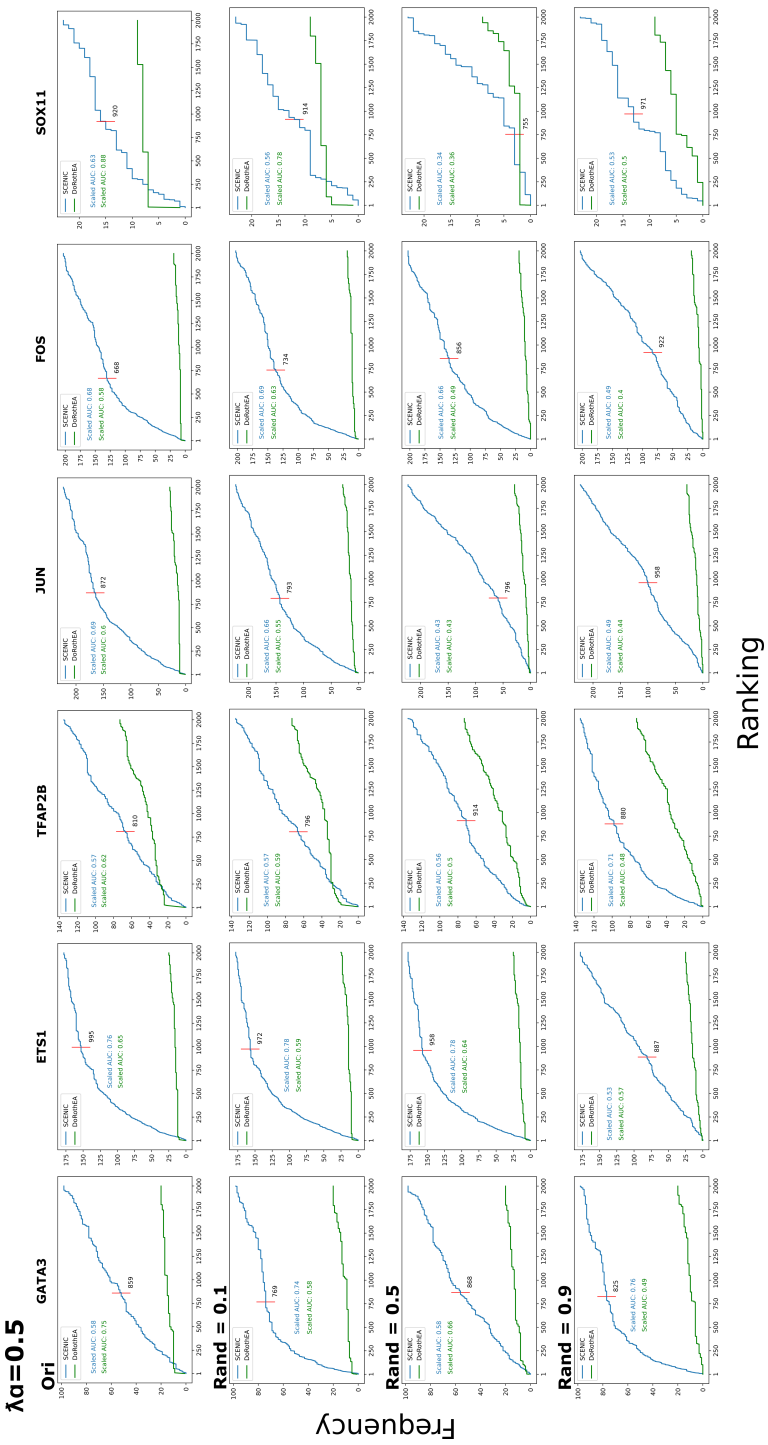

Figure B.8: The caption is on the next page.

Figure B.8: **Sanity check on prior knowledge randomization testing** — This figure is to provide more evidence for that the incorrect prior worsen the inference performance of the regularized decoder. Instead of using the model with $\lambda\alpha = 0.9$, we used the model with $\lambda\alpha = 0.5$ to investigate the changes in the dataset-specific GRN inferences when the non-context-specific DoRothEA regulons were randomized to varied degrees. The x-axis indicates the ranking of weights of a certain TF and if the corresponding gene was annotated in the SCENIC regulon (blue curves) or in the DoRothEA regulon (green curves) of the TF, the frequency increased 1 in the y-axis. Note that we will focus on blue curves for observing the inference capacity changes. The columns and the rows show the different TFs and the different degrees of randomization of the prior. Similarly to the model with $\lambda\alpha = 0.9$, there was no evident pattern of the changes in the inference performance when the prior was 10% and 50% randomized and most of the GRN inferences, except GATA3 and TFAP2B, were lost when the prior was 90% randomized. The red bar on a curve indicates the number of non-zero weights of a certain TF (i.e., the putative regulon of the TF). Note that the zero-valued weights were randomly ranked. Ori indicates the original prior and Rand indicates the degree of randomization of the prior.

Figure B.9: **Overview of inference capacity of regularized decoder across different** $\lambda\alpha$ — We trained the model with the regularized decoder on the human adrenal medulla dataset and using the non-context-specific DoRothEA regulons as prior knowledge to infer more dataset-specific regulons. We computed AUC scores on recovery curves generated from decoder weights of each GMV based on the SCENIC adrenal medulla regulons to study the inference capacity of the regularized decoder. We display the AUC scores of each GMV (TF) from the individual models with different $\lambda\alpha$ values via heatmaps to provide a general picture of the changes in the inference capacity across different values of $\lambda\alpha$ used. The column and the row indicate the $\lambda\alpha$ values and the TFs. The result shows that, with different $\lambda\alpha$ values, the overall inference capacity of the regularized decoder did not differ much. This was proved by computing the median of the AUC scores from each column, where $\lambda\alpha = 10^{-3}$ is 0.53, $\lambda\alpha = 10^{-2}$ is 0.52, $\lambda\alpha = 0.1$ is 0.53, $\lambda\alpha = 0.5$ is 0.53, $\lambda\alpha = 0.9$ is 0.51 and $\lambda\alpha = 1$ is 0.48. Besides, a majority of TFs were not precisely inferred to more dataset-specific regulons and there was no clear pattern of how the regularized decoder treats the individual TFs with different $\lambda\alpha$ values. Interestingly, the top white block containing zero-valued AUC scores including all GMVs without any connection to the genes in the output layer, which indicates the model with the regularized decoder can exclude these GMVs that may hold wrong information automatically.

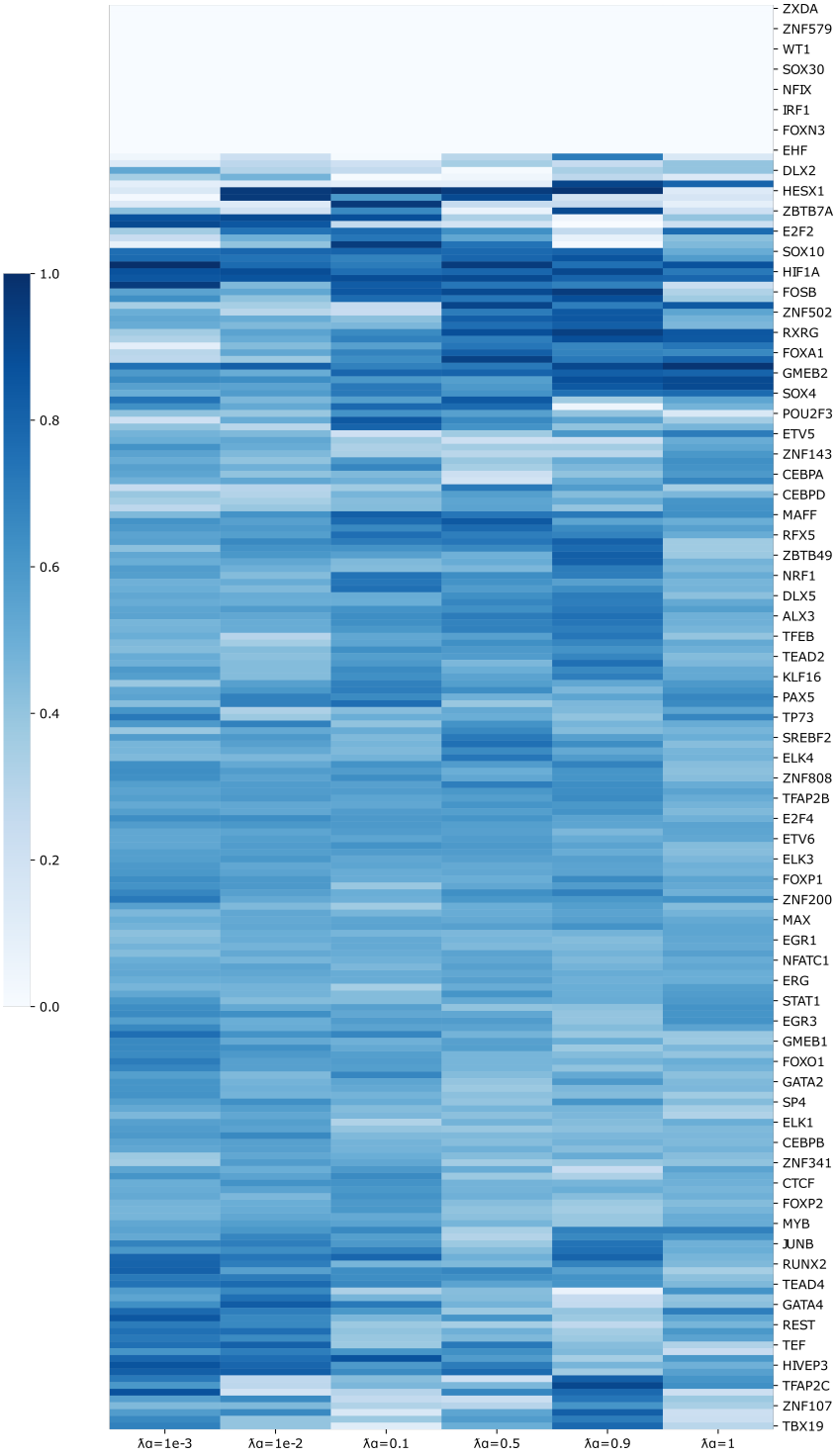

Figure B.9: The caption is on the previous page.

Bibliography

[1] Aleksandra A. Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C. Marioni, and Sarah A. Teichmann. The technology and biology of single-cell rna sequencing. *Molecular Cell*, 58:610–620, 2015.

[2] Scott J Emrich, W Brad Barbazuk, Li Li, and Patrick S Schnable. Gene discovery and annotation using lcm-454 transcriptome sequencing. *Genome Research*, 17:69–73, 2007.

[3] Ashraful Haque, Jessica Engel, Sarah A. Teichmann, and Tapio Lönnberg. A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. *Genome Medicine*, 9:75, 2017.

[4] Itay Tirosh, Benjamin Izar, Sanjay M Prakadan, Marc H Wadsworth 2nd, Daniel Treacy, John J Trombetta, Asaf Rotem, Christopher Rodman, Christine Lian, George Murphy, Mohammad Fallahi-Sichani, Ken Dutton-Regester, Jia-Ren Lin, Ofir Cohen, Parin Shah, Diana Lu, Alex S Genshaft, Travis K Hughes, Carly G K Ziegler, Samuel W Kazer, Aleth Gaillard, Kellie E Kolb, Alexandra-Chloé Villani, Cory M Johannessen, Aleksandr Y Andreev, Eliezer M Van Allen, Monica Bertagnolli, Peter K Sorger, Ryan J Sullivan, Keith T Flaherty, Dennie T Frederick, Judit Jané-Valbuena, Charles H Yoon, Orit Rozenblatt-Rosen, Alex K Shalek, Aviv Regev, and Levi A Garraway. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science*, 352:189–96, 2016.

[5] Michael J T Stubbington, Tapio Lönnberg, Valentina Proserpio, Simon Clare, Anneliese O Speak, Gordon Dougan, and Sarah A Teichmann. T cell fate and clonality inference from single-cell transcriptomes. *Nature Methods*, 13:329–332, 2016.

[6] Di Ran, Shanshan Zhang, Nicholas Lytal, and Lingling An. scdoc: correcting drop-out events in single-cell rna-seq data. *Bioinformatics*, 36:4233–4239, 2020.

[7] Hoa Thi Nhu Tran, Kok Siong Ang, Marion Chevrier, Xiaomeng Zhang, Nicole Yee Shin Lee, Michelle Goh, and Jinmiao Chen. A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome Biology*, 21:12, 2020.

[8] Serena Liu and Cole Trapnell. Single-cell transcriptome sequencing: recent advances and remaining challenges [version 1; peer review: 2 approved]. *F1000Research*, 5(F1000 Faculty Rev):182, 2016.

[9] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc.*, 374:20150202, 2016.

[10] Andres Quintero, Daniel Hübschmann, Nils Kurzawa, Sebastian Steinhauser, Philipp Rentzsch, Stephen Krämer, Carolin Andresen, Jeongbin Park, Roland Eils, Matthias Schlesner, and Carl Herrmann. Shinybutchr: Interactive nmf-based decomposition workflow of genome-scale datasets. *Biology Methods and Protocols*, pages 1–7, 2020.

[11] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.

[12] Dongfang Wang and Jin Gu. Vasc: Dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, Proteomics and Bioinformatics*, 16:320–331, 2018.

[13] Thomas A. Geddes, Taiyun Kim, Lihao Nan, James G. Burchfield, Jean Y. H. Yang, Dacheng Tao, and Pengyi Yang. Autoencoder-based cluster ensembles for single-cell rna-seq data analysis. *BMC Bioinformatics*, 20:660, 2019.

[14] Gökcen Eraslan, Lukas M. Simon, Maria Mircea, Nikola S. Mueller, and Fabian J. Theis. Single-cell rna-seq denoising using a deep count autoencoder. *Nature Communications*, 10:390, 2019.

[15] Alexander Van de Kleut. Variational autoencoders (vae) with pytorch, May 2020.

[16] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[17] Joseph Rocca and Baptiste Rocca. Understanding variational autoencoders (vaes), Sep 2019.

[18] Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15:1053–1058, 2018.

[19] Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16:715–721, 2019.

[20] Jianzhu Ma, Michael Ku Yu, Samson Fong, Keiichiro Ono, Eric Sage, Barry Demchak, Roded Sharan, and Trey Ideker. Using deep learning to model the hierarchical structure and function of a cell. *Nature Methods*, 15:290–298, 2018.

[21] Florian Buettner, Naruemon Pratanwanich, Davis J. McCarthy, John C. Marioni, and Oliver Stegle. f-sclvm: scalable and versatile factor analysis for single-cell rna-seq. *Genome Biology*, 18:212, 2017.

[22] Valentine Svensson, Adam Gayoso, Nir Yosef, and Lior Pachter. Interpretable factor models of single-cell rna-seq via variational autoencoders. *Bioinformatics*, 36:3418–3421, 2020.

[23] Marc Gillespie, Bijay Jassal, Ralf Stephan, Marija Milacic, Karen Rothfels, Andrea Senff-Ribeiro, Johannes Griss, Cristoffer Sevilla, Lisa Matthews, Chuqiao Gong, Chuan Deng, Thawfeek Varusai, Eliot Ragueneau, Yusra Haider, Bruce May, Veronica Shamovsky, Joel Weiser, Timothy Brunson, Nasim Sanati, Liam Beckman, Xiang Shao, Antonio Fabregat, Konstantinos Sidiropoulos, Julieth Murillo, Guilherme Viteri, Justin Cook, Solomon Shorser, Gary Bader, Emek Demir, Chris Sander, Robin Haw, Guanming Wu, Lincoln Stein, Henning

Hermjakob, and Peter D'Eustachio. The reactome pathway knowledgebase 2022. *Nucleic Acids Research*, 50:D687–D692, 2022.

[24] Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P. Mesirov, and Pablo Tamayo. The molecular signatures database (msigdb) hallmark gene set collection. *Cell Systems*, 1:417–425, 2015.

[25] Sara Aibar, Carmen Bravo González-Blas, Thomas Moerman, Vân Anh Huynh-Thu, Hana Imrichova, Gert Hulselmans, Florian Rambow, Jean-Christophe Marine, Pierre Geurts, Jan Aerts, Joost van den Oord, Zeynep Kalender Atak, Jasper Wouters, and Stein Aerts. Scenic: single-cell regulatory network inference and clustering. *Nature Methods*, 14:1083–1086, 2017.

[26] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7:S7, 2006.

[27] Lucas Seninge, Ioannis Anastopoulos, Hongxu Ding, and Joshua Stuart. Vega is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nature Communications*, 12:5684, 2021.

[28] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 78, New York, NY, USA, 2004. Association for Computing Machinery.

[29] Sergei Rybakov, Mohammad Lotfollahi, Fabian J. Theis, and F. Alexander Wolf. Learning interpretable latent autoencoder representations with annotations of feature sets. *bioRxiv*, 2020.

[30] Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, Rachel E Gate, Sara Mostafavi, Alexander Marson, Noah Zaitlen, Lindsey A Criswell, and Chun Jimmie Ye. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature Biotechnology*, 36(1):89–94, 2018.

[31] Bijay Jassal, Lisa Matthews, Guilherme Viteri, Chuqiao Gong, Pascual Lorente, Antonio Fabregat, Konstantinos Sidiropoulos, Justin Cook, Marc Gillespie, Robin Haw, Fred Loney, Bruce May, Marija Milacic, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The reactome pathway knowledgebase. *Nucleic Acids Research*, 48(D1):D498–D503, 2020.

[32] Selina Jansky, Ashwini Kumar Sharma, Verena Körber, Andrés Quintero, Umut H. Toprak, Elisa M. Wecht, Moritz Gartlgruber, Alessandro Greco, Elad Chomsky, Thomas G. P. Grünewald, Kai-Oliver Henrich, Amos Tanay, Carl Herrmann, Thomas Höfer, and Frank Westermann. Single-cell transcriptomic analyses provide insights into the developmental origins of neuroblastoma. *Nature Genetics*, 53:683–693, 2021.

[33] Luz Garcia-Alonso, Christian H. Holland, Mahmoud M. Ibrahim, Denes Turei, and Julio Saez-Rodriguez. Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome Research*, 29:1363–1375, 2019.

[34] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *Preprint at arXiv*, 2020.

[35] Candice Mazewski, Ricardo E. Perez, Eleanor N. Fish, and Leonidas C. Platanias. Type i interferon (ifn)-regulated activation of canonical and non-canonical signaling pathways. *Frontiers in Immunology*, 11:606456, 2020.

[36] Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:773–795, 1995.

[37] Leonhard Held and Manuela Ott. On p-values and bayes factors. *Annual Review of Statistics and Its Application*, 5(1):393–419, 2018.

[38] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1:127–239, 2014.

[39] Chenchen Feng, Chao Song, Yuejuan Liu, Fengcui Qian, Yu Gao, Ziyu Ning, Qiuyu Wang, Yong Jiang, Yanyu Li, Meng Li, Jiaxin Chen, Jian Zhang, and Chunquan Li. Knocktf: a comprehensive human gene expression profile database with knockdown/knockout of transcription factors. *Nucleic Acids Research*, 48:D93–D100, 2020.

[40] ENCODE Project Consortium. The encode (encyclopedia of dna elements) project. *Science*, 306:636–40, 2004.

[41] ENCODE Project Consortium. A user's guide to the encyclopedia of dna elements (encode). *PLOS Biology*, 9:e1001046, 2011.

[42] Andrew D. Rouillard, Gregory W. Gundersen, Nicolas F. Fernandez, Zichen Wang, Caroline D. Monteiro, Michael G. McDermott, and Avi Ma'ayan. The harmonizome: a collection of processed datasets gathered to serve and mine knowledge about genes and proteins. *Database*, 2016, 2016. baw100.

[43] Alexander Lachmann, Huilei Xu, Jayanth Krishnan, Seth I Berger, Amin R Mazloom, and Avi Ma'ayan. Chea: transcription factor regulation inferred from integrating genome-wide chip-x experiments. *Bioinformatics*, 26:2438–44, 2010.

[44] Victor Cumer. The basics of monte carlo integration, Oct 2020.

[45] James M. Joyce. *Kullback-Leibler Divergence*, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[46] Arxiv Insights. Variational autoencoders. `https://www.youtube.com/watch?v=9zKuYvjFFS8&t=520s`, Feb 2018.

[47] math et al. Naive monte carlo integration + r demo. `https://www.youtube.com/watch?v=S11p4KAHdcQ`, Oct 2021.

[48] F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:15, 2018.

[49] 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*, 177:1888–1902.E21, 2019.

[50] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck, Shiwei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P. Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M. Fleming, Bertrand Yeung, Angela J. Rogers, Juliana M. McElrath, Catherine A. Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, 184:3573–3587.e29, 2021.

[51] 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*, 16:1289–1296, 2019.

[52] Isaac Virshup, Sergei Rybakov, Fabian J. Theis, Philipp Angerer, and F. Alexander Wolf. anndata: Annotated data. *Journal of Open Source Software*, 2021.

[53] Vladimir Kiselev and Ni Huang. A package to help convert different single-cell data formats to each other, 2020.

[54] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[55] Mohammad Lotfollahi, Sergei Rybakov, Karin Hrovatin, Soroor Hediyeh-zadeh, Carlos Talavera-López, Alexander V Misharin, and Fabian J. Theis. Biologically informed deep learning to infer gene program activity in single cells. *bioRxiv*, 2022.

[56] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, page 1–18, 2019.

[57] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[58] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

[59] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[60] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[61] Jeff Reback, jbrockmendel, Wes McKinney, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, Simon Hawkins, gfyoung, Matthew Roeschke, Sinhrks, Adam Klein, Terji Petersen, Jeff Tratner, Chang She, William Ayd, Patrick Hoefler, Shahar Naveh, Marc Garcia, Jeremy Schendel, Andy Hayden, Daniel Saxton, JHM Darbyshire, Richard Shadrach, Marco Edward Gorelli, Fangchen Li, Vytautas Jancauskas, Ali McMaster, Matthew Zeitlin, Pietro Battiston, and Skipper Seabold. pandas-dev/pandas: Pandas 1.3.3, September 2021.

[62] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[63] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[64] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[65] Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, Yining Liu, Jules Samaran, Gabriel Misrachi, Achille Nazaret, Oscar Clivio, Chenling Xu, Tal Ashuach, Mariano Gabitto, Mohammad Lotfollahi, Valentine Svensson, Eduardo da Veiga Beltrame, Vitalii Kleshchevnikov, Carlos Talavera-López, Lior Pachter, Fabian J. Theis, Aaron Streets, Michael I. Jordan, Jeffrey Regier, and Nir Yosef. A python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, Feb 2022.

[66] neptune.ai. Neptune: experiment management and collaboration tool, 2020.