SCIENTIFIC REPORTS
nature research

OPEN

Unsupervised generative and graph representation learning for modelling cell differentiation

Ioana Bica[1,2,4,7] ✉, Helena Andrés-Terré[2,7] ✉, Ana Cvejic[3,5,6,8] ✉ & Pietro Liò[2,8]

Using machine learning techniques to build representations from biomedical data can help us understand the latent biological mechanism of action and lead to important discoveries. Recent developments in single-cell RNA-sequencing protocols have allowed measuring gene expression for individual cells in a population, thus opening up the possibility of finding answers to biomedical questions about cell differentiation. In this paper, we explore unsupervised generative neural methods, based on the variational autoencoder, that can model cell differentiation by building meaningful representations from the high dimensional and complex gene expression data. We use disentanglement methods based on information theory to improve the data representation and achieve better separation of the biological factors of variation in the gene expression data. In addition, we use a graph autoencoder consisting of graph convolutional layers to predict relationships between single-cells. Based on these models, we develop a computational framework that consists of methods for identifying the cell types in the dataset, finding driver genes for the differentiation process and obtaining a better understanding of relationships between cells. We illustrate our methods on datasets from multiple species and also from different sequencing technologies.

As technology continues to drive biomedical research forward, new challenges arise with the surge of high volume, information-dense and multivariate data that are generated. The extraction of critical information from such data remains an open problem in biomedical research, which can be significantly aided by the incorporation of machine learning techniques. In particular, unsupervised learning methods have the potential to uncover the underlying structure in biomedical data and therefore propel research on biological processes and diseases that have not yet been fully understood.

This paper aims to build unsupervised neural methods that can be applied to understand cell differentiation using gene expression data. Recent technology for performing single-cell RNA sequencing has resulted in high-throughput experiments capable of measuring gene expression levels for individual cells in a population, thus achieving a granularity not previously possible. In-depth analysis of this high-dimensional and complex gene expression data about the cells can lead to important biomedical discoveries about the factors influencing the differentiation process. However, gene expression data is in general high dimensional, as there are thousands of gene expression measurements for each cell, and very complex.

We propose using a disentangled generative probabilistic framework to model single-cell RNA sequencing gene expression data and build a low dimensional representation that can help us discovering latent biological mechanisms. In this framework, we develop novel methodology that can be used to identify the different cell types in such single-cell RNA-seq datasets using the learned latent representations. We show that we can correctly identify the different cell types in two datasets: one dataset consisting of hematopoietic stem and differentiated cells in zebrafish obtained using Smart-Seq2[1], and another dataset consisting of humans pancreatic cells obtained using CEL-Seq2[2]. We also show and discuss some limitations of our methods on a dataset with human

[1]Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ, United Kingdom. [2]Department of Computer Science and Technology, University of Cambridge, Cambridge, CB3 0FD, United Kingdom. [3]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Cambridge, CB10 1SA, United Kingdom. [4]The Alan Turing Institute, London, NW1 2DB, United Kingdom. [5]Department of Haematology, University of Cambridge, Cambridge, CB2 0XY, United Kingdom. [6]Wellcome Trust - Medical Research Council Stem Cell Institute, Cambridge, CB2 0AW, United Kingdom. [7]These authors contributed equally: Ioana Bica and Helena Andrés-Terré. [8]These authors jointly supervised this work: Ana Cvejic and Pietro Liò. ✉e-mail: ioana.bica@eng.ox.ac.uk; ha376@cam.ac.uk; as889@cam.ac.uk

hematopoietic cells[3]. In addition, we explore performing perturbations to the latent representation to study how the stem or progenitor cells can be changed into differentiated cells. Moreover, we propose a graph representation learning method based on an autoencoder consisting of graph convolutional layers that can be used to analyze links between single cells.

Current methods for analysing single-cell RNA-seq data are based on the combination of dimensionality reduction techniques and clustering algorithms, at either gene or cell level analysis. The identification of cell lineages and trajectories is one of the main fields of study where single-cell scRNA-seq has had a great influence. The most widespread computational tools include Waterfall or Wishbone[4,5], which are based on principal component analysis (PCA). Monocle uses independent component analysis (ICA) and SCUBA pseudotime focuses on t-distributed stochastic neighbour embedding (tSNE)[6,7]. However, some of these methods, particularly the ones based on linear approaches such as PCA, are not able to capture the complex relationships between the input dimensions and can disregard meaningful information within the data. In addition, Yeung and Ruzzo[8] showed that using PCA before clustering gene expression data has a negative effect on the quality of the clusters. Despite these findings, a lot of research in gene expression analysis[9–11] is based on applying PCA before clustering cells to identify their types.

Autoencoders can be used to perform non-linear dimensionality reduction, but also to extract biologically relevant latent features from transcriptomics data. Related work shows the effectiveness of these models in analysing gene expression data. Way and Greene[12] trained a variational autoencoder on pan-cancer RNA-seq data from The Cancer Genome Atlas[13] to explore the biological relevance of the latent space produced by the autoencoder. Tan *et al.*[14] built a denoising autoencoder capable of modelling the response of cells to low oxygen and finding differences between strains in gene expression from Pseudomonas aeruginosa. Eraslan *et al.*[15] used autoencoders for denoising purposes, developing a method that is linearly scalable with the number of cells and outperforms existing methods for data imputation. Talwar *et al.*[16] proposed an autoencoder-based method to perform gene expression imputation, while Wang and Gu[17] use variational autoencoders for dimensionality reduction and visualization of single-cell data. Finally, Rashid *et al.*[18] used a variational autoencoder to identify tumour subpopulations, marker genes, as well as differentiation trajectories for the malignant cells using scRNA-seq genomic data. Compared to our proposed models, these methods do not use a training objective to enforce disentanglement in the latent representation[12,18] and focus on different applications such as denoising[14,15], visualization[17] and missing data imputation[16].

This work represents the first application, to the best of our knowledge, of disentanglement, perturbation and graph-based methods for variational autoencoders with the aim of analysing cell differentiation using single-cell RNA-seq data. We emphasise the importance of building interpretable models, by analysing the relationship between the embedding and gene expression spaces. We also explore the robustness and variability of the latent space by introducing perturbations. Graph representation learning represents a new powerful generation of methodologies for graphs. We show how predicting links between cells can provide insights into differentiation trajectories.

Disentangled generative probabilistic framework

We propose using a generative probabilistic framework[19] to model the biological processes that lead to the changes in the observed gene expression for cells at different stages in the differentiation process. Let $\mathscr{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ be a high-dimensional single-cell RNAseq dataset consisting of the gene expression of N i.i.d cells. Each gene expression vector $\mathbf{x}^{(i)}$ is an observation from a continuous random variable $\mathbf{x}$, having distribution $p_{data}(\mathbf{x})$. The gene expression data is assumed to be generated by some random process, modelled by an unobserved continuous random variable $\mathbf{z}$ with parametrised prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$. The marginal likelihood $\boldsymbol{p_\theta}(\mathbf{x})$, also known as the evidence, is computed by integrating over the possible latent representations:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbf{z}\in\mathscr{Z}} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\mathrm{d}\mathbf{z} = \int_{\mathbf{z}\in\mathscr{Z}} p_{\theta}(\mathbf{x}|\mathbf{z})p_{\theta}(\mathbf{z})\mathrm{d}\mathbf{z}. \tag{1}$$

Computing the integral involves spanning the space of values for $\mathbf{z}$ which is often intractable. For inference, the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = (p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\theta}(\mathbf{z}))/p_{\boldsymbol{\theta}}(\mathbf{x})$ has to be computed, which is also intractable, as it requires the marginal likelihood.

To learn in such a framework we use variational inference and we approximate the posterior using the variational distribution $\boldsymbol{q}_{\phi}(\mathbf{z}|\mathbf{x})$. We thus build a variational autoencoder model[19] and we use a multivariate Gaussian $\mathscr{N}(\mathbf{z};\ \mu,\ \mathrm{diag}(\boldsymbol{\sigma}^2))$ distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ to approximate $\mathbf{q}_{\phi}(\mathbf{z}|\mathbf{x})$. An encoder neural network is trained to estimate $\boldsymbol{q}_{\phi}(\mathbf{z}|\mathbf{x})$. In addition, an isotropic multivariate Gaussian prior is assigned to the latent representation: $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathscr{N}(\mathbf{z};\ \mathbf{0},\ \mathbf{I})$. The decoder neural network is trained to reconstruct (generate) the gene expression data from the latent representation and thus estimate $\boldsymbol{p_\theta}(\mathbf{x}|\mathbf{z})$. See Fig. 1a for a graphical illustration of the model. The training objective of the standard variational autoencoder model[19] penalises the mutual information between the input gene expression and the latent representation[20] and it also does not encourage disentaglement in the latent representation[21]. Disentanglement is desirable in our case because, ideally, the latent representation $\mathbf{Z}$ should be able to separate the biological factors that have led to the development of various cell types.

We introduce DiffVAE, a variational autoencoder that can be used to model and study the differentiation of cells using gene expression data. DiffVAE is an MMD-VAE, part of the InfoVAE family of autoencoders[21] and is trained to maximize the following objective:

$$\mathscr{L}_{\mathrm{DiffVAE}}(\boldsymbol{\theta},\ \phi;\ \mathbf{x}) = \mathbb{E}_{q_{\phi(\mathbf{z}|\mathbf{x})}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \mathrm{MMD}(q_{\phi}(\mathbf{z})\|p_{\boldsymbol{\theta}}(\mathbf{z})), \tag{2}$$

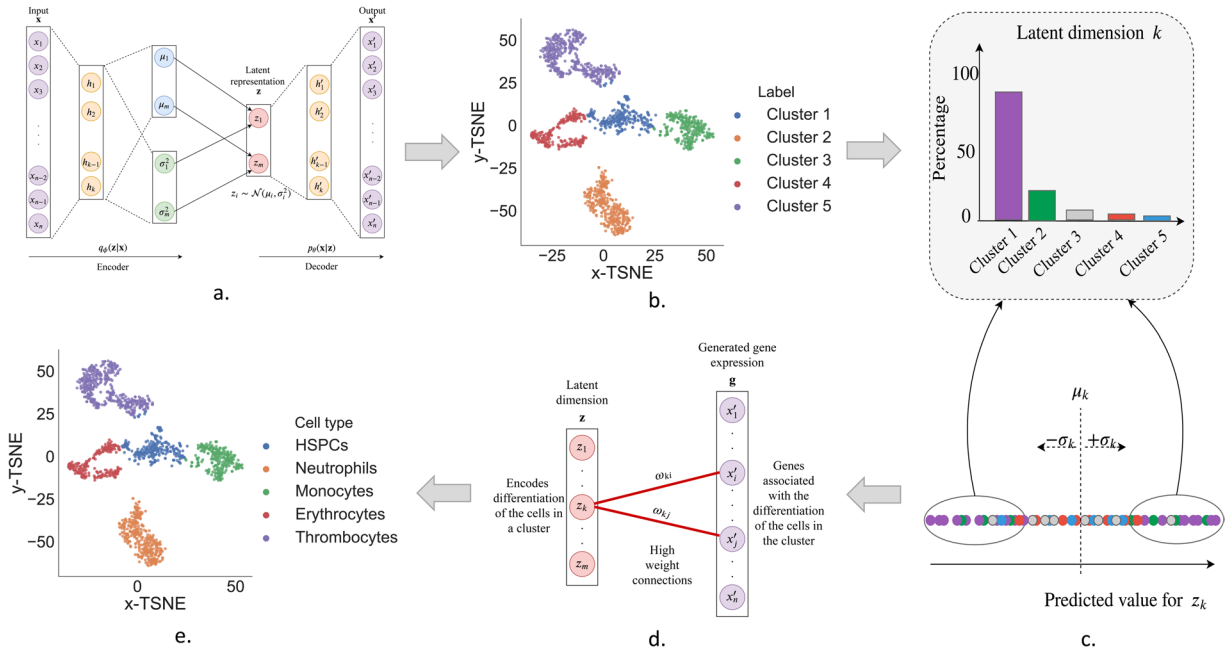


Figure 1. Pipeline for identifying the cell types in a dataset using DiffVAE. Illustration on the zebrafish dataset. (**a**) Train DiffVAE to map the gene expression measurements for each cell to a m-dimensional latent representation $\mathbf{z}$. (**b**) Apply T-SNE on the latent representation $\mathbf{z}$ and clustering to find the different cell clusters in the dataset. (**c**) Identify which latent dimensions in $\mathbf{z}$ encode the differentiation of the cells in each cluster. (**d**) Find the high weights genes for the relevant latent dimensions. (**e**) Map the clusters to cell types based on the high weight genes for each cluster.

where $\mathbb{E}_{q_{\phi(\mathbf{z}|\mathbf{x})}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]$ represents the reconstruction accuracy and the maximum mean discrepancy (MMD)[22–24] divergence between $\boldsymbol{q}_{\phi}(\mathbf{z})$ and $\boldsymbol{p}_{\boldsymbol{\theta}}(\mathbf{z})$ measures how different the moments of two probability distributions are. The intuition behind the MMD divergence is given by the fact that two probability distributions are identical if and only if their moments match. Zhao *et al.*[21] prove that using this training objective will always prefer to maximizes mutual information between the input and the latent representation. Moreover, minimising the divergence $MMD(\boldsymbol{q}_{\phi}(\mathbf{z})||\boldsymbol{p}_{\boldsymbol{\theta}}(\mathbf{z}))$, will encourage $\boldsymbol{q}_{\phi}$ to be similar to the prior $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ with diagonal covariance matrix, which will lead to disentanglement in the latent dimension. Using the MMD to measure discrepancy between distributions achieves best performance in the InfoVAE model family[21]. The MMD also achieved good results when used in the training objective of other autoencoder models[23,25,26]. For further analysis we use as the latent representation $\mathbf{z}$ the mean $\boldsymbol{\mu}$ of the distribution $q(\mathbf{z}|\mathbf{x})$ learnt by DiffVAE. The reason for this choice is the fact that the mean of the distribution $q(\mathbf{z}|\mathbf{x})$ represents the maximum likelihood estimate of the latent distribution learnt by DiffVAE for the cells in the dataset.

The DiffVAE model consists of two fully connected layers in the decoder and encoder networks. See the Methods section for more details about the DiffVAE model. The models in this paper were implemented in Python using Keras[27].

Identifying cell types using DiffVAE

Data details and pre-processing. The unsupervised models and the methodology developed in this paper are used to analyse single-cell gene expression data from hematopoietic stem and differentiated cells in zebrafish[1] and in human[3] and also from human pancreatic cells[2]. Let a scRNA-seq dataset be denoted as $\mathscr{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$, where $\mathbf{x}^{(i)} = [x_1^{(i)} x_2^{(i)} \dots x_k^{(i)}]^T$ consists of the transcriptomics data for cell (i). The zebrafish dataset consists of $k = 1845$ gene expression measurements from $N = 1422$ cells. We used the same 1845 genes identified by[1] to be the most highly variable ones among the 1422 zebrafish single cells. The dataset with human pancreatic cells consists of $N = 2285$ cells with measurements from the $k = 4000$ most highly variable genes. The dataset with human hematopoietic cells contains $N = 1034$ cells with $k = 700$ measurements from the most variable genes. In all cases, we consider that the cell states are initially unknown and we show how the methodology developed in this paper can be used to identify them. Note that some of the results on these datasets are also presented in the supplementary materials.

The transcriptomics data used is log-normalized. However, to use the transcriptomics data as input to DiffVAE, we performed additional normalization through Min-Max scaling such that the expression values for the genes were scaled to the range [0, 1]. This way we model the gene expression for each cell as a multivariate Bernoulli distribution in our probabilistic framework.

Cluster 1 (HSPCs)	Cluster 2 (Neutrophils)	Cluster 3 (Monocytes)	Cluster 4 (Erythrocytes)	Cluster 5 (Thrombocytes)
si:ch211-161c3.6[1], *cad*[1,40], *pcna*[40]	*illr4*[1], *ponzr6*, *npsn*[41], *abcb9*[42], *lyz*[43]	*lgals2a*[1], *c1qc*, *c1qa*[1], *s100a10b*, *mafbb*[44,45]	*alas2*[1], *ba1l*[1], *aqp1a.1*[1], *hbaa1*[1], *slc4a1a*[46,47], *ba1*[1], *si:xx-by187g17.1*	*fn1b*[1], *itga2b*[1,48], *bmp6*, *thbs1b*[1], *fhl1a*, *ctgfa*, *apln*

Table 1. Zebrafish. High weight genes computed using the high weight connections to the latent dimensions with the highest percentage for differentiating the corresponding cell type. Using references from scientific literature each cluster found using DiffVAE is mapped to a cell type.

Pipeline for identifying the cell types. In this section, we describe how DiffVAE can be used to find the different cell types in each dataset. Figure 1 shows the methodological pipeline for this process, with the specific details described in further subsections.

Using DiffVAE to obtain cell clusters. DiffVAE was trained to map the gene expression data for the single cells to a latent representation of m dimensions (Fig. 1a). For the datasets with hematopoietic cells (both zebrafish and human), we used $m = 50$ latent dimensions, while for the dataset with the human pancreatic cells, we used $m = 100$ latent dimensions. The large number of latent dimensions is needed to capture the complex biological processes influencing cell differentiation. To visualize the data and identify the different cells, we further use t-Distributed Stochastic Neighbour Embedding (t-SNE)[28] to obtain a 2-dimensional embedding for each cell. In the zebrafish dataset, K-means clustering is applied to the t-SNE embedding to obtain 5 cell clusters (Fig. 1b). In the datasets with human cells, we used DBSCAN clustering. We further develop the methodology for mapping each cluster to a cell type.

Latent dimensions encoding cell differentiation. DiffVAE was designed to model the data generating process giving rise to the observations in our dataset $\mathscr{D}$. Thus, this method should be able to identify the biological mechanisms that result in the observed gene expression value for our cells. Consider the analysis of a latent dimension k for any of the models. Let $\mathbf{z}_k = [z_k^{(1)} z_k^{(2)} \dots z_k^{(N)}]^T$ be the predicted value of the encoder for z_k across all of the cells in the dataset. Let $\boldsymbol{\mu}_k$ and σ_k be the mean and standard deviation of $\mathbf{z}_k$. We define:

$$\mathscr{D}_k = \{\mathbf{x}^{(i)} \in \mathscr{D} | z_k^{(i)} \geq \mu_k + \sigma \vee z_k^{(i)} \leq \mu_k - \sigma\} \tag{3}$$

as the set of cells at least a standard deviation from the mean in latent dimension k. By computing the percentage distribution of the cells in $\mathscr{D}_k$ across the distinct cell clusters found in the dataset, we can evaluate how well the latent dimension is encoding the differentiation of the cells in a particular cluster (Fig. 1c). Thus, for each cluster C we compute the percentage of cells from cluster C in each of $\mathscr{D}_k$, $k \in \{1,2,\dots,50\}$. The latent dimensions relevant for the differentiation of cells in cluster C will be the ones with the top 10 highest percentage of cells from cluster C in $\mathscr{D}_k$.

Identifying high weight genes. The decoder in DiffVAE learns to reconstruct the original gene expression data, and therefore, the weights in the decoder indicate the contribution of each gene in the biological process. By finding the high weight connections between the latent dimensions relevant for each cell cluster and the reconstructed gene expression, we can identify the marker genes for each cell cluster. This will help us identify the cell types.

The high weight connections can be obtained using the weight matrices in the decoder. The decoder consists of a two fully connected layers. Let $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{h}^{(1)} \in \mathbb{R}^{n_1}$, $\mathbf{h}^{(2)} \in \mathbb{R}^{n_2}$, $\mathbf{x}' \in R^n$, be the sequence of layer activations in the decoder, where the latent dimension $\mathbf{z}$ represents the input, $\mathbf{h}^{(1)}$, $\mathbf{h}^{(2)}$ are the hidden layers and $\mathbf{x}'$ is the output. The weight matrices for the connections between the layers in the decoder can be described by $\mathbf{W}^{(0)} \in \mathbb{R}^{m \times n_1}$, $\mathbf{W}^{(1)} \in \mathbb{R}^{n_1 \times n_2}$, $\mathbf{W}^{(2)} \in \mathbb{R}^{n_2 \times n}$. Let $\boldsymbol{\omega} \in \mathbb{R}^{m \times n}$ be the weight matrix for the connections between the latent dimension and the output. ω can be computed by multiplying the weight matrices between the individual fully connected layers, as follows: $\omega = \mathbf{W}^{(0)} \cdot \mathbf{W}^{(1)} \cdot \mathbf{W}^{(2)}$, where the matrix element ω_{ij} indicates the weight of the connection between latent dimension i and gene j. For each latent dimension, the genes are sorted by the absolute value of their weight. The genes having the highest of such weights are referred to as the *high weight genes* (Fig. 1d).

For each cluster, we selected the latent dimensions that distinguished the best the cells in the clusters and then computed the high weight genes. The high weight genes found for the clusters in the zebrafish dataset are given in Table 1. Using knowledge from biomedical literature about marker genes for blood cells, we mapped each cluster to a cell type. Thus, Cluster 1 corresponds to HSPCs, Cluster 2 to Neutrophils, Cluster 3 to Monocytes, Cluster 4 to Erythrocytes and Cluster 5 to Thrombocytes. The same process was used to map the clusters to cell types in the dataset with human pancreatic cells; see Supplementary Table 1 for the high weight genes found for the clusters in the human pancreatic dataset and their mapping to cell types.

Our results for identifying the different cell types in the zebrafish dataset are validated by[1] who computationally reconstructed the differentiation trajectories using the Monocle2 algorithm[29] and found the same cellular states. In particular, there is 89.9% overlap between the cell types identified using DiffVAE and the cell types obtained by[1]. Conversely, for the dataset with the human pancreatic cells, we found there is a 96.2% overlap between the cells types obtained using DiffVAE and the ones reported by Murano *et al.*[2]. In addition, DiffVAE identified all the different cell types in the dataset except for the epsilon cells. However, note that are only 4 epsilon cells in the dataset and Murano *et al.*[2] also did not identify them computationally, but rather based on the

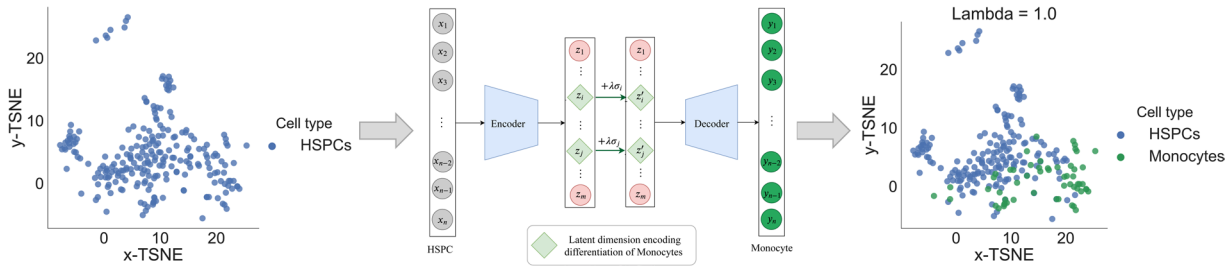


Figure 2. Methodology proposed for changing the cellular states: HSPCs can be converted into Monocytes by shifting the latent dimensions differentiating Monocytes by a factor λ multiplied with their standard deviation. Increasing the shifting parameter λ will result in more of the HSPCs to be subsequently classified as Monocytes.

expression of the GHRL gene. See Supplementary Fig. 1 for the clusters found using DiffVAE on the dataset with human pancreatic cells.

The representations built by DiffVAE on the human hematopoietic cells do not display separable clusters, which makes it difficult to identify all of the cell types. Velten *et al.*[3] also indicate that the hematopoietic stem cells, multipotent progenitors and multilymphoid progenitors cells form a unique continuous group when applying clustering methods to the dataset. Refer to Supplementary Fig. 2 and the corresponding section for a discussion of the limitations of DiffVAE in this case and directions for future work.

Characterization of cell states. For the zebrafish dataset, we also explored the possibility of changing the state of cells through perturbations on the latent dimension. This could help us learn more about the type of biological changes in gene expression that cause a less specialised cell such as an HSPC to differentiate in a more specialised cell such as a Monocyte. For this, we trained a neural network classifier capable of labelling Monocytes, Neutrophils, Erythrocytes and Thrombocytes using the full gene expression data with 99.5% accuracy. See the Methods section for more details.

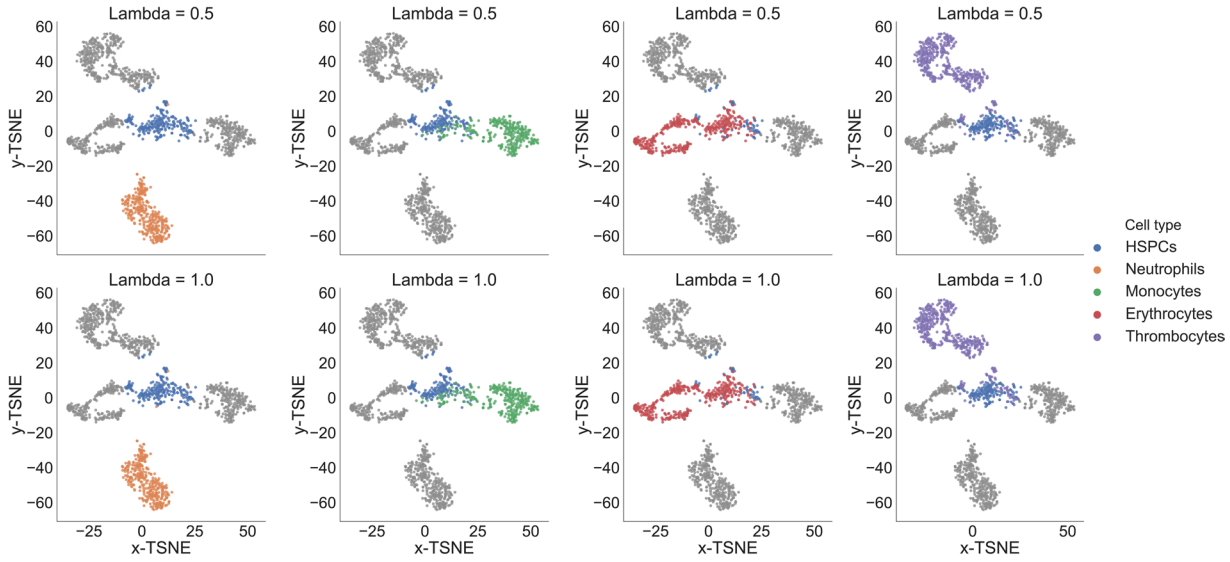
Assume that we have identified, that latent dimension j encodes the differentiation of a type of mature blood cells, such as Monocytes. Let $\boldsymbol{\mu}_j$ and σ_j be the mean and standard deviation of $\mathbf{z}_j = [z_j^{(1)} z_j^{(2)} \dots z_j^{(N)}]^T$ the predicted value of the encoder for z_j across all of the cells in the dataset. We can say that if latent dimension j identifies Monocytes, it means that the ratio of the number of Monocytes in $\mathscr{D}_j = \{\mathbf{x}^{(i)} \in \mathscr{D} | z_j^{(i)} \geq \mu_j + \sigma_j \vee z_j^{(i)} \leq \mu_j - \sigma_j\}$ is larger than for the other cells. This strongly suggests that shifting $z_j^{(i)}$ by the standard deviation σ_j of latent dimension j could potentially change the cell $\mathbf{x}^{(i)}$ label into a Monocyte.

The method proposed for changing a less specialised cell (an HSPC) into Monocytes involves shifting several of the latent dimensions encoding the differentiation of Monocytes proportionally with their standard deviations. The proportionality factor is the parameter λ. The method is illustrated in Fig. 2 and it can be generalised to any of the mature cell types that the embedding can separate.

Figure 3 shows the results after performing this kind of perturbations to change HSPCs into all of the mature blood cells in our dataset. For each cell type, we shifted the top 5 latent representation encoding their differentiation. We illustrate the results for both $\lambda = 0.5$ and $\lambda = 1.0$. We notice that there is a difference in how easy is to change the HSPCs into the different mature blood cell types which indicates that there is some heterogeneity among the HSPCs. In this context, we can also learn the minimum number of genes that need to be modified to change the cell type, and particularly, to determine which genes get upregulated and which genes get downregulated in this processes.

Let $\mathbf{x}^{(i)}$ be the input gene expression measurements for cell (i). After performing the perturbations on the latent representation $\mathbf{z}^{(i)}$ of cell (i) and putting the results through the decoder, we obtain the reconstructed gene expression measurement $\mathbf{y}^{(i)}$. Assume $\mathbf{y}^{(i)}$ is then classified by the neural network as a mature blood cell. By looking at the difference $\mathbf{y}^{(i)} - \mathbf{x}^{(i)}$ we can learn which genes have changed the most in the process of performing perturbations on the latent dimension. Then, by only changing the expression of these genes in $\mathbf{x}$ and leaving the other ones the same, we can compute the maximum number of genes that need to be changed to reprogram HSPCs into mature blood cells.

For the shifting parameter $\lambda = 1$, we analyze the HSPCs that were classified as mature blood cells after the operations on their latent representation. Our analysis shows that for the 175 HSPCs that were converted into Erythrocytes, we needed to change a relatively small number of genes (up to 25) for each HSPC. Conversely, the 70 HSPCs changed into Monocytes, gene perturbations needed to be performed on ~70% of the genes to change the cell type. For the 70 HSPCs that were classified as Thrombocytes after the perturbations, 50% of the cells were changed with modifications to only 20 genes; to change 100% of the 70 HSPCs into Thrombocytes almost all of the genes needed to be perturbed. Finally, only 3 HSPCs were changed into Neutrohpils so we did not perform any further analysis in this case. We would like to emphasize that these results are entirely computational and show how performing perturbations on the latent representation obtained from DiffVAE allows us to explore changing cell states. Nevertheless, biological experiments are required to validate such hypothesis generated by DiffVAE about cell reprogramming.



a. HSPCs → Neutrophils b. HSPCs → Monocytes c. HSPCs → Erythrocytes d. HSPCs → Thrombocytes

Figure 3. Results obtained after performing cell perturbations. We show in colour the cells of interested for each subfigure and in grey the rest of the cells. Each subfigure indicates how many of the HSPCs were converted into each type of mature blood cell after performing perturbations to the latent representations of DiffVAE. Notice that increasing the shifting parameter λ in the perturbations will result in more cells to be changed.

Clustering method	Dim size (m)	Latent representation				T-SNE embedding of latent representation			
		DiffVAE	VAE	AE	PCA	DiffVAE	VAE	AE	PCA
k-means	20	**0.803**	0.771	0.799	0.633	**0.809**	0.738	0.699	0.717
	50	**0.829**	0.775	0.811	0.629	**0.831**	0.801	0.759	0.709
	100	**0.844**	0.831	0.815	0.627	**0.815**	0.796	0.806	0.680
DBSCAN	20	0.007	0.004	0.001	0.0002	**0.753**	0.717	0.556	0.506
	50	0.474	0.243	0.223	0.0009	**0.710**	0.667	0.573	0.590
	100	0.154	0.018	0.011	0.002	**0.813**	0.799	0.749	0.570

Table 2. Zebrafish. Mean ARI obtained for clustering the latent representation and the t-SNE embedding of the latent representation for three settings of the reduced dimension size m. The clustering algorithms used are k-means and Gaussian Mixture Models.

Comparison of DiffVAE with other dimensionality reduction methods

After performing dimensionality reduction, standard single-cell RNA-seq workflows for identifying cell types involve clustering of the lower representation obtained for the gene expression data[30]. Using the zebrafish cell types found by Athanasiadis *et al.*[1] and the human pancreatic cell types found by Muraro *et al.*[2] as true labels, we compare DiffVAE with a standard variational autoencoder (VAE), a simple autoencoder (AE) and Principle Component Analysis (PCA) in terms of clustering performance. Their performance is compared using two clustering algorithms that use different approaches in defining clusters: k-means and DBSCAN. We will cluster both the raw data obtained through dimensionality reduction for $m \in \{20, 50, 100\}$ latent dimensions, as well as the 2-dimensional embedding produced using t-SNE.

For each setting of m (size of latent dimension), the clustering algorithms (including the computation of the t-SNE embedding) were performed 50 times and each time the ARI between the true labels and the cluster labels was computed. The results reported in Table 2 represent mean ARI obtained on the zebrafish dataset. See Supplementary Table 2 for the results on the dataset with human pancreatic cells. For both datasets, the representation built by DiffVAE gives the best overall clustering performance. In addition, computing the t-SNE embedding on top of the latent representation improves the clustering results.

Exploring links between cells

In this section, we shift the focus from just modelling the stochastic behaviour of gene expression across cell types and we also explore modelling the relations between different cell types. For this purpose, we propose Graph-DiffVAE, a graph variational autoencoder where the encoder and the decoder networks are graph convolutional networks. Graph-DiffVAE is based on the graph variational autoencoder proposed by Kipf and Welling[31] and on the Graphite model developed by Grover *et al.*[32].

In this context, we will consider the different cells in the zebrafish dataset as nodes in a graph, represented by the adjacency matrix **A**. The gene expression measurements for each cell will form the node features **X**. The encoder part of Graph-DiffVAE takes as input an initial graph structure for the cells and the input node features and computes a latent representation for cell $q_\phi(\mathbf{Z}|\mathbf{A}, \mathbf{X})$, which in this case will be denoted as latent node features. The decoder uses these latent node features and the initial adjacency matrix to predict additional links between the cells, which will be similar to the ones in the input graph.

The input graph can depend on specific applications. One option is to incorporate biological knowledge in the graph, where for instance, edges can represent potential differentiation trajectories for the cells in the dataset. The proposed architecture and training objective for Graph-DiffVAE results in additional edges between cells to be predicted in the output adjacency matrix. The predicted relationships between cells are similar to the ones in the initial graph given as input to the model.

In this paper, we aim to show a proof of concept for using Graph-DiffVAE with single-cell gene expression data. Thus, we propose building an initial graph for the cells where there is an edge between each cell and the cell most similar to it. For this purpose, we will use the Pearson correlation coefficient to measure the similarity between cells. This initial graph is undirected and is represented by a binary adjacency matrix where 1 indicates that there is an edge between two nodes (cells). For each cell in the dataset, we computed the Pearson correlation coefficient between its gene expression vector and the feature vectors of the rest of the cells in the dataset and we added an edge to connect it to the highest positively correlated cell.

Figure 4a illustrates the pipeline for using Graph-DiffVAE and Fig. 4b. shows the 2-dimensional t-SNE embedding of the node features predicted by Graph-DiffVAE, as well as specific links (both initial and predicted) between the HSPCs and differentiated cells. Figure 4c shows the adjacency matrix for the zebrafish dataset used as input to Graph-DiffVAE and Fig. 4d illustrates the predicted adjacency matrix. In Fig. 4b, it is noticeable that the latent representation built in the encoder exhibits a clustering structure between the different types of cells. This is expected and validates the behaviour of the model, as the cells that are highly correlated to each other are more likely to be part of the same cluster. We can also notice that having an initial edge between these types of cells encourages the prediction of similar types of edges. Moreover, in Fig. 4d this clustering behaviour represented in the encoder is emphasised in the output of the decoder, which predicts relatively well-defined clusters for the Monocytes, Neutrophils, Erythrocytes and Thrombocytes.

An interesting aspect of the predicted graph in Fig. 4c is that the HSPCs do not cluster together well. In particular, there are clear links between several HSPCs and all of the other cells in a cluster of mature blood cells. This means that among the HSPCs there are cells that have already started the process of differentiation towards one of the specific mature cells. Additionally, we can notice that Graph-DiffVAE predicted more edges between HSPCs and Erythrocytes compared to the other differentiated cells.

For comparison in Fig. 4e we also illustrate a co-expression matrix built by computing the absolute value of the Pearson correlation between all cells. While the co-expression matrix also shows clustering behaviour, the clusters are less well-defined. Using different thresholds for the correlation to select edges in the co-expression matrix will also result in different connections between cells. Moreover, note that such co-expression matrix only accounts for linear relationships between cells, while Graph-DiffVAE allows us to model non-linearities.

Another important difference is that the predictions of Graph-DiffVAE are highly dependent on the input graph. If prior biological knowledge is available about existing links between cells, this can be incorporated into the input graph. Based on this, Graph-DiffVAE will be able generate hypothesis about other links between cells that share the same biological meaning as the input ones.

Discussion

In this paper, we explored unsupervised generative and graph representation learning methods for modelling single-cell gene expression data and understanding cell differentiation by developing the DiffVAE and Graph-DiffVAE models. The two different models succeed in characterising different states of cell differentiation based on single-cell RNA-Seq data. We illustrated how to identify cell types using DiffVAE through a pipeline that involves clustering the latent representation, detecting important genes for each cluster and mapping from clusters to cell type. Many of the high-weight genes found by DiffVAE are well-known in the literature as key haematopoietic genes. In addition to these "usual suspects", our method identified a number of novel genes that can be further explored for their role during cell differentiation. We have also shown that the pipeline is applicable to datasets of different nature, providing powerful insight into the noisy information concealed by single-cell genetic data.

The embeddings obtained from DiffVAE can be used to generate artificial samples, allowing further exploration and expansion of the current datasets. Moreover, we explored perturbations over the generative latent space to then analyse the effect on the gene expression and changes in cellular states. The computational results on performing perturbations can help us understand better how easy/difficult it is for the hematopoietic stem and progenitor cells to change into differentiated cells. Additional information can be gained in terms of the number of genes that need to be up-regulated or down-regulated to change cellular state. That can lead to future studies on the stability of cellular states, and robustness over genetic stochasticity.

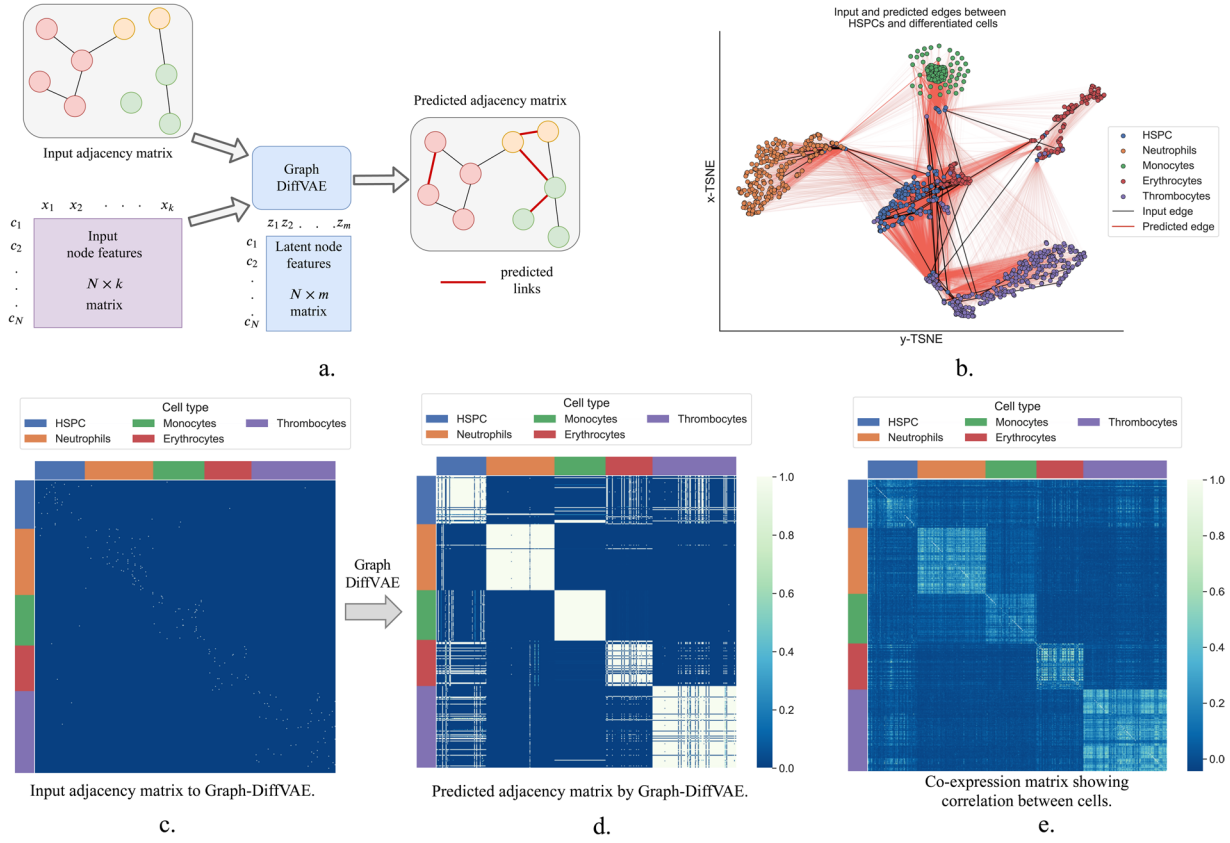


Figure 4. Methodology proposed for analyzing links between cells. (**a**) Graph-DiffVAE uses an initial adjacency matrix and individual node features to predict more links between cells. (**b**) Projection of cells onto 2-dimensional t-SNE embedding of the latent node features learnt by Graph-DiffVAE and illustration of initial and predicted links between HSPCs and differentiated cells. (**c**) Adjacency matrix with input links between cells (the colour white indicates an input edge); each cell is connected to the highest positively correlated cell. (**d**) Adjacency matrix with predicted links between all cells by Graph-DiffVAE (the colour white indicates a predicted edge). (**e**) Co-expression matrix between all cells; each entry represents the absolute value of the Pearson correlation coefficient.

Through Graph-DiffVAE we explored a way of understanding the connections between cells, and in particular between HSPCs and differentiated cells. Investigating the predicted links of Graph-DiffVAE between HSPCs and the other differentiated cells could inform us about the HSPCs which have already chosen a lineage and have started differentiated. For instance, if in the adjacency matrix predicted by Graph-DiffVAE, an HSPC cell is strongly connected to differentiated cells of a single type, such as Erythrocytes, we can hypothesize that this HSPC cell may also differentiate into an Erythrocyte. Similarly, if the HSPC cell is connected to two types of differentiated cells that might indicate that the HSPC cell has the potential to become either of these two differentiated cells. By analyzing the patterns in the gene expression of these types of cells may allow us to distinguish between the truly stem cells and the cells that have already started the differentiation process. Further analysis in this direction can also allow us to better understand cell differentiation trajectories. From a methodological perspective, future work could involve combining DiffVAE and Graph-DiffVAE into a single multitask learning framework[33] and using Graph Attentional Layers[34] as part of Graph-DiffVAE to better quantifying the importance of the links between cells.

Methods

DiffVAE. DiffVAE receives as input expression levels from k genes. The autoencoder model was constructed such that both the encoder and decoder consist of two fully connected hidden consisting of h_1 and h_2 neurons respectively. The incorporation of multiple hidden layers helps to build a hierarchical representation of features, thus obtaining a more complex model. The size of the hidden layers is symmetric between the encoder and decoder. The latent representation $\mathbf{z}$ has m dimensions. The ReLU activation was applied in the hidden layers of both the encoder and decoder in order to introduce non-linearity in the network. The specific operations performed by DiffVAE are as follows:

Encoder (Inference model): $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathrm{z};\ \boldsymbol{\mu},\ \mathrm{diag}(\boldsymbol{\sigma}^2))$. The encoder consists of fully connected layers and has a Gaussian output. For numerical stability, the encoder network learns $\log(\boldsymbol{\sigma}^2)$ instead of $\boldsymbol{\sigma}^2$. The input to the

encoder is $\mathbf{x} \in \mathbb{R}^{1\times 1845}$, which, in our case, represents the gene expression data. The operations performed by the encoder network are summarised by:

$$\mathbf{x}_{\text{enc}}^{(1)} = \text{ReLU}(\mathbf{W}_{\text{enc}}^{(0)}\mathbf{x} + \mathbf{b}_{\text{enc}}^{(1)}), \tag{4}$$

$$\mathbf{x}_{\text{enc}}^{(2)} = \text{ReLU}(\mathbf{W}_{\text{enc}}^{(1)}\mathbf{x}_{\text{enc}}^{(1)} + \mathbf{b}_{\text{enc}}^{(2)}), \tag{5}$$

$$\boldsymbol{\mu} = \text{ReLU}(\mathbf{W}_{\mu}\mathbf{x}_{\text{enc}}^{(2)} + \mathbf{b}_{\mu}), \tag{6}$$

$$\log\boldsymbol{\sigma}^2 = \text{ReLU}(\mathbf{W}_{\sigma}\mathbf{x}_{\text{enc}}^{(2)} + \mathbf{b}_{\sigma}), \tag{7}$$

where $\mathbf{W}_{\text{enc}}^{(0)} \in \mathbb{R}^{k\times h_2}$, $\mathbf{b}_{\text{enc}}^{(1)} \in \mathbb{R}^{1\times h_2}$, $\mathbf{W}_{\text{enc}}^{(1)} \in \mathbb{R}^{h_2\times h_1}$, $\mathbf{b}_{\text{enc}}^{(2)} \in \mathbb{R}^{1\times h_1}$, $\mathbf{W}_{\mu} \in \mathbb{R}^{h_1\times m}$, $\mathbf{b}_{\mu} \in \mathbb{R}^{1\times m}$, $\mathbf{W}_{\sigma} \in \mathbb{R}^{h_1\times m}$, $\mathbf{b}_{\sigma} \in \mathbb{R}^{1\times m}$are the trainable parameters in the encoder. The encoder also uses batch normalization[35] to overcome the problem of internal covariate shift.

Directly sampling the latent representation $\mathbf{z}$ can cause problems to the standard gradient-based algorithm, as it is not possible to compute gradients through the random sampling of $\mathbf{z}$. To overcome these issues, Kingma and Welling[19] proposed the reparameterisation trick that involves parameterising the latent code as follows:

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\varepsilon} \odot \boldsymbol{\sigma}. \tag{8}$$

Decoder (generative model): $p_{\theta}(\mathbf{x}|\mathbf{z})$. The output of the decoder has to reward the likelihood of the data we want to generate with this model. In our case, for each data point, the gene expression values can be modelled as samples from a multivariate Bernoulli distribution. Intuitively, each input gene is modelled as a Bernoulli random variable, and a sample from this distribution indicates whether the gene is expressed or not. To build a decoder with Bernoulli output, we need to apply the logistic activation function to compute the output of the decoder because it takes values in the range [0, 1].

The input to the decoder is the latent representation $\mathbf{z}$. The decoder performs the following operations in order to obtain the reconstructed input $\mathbf{x}$':

$$\mathbf{x}_{\text{dec}}^{(1)} = \text{ReLU}(\mathbf{W}_{\text{dec}}^{(0)}\mathbf{z} + \mathbf{b}_{\text{dec}}^{(1)}), \tag{9}$$

$$\mathbf{x}_{\text{dec}}^{(2)} = \text{ReLU}(\mathbf{W}_{\text{dec}}^{(1)}\mathbf{x}_{\text{dec}}^{(1)} + \mathbf{b}_{\text{dec}}^{(2)}), \tag{10}$$

$$\mathbf{x}' = \sigma(\mathbf{W}_{\text{out}}\mathbf{x}^{(2)} + \mathbf{b}_{\text{out}}), \tag{11}$$

where σ is the logistic activation function and $\mathbf{W}_{\text{dec}}^{(0)} \in \mathbb{R}^{m\times h_1}$, $\mathbf{b}_{\text{dec}}^{(1)} \in \mathbb{R}^{1\times h_1}$, $\mathbf{W}_{\text{dec}}^{(1)} \in \mathbb{R}^{h_1\times h_2}$, $\mathbf{b}_{\text{dec}}^{(2)} \in \mathbb{R}^{1\times h_2}$, $\mathbf{W}_{\text{out}} \in \mathbb{R}^{h_2\times k}$, $\mathbf{b}_{\text{out}} \in \mathbb{R}^{1\times k}$are the trainable parameters in the decoder. As $\mathbf{x}$' is not sampled, we provide a maximum likelihood estimate for the reconstruction.

DiffVAE was trained using minibatch stochastic gradient descent to minimize $-\mathcal{L}_{\text{DiffVAE}}$:

$$\mathcal{L}_{\text{DiffVAE}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathbb{E}_{\boldsymbol{q}_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \text{MMD}(\boldsymbol{q}_{\boldsymbol{\phi}}(\mathbf{z})\|\boldsymbol{p}_{\boldsymbol{\theta}}(\mathbf{z})), \tag{12}$$

We used the Adam Optimizer[36] and we trained DiffVAE for 100 epochs. The learning rate and batch size were selected as part of the hyperparameter optimization process.

Hyperparameter selection for DiffVAE. DiffVAE consists of the following hyperparameters that need to be optimized before using it a new dataset: number of neurons in the hidden layers (h_1 and h_2), latent representation size (m),learning rate (α) and batch size (B). The original dataset was split such that 80% of the data points were used for training and 20% for validation. Each hyperparameter was optimised in a range of possible values, with the final one chosen based on the validation loss. A similar approach for hyperparameter optimization has been used by other methods modelling single-cell gene expression data using deep generative architectures[37].

The possible values used for each hyperparameter are as follows: number of neurons $h_1 \in \{256, 128\}$, latent representation size $m \in \{50, 100\}$, learning rate $\alpha \in \{0.01, 0.001, 0.0001\}$ and batch size $B \in \{64, 128, 256\}$. We set $h_2 = 2 \cdot h_1$. The final hyperparameter values used for the zebrafish dataset are $h_1 = 256$, $h_2 = 512$, $m = 50$, $\alpha = 0.001$ and $B = 128$. For the dataset with human pancreatic cells we used $h_1 = 256$ and $h_2 = 512$, $m = 100$, $\alpha = 0.001$ and $B = 256$. Finally, for the dataset with human hematopoietic cells, we used $h_1 = 256$ and $h_2 = 512$, $m = 50$, $\alpha = 0.001$ and $B = 128$.

To apply DiffVAE to a new dataset, a similar approach as described in this section can be used for selecting the most appropriate hyperparameters.

Additional implementation details for identifying cell types. In the process of identifying cells using DiffVAE additional design choices came into play in terms of the number of latent dimensions and the number of high weight genes to select for each cluster. In the dataset with zebrafish cells, we have used the top 10 latent dimensions that were encoding the differentiation of cells in each cluster and for each latent dimension, we have investigated the top 3 high weight genes. On the other hand, for the dataset with human pancreatic cells, due to

the large number of clusters, we have only used the top 5 latent dimensions with the top 5 high weight genes. In Table 1 and Supplementary Table 1, we have reported the high weight genes that were common among the selected latent dimensions.

The main purpose of this process is to be able to identify the cell types in the dataset. In practice, the choice for the number of latent dimensions to analyze should depend on their ability to differentiate between the cells in the different clusters, as well as on the total number of clusters. Similarly, the selected number of high-weight genes should provide enough information to map clusters to cell types, but also allow for uncovering biological knowledge about new marker genes important for the differentiation process.

Additional models used as benchmarks. To assess the performance of DiffVAE on clustering, we compare it against the following benchmarks: standard VAE, standard autoencoder and PCA. For the standard VAE and the standard autoencoder, we used the same number of layers and neurons as we used for DiffVAE. For both models, we also used the Adam Optimizer with a learning rate of 0.001, a batch size of 128 and we trained them for 100 epochs.

The objective function maximised by the variational autoencoder is:

$$\mathscr{L}_{\mathrm{VAE}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathbb{E}_{q_{\phi(\mathbf{z}|\mathbf{x})}}[\log p_{\theta}(\mathbf{x}|\mathbf{z})] - D_{KL}(\boldsymbol{q}_{\phi}(\mathbf{z}|\mathbf{x})\|\boldsymbol{p}_{\theta}(\mathbf{z})). \tag{13}$$

The standard autoencoder model is trained to minimise the reconstruction error, which can be measured by the mean squared error loss function $l(\mathbf{x}, \omega) = \|\mathbf{x} - dec(enc(\mathbf{x}))\|^2$, where ω consists of all of the parameters in the encoder and decoder networks.

Neural network for characterizing cell states. The neural network trained for classifying the mature cell types takes as input the gene expression data for the cell and outputs the probability that the particular cell is a Monocyte, Neutrophil, Thrombocyte or Erythrocyte. The model consists of three hidden layers of sizes 256, 512, 256 neurons with ReLU activation and an output layer with 4 neurons and softmax activation. The model was also trained using the Adam Optimizer for 300 epochs with a learning rate of 0.001 and a batch size of 128. The model architecture was chosen through hyperparameter optimization similarly to DiffVAE.

GraphDiffVAE. Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, with $|\mathscr{V}| = N$ be an undirected and unweighted initial graph built from the cells, defined by the binary adjacency matrix $\mathbf{A} \in \{0, 1\}^{N\times N}$. Such an initial graph for the cells can be already available or it can be artificially built using the Pearson correlation between cells.

Let $\mathbf{X}$ be an $N \times F$ matrix consisting of node features, where F is the number of features for each node. In our case, the nodes are the different cells in the dataset, and the features are represented by the gene expression for each cell. Assume that each node is connected to itself, so that $\mathbf{A}$ has diagonal entries $A_{ii} = 1$. Let $\mathbf{D}$ be the diagonal degree matrix of $\mathbf{A}$: $D_{ii} = \sum_j A_{ij}$.

Graph convolutional networks (GCN) were proposed by Kipf and Welling[38] with the following layer wise propagation rule:

$$\mathbf{X}^{(l+1)} = \tau(\widetilde{\mathbf{A}}\mathbf{X}^{(l)}\mathbf{W}^{(l)}) = \mathrm{GCN}_{\tau,l}(\mathbf{A}, \mathbf{H}^{(l)}), \tag{14}$$

where $\widetilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, τ is the activation function applied, $\mathbf{X}^{(l)}$ and $\mathbf{X}^{(l+1)}$ are the activations of the layers (l) and $(l+1)$ respectively, and $\mathbf{W}^{(l)}$ are the weights. $\mathbf{X}^{(0)}$ represents the input feature matrix $\mathbf{X}$. The size of layer n_l represents by the number of node features computed at layer (l).

Through the layer-wise propagation rule, the graph convolutional network performs spectral graph convolutions. The model can be regarded as the differentiable and generalised version of the algorithm proposed by Weisfeiler-Lehman on graphs[39]. In particular, the layer-wise propagation rule can be viewed as a message passing computation over the graph structure. Through one hidden layer, nodes in the graph pass information about their local structure to neighbours that are 1-hop away. Based on the information received from the neighbours, the nodes update their node features.

Graph-DiffVAE is a graph variational autoencoder where the encoder and the decoder networks are graph convolutional networks applying the layer-wise propagation rule. The architecture of Graph-DiffVAE is based on the ones in the graph variational autoencoder proposed by Kipf and Welling[31] and in the Graphite model developed by Grover *et al.*[32].

Inference model (encoder): $q_{\phi}(Z|A, X)$. The encoder in Graph-DiffVAE is represented by a graph convolutional network with multiple layers and with Gaussian output. The input to the encoder consists of the matrix with node features $\mathbf{X}$ and of the graph adjacency matrix $\mathbf{A}$. The layers in the encoder network perform the following operations:

$$\mathbf{X}^{(1)}_{\mathrm{enc}} = GCN_{\tau_1,1}(\mathbf{A}, \mathbf{X}), \tag{15}$$

$$\boldsymbol{\mu} = GCN_{\tau_2,\mu}(\mathbf{A}, \mathbf{X}^{(1)}_{\mathrm{enc}}), \tag{16}$$

$$\log\boldsymbol{\sigma}^2 = GCN_{\tau_2,\sigma}(\mathbf{A}, \mathbf{X}^{(1)}_{\mathrm{enc}}). \tag{17}$$

where τ_1 is the ReLU activation function and τ_2 is the linear activation function. The number of node features computed in $\mathbf{X}_{\text{enc}}^{(1)}$ is 512 and the number of node features in $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ is 50, thus forming a latent representation $\mathbf{Z}$ where each node has $M = 50$ features.

The encoder represents a factorised multivariate Gaussian distribution, such that:

$$q_\phi(\mathbf{Z}|\mathbf{A}, \mathbf{X}) = \prod_{i=1}^{N} q_\phi(\mathbf{z}_i|\mathbf{Z}, \mathbf{A}), \quad q_\phi(\mathbf{z}_i|\mathbf{X}, \mathbf{A}) = \mathcal{N}(\mathbf{z}_i|\mu_i, \text{diag}(\sigma_i^2)). \tag{18}$$

The reparametrisation trick is used again to sample each $\mathbf{z}_i$:

$$\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{z}_i = \boldsymbol{\mu}_i + \varepsilon \odot \boldsymbol{\sigma}_i. \tag{19}$$

It is important to notice that the latent representation $\mathbf{Z}$ build by the Graph-DiffVAE encoder contains information from both the graph structure and the node features. $\mathbf{Z}$ encompass the latent representations for each node in the graph.

Generative model (decoder): $p_\phi(\boldsymbol{A}|\boldsymbol{Z}, \boldsymbol{X})$. The output of the decoder is an adjacency matrix $\hat{\mathbf{A}}$ representing an undirected and unweighted graph with predicted edges between nodes. Such an adjacency matrix can be represented by a factorised Bernoulli distribution.

The decoder network uses as input the initial adjacency matrix $\mathbf{A}$ and a concatenation of the input node features $\mathbf{X}$ and the latent node features computed by the encoder $\mathbf{Z}$, described by $[\mathbf{Z}|\mathbf{X}]$. The layers in the decoder network perform the following operations:

$$\mathbf{X}_{\text{dec}}^{(1)} = GCN_{\tau_1,1}(\mathbf{A}, [\mathbf{Z}|\mathbf{X}]), \quad \mathbf{Z}^* = \frac{1}{2}(\mathbf{Z}' + \mathbf{Z}), \tag{20}$$

$$\mathbf{Z}' = GCN_{\tau_1,2}(\mathbf{A}, \mathbf{X}_{\text{dec}}^{(1)}), \quad \hat{\mathbf{A}} = \sigma(\mathbf{Z}_i^* \mathbf{Z}_j^*), \tag{21}$$

where τ_1 is the ReLU activation function. The number of node features computed in $\mathbf{X}_{\text{dec}}^{(1)}$ is 512. The decoder builds its own latent representation $\mathbf{Z}'$ consisting of 50 node features which it then adds to the representation constructed through the encoder to obtain $\mathbf{Z}^*$.

Similarly with the standard framework of the variational autoencoder, we optimize the following objective for Graph-DiffVAE:

$$\mathcal{L}_{\text{Graph}-\text{DiffVAE}}(\boldsymbol{\theta}, \phi; \mathbf{X}, \mathbf{A}) = \mathbb{E}_{q_{\phi(\mathbf{Z}|\mathbf{A},\mathbf{X})}}[\log p_{\boldsymbol{\theta}}(\mathbf{A}|\mathbf{Z})] - \text{KL}(q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\|p_{\boldsymbol{\theta}}(\mathbf{z})), \tag{22}$$

The model was trained for 200 epochs with a learning rate of 0.0001. The proposed architecture and training objective for Graph-DiffVAE results in additional edges between cells to be predicted in the output adjacency matrix $\hat{\mathbf{A}}$. The predicted relationships between cells are similar to the ones in the initial graph given as input to the model.

Software

The code for DiffVAE and Graph-DiffVAE is publicly available in the GitHub repository: https://github.com/ioanabica/DiffVAE.

Data availability

The zebrafish dataset used for this paper is made publicly available by Athanasiadis *et al.*[1] on ArrayExpress under the accession numbers E-MTAB-3947, E-MTAB-4617 and E-MTAB-5530 and also at https://www.sanger.ac.uk/science/tools/basicz. Similarly, the dataset with human pancreatic cells is made publicly available by Murarot al[2]. under accession number GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE85241GSE85241. Moreover, the dataset with human hematopoietic cells was made publicly available by Velten *et al.*[3] under accession code GEO: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75478GSE75478.

Received: 1 September 2019; Accepted: 10 February 2020;
Published online: 17 June 2020

References

1. Athanasiadis, E. *et al.* Single-cell rna-sequencing uncovers transcriptional states and fate decisions in haematopoiesis. *Nature communications* **8**, 2045 (2017).
2. Muraro, M. J. *et al.* A single-cell transcriptome atlas of the human pancreas. *Cell systems* **3**, 385–394 (2016).
3. Velten, L. *et al.* Human haematopoietic stem cell lineage commitment is a continuous process. *Nature cell biology* **19**, 271 (2017).
4. Shin, J. *et al.* Single-cell rna-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell stem cell* **17**, 360–372 (2015).
5. Setty, M. *et al.* Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nature biotechnology* **34**, 637 (2016).
6. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology* **32**, 381 (2014).
7. Marco, E. *et al.* Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proceedings of the National Academy of Sciences* **111**, E5643–E5650 (2014).
8. Yeung, K. Y. & Ruzzo, W. L. Principal component analysis for clustering gene expression data. *Bioinformatics* **17**, 763–774 (2001).
9. Guibentif, C. *et al.* Single-cell analysis identifies distinct stages of human endothelial-to-hematopoietic transition. *Cell reports* **19**, 10–19 (2017).

10. McKinney-Freeman, S. *et al*. The transcriptional landscape of hematopoietic stem cell ontogeny. *Cell stem cell* **11**, 701–714 (2012).
11. Kluger, Y. *et al*. Lineage specificity of gene expression patterns. *Proceedings of the National Academy of Sciences of the United States of America* **101**, 6508–6513 (2004).
12. Way, G. P. & Greene, C. S. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *bioRxiv* 174474 (2017).
13. Weinstein, J. N. *et al*. The cancer genome atlas pan-cancer analysis project. *Nature genetics* **45**, 1113 (2013).
14. Tan, J., Hammond, J. H., Hogan, D. A. & Greene, C. S. Adage-based integration of publicly available pseudomonas aeruginosa gene expression data with denoising autoencoders illuminates microbe-host interactions. *mSystems* **1** (2016).
15. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications* **10**, 390 (2019).
16. Talwar, D., Mongia, A., Sengupta, D. & Majumdar, A. Autoimpute: Autoencoder based imputation of single-cell rna-seq data. *Scientific reports* **8**, 16329 (2018).
17. Wang, D. & Gu, J. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics* **16**, 320–331 (2018).
18. Rashid, S., Shah, S., Bar-Joseph, Z. & Pandya, R. Project dhaka: Variational autoencoder for unmasking tumor heterogeneity from single cell genomic data. *bioRxiv* 183863 (2018).
19. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)* (2014).
20. Tishby, N. & Zaslavsky, N. Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, 1–5 (IEEE, 2015).
21. Zhao, S., Song, J. & Ermon, S. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, 5885–5892 (2019).
22. Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B. & Smola, A. J. A kernel method for the two-sample-problem. In *Advances in neural information processing systems (NeurIPS)*, 513–520 (2007).
23. Li, Y., Swersky, K. & Zemel, R. Generative moment matching networks. In *International Conference on Machine Learning (ICML)*, 1718–1727 (2015).
24. Dziugaite, G. K., Roy, D. M. & Ghahramani, Z. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906* (2015).
25. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I. & Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).
26. Tolstikhin, I., Bousquet, O., Gelly, S. & Schoelkopf, B. Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558 (2017).
27. Chollet, F. *et al*. Keras (2015).
28. Maaten, L. V. D. & Hinton, G. Visualizing data using t-sne. *Journal of machine learning research* **9**, 2579–2605 (2008).
29. Qiu, X. *et al*. Single-cell mrna quantification and differential analysis with census. *Nature methods* **14**, 309 (2017).
30. Luecken, M. D. & Theis, F. J. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular systems biology* **15** (2019).
31. Kipf, T. N. & Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
32. Grover, A., Zweig, A. & Ermon, S. Graphite: Iterative generative modeling of graphs. *International Conference on Machine Learning (ICML)* (2019).
33. Zhang, Y. & Yang, Q. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).
34. Veličković, P. *et al*. Graph attention networks. International Conference on Learning Representations (ICLR) (2018).
35. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning (ICML)* (2015).
36. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)* (2015).
37. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods* **15**, 1053 (2018).
38. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)* (2017).
39. Shervashidze, N., Schweitzer, P., Leeuwen, E. J. V., Mehlhorn, K. & Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* **12**, 2539–2561 (2011).
40. Leung, A. Y. *et al*. Proliferating cell nuclear antigen (pcna) as a proliferative marker during embryonic and adult zebrafish hematopoiesis. *Histochemistry and cell biology* **124**, 105–111 (2005).
41. Patil, P., Uechi, T. & Kenmochi, N. Incomplete splicing of neutrophil-specific genes affects neutrophil development in a zebrafish model of poikiloderma with neutropenia. *RNA biology* **12**, 426–434 (2015).
42. Foulkes, M. J. *et al*. Expression and regulation of drug transporters in vertebrate neutrophils. *Scientific reports* **7**, 4967 (2017).
43. Harvie, E. A. & Huttenlocher, A. Neutrophils in host defense: new insights from zebrafish. *Journal of leukocyte biology* **98**, 523–537 (2015).
44. Tran, M. T. N. *et al*. Mafb is a critical regulator of complement component c1q. *Nature communications* **8**, 1700 (2017).
45. Kelly, L. M., Englmeier, U., Lafon, I., Sieweke, M. H. & Graf, T. Mafb is an inducer of monocytic differentiation. *The EMBO journal* **19**, 1987–1997 (2000).
46. Pimtong, W., Datta, M., Ulrich, A. M. & Rhodes, J. Drl. 3 governs primitive hematopoiesis in zebrafish. *Scientific reports* **4**, 5791 (2014).
47. Moore, F. E. *et al*. Single-cell transcriptional analysis of normal, aberrant, and malignant hematopoiesis in zebrafish. *Journal of Experimental Medicine* jem–20152013 (2016).
48. Khandekar, G., Kim, S. & Jagadeeswaran, P. Zebrafish thrombocytes: functions and origins. *Advances in hematology* **2012** (2012).

Acknowledgements

I.B. is funded by The Alan Turing Institute Doctoral Studentship, under the EPSRC grant EP/N510129/1. A.C. is funded by the European Research Council (project 677501).

Author contributions

I.B. and H.A.T. led the study, developed the methods and contributed to the analysis and design of the experiments. A.C. contributed to the the interpretation of the experimental results and P.L. supervised the development of the methodology. All authors read and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41598-020-66166-8.

Correspondence and requests for materials should be addressed to I.B., H.A.-T. or A.C.