Patterns

CellPress
OPEN ACCESS

Review

Latent representation learning in biology and translational medicine

Andreas Kopf[1] and Manfred Claassen[2,3,4,*]
[1]Institute of Molecular Systems Biology, ETH Zürich, 8093 Zürich, Switzerland
[2]Division of Clinical Bioinformatics, Department of Internal Medicine I, University Hospital Tübingen, 72076 Tübingen, Germany
[3]Computer Science Department, Eberhard Karls University of Tübingen, 72076 Tübingen, Germany
[4]Cluster of Excellence Machine Learning (EXC 2064), Eberhard Karls University of Tübingen, 72076 Tübingen, Germany
*Correspondence: manfred.claassen@med.uni-tuebingen.de

https://doi.org/10.1016/j.patter.2021.100198

THE BIGGER PICTURE Current data generation capabilities in the life sciences render scientists in an apparently contradicting situation. While it is possible to simultaneously measure an ever-increasing number of systems parameters, the resulting data are becoming increasingly difficult to interpret. Latent variable modeling has proved to be a formal machine learning paradigm to achieve such interpretation by learning non-measurable hidden variables from observations. This review summarizes concepts and applications of this paradigm in the life sciences.

SUMMARY

Current data generation capabilities in the life sciences render scientists in an apparently contradicting situation. While it is possible to simultaneously measure an ever-increasing number of systems parameters, the resulting data are becoming increasingly difficult to interpret. Latent variable modeling allows for such interpretation by learning non-measurable hidden variables from observations. This review gives an overview over the different formal approaches to latent variable modeling, as well as applications at different scales of biological systems, such as molecular structures, intra- and intercellular regulatory up to physiological networks. The focus is on demonstrating how these approaches have enabled interpretable representations and ultimately insights in each of these domains. We anticipate that a wider dissemination of latent variable modeling in the life sciences will enable a more effective and productive interpretation of studies based on heterogeneous and high-dimensional data modalities.

INTRODUCTION

Latent representation learning (LRL), or latent variable modeling (LVM), is a machine learning technique that attempts to infer latent variables from empirical measurements. Latent variables are variables that cannot be measured directly and therefore have to be inferred from the empirical measurements. In biomedicine or biomedical applications, directly measurable variables are related to physical and biological characteristics, such as weight, height, body temperature, pH, hemoglobin, blood count, metabolism, and many more. However, many variables of interest are not directly measurable, and examples include variables like pain, satisfaction, abilities to perform activities of daily living, stress, burnout or well-being, and health.[1] Such variables are modeled as latent variables of a LVM. In general, one or many latent variables jointly constitute a latent space or latent representation. This representation is usually a compressed form of the empirical measurements; it consists of fewer latent variables than the dimensionality of the measurements (i.e., the number of different measurement modalities).

Distinguishing between healthy or diseased patients is an illustrative example for LVM. This distinction typically involves a physician performing many assessments; i.e., empirical measurements such as visual tests, measuring temperature, physical tests. In the end, the physician integrates this information to conclude on the health state of the patient. This conclusion can also be conceived as the inference of the latent patient health state, and, in this conceptual example, the physician performs the task of an LVM (Figure 1). The diagnosis of multiple sclerosis (MS) is a specific instance of this inference. No single test exists for diagnosis of MS, but, via diagnosis of exclusion (McDonald criteria 2017[2]), many variables, such as clinical relapses, MRI lesions in specific locations, or oligoclonal bands in cerebrospinal fluid, are integrated for the physician's diagnosis of this disease.

In biology or translational studies, different measurement types are commonly recorded; for example, single-cell omics, imaging, structural data, time-series, or text data. Due to technological developments, the dimensionality of the data as well as the sample size has been steadily increasing, in principle

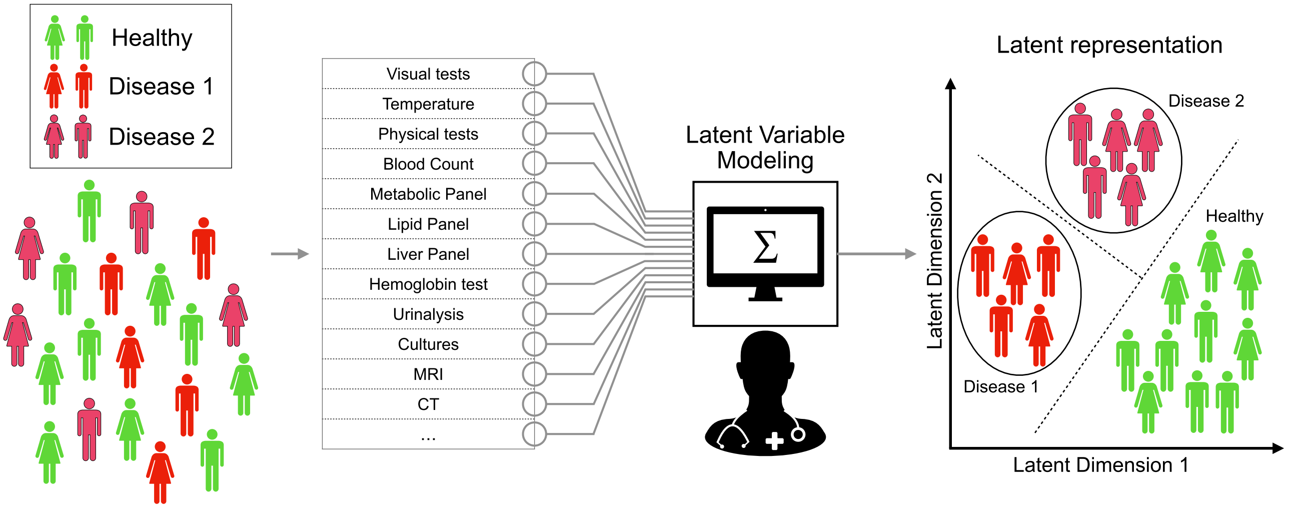

Figure 1. Visualizing the concept behind LVM based on a toy example of patient stratification
The state of health is a variable that cannot be measured directly, in particular in relation to diseases that cannot be diagnosed with one specific test, such as MS. Many different tests can be performed and taken together to infer the state of health of patients. Similar test results lead to patients mapping closely together in the latent representation and therefore to a similar diagnosis.

opening unprecedented possibilities to infer so far inaccessible latent variables. To infer latent variables of interest, it is advisable to focus on recording informative variables instead of as many variables as possible. Hence, a well-prepared study design is essential to be able to infer potential latent variables of interest. Additionally, LVM can be used to combine different data modalities for inferring a shared latent variable; e.g., to resolve the cell cycle stage of a cell in a combined analysis of transcriptomic and proteomic measurements. Inferring the lower-dimensional latent representation typically makes it easier to handle data visualization, non-numerical data types, or finding similarities in highly complex data.

In this review, we aim to provide an overview of recent and forthcoming LRL approaches and how these have been applied in the past. The journey involves the development of classical factor analysis (FA) models[3] to Gaussian process (GP) LVM[4–6] and, for now, ends with the breakthrough of deep learning, including many different deep model architectures, such as autoencoders (AEs)[7] variational AEs (VAEs),[8,9] or generative adversarial networks.[10,11] In this work, we survey LVM from two perspectives. First, we give an overview of the mathematical concepts to infer a latent space, introducing the most popular models in a shared notation. Second, we discuss LVM from the perspective of a hierarchy of applications defined by conceptual similarity of application domains in the life sciences. Finally, we discuss advantages, shortcomings, and potential orientation for future work in LVM.

BASIC CONCEPTS OF LVM

Here, we aim to describe the main concepts of LVM approaches and variants of them mostly developed for more specific data requirements or better interpretability of the latent variables. We cover the idea behind the main approaches of FA and GP-LVM. Further, we introduce the different variations of LVM using deep learning approaches as AEs, VAEs, standard (deep) neural networks (DNNs), or generative adversarial networks.

As a general mathematical notation, we define the following main variables that are used throughout this work:

- Factors, latent variable/representation:

$$\{\boldsymbol{z}_i\}_{i=1,\ldots,N} = \boldsymbol{Z} \in \mathbb{R}^{K \times N}$$

- Data matrix:

$$\{\boldsymbol{x}_i\}_{i=1,\ldots,N} = \boldsymbol{X} \in \mathbb{R}^{P \times N}$$

- Data matrix reconstructed:

$$\{\boldsymbol{x}_{R,i}\}_{i=1,\ldots,N} = \boldsymbol{X}_R \in \mathbb{R}^{P \times N}$$

- Data matrix generated:

$$\{\boldsymbol{x}_{G,i}\}_{i=1,\ldots,N} = \boldsymbol{X}_G \in \mathbb{R}^{P \times N}$$

In general, we always assume $K<P$, where K is the dimensionality of the latent representation and P the number of variables in the data. Further, N is the number of independent samples in the data matrix. The variables $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ stand for mean and covariance, respectively.

Factor analysis

The most basic approach to infer latent variables is called FA[3] and can be used to derive many variants of it, such as probabilistic principal component analysis (PPCA), where the difference

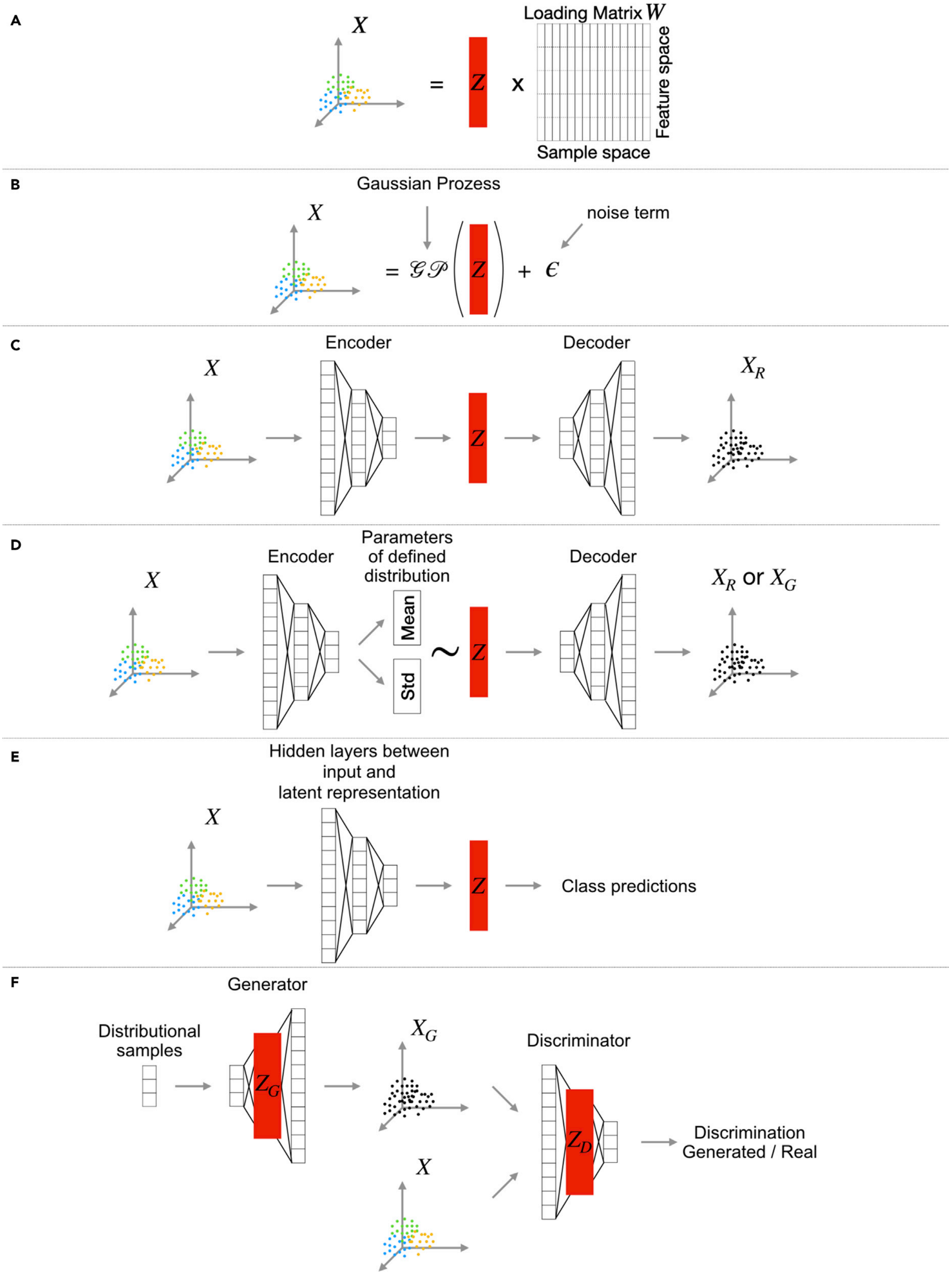

Figure 2. Schematic visualization of state-of-the-art methods used for latent representation learning

In red we visualize the central latent representation *Z* for each model type inferred from data *X*.

(A) FA where the loading matrix ***W*** defines the feature importance for each sample.

(legend continued on next page)

can be found in assumptions on the noise parameterization. Having our data matrix $\{\boldsymbol{x}_i\}_{i=1,\ldots,N} = \boldsymbol{X} \in \mathbb{R}^{P \times N}$ containing N samples with P features, we can define

$$\boldsymbol{X} = \boldsymbol{WZ} \qquad \text{(Equation 1)}$$

where $\boldsymbol{W} \in \mathbb{R}^{P \times K}$ is called the factor loading matrix with $K < P$ and $\boldsymbol{Z} \in \mathbb{R}^{K \times N}$ contains the factors or latent variables. The loading matrix captures the correlation between the observed variables for every single factor. Those loading matrices can be used to interpret the type of variance the factor is capturing. Argelaguet et al.,[12] for example, used the factor loadings to understand which genes influence a specific factor the most, including directionality, whereas Klami et al.[13] used the factor loadings to identify the relationship between groups of variables. The prior distribution over the factor matrix $\{\boldsymbol{z}_i\}_{i=1,\ldots,N} = \boldsymbol{Z} \in \mathbb{R}^{K \times N}$ in FA is assumed to follow

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i | 0, \boldsymbol{I}) \qquad \text{(Equation 2)}$$

where $0 \in \mathbb{R}^K$ is the zero mean vector and $\boldsymbol{I} \in \mathbb{R}^{K \times K}$ stands for the identity matrix. Hence, the latent variables are standard normal distributed in the classical FA model. Finally, we can write

$$p(\boldsymbol{x_i} | \boldsymbol{z_i}) = \mathcal{N}(\boldsymbol{x_i} | \boldsymbol{Wz_i} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \qquad \text{(Equation 3)}$$

where the mean is defined by a general linear function of $\boldsymbol{z}_i$ plus the sample mean $\boldsymbol{\mu}$, and the covariance matrix $\boldsymbol{\Psi}$ is given by a $P \times P$ diagonal matrix capturing for each variable the variance of the noise. This Gaussian noise assumption is as mentioned above the link to PPCA, where for each variable the same variance is assumed. Maximizing the log likelihood of the FA model is one option for learning the latent variables, but can also be done via variational inference (VI)[14] or Markov Chain Monte Carlo (MCMC) sampling.[3] Figure 2A shows symbolically the FA model.

Many variants of FA with different applications in life sciences were developed; for example, a model called zero-inflated FA (ZIFA),[15] which specifically was designed to meet the requirements of the so-called dropout characteristics of single-cell RNA sequencing (scRNA-seq) data. Those dropouts lead to many zero entries in the data matrix, which are modeled with an additional zero-inflation modulation layer in the FA framework. Buettner et al.[16] introduce slalom, which is an FA model trying to decompose the heterogeneity of the data in biological and technical factors by introducing prior knowledge from gene set annotation databases via priors on the factor weights. In addition to that, slalom is not only modeling the dropout effect but supports alternative different noise assumptions in different RNA-seq protocols. FA was also adopted to incorporate different data types (views) into one model to capture variance across data types. The model is called multi-omics FA (MOFA) and shares a single factor matrix across the different data views, which have assigned their respective factor loading matrices. A very similar motivation can be assigned to group FA (GFA) models where different variants have been introduced.[13,17] GFA groups variables and tries to find relationships between groups of variables (i.e., pathways) instead of correlating individual variables.

FA can be adopted for various applications or data requirements, as shown above. Different priors on the latent variables or adding additional terms allow modeling the special features different data types might bring with them as well as to infer latent variables of interest. A big advantage of FA is the interpretability of the factor loadings, which provides a way to interpret the kind of hidden variability that has been inferred from the data. Further, the measurement noise of an FA model can be defined explicitly. The standard FA model assumes normally distributed observation noise, different for every variable. Variants include data-type-specific noise assumptions, such as Bernoulli distributed noise models[12,16] or dropout noise.[15,16]

Gaussian process LVM

When data are very complex and difficult to interpret, non-parametric models might be the right choice for fitting the data. In the case of LVM, GP-LVMs[4–6] are such non-parametric models to infer a latent space from the data. The GP models a finite set of random function variables $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]$ as a joint Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^P$ and covariance $\boldsymbol{K} \in \mathbb{R}^{P \times P}$. Here, $\boldsymbol{x}_i$ is the ith data input to the function f. If we define f to follow a GP prior, we can write

$$f \sim \mathcal{GP}(\boldsymbol{\mu}, \boldsymbol{K}). \qquad \text{(Equation 4)}$$

Often, the mean is chosen to be zero $(\boldsymbol{\mu} = 0)$ and the covariance matrix is represented by a kernel matrix or function $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') : \mathcal{X} \times \mathcal{X}' \mapsto \mathbb{R}$, which measures the similarity between the inputs $\boldsymbol{x}$ and $\boldsymbol{y}$. Typical choices for kernel functions are the linear kernel or radial basis function (RBF) kernel. GP can, for example, be used for regression analysis, where we try to model a response variable $\boldsymbol{Y} \in \mathbb{R}^N$. This response variable can then be defined as

$$y_i \sim \mathcal{N}\left(f(\boldsymbol{x}_i), \sigma^2\right), \qquad \text{(Equation 5)}$$

where y_i is ith response modeled as a noisy version of the function value $f(\boldsymbol{x}_i)$ and the distribution of noise is Gaussian $\mathcal{N}(0, \sigma^2)$. In this work, we are interested in inferring latent variables of data sample $\boldsymbol{x}_i$, where also GPs can be used in a slightly different formulation. The GP-LVM can be written as

$$\boldsymbol{x}_i = f(\boldsymbol{z}_i) + \varepsilon \qquad \text{(Equation 6)}$$

with $\boldsymbol{z}_i$ being the lower-dimensional latent representation, with the noise assumed to follow a Gaussian distribution $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, and with f defined as a non-linear function with GP prior

(B) GP-LVM, where $\mathcal{GP}$ defines a Gaussian process prior and ε a noise term usually normally distributed around zero.
(C) AE consisting of an encoder and decoder network, where $\boldsymbol{X}_R$ stands for the reconstructed data.
(D) VAE, where the encoder predicts the parameter's mean and SD of a normally distributed latent representation. Different distributions of the latent representation can be modeled. The variables $\boldsymbol{X}_R$ and $\boldsymbol{X}_G$ stand for reconstructed and generated data, respectively.
(E) DNN where the latent representation is defined as the last hidden layer before label prediction.
(F) GAN, where the latent representation can be defined in the generator or discriminator networks. The variable X_G stands for the generated data.

$f \sim \mathcal{GP}(0, \boldsymbol{K})$, where the kernel matrix $\boldsymbol{K}$ serves as covariance matrix. In comparison with the linear FA, GP-LVM allows us to infer complex non-linear variables from the data if the kernel matrix $\boldsymbol{K}$ is defined as such; e.g., when using an RBF-kernel. The flexible GP-LVM framework allows us to modify the noise assumption of the model straightforwardly; for example, for ordinal or count data,[5] such as scRNA-seq data where the noise does not follow a Gaussian distribution. In general, non-Gaussian distributed noise is the often the case in biological applications, which makes GP-LVMs a valuable modeling approach. GP-LVM models can be trained via optimizing the log likelihood function and finding its maximum[6] or in a more efficient way via VI,[18] where the posterior distribution of the model is approximated. Using a standard linear kernel $\boldsymbol{K}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{X}^T$ and optimizing the model via maximum likelihood, the GP-LVM model reproduces the classic principal component analysis (PCA). However, GP-LVM corresponds to a probabilistic PCA with a VI due to the Bayesian formulation of the optimization problem. A rather schematic overview of this model can be seen in Figure 2B.

Different variants and their applications of GP-LVMs can be found in Buettner et al.,[19] where single-cell qPCR expression data are analyzed with a novel framework of GP-LVMs, introducing gene-relevant maps and gradient plots for better interpretation of the data. García et al.,[20] on the other hand, introduce an analysis of time-series data investigating physiological behavior. A GP-LVM was used to embed the dynamics of the data in a latent representation used for classification.

Deep learning

Here, we go more into detail on the different deep learning approaches for LVM and show different variants used in life sciences and translational studies. In general, the big advantage of deep learning models is the highly non-linear inference of the latent representation compared with FA.

AE

When looking for a non-linear dimensionality reduction for many different kind of data types, the AE is a model to think about. In other words, it can briefly be explained by sending the data through a bottleneck, which forms the latent representation, and is used to reconstruct the original data. This bottleneck ensures that only the important variation of the data is captured. In the simplest form, an AE[7] only consists of two layers and maps the data $\{\boldsymbol{x}_i\}_{i=1,\dots,N} = \boldsymbol{X} \in \mathbb{R}^{P \times N}$ into the latent representation $\{\boldsymbol{z}_i\}_{i=1,\dots,N} = \boldsymbol{Z} \in \mathbb{R}^{K \times N}$ as follows

$$\boldsymbol{z}_i = \sigma(\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{b}) \qquad \text{(Equation 7)}$$

where $\boldsymbol{W}$ is the weight matrix, $\boldsymbol{b}$ is a bias vector, and $\sigma(\bullet) : \mathcal{R} \mapsto \mathcal{R}_\sigma$ is called the activation function, where $\mathcal{R}_\sigma = [0, 1]$ when using a sigmoid activation function. This part of the network is called the encoder. To complete the two-layer AE we define

$$\boldsymbol{x}_{R,i} = \widehat{\sigma}(\widehat{\boldsymbol{W}}\boldsymbol{z}_i + \widehat{\boldsymbol{b}}) \qquad \text{(Equation 8)}$$

which maps from the latent representation back to the original data space $\{\boldsymbol{x}_{R,i}\}_{i=1,\dots,N} = \boldsymbol{X}_R \in \mathbb{R}^{P \times N}$ via reconstruction. The part of the network that maps from the latent representation back to the original data space is called decoder, where the weight matrix $\widehat{\boldsymbol{W}}$ and the bias vector $\widehat{\boldsymbol{b}}$ are the trainable parameters and $\widehat{\sigma}$ is the activation function, which can be similarly defined as σ above. To train an AE, typically the distance between original data points $\boldsymbol{x}_i$ and the reconstructed data points $\boldsymbol{x}_{R,i}$ is minimized, which can be written as a loss function

$$\mathcal{L}(x_i, x_{R,i}) = \|\boldsymbol{x}_i - \boldsymbol{x}_{R,i}\|^2 \qquad \text{(Equation 9)}$$

$$= \|\boldsymbol{x}_i - \widehat{\sigma}(\widehat{\boldsymbol{W}}\sigma(\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{b}) + \widehat{\boldsymbol{b}})\|^2 \qquad \text{(Equation 10)}$$

where $\| \bullet \|^2$ is the squared error. Generally, AEs are optimized using backpropagation,[21] which also applies for the upcoming models here, which means in this simple case that the parameters $\theta = \{\boldsymbol{W}, \widehat{\boldsymbol{W}}, \boldsymbol{b}, \widehat{\boldsymbol{b}}\}$ are optimized during training. It has been shown that an AE with a single fully connected hidden layer, a linear activation function, and a squared error cost function can be used for a simple PCA.[22] Adding more layers just means stacking them one after another, as in Equation 10. Figure 2C shows a schematic visualization of an AE with the latent representation $\boldsymbol{Z}$ in the middle of the encoder and decoder network.

AE models can vary a lot in their architecture and what they are intended to achieve. Apart from the one layer AE above, mostly deeper AE architectures are used and can include regularization of different natures. This regularization can be an additional loss term that, for example, shapes the latent space to help with better classification[23–26] or to solve data-specific tasks or requirements better.[27,28] Regularization can be added on any layer, not only the latent representation,[29] and can be used to define a sparse AE.[30] There are also many architectural variations, including, for example, stacked AE,[31,32] a group of AEs that forward the latent representation always to the next AE until the last provides the final latent space. Another variation are symmetrical AEs,[33] where the weights of encoder and decoder are shared. It is also popular to couple an AE with other model types, such as generative adversarial networks, to be able to sample the latent representation;[34] reinforcement learning models;[30] or logistic regression,[23] since an AE can shape the latent space for the respective needs. In many cases, AE architectures are also adapted to their input, such as for varying length of data input,[24] or convolutional AE,[26,31] such as to extract local features. Another example would be using recurrent NNs (RNNs) as encoder and decoder networks for time-dependent data.[35] RNNs were also used for data imputation tasks on an AE reconstructed output.[36] AEs are also used to have a combined latent representation of different input data types.[33]

VAE

A VAE[8,9] still consists of an encoder and decoder network as the classical AE introduced above in the section on AE, but VAEs differ significantly from an AE model when putting the architectural similarities aside. VAEs are generative directed probabilistic graphical models with a distributional assumption on the latent variables and optimize a VI problem. The VAE learns a stochastic mapping from the original observed data space to the latent representation $\boldsymbol{z}$ via the encoder $q_\varphi(\boldsymbol{z}|\boldsymbol{x})$ and back to the original data space when reconstructing or generating via the decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$, where φ and $\boldsymbol{\theta}$ are the parameters of the neural networks. In this case, $q_\varphi(\boldsymbol{z}|\boldsymbol{x})$ defines the approximation of the true posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$, which is in general intractable, and the neural network parameters φ are the variational parameters for the inference. The distributional assumption on the latent space is incorporated in the model via the prior distribution $p(\boldsymbol{z})$. More practically, we can define

$$(\boldsymbol{\mu}, \log\boldsymbol{\sigma}) = \mathrm{Encoder}_{\varphi}(\boldsymbol{x}) \qquad \text{(Equation 11)}$$

$$q_{\varphi}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma})) \qquad \text{(Equation 12)}$$

$$\boldsymbol{z}_R \sim q_{\varphi}(\boldsymbol{z}|\boldsymbol{x}) \qquad \text{(Equation 13)}$$

$$\boldsymbol{x}_R = \mathrm{Decoder}_{\boldsymbol{\theta}}(\boldsymbol{z}_R) \qquad \text{(Equation 14)}$$

$$\boldsymbol{z}_G \sim p(\boldsymbol{z}) \qquad \text{(Equation 15)}$$

$$\boldsymbol{x}_G = \mathrm{Decoder}_{\boldsymbol{\theta}}(p(\boldsymbol{z}_G)) \qquad \text{(Equation 16)}$$

where diag is the diagonal of a matrix, $\boldsymbol{x}_R$ is the reconstructed data where the latent space is sampled from $q_{\varphi}(\boldsymbol{z}|\boldsymbol{x})$ via the encoder network and $\boldsymbol{x}_G$ is generated data, where the latent space is sampled from the prior distribution $p(\boldsymbol{z})$. The encoder network predicts the mean $\boldsymbol{\mu}$ and diagonal $\log\boldsymbol{\sigma}$ of the covariance matrix of the distribution of the latent representation, if we assume a normally distributed latent representation. Optimizing a VAE model requires maximizing the evidence lower bound (ELBO)

$$\mathcal{L}_{\boldsymbol{\theta},\varphi}(\boldsymbol{x}) = \mathbb{E}_{q_{\varphi}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q_{\varphi}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \qquad \text{(Equation 17)}$$

where $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is the log likelihood of the reconstructed data and D_{KL} is the Kullback-Leibler divergence, which ensures the distribution of the latent representation to follow $p(\boldsymbol{z})$ when optimized. In many cases, the prior on the latent representation is trained to be multivariate standard Gaussian distributed $p(\boldsymbol{z}) = \mathcal{N}(0, \boldsymbol{I})$, where **0** is a zero mean vector and $\boldsymbol{I}$ the identity matrix as covariance, which makes it possible to derive a closed form solution for the Kullback-Leibler divergence. The prior distribution for the latent representation can also be chosen differently, which requires an alignment on the encoder network to predict the distributional parameters of the latent representation. For more details on the derivation of the loss function $\mathcal{L}_{\boldsymbol{\theta},\varphi}(\boldsymbol{x})$, we refer to Kingma and Welling[8] and Welling and Kingma.[9] Schematic visualization of a VAE model is provided in Figure 2D.

Many VAE models have been proposed for different modeling purposes or to meet certain data demands. VAEs are used to infer a latent representation specifically trained for respective prediction tasks,[37–40] where mostly additional optimization terms are added to the loss function, validating the classification performance. Erroneous predictions influence the latent representation and will be penalized during training via the loss function. Further, many VAE model architectures were designed to fulfill the needs of specific data requirements; for example, using RNNs in the encoder and decoder networks for time-series data,[41] having a Student's t-distributed data distribution,[42] or to graph data-specific VAE network architectures.[43] Those adjustments of the architecture of the VAE model aim at achieving a better fit of the model to the data. To combine multiple data-specific tasks, such as batch correction, visualization, clustering, and differential expression analysis, single-cell VI (scVI) was developed.[44] This scalable model framework assumes a zero-inflated negative binomial distribution conditioned on batch annotations for decoding the single-cell data accounting for the so-called dropout effect (similar to ZIFA,[15] introduced in the section on Factor analysis). To generate data of a requested data sub-type more precisely, conditional VAEs[45] are used, whereby a combination of the original input data and the label is forwarded to the latent representation and concatenated to form the input for the decoder network to generate data. VAEs were also designed to learn latent representations that form clusters of the input data in an unsupervised fashion. Therefore, similarity features of the input data in combination with a mixture of Gaussian assumption on the latent representation were used,[46] or the optimization of self-organizing maps with probabilistic cluster assignments was suggested.[47] Finally, the denoised reconstructed output of VAEs is used as input for RNNs for data generation purposes.[48]

DNN

When labels for the data are available, neural networks (NNs) can be used for LVM extracting the hidden layers $\boldsymbol{z}_{i,j}$ between the input layer and the output layer of the NN as latent representation. In this case, variable $i = 1, \ldots, N$ is the sample index of N independent input samples and $j = 1, \ldots, D$ is the index of the hidden layer, where D is the total number of hidden layers in the NN. In general, if D>1, the NN is usually called a DNN. Generally, the last hidden layer $\boldsymbol{z}_{i,D}$ is used as a latent representation, since this layer captures the most detailed lower-dimensional representation of the data. The more shallow layers at the beginning of the network typically have still-higher dimensions and therefore might spread the variance of the data across to many variables, but this depends on the architecture of the DNN. In common cases, the hidden layer with the lowest number of units, or in other words lowest dimensionality, is used as a latent representation. Figure 2E shows a schematic visualization of a DNN used for LVM with $j = D$, the latent representation as to the last hidden layer before prediction.

There are many variants of NNs used for LRL reported in the literature; for example, the classic case of using the last hidden layer before the classification output layer as latent representation.[49] In other cases, the hidden layers of the DNN are trained semi-supervised via the loss function; for example, using the mutual information criterion, which enhances discrimination of unsupervised points, and adding a multinomial logistic regression for samples with labels.[50] There also exist examples where two model types are combined, such as the combination of the word embedding model word2vec[51] with classical convolutional layers for prediction.[52] Others use the encoder part of a VAE to learn a latent representation with a defined distribution, which is directly used for prediction without reconstructing as in the classical VAE.[53] Li et al.[54] introduce a Siamese network where the hidden layers are trained in an adversarial fashion to adapt to each other for integrating a source and target domain into the same latent representation. This goes in a similar direction as DNN architectures that have multiple input layers for different input types, where, in the deeper regions of the DNN, the hidden layers get concatenated to one specific latent representation of the multiple views from the input.[55,56] In general, no architectural limits exist for LVM with DNNs where, for example, the depth of the network, types of layers or regularization used, and combinations of models vary a lot and provide an interpretable latent representation of complex high-dimensional data.

Generative adversarial network

A generative adversarial network (GAN)[10,11] consists of two competitive NNs playing a game, and they try to fool each other during training. On the one hand, a generator network $G(\cdot): \mathbb{R}^D \mapsto \mathbb{R}^P$ is trained to generate fake data $\boldsymbol{x}_G$, which is close in terms of similarity to the input data $\boldsymbol{x}$, from a sample $\zeta \in \mathbb{R}^D$ of a simple prior distribution. This prior distribution can, for example, be multivariate standard Gaussian distributed. Therefore, we can write for the generation process

$$\boldsymbol{x}_G = G(\zeta). \qquad \text{(Equation 18)}$$

On the other hand, a discriminator network $D(\cdot): \mathbb{R}^P \mapsto \mathbb{R}_{[0,1]}$ is trained to find the differences between real and fake data, which is generated by the generator network, and predicts whether the input is fake or real. The lower-dimensional latent representation $\boldsymbol{z}_i$ of data point $\boldsymbol{x}_i$ that we are interested in can be found similarly as with the DNNs in the hidden layers of the GAN itself. Mostly the last hidden layer of the discriminator network before classification is therefore used. This game between generator and discriminator comes with two cost functions, one for each of both networks. The cost function for the discriminator network

$$L^D_{\theta_D,\theta_G}(\boldsymbol{x},\zeta) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}\log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\zeta}\log(1 - D(G(\zeta))) \qquad \text{(Equation 19)}$$

which is nothing else but the standard cross-entropy cost function for classification tasks, where $\boldsymbol{\theta}_D$ and $\boldsymbol{\theta}_G$ are the model parameters for the discriminator and generator network, respectively. It is important to mention that the discriminator is trained on two minibatches of data, one with label 1 for the batch coming from the training data $\boldsymbol{x}$ itself and one minibatch with label 0 coming from the generator network $G(\zeta)$ and samples from the prior distribution. When the discriminator network is optimized, the parameters for the generator network $\boldsymbol{\theta}_G$ are fixed and vice versa. The simplest game between discriminator and generator is the so-called zero-sum game (or minimax game since the solution involves minimization and maximization), where

$$L^G_{\theta_D,\theta_G} = -L^D_{\theta_D,\theta_G}. \qquad \text{(Equation 20)}$$

There are many more optimization methods, which usually involve a different cost function for the generator network, such as the heuristic non-saturating game or the maximum likelihood game. Optimization of both networks requires finding a so-called Nash-equilibrium, which is in general more difficult and unstable than optimizing an objective function, such as with a VAE, which is also a generative model. In comparison, GANs can suffer from mode collapse where the generator specializes to fool the discriminator with a specific data mode, whereas VAEs tend to generate more blurry and less sharp data. Hence, both techniques can be combined in so-called adversarial AEs.[57] A schematic overview of GANs can be found in Figure 2F.

GAN architectures are discussed widely in the literature and are beyond the scope of this review, but Goldsborough et al.[58] classically used the last hidden layer of the discriminator network as a latent representation of the data. Ghahramani et al.[59] did the very same thing, but also used the hidden layer of the generator network as the latent representation for downstream analysis.

APPLICATIONS IN BIOLOGY AND TRANSLATIONAL MEDICINE

In general, multiple data types with different properties have been used to infer latent variables, such as imaging, single-cell omics, graph-structured, time-series, or text data. The latent variables capture different kinds of information, which we want to discuss in the following section. With the different types of input data, many general tasks have been performed; for example, pre-processing of the data, classification, clustering, visualization of the data in lower dimensions, variance decomposition with or without prior knowledge, downstream analysis based on the latent space, or data generation for augmentation or design of new data. In principle, all the data types can be combined with all the different tasks mentioned above. A schematic overview of these various combinations is depicted in Figure 3A.

In the remainder of this section, we provide an overview of LVM from an application's point of view. We discuss applications at varying scales of biological systems, and start by discussing latent variables inferred from measurements of individual biomolecules in and on the surface of cells, followed by analyzing measurements of collections of cellular components such as proteomes. Afterward, we consider a tissue-level perspective and review the various applications of LVM on single-cell omics measurements as well as medical imaging data. We conclude with the application of LVM on clinical physiological patient data with the aim of explaining phenotypes that cannot be measured directly. We depict the structure of this section in Figure 3B and link all the different models with their application and investigated data types in Table 1.

Designing new biomolecule target structures

The structure of a protein or biomolecule can be described as the arrangement of atoms. Those structures are typically defined in discrete graph domains. Defining a latent representation for graph inputs is challenging since it involves finding an optimal object in a finite and large set of objects. The finite set of potential molecules is estimated to be 10^{23}, whereas only about 10^8 substances have ever been synthesized,[37] Naive exhaustive search through this discrete space to create objects with desired properties, such as activity against a specific protein, solubility, or ease of synthesis, is intractable.[38] LVM provides an elegant solution, where the discrete molecular structured data are embedded in a continuous latent representation.[32–34,37,38,43,45,48,60–62] The latent space is supposed to capture in the latent variables the similarity of the graph structures. This is not a trivial task due to non-trivial choice of the many different available graph distance measures. Once the graphs are embedded in the continuous space, the exact distances and similarities between two discrete graph structures can be efficiently computed. Once the continuous latent representation is trained and defined, one can then optimize the representation and search efficiently for potential

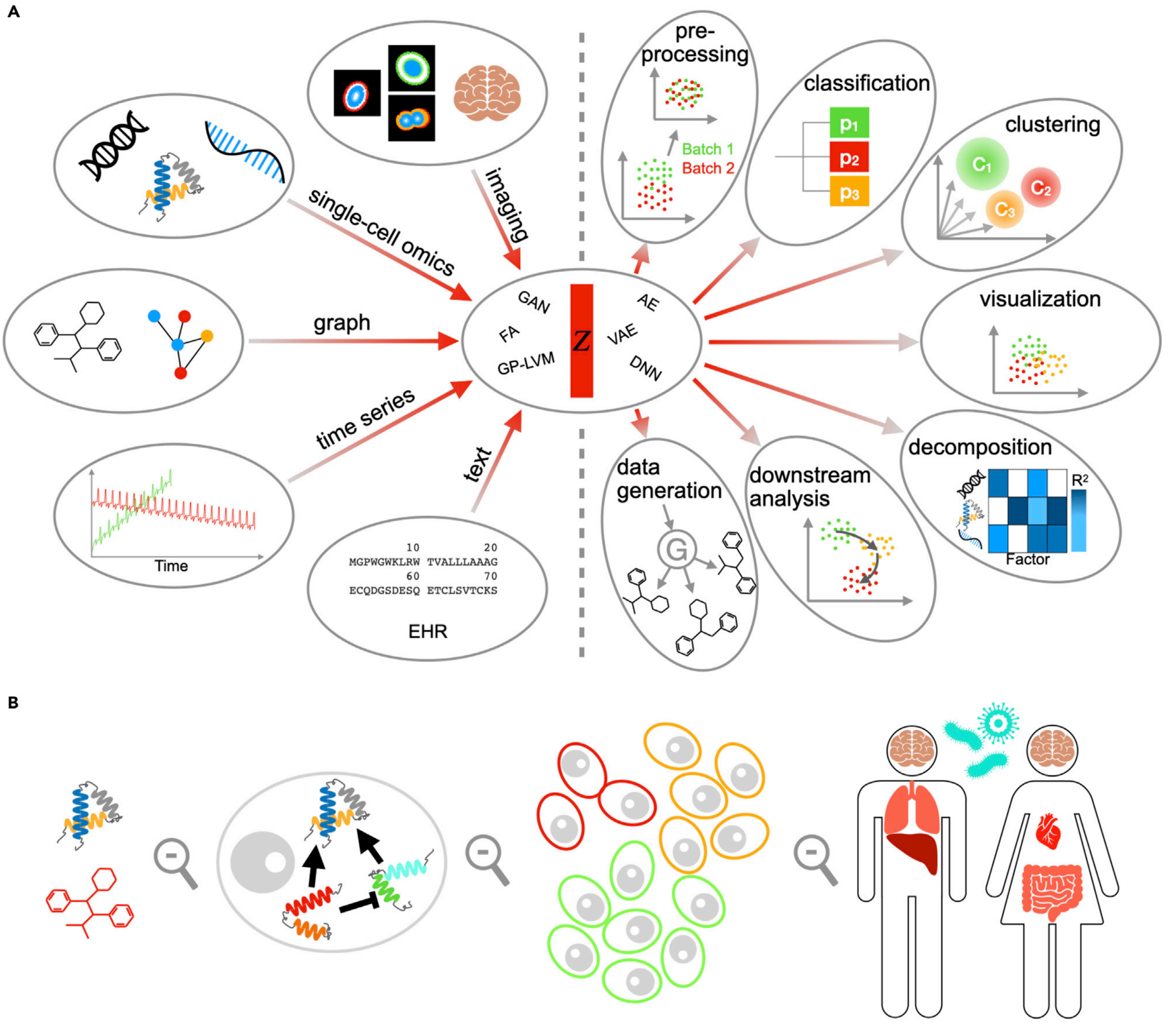

Figure 3. Schematic overview of LVM applications associated to different levels of biology and translational medicine

(A) Different data types (left) are used as input and several applications (right) are reported such as pre-processing, classification, clustering, visualization, decomposition, downstream analysis, and data generation. Typical data types, such as imaging data, single-cell omics, graphs, time series, or text data, were commonly modeled.

(B) LVM was applied at different levels in biological or translational studies, starting with analyzing particles within the cell (see section on Designing New Biomolecule Target Structures) and how those interact with each other (Learning function-associated cellular interaction networks from sequentially embedded data). LVM was also heavily applied to measurements describing cells as a whole (see section on Inferring Cellular Variation Across Several Axes, Including Space, Phenotype, Or Time). Last but not least, LVM was used for modeling clinical trials (see section on Clinical applications of LVM).

targets, which then can be generated and transferred back into the discrete visualizable format. There are different ways the new design of the molecule can be achieved, one of which is by perturbing the latent representation before generating from it or interpolating between molecules.[37,48] Gradient-based optimizations are more powerful, where efficient guidance toward functional compounds is performed via trained classifiers on the latent representation,[37,60,61] transfer learning approaches (using learned knowledge on related tasks),[34] or to learn a conditional latent representation.[38] For the later approaches, labels need to be provided for training purposes. For the application of generating and designing new data with desirable properties, LVM provides a huge boost, especially with the growth of deep learning approaches since they provide a highly non-linear way to embed discrete data types in continuous latent representations, as discussed above.

Learning function-associated cellular interaction networks from sequentially embedded data

DNA, RNA, or protein information can be captured in sequential data (i.e., nucleotide or amino acid sequences), a data type difficult to analyze. The challenges can arise due to the variable

Table 1. Overview of all cited LVM approaches and their applications

Section	Method	Type	Data type	Functionality	Citation
Designing new biomolecule target structures	–	VAE	SMILES	downstream analysis	Gomez-Bombarelli R. et al.[37]
	ECAAE	entangled conditional adversarial AE	molecular graphs	data generation	Polykovskiy D. et al.[38]
	–	graph convolutional network	protein structure data	downstream analysis	Aumentado-Armstrong T.[60]
	–	VAE	protein structure data	data generation	Greener J. G. et al.[45]
	–	AE	Graphs	graph representation learning	Tran P. V.[33]
	NEVAE	VAE	molecular graphs	representation learning	Samanta et al.[43]
	–	AE	protein structure data	dimensionality reduction	Alam et al.[32]
	LatentGAN	AE, GAN	molecular graphs	data generation	Prykhodko et al.[34]
	GraphNVP	invertible normalizing flow	molecular graphs	representation learning	Madhawa et al.[61]
	–	VAE, convolutional AE, RNN	SMILES	data generation	Skalic et al.[48]
	–	AE	SMILES, IUPAC	domain transfer, classification	Winter et al.[62]
Learning function-associated cellular interaction networks from sequentially embedded data	–	AE	DNA sequence	representation learning	Agarwal et al.[24]
	–	VAE	protein sequence	downstream analysis	Ding et al.[63]
	BindSpace	StarSpace	DNA sequence	domain adaption, classification	Yuan et al.[64]
	–	CNN	RNA sequences	classification	Pan and Shen,[52]
	PIPR	Siamese residual RCNN	protein sequence	similarity learning	Chen et al.[65]
	DeepConv-DTI	neural network	molecular graphs	classification, latent representation	Lee et al.[55]
	Dr. VAE	VAE	drug response, transcriptomics	prediction	Rampášek et al.[40]
	–	FA, matrix factorization	drug response	classification	Gönen and Margolin,[66]
	cwKBMF	kernelized Bayesian matrix factorization	drug response	multi-view representation learning	Ammad-ud-din et al.[67]
	GFA	group FA	fMRI, drug response	representation learning	Klami et al.[13]
Inferring cellular variation across several axes, including space, phenotype, or time	–	neural network	RNA-seq	clustering	Lin et al.[49]
	scScope	RNN	RNA-seq	clustering	Deng et al.[36]
	scVI	deep probabilistic model	RNA-seq	dimensionality reduction, imputation, clustering, normalization, batch correction	Lopez et al.[44]
	SIMLR	multikernel learning	RNA-seq	dimensionality reduction	Wang et al.[68]
	t-SNE	stochastic neighbor embedding	multi-omics	dimensionality reduction, visualization	van der Maaten et al.[69]
	UMAP	uniform manifold approximation	multi-omics	dimensionality reduction, visualization	McInnes et al.[70]
	SISUA	VAE	multi-omics	heterogeneity decomposition	Trong et al.[39]
	MoE-Sim-VAE	VAE	image, CyTOF	clustering	Kopf et al.[46]
	Slalom	FA	RNA-seq	heterogeneity decomposition	Buettner et al.[16]
	ZIFA	FA	RNA-seq	dimensionality reduction	Pierson and Yau,[15]

(*Continued on next page*)

Table 1. *Continued*

Section	Method	Type	Data type	Functionality	Citation
	Scvis	VAE	RNA-seq	dimensionality reduction	Ding et al.[42]
	Tybalt	VAE	RNA-seq	downstream analysis	Way and Greene,[71]
	Dhaka	VAE	RNA-seq	representation learning	Rashid et al.[72]
	MDA-CNN	AE, neural network	miRNA	similarity learning, classification	Peng et al.[25]
	MOFA	FA	multi-omics	downstream analysis, heterogeneity decomposition	Argelaguet et al.[12]
	–	AE, supervised AE	image	classification	Zeune et al.[26]
	–	AE	RNA-seq	disentangled representation	Wang et al.[73]
	SAUCIE	AE	mass cytometry	clustering, imputation, batch correction	Amodio et al.[29]
	CellCNN	CNN	mass cytometry	cell type identification	Arvaniti et al.[74]
	SIMLR	multikernel learning	RNA-seq	dimensionality reduction	Wang et al.[68]
	–	GP-LVM	qPCR	representation learning	Buettner and Theis,[19]
Clinical applications of LVM	deep RIT	neural network, RIT	MRI	segmentation, classification	Deng et al.[50]
	InfoMask	variational neural network	image	segmentation	Taghanaki et al.[53]
	–	latent topic model	image	automatic annotation	Cruz-Roa et al.[75]
	–	convolutional AE, AE	MRI	representation learning	Jaiswal et al.[31]
	–	VAE	fMRI	classification, domain transfer	Han et al.[76]
	–	AE, logistic regression	fMRI	classification	Bzdok et al.[23]
	–	neural network	text	word embeddings	Wehbe et al.[77]
	–	neural network	image	pseudo-time	Eulenberg et al.[78]
	–	neural network, AE	image	classification, survival analysis	Bello et al.[27]
	–	VAE	ECG	clustering	Rajan and Thiagarajan,[41]
	SOM-VAE	VAE	time series	clustering	Fortuin et al.[79]
	DPSOM	VAE, self-organizing maps	time series	clustering	Laura Manduchi et al.[47]
	–	GP-LVM	multimodal	classification	García et al.[20]
	Deep Patient	AE	EHRs	representation learning	Miotto et al.[80]
	–	unsupervised neural language model	EHRs	representation learning	Stojanovic et al.[81]

The column Type provides information about the modeling approach used, whereas the column Data Type summarizes the data types the model was applied on. In the column Application we summarize which modeling goals have been tackled with the respective approach. CNN, convolutional neural network; GAN, generative adversarial network; RIT, robust information theoretic; SMILES, simplified molecular-input line-entry system.

length of the sequences or when defining a proper similarity between them. LVM provides a solution to this problems, since sequences can be embedded in continuous latent representations, with latent variables capturing biologically relevant variability and similarity, such as for splice site classification from DNA sequences[24] or analyzing evolutionary properties, fitness, and stability from protein sequences.[63] To predict multi-class transcription factor binding sites, the target sequence of the transcription factors and the DNA sequences have been embedded in a shared latent representation.[64] Another study focuses on RNA-protein interfaces, which play critical roles in processes such as mRNA degradation and stability, as well as alternative splicing. Those RNA-protein binding sites have been predicted from human RNA sequences of various lengths via a latent representation inferred using LVMs.[52] Finally, protein interaction partners were predicted from a latent representation that has been inferred only using protein sequences.[65]

Protein sequences constitute a basis for drug target discovery. The high level of difficulty to experimentally identify suitable protein drug targets renders more efficient computational approaches attractive alternatives. Latent representations from protein sequence embedding have allowed prediction of drug–target interactions, and to detect potential binding sites for drugs.[55] The response of drugs on gene expression levels in terms of viability and transcriptomic perturbation[40] or drug susceptibility[66] have been analyzed and predicted using LVMs. The interaction between genes and proteins captured in pathways has been exploited for genomic drug response predictions.[67] Moreover, LVMs clustered specific cancer cell lines and chemical descriptors based on the gene expression response due to drug treatment.[13]

Inferring cellular variation across several axes, including space, phenotype, or time

Recent developments of single-cell measurement technologies allowed to achieve a better understanding of tissue- or phenotype-specific differences between single cells. While such technologies are capable of defining the high-dimensional molecular profile of a single cell, it is not possible to directly measure which cell or molecular makeup conveys tissue function or other complex phenotypes. However, these cell characteristics can be inferred from gene or protein expression, which can nowadays be measured in high dimensions with various techniques.

The identification of different cell types is a popular application of LVM and has been applied extensively.[36,44,49,68] The latent variable cell type is thereby inferred when combining multiple measurements of gene or protein expressions. State-of-the-art dimensionality reduction techniques frequently applied to group cell types in a latent representation are t-distributed stochastic neighbor embedding (t-SNE)[69] and uniform manifold approximation and projection (UMAP).[70] Another study combined gene and protein expressions, such as using protein quantification for constraining the learning process of the lower-dimensional representation of scRNA-seq data[39] to separate the cell types in the latent space. Cell subpopulations in peripheral blood mononuclear cells (PBMCs) have also been modeled and clustered via mass cytometry incorporating prior knowledge about the similarity of the data, which influences the training of the latent representation.[46] Prior knowledge in terms of pathway annotations has also been used for inferring latent representations for variance decomposition between biological and technical variation, novel subpopulation discovery, and interpretability of them via scRNA-seq data.[16] Technology-specific measurement noise characteristics have been explicitly taken into account. For instance, scRNA-seq measurements exhibit the dropout effect, where the mRNA of random genes of single cells could not be amplified, resulting in false-zero signals. This effect was in particular modeled with an LVM for dimensionality reduction and cell type identification,[15] since falsely missing gene measurements can strongly influence the cell type definition, in particular for rare cell types. Mapping cell measurements from high to a lower dimensional representations requires preservation of local and global distances and was modeled using a probabilistic LVM.[42]

Tissue- or organism-level phenotypes, such as disease states, are frequently conveyed by, and are therefore associated with, specific cell types or subpopulations. These associations have been inferred by LVM approaches. Gene and protein expression play a central role when inferring the latent variable explaining the phenotype variation of interest, since they define the potential subpopulations, and potentially constitute the basis of the mechanism conferring their association with the phenotype. LVM has been applied to single-cell measurements of tissues originating from different cancer types to infer cancer specific subpopulations.[25,42,71,72] Argelaguet et al.[12] developed an LVM to infer the variation of chronic lymphocytic leukemia patients across multiple omics levels, unraveling the variation shared and also specific for the respective measurement types of the cancer patients. In another study, different classes of circulating tumor cells from blood samples have been identified via fluorescence imaging techniques via a deep LVM approach.[26] Different phenotypes have also been reported to be detected via LVM; e.g., different treatment conditions,[73] subpopulations in 11 million T cells from dengue patients in India,[29] or the inference of a latent variable that associates to each cell a disease onset association measure applied on PBMC samples from HIV patients.[74] The very same approach was applied to PBMC samples of MS patients in comparison with healthy donors measured via mass cytometry to identify a disease-associated subpopulation identifiable via a disease-associated latent variable.[82]

Latent variables have been used to capture variation along time or developmental stages of biological processes using snapshot single-cell expression data of genes or proteins. A specific example is the inference of latent variables capturing the different stages in the cell cycle.[16,68] In another study, Buettner et al.[19] inferred a latent space that allowed them to resolve the differences in gene expressions for all developmental stages and identification of new subpopulations in mouse fetal development, from zygote to blastocyst.

Clinical applications of LVM

LVMs have been used to infer the health or disease state of patients on the basis of health records, physiological parameters, and radiological data.[1] Typical for such studies is the difficulty of directly measuring disease manifestations in patients. LVM provides a solution by deriving latent variables for these manifestations from measurements, such as imaging of organs, physiological signals, or clinical parameters.

Radiological approaches have enabled multimodal imaging of every organ in the human body. MRI or fMRI is extensively used to monitor brain activity. The resulting images allow three-dimensional reconstruction of brain tissue composition. Deng et al.[50] presented a study utilizing LVM for tissue type segmentation, overcoming the time-consuming and difficult manual annotation of image structures and color details. Further applications of image segmentation include localization of pneumonia from patients' chest X-ray images to pin down disease localizations[53] or for automatic annotation of histopathological images[75] from latent representations, respectively. Another common task in image analysis is feature extraction, traditionally requiring expert domain knowledge. Extracting features from brain MRI using LVM without the aforementioned expert knowledge helped to classify patient status into healthy, Alzheimer diseased, or having mild cognitive impairment or autism spectrum disorder from the brain structure only.[31] Using fMRI techniques, functional phenotypes, such as activity or cognitive tasks during various stimuli, have been associated with specific brain regions using latent representations inferred from the fMRI,[23,76,77] Most diseases can be classified in different stages, and therefore the progression of such diseases can be built up in temporal order, such as diabetic retinopathy, possibly leading to blindness. Eulerberg et al.[78] have shown that LVM has the potential to evince the disease progression from snapshot color fundus photographs of the eyes only.

The progression of diseases can often be inferred from time-series data. Temporal dependencies and models can be trained to recognize patterns along the time axis responsible for the presence or development of disease phenotypes. For instance, time-resolved imaging of cardiac motion from patients diagnosed with pulmonary hypertension was the basis to predict survival via LVM.[27] Multi-channel electrocardiogram (ECG) was used to predict disease status[41] or vital-sign time-series measurements of the intensive care unit (ICU), which have been clustered with respect to the patients' future physiological states[47,79] via LVM. Time-series data typically need to be modeled differently compared with other data formats due to their sequential structure possibly comprising temporal dependencies. LVM can be utilized to find similarities between different time series in latent variables and extract those into the latent representation. Emotions constitute a canonical example for latent variables, since they cannot be measured with any device directly. LVM has, for example, been applied to infer the affective state of an emotional process from multimodal physiological signals.[20] In contrast to the approaches mentioned earlier, Weng et al.[30] used LVMs to infer personalized optimal glycemic trajectories for septic patients from the patients' clinical features.

Treatment details of patients, disease state and progression, change in phenotype, and many more details are so far recorded in written form. Those electronic health records (EHRs) constitute a source of information about successful treatments, possibly suggested or at least supported via automatic approaches. To detect subtle symptoms at early disease stages, LVM can be applied to text data such as EHR to embed those into a continuous representation, which has many advantages, such as capturing semantic meaning and defining a similarity measure. The resulting features on the latent representation of EHR have helped in recent studies to, for example, assess probabilities for patients to develop various diseases for early recognition,[80] or predicting healthcare quality such as length of stay, total incurred charges, or mortality rates of patients.[81]

REMARKS AND CONCLUSION

LRL has been a field of research, possibly under varying names, and applied for a long time. Recent dramatic advances in deep learning yielded a strong impetus to LVM, in particular due to the introduction of model types such as AE, VAE, or GANs. Those models found various applications in biological and translational studies and resulted in numerous new findings in the respective fields. The big advantage of deep learning approaches over classical FA approaches is the ability to extract highly non-linear properties and the local feature extraction via convolutional computations, in particular for images. Further, deep learning allowed straightforward regularization of latent representations, such as distributional assumptions (see the section on VAE) or regularization to preserve local and global structures of the original data.[42]

On the other hand, there is also still room for improvement concerning the interpretability of the latent variables inferred from deep learning approaches. While classical FA methods allow us to investigate the factor loading matrix and hence identify features that are most important for the respective factor, the highly non-linear structure of stacked layers in deep learning models does not allow us to straightforwardly extract this information. Nevertheless, there exist methods for deep learning that allow us to extract features from the original data responsible for variation in the latent representation. Mostly, those features can be traced back using gradient-based methods[83–88] or perturbation-based methods,[89] which are in general slower. Furthermore, we believe that a theoretical understanding of deep learning, in general, will also help in the future for improving the interpretability of LVM. The idea of being able to infer a biologically disentangled representation[90–92] is appealing. This would mean that single variables of the latent representation encode very specific biologically relevant features, such as cell type, differentiation state, cell cycle stage, disease state, and many more.[93] Coupled with generative models, single cells with very specific properties could be generated and amplified straight away; for example, for the analysis of very rare cell types.

Overall, the use of LVM approaches often is motivated by the lack of labels or phenotypes for the data. Hence, unsupervised modeling approaches, such as FA or (variational) AEs, are of great importance. They allow us to gather a better understanding of the data due to reduced dimensions and grouping of objects based on similarities. The choice of the model type typically depends on the data properties defined by the type of the data. scRNA-seq data, for instance, consist of gene counts with non-Gaussian measurement noise. Many approaches discussed above (Table 1) explicitly assume parametric noise models for the data, making up a pivotal part of defining sensible models for biological data analysis. In addition, the scientific questions and the corresponding modeling goal affect the selection of the LVM type. Many models, such as (variational) AE or DNNs allow us to take into account supervision in terms of labels in the learning process, and therefore enable us to derive a latent representation tailored for label predictions. Semi-supervised learning paradigms can be applied using deep learning-based approaches if only a few data samples are labeled. Classical and deep learning models also have been combined, such as FA with a VAE[94] to combine the best of both worlds; e.g., scalability with increasing number of samples, as ensured via the VAE and interpretability as provided via FA.

Latent representations will be increasingly used to derive fundamental physical quantities such as process coordinates. For instance, latent representations of high-dimensional single-cell data have been interpreted as a process coordinate; i.e., pseudo-time of cellular differentiation processes. Different approaches have been proposed to this end, with the common concept of converting the high-dimensional data into a more interpretable latent representation from which the pseudo-time is then inferred. Some methods are based on GP-LVMs,[95,96] where the choice of the kernel plays an important role in inferring the pseudo-time. Others make use of t-SNE,[97] diffusion maps,[98–100] locally linear embedding,[101] PCA,[97,99,102–104] or independent component analysis.[105] Most commonly, the pseudo-times are inferred from high-dimensional scRNA-seq or mass cytometry measurements.

In summary, this review exemplifies the increasingly widespread use of LVM across a plethora of applications in the life sciences. We expect their application scope to widen and their formal concepts to be further developed at the fundamental level as well as in task-specific terms, and thereby ultimately support the interpretation and progress in the different domains of the life sciences.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback and insightful suggestions to improve this review. A.K. was supported by the grants SystemsX.ch HDL-X and PHRT 2017-103.

AUTHOR CONTRIBUTIONS

Conceptualization, A.K. and M.C.; investigation, A.K.; writing A.K. and M.C.; funding acquisition, M.C.; resources, M.C.; supervision, M.C.

REFERENCES

1. Curado, M.A.S., Teles, J., and Marôco, J. (2014). Analysis of variables that are not directly observable: influence on decision-making during the research process. Revista da Escola de Enfermagem da USP *48* (1), 146–152.

2. Thompson, A.J., Banwell, B.L., Barkhof, F., Carroll, W.M., Coetzee, T., and Comi, G. (2018). Diagnosis of multiple sclerosis: 2017 revisions of the mcdonald criteria. Lancet Neurol. *17* (2), 162–173.

3. Bishop, C.M. (2006). Pattern Recognition and Machine Learning, *volume 4* (Springer).

4. N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. NIPS'03: Proceedings of the 16th International Conference on Neural Information Processing Systems, Page 329–336, 2004.

5. Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. J. Mach. Learn. Res. *6*, 1783–1816.

6. Li, P., and Chen, S. (2016). A review on Gaussian process latent variable models. CAAI Trans. Intell. Technol. https://doi.org/10.1016/j.trit.2016.11.004.

7. Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning internal representations by error propagation. . Parallel Distributed Processing, *Vol 1* (Foundations. MIT Press).

8. D.P. Kingma and M. Welling. Auto-encoding variational bayes. 2nd International Conference on Learning Representations, ICLR 2014, 2014.

9. Welling, M., and Kingma, D.P. (2019). An introduction to variational autoencoders. Mach. Learn. *12*, 307–392, https://doi.org/10.1561/2200000056.

10. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. . Advances in Neural Information Processing Systems, *27* (NIPS).

11. Ian, J. (2016). Goodfellow. Nips 2016 Tutorial: Generative Adversarial Networks (NIPS).

12. Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J.C., Buettner, F., Huber, W., and Stegle, O. (2018). Multi-omics factor analysis–a framework for unsupervised integration of multi-omics data sets. Mol. Syst. Biol. *14* (6), e8124.

13. Klami, A., Virtanen, S., Leppaaho, E., and Kaski, S. (2014). Group factor analysis. arXiv, arXiv:1411.5799.

14. Ghahramani, Z., and Matthew, J.B. (1999). Variational inference for bayesian mixtures of factor analysers. In Advances in Neural Information Processing Systems, *12* (NIPS).

15. Pierson, E., and Yau, C. (2015). Zifa: dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biol. *16*, 241.

16. Buettner, F., Pratanwanich, N., McCarthy, D.J., Marioni, J.C., and Stegle, O. (2017). f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq. Genome Biol. *18*, 212.

17. Nikolova, O., Moser, R., Kemp, C., Gönen, M., and Margolin, A.A. (2017). Modeling gene-wise dependencies improves the identification of drug response biomarkers in cancer studies. Bioinformatics *33* (9), 1362–1369.

18. Titsias, M., and Neil, D. (2010). Lawrence. Bayesian Gaussian process latent variable model. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, *1*, pp. 366–376.

19. Buettner, F., and Theis, F.J. (2012). A novel approach for resolving differences in single-cell gene expression patterns from zygote to blastocyst. Bioinformatics *28* (18), i626–i632.

20. García, H.F., Álvarez, M.A., and Orozco, Á.A. (2016). Gaussian process dynamical models for multimodal affect recognition. In 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 850–853, https://doi.org/10.1109/EMBC.2016.7590834.

21. Linnainmaa, S. (1970). The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors, Master Thesis (University of Helsinki).

22. Plaut, E. (2018). From principal subspaces to principal components with linear autoencoders. arXiv, arXiv:1804.10253.

23. Bzdok, D., Eickenberg, M., Grisel, O., Bertrand, T., and Varoquaux, G. (2015). Semi-supervised factored logistic regression for high-dimensional neuroimaging data. In Advances in Neural Information Processing Systems, *28* (NIPS).

24. Agarwal, V., Jayanth Kumar Reddy, N., and Anand, A. (2019). Unsupervised representation learning of DNA sequences. arXiv, arXiv:1906.03087.

25. Peng, J., Hui, W., Li, Q., Chen, B., Hao, J., Jiang, Q., Shang, X., and Wei, Z. (2019). A learning-based framework for miRNA-disease association identification using neural networks. Bioinformatics *35*, 4364–4371.

26. Zeune, L.L., Boink, Y.E., van Dalum, G., Nanou, A., de Wit, S., Andree, K.C., Swennenhuis, J.F., van Gils, S.A., Terstappen, L.W.M.M., and Christoph, B. (2020). Deep learning of circulating tumour cells. Nat. Mach Intell. *2*, 124–133.

27. Bello, G.A., Dawes, T.J.W., Duan, J., Biffi, C., de Marvao, A., Howard, L.S.G.E., Gibbs, J.S.R., Wilkins, M.R., et al. (2019). Deep-learning cardiac motion analysis for human survival prediction. Nat. Mach Intell. *1*, 95–104.

28. H. He, C. Liu, and H. Liu. Model reconstruction from small-angle x-ray scattering data using deep learning methods. iScience *Volume* 23, Issue 3, 100906, 2020. doi:https://doi.org/10.1016/j.isci.2020.100906.

29. Amodio, M., van Dijk, D., Srinivasan, K., Chen, W.S., Mohsen, H., Moon, K.R., Campbell, A., Zhao, Y., Wang, X., Venkataswamy, M., et al. (2019). Exploring single-cell data with deep multitasking neural networks. Nat. Methods *16*, 1139–1145, https://doi.org/10.1038/s41592-019-0576-7.

30. Weng, W.-H., Gao, M., He, Z., Yan, S., and Szolovits, P. (2017). Representation and reinforcement learning for personalized glycemic control in septic patients. arXiv, arXiv:1712.00654.

31. Jaiswal, A., Guo, D., Raghavendra, C.S., and Thompson, P. (2018). Large-scale unsupervised deep representation learning for brain structure. arXiv, arXiv:1805.01049.

32. Alam, F.F., Rahman, T., and Shehu, A. (2019). Learning reduced latent representations of protein structure data. In Proceedings of the 10th ACM International Conference on Bioinformatics, pp. 592–597, https://doi.org/10.1145/3307339.3343866.

33. Tran, P.V. (2018). Learning to make predictions on graphs with autoencoders. arXiv. arXiv:1802.08352. https://doi.org/10.1109/DSAA.2018.00034.

34. Prykhodko, O., Johansson, S.V., Kotsias, P.-C., ArÃ°s-Pous, J., Bjerrum, E.J., Engkvist, O., and Chen, H. (2019). A de novo molecular generation method using latent vector based generative adversarial network. J. Cheminform. *11*, 74.

35. Kimmel, J., Brack, A., and Marshall, W. (2019). Deep convolutional and recurrent neural networks for cell motility discrimination and prediction.

IEEE/ACM Trans. Comput. Biol. Bioinform. https://doi.org/10.1109/TCBB.2019.2919307.

36. Deng, Y., Bao, F., Dai, Q., Wu, L.F., and Altschuler, S.J. (2019). Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. Nat. Methods *16*, 311–314.

37. Gómez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent. Sci. *4*, 268–276.

38. Polykovskiy, D., Zhebrak, A., Vetrov, D., Ivanenkov, Y., Aladinskiy, V., Mamoshina, P., Bozdaganyan, M., Aliper, A., Zhavoronkov, A., and Kadurin, A. (2018). Entangled conditional adversarial autoencoder for de novo drug discovery. Mol. Pharmaceutics *15*, 4398–4405.

39. Trong, T.N., Mehtonen, J., Gonzalez, G., Kramer, R., Hautamäki, V., and Heinäniemi, M. (2019). Semisupervised generative autoencoder for single-cell data. J. Comput. Biol. *27*, https://doi.org/10.1089/cmb.2019.0337.

40. Rampášek, L., Hidru, D., Smirnov, P., Haibe-Kains, B., and Goldenberg, A. (2019). Dr.vae: improving drug response prediction via modeling of drug perturbation effects. Bioinformatics *35*, 3743–3751.

41. Rajan, D., and Thiagarajan, J.J. (2018). A generative modeling approach to limited channel ecg classification. arXiv, arXiv:1802.06458.

42. Ding, J., Condon, A., and Shah, S.P. (2018). Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nat. Commun. *9*, 2002.

43. Samanta, B., De, A., Jana, G., Chattaraj, P.K., Ganguly, N., and Gomez Rodriguez, M. (2019). Nevae: a deep generative model for molecular graphs. In Proceedings of the AAAI Conference on Artificial Intelligence.

44. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. Nat. Methods *15*, 1053–1058.

45. Greener, J.G., Moffat, L., and Jones, D.T. (2018). Design of metalloproteins and novel protein folds using variational autoencoders. Sci. Rep. *8*, 16189.

46. Kopf, A., Fortuin, V., Ram Somnath, V., and Claassen, M. (2019). Mixture-of-experts variational autoencoder for clustering and generating from similarity-based representations. arXiv, arXiv:1910.07763.

47. Manduchi, L., Hüser, M., Vogt, J., Rätsch, G., and Fortuin, V. (2020). DPSOM: deep probabilistic clustering with self-organizing maps. arXiv, arXiv:1910.01590.

48. Skalic, M., Jiménez, J., Sabbadin, D., and De Fabritiis, G. (2019). Shape-based generative modeling for de novo drug design. J. Chem. Inf. Model. *59*, 1205–1214.

49. Lin, C., Jain, S., Kim, H., and Bar-Joseph, Z. (2017). Using neural networks for reducing the dimensions of single-cell RNA-seq data. Nucleic Acids Res. *45*, e156.

50. Deng, Y., Bao, F., Deng, X., Wang, R., Kong, Y., and Dai, Q. (2016). Deep and structured robust information theoretic learning for image analysis. IEEE Trans. Image Process. *25*, 4209–4221, https://doi.org/10.1109/TIP.2016.2588330.

51. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv, arXiv:1301.3781.

52. Pan, X., and Shen, H.-B. (2018). Learning distributed representations of RNA sequences and its application for predicting RNA-protein binding sites with a convolutional neural network. Neurocomputing *305*, 51–58.

53. Taghanaki, S.A., Havaei, M., Berthier, T., Dutil, F., Di Jorio, L., Hamarneh, G., and Bengio, Y. (2019). Infomask: masked variational latent representation to localize chest disease. arXiv, arXiv:1903.11741.

54. Li, J., Qiu, S., Du, C., Wang, Y., and He, H. (2019). Domain adaptation for EEG emotion recognition based on latent representation similarity. IEEE Trans. Cogn. Developmental Syst. *12*, 344–353, https://doi.org/10.1109/TCDS.2019.2949306.

55. Lee, I., Keum, J., and Nam, H. (2019). DeepConv-DTI: prediction of drug-target interactions via deep learning with convolution on protein sequences. PLoS Comput. Biol. https://doi.org/10.1371/journal.pcbi.1007129.

56. Schreiber, J., Durham, T., Bilmes, J., and Noble, W.S. (2020). Avocado: a multi-scale deep tensor factorization method learns a latent representation of the human epigenome. Genome Biol. *21*, 81.

57. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. arXiv, arXiv:1511.05644.

58. Goldsborough, P., Pawlowski, N., Caicedo, J.C., Singh, S., and Carpenter, A.E. (2017). Cytogan: generative modeling of cell images. bioRxiv. https://doi.org/10.1101/227645.

59. Ghahramani, A., Watt, F.M., and Luscombe, N.M. (2018). Generative adversarial networks uncover epidermal regulators and predict single cell perturbations. bioRxiv. https://doi.org/10.1101/262501.

60. Aumentado-Armstrong, T. (2018). Latent molecular optimization for targeted therapeutic design. arXiv, arXiv:1809.02032.

61. Madhawa, K., Ishiguro, K., Nakago, K., and Abe, M. (2019). GraphNVP: an invertible flow model for generating molecular graphs. arXiv, arXiv:1905.11600.

62. Winter, R., Montanari, F., Noé, F., and Clevert, D.-A. (2019). Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. Chem. Sci. *10*, 1692–1701.

63. Ding, X., Zou, Z., Charles, L., and Brooks, III. (2019). Deciphering protein evolution and fitness landscapes with latent space models. Nat. Commun. *10*, 5644.

64. Yuan, H., Kshirsagar, M., Zamparo, L., Lu, Y., and Leslie, C.S. (2019). Bindspace decodes transcription factor binding signals by large-scale sequence embedding. Nat. Methods *16*, 858–861.

65. Chen, M., Ju, C.J.T., Zhou, G., Chen, X., Zhang, T., Chang, K.-W., Zaniolo, C., and Wang, W. (2019). Multifaceted protein–protein interaction prediction based on siamese residual rcnn. Bioinformatics *35*, i305–i314.

66. Gönen, M., and Margolin, A. (2014). Drug susceptibility prediction against a panel of drugs using kernelized bayesian multitask learning. Bioinformatics *30* (17), i556–i563, https://doi.org/10.1093/bioinformatics/btu464.

67. Ammad ud din, M., Khan, S.A., Malani, D., Murumägi, A., Kallioniemi, O., Aittokallio, T., and Kaski, S. (2016). Drug response prediction by inferring pathway-response associations with kernelized bayesian matrix factorization. Bioinformatics *32*, i455–i463.

68. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D., and Batzoglou, S. (2017). Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. Nat. Methods *14*, 414–416.

69. van der Maaten, L.J.P., and Hinton, G.E. (2008). Visualizing high-dimensional data using t-SNE. J. Mach. Learn. Res. *9*, 2579–2605.

70. McInnes, L., and Healy, J. (2018). UMAP: uniform manifold approximation and projection for dimension reduction. arXiv, arXiv:1802.03426.

71. Way, G.P., and Greene, C.S. (2018). Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. Pac. Symp. Biocomput. *23*, 80–91.

72. Rashid, S., Shah, S., Bar-Joseph, Z., and Pandya, R. (2019). Dhaka: variational autoencoder for unmasking tumor heterogeneity from single cell genomic data. Bioinformatics, btz095, https://doi.org/10.1093/bioinformatics/btz095.

73. Wang, Z., Grace, H., Yeo, T., Sherwood, R., and Gifford, D. (2019). Disentangled representations of cellular identity. Int. Conf. Res. Comput. Mol. Biol. *11467*, 256–271.

74. Arvaniti, E., and Claassen, M. (2017). Sensitive detection of rare disease-associated cell subsets via representation learning. Nat. Commun. *8*, 14825.

75. Cruz-Roa, A., Diaz, G., Romero, E., and Gonzalez, F.A. (2011). Automatic annotation of histopathological images using a latent topic model based

on non-negative matrix factorization. J. Pathol. Inform. *2*, https://doi.org/10.4103/2153-3539.92031.

76. Han, K., Wen, H., Shi, J., Lu, K.H., Zhang, Y., Fu, D., and Liu, Z. (2019). Variational autoencoder: an unsupervised model for modeling and decoding fmri activity in visual cortex. Neuroimage *198*, 125–136, https://doi.org/10.1016/j.neuroimage.2019.05.039.

77. Wehbe, L., Vaswani, A., Knight, K., and Mitchell, T. (2014). Aligning context-based statistical models of language with brain activity during reading. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 233–243, https://doi.org/10.3115/v1/D14-1030.

78. Eulenberg, P., Köhler, N., Blasi, T., Filby, A., Carpenter, A.E., Rees, P., Theis, F.J., and Alexander Wolf, F. (2017). Reconstructing cell cycle and disease progression using deep learning. Nat. Commun. *8*, 463.

79. Fortuin, V., Hüser, M., Locatello, F., Strathmann, H., and Rätsch, G. (2019). SOM-VAE: interpretable discrete representation learning on time series. ICLR, 2018.

80. Miotto, R., Li, L., Kidd, B.A., and Dudley, J. (2016). Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. Sci. Rep. *6*, 26094.

81. Stojanovic, J., Gligorijevic, D., Radosavljevic, V., Djuric, N., Grbovic, M., and Obradovic, Z. (2017). Modeling healthcare quality via compact representations of electronic health records. IEEE/ACM Trans. Comput. Biol. Bioinform. *14*, 545–554, https://doi.org/10.1109/TCBB.2016.2591523.

82. Galli, E., Hartmann, F.J., Schreiner, B., Ingelfinger, F., Arvaniti, E., Diebold, M., Mrdjen, D., van der Meer, F., Krieg, C., Al Nimer, F., et al. (2019). GM-CSF and CXCR4 define a T helper cell signature in multiple sclerosis. Nat. Med. *25*, 1290–1300.

83. Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps (ICLR Workshop).

84. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One. https://doi.org/10.1371/journal.pone.0130140.

85. Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. (2016). Not just a black box: learning important features through propagating activation differences. arXiv, arXiv:1605.01713.

86. Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. Proc. 34th Int. Conf. Machine Learn.

87. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recogn. https://doi.org/10.1016/j.patcog.2016.11.008.

88. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2019). Grad-CAM: visual explanations from deep networks via gradient-based localization. Int. J. Computer Vis. 2019.

89. Matthew, D. (2014). Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Network (European Conference on Computer Vision).

90. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In 30th Conference on Neural Information Processing Systems (NIPS).

91. Yan, X., Yang, J., Sohn, K., and Lee, H. (2016). Attribute2Image: conditional image generation from visual attributes (European Conference on Computer Vision (ECCV)).

92. Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., and Samaras, D. (2017). Neural face editing with intrinsic image. arXiv, arXiv:1704.04131.

93. J.C. Kimmel. Disentangling latent representations of single cell RNA-seq experiments.bioRxiv bioRxiv, 2020. doi:doi.org/10.1101/2020.03.04.972166.

94. Svensson, V., Gayoso, A., Yosef, N., and Pachter, L. (2020). Interpretable factor models of single-cell RNA-seq via variational autoencoders. Bioinformatics *36*, 3418–3421.

95. Reid, J., and Wernisch, L. (2016). Pseudotime estimation: deconfounding single cell time series. Bioinformatics *32*, 2973–2980.

96. Ahmed, S., Rattray, M., and Boukouvalas, A. (2019). GrandPrix: scaling up the bayesian GPLVM for single-cell data. Bioinformatics *35*, 47–54.

97. Marco, E., Karp, R.L., Guo, G., Robson, P., Hart, A.H., Trippa, L., and Yuan, G.-C. (2014). Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. Proc. Natl. Acad. Sci. U S A *111*, E5643–E5650.

98. Haghverdi, L., Büttner, M., Wolf, F.A., Buettner, F., and Theis, F.J. (2016). Diffusion pseudotime robustly reconstructs lineage branching. Nat. Methods *13*, 845–848.

99. Setty, M., Tadmor, M.D., Reich-Zeliger, S., Angel, O., Salame, T.M., Kathail, P., Choi, K., Bendall, S., Friedman, N., and Pe'er, D. (2016). Wishbone identifies bifurcating developmental trajectories from single-cell data. Nat. Biotechnol. *34*, 637–645.

100. Macnair, W., De Vargas Roditi, L., Ganscha, Stefan, and Claassen, Manfred (2019). Tree-ensemble analysis assesses presence of multifurcations in single cell data. Mol. Syst. Biol. *15*, https://doi.org/10.15252/msb.20188552.

101. Welch, J.D., Hartemink, A.J., and Prins, J.F. (2016). Slicer: Inferring branched, nonlinear cellular trajectories from single cell RNA-seq data. Genome Biol. *17*, 106.

102. Matsumoto, H., and Kiryu, H. (2016). Scoup: a probabilistic model based on the Ornstein–Uhlenbeck process to analyze single-cell expression data during differentiation. BMC Bioinformatics *17*, 232.

103. Ji, Z., and Ji, H. (2016). TSCAN: pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. Nucleic Acids Res. *44*, e117.

104. Shin, J., Berg, D.A., Zhu, Y., Shin, J.Y., Song, J., Bonaguidi, M.A., Enikolopov, G., Nauen, D.W., Christian, K.M., Ming, G.L., and Son, H. (2015). Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. Cell Stem Cell *17* (3), 360–372, https://doi.org/10.1016/j.stem.2015.07.013.

105. Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N.J., Livak, K.J., Mikkelsen, T.S., and Rinn, J.L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat. Biotechnol. *32*, 381–386.