Tutorial: Cell identification from scRNA-seq data with Variational Autoencoders

Hugo Gangloff

INRAE

MIA
PARIS-SACLAY

HappyR Session, 14 avril 2023

Outline

1. Variational Autoencoders
2. Gaussian Mixture VAE
3. Data
4. Code organization
5. Time to experiment
6. Analysis of the results

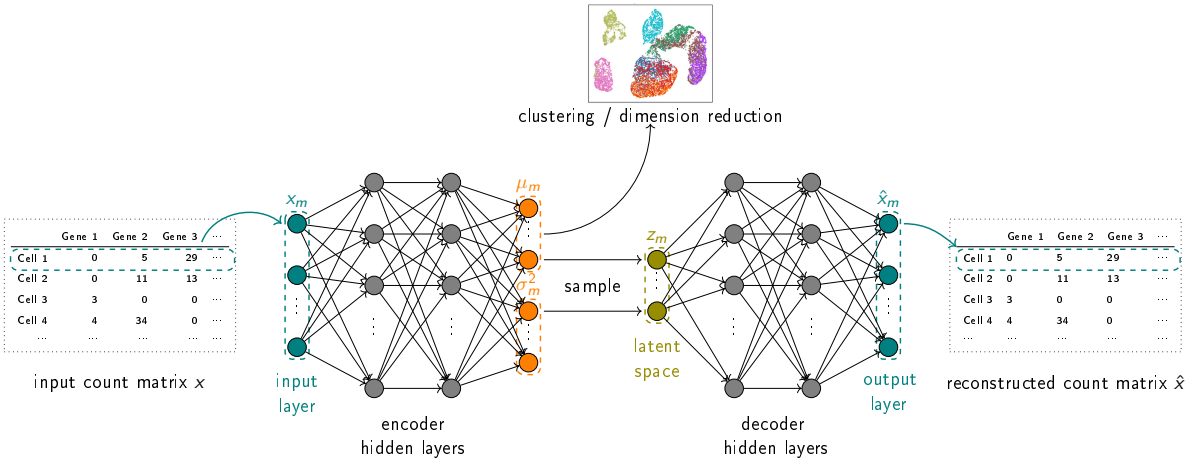
The goals of this tutorial are:

- Conception of a Variational Autoencoder architecture with `Pytorch`
- Conception of a Gaussian Mixture Variational Autoencoder architecture with `Pytorch`
- Apply the new model to scRNA-seq cell identification with `scanpy` and `anndata`

This presentation accompanies the two jupyter-notebooks

Variational Autoencoders

Variational Autoencoder (VAE): deep learning view



VAE for scRNA-seq data

Variational Autoencoder (VAE): deep learning view

For an unnormalized count matrix x with M rows (cells) and N columns (lines):

- a row $x_m \in \mathbb{N}^N, m \in \{1, \dots, M\}$ is the VAE input
- x_m goes through a fully connected neural network, the encoder, with K^e layers and parameters φ (weights and biases)
- the outputs of the encoder are $\mu_m \in \mathbb{R}^P$ and $\sigma_m^2 \in (\mathbb{R}_*^+)^P$
- the input of the decoder, $z_m \in \mathbb{R}^P$ (latent space random variables) are sampled from a Gaussian with mean μ_m and variance σ_m^2.
- z_m goes through a fully connected neural network, the decoder, with K^d layers and parameters θ (weights and biases)
- the outputs of the decoder are $\hat{x}_m \in \mathbb{N}^N$.

→ **Parallel computations on GPU: process a batch of x_m at the same time!**

VAE: probabilistic view

- In VAEs, for each input vector x_m of size N, we have the following generative model:

$$p_\theta(x_m, z_m) = p_\theta(z_m)p_\theta(x_m|z_m) \text{ where } \begin{cases} p_\theta(z_m) & = \mathcal{N}(z_m, 0, I_P), \\ p_\theta(x_m|z_m) & = \prod_{n=1}^{N} p_\theta(x_{m,n}|z_m) \rightarrow \textit{likelihood} \end{cases}$$

  $\rightarrow$ **The likelihood is parametrized by the output of the decoder.**

- We also have the inference model, $q_\varphi(z_m|x_m)$, where

$$q_\varphi(z_m|x_m) = \mathcal{N}(z_m, \mu_m, \sigma_m^2 I_P) \rightarrow \textit{variational distribution}$$

  $\rightarrow$ The variational distribution q_φ needs to be learnt in order to approximate the true intractable posterior $p_\theta(z_m|x_m)$. **The variational distribution is parametrized by the output of the encoder.**

VAE: probabilistic view

- The VAE is trained by maximizing the evidential lower bound (ELBO), it is a lower bound of the marginal log-likelihood $\log p_\theta(x_m)$,

$$\log p_\theta(x_m) \geqslant \mathcal{L}_{\theta,\varphi}(x_m) = \mathbb{E}_{q_\varphi(z_m|x_m)}[\log p_\theta(x_m|z_m)] - D_{KL}(q_\varphi(z_m|x_m)||p_\theta(z_m))$$

  (it follows $\log p_\theta(x_1, \ldots, x_M) = \sum_{m=1}^{M} \log p_\theta(x_m) \geqslant \sum_{m=1}^{M} \mathcal{L}_{\theta,\varphi}(x_m)$)
  $\rightarrow$ VAEs are trained in a unsupervised way
- The likelihood $p_\theta(x_m|z_m)$ is dependent on the application. We now present 3 classical distributions that will be tested on the scRNA-seq data.
- More details about VAEs can be found in (Kingma et al. 2013)

Continuous Bernoulli (CB) likelihood

- The output of the VAE can be modeled as independent CB variables.
- This supposes that $x_{m,n}$ are in $[0, 1] \rightarrow$ the count matrix needs to be normalized.
- Let $\lambda \in [0, 1]^N$ be the output of the decoder, then:

$$p_\theta(x_m|z_m) = \prod_{n=1}^{N} p_\theta^{CB}(x_{m,n}|\lambda_n) = \prod_{n=1}^{N} C(\lambda_n)\lambda_n^{x_{m,n}}(1-\lambda_n)^{1-x_{m,n}}$$

where

$$C(\lambda_n) = \frac{2tanh^{-1}(1-2\lambda_n)}{1-2\lambda_n}$$

- (Loaiza-Ganem et al. 2019) uses this distribution for VAEs

Negative Binomial (NB) likelihood

- In case of counting data, the output of the VAE can be modeled as independent NB variables
- Let $r \in (\mathbb{R}_*^+)^N$ and $p \in [0,1]^N$ be the outputs of the decoder, then:

$$p_\theta(x_m|z_m) = \prod_{n=1}^{N} p_\theta^{NB}(x_{m,n}|r_n, p_n) = \prod_{n=1}^{N} \frac{\Gamma(x_{m,n} + r_n)}{\Gamma(r_n)x_{m,n}!} p_n^{r_n}(1-p_n)^{x_{m,n}}$$

- (Zhao et al. 2020) uses this distribution for VAEs

Zero Inflated Negative Binomial (ZINB) likelihood

- In case of counting data, the output of the VAE can be modeled as independent ZINB variables
- Let $r \in (\mathbb{R}_*^+)^N$, $p \in [0,1]^N$ and $\rho \in [0,1]^N$ be the outputs of the decoder, then:

$$p_\theta(x_m|z_m) = \prod_{n=1}^{N} p_\theta^{ZINB}(x_{m,n}|r_n, p_n, \rho_n) = \prod_{n=1}^{N} \begin{cases} \rho_n + (1-\rho_n)p_\theta^{NB}(x_{m,n}|r_n, p_n), & x = 0, \\ (1-\rho_n)p_\theta^{NB}(x_{m,n}|r_n, p_n), & x > 0. \end{cases}$$

- (Grønbech et al. 2020) uses this distribution for VAEs

Poisson (Poiss) likelihood

- In case of counting data, the output of the VAE can be modeled as independent Poisson variables
- Let $\lambda \in (\mathbb{R}_*^+)^N$ be the output of the decoder, then:

$$p_\theta(x_m|z_m) = \prod_{n=1}^{N} p_\theta^{Poiss}(x_{m,n}|\lambda_n) = \prod_{n=1}^{N} \frac{\lambda_n^{x_{m,n}} e^{-\lambda_n}}{\Gamma(x_{m,n}+1)}.$$

Zero Inflated Poisson (ZIPoiss) likelihood

- In case of counting data, the output of the VAE can be modeled as independent ZIPoisson variables
- Let $\lambda \in (\mathbb{R}_*^+)^N$ and $\rho \in [0,1]^N$ be the outputs of the decoder, then:

$$p_\theta(x_m|z_m) = \prod_{n=1}^{N} p_\theta^{ZIPoiss}(x_{m,n}|\lambda_n, \rho_n) = \prod_{n=1}^{N} \begin{cases} \rho_n + (1-\rho_n)p_\theta^{Poiss}(x_{m,n}|\lambda_n), & x = 0, \\ (1-\rho_n)p_\theta^{Poiss}(x_{m,n}|\lambda_n), & x > 0. \end{cases}$$

Gaussian Mixture VAE

Gaussian Mixture VAE I

- Let C be the number of clusters
- Using the previous notations, we further introduce y_m, an hidden categorical latent random variable with values in $\{1, \dots, C\}$
  $\rightarrow$ **Gaussian Mixture prior** to better structure the latent space
- Following (Grønbech et al. 2020), we have the generative network:

$$p_\theta(x_m, y_m, z_m) = p_\theta(y_m)p_\theta(z_m|y_m)p_\theta(x_m|y_m, z_m)$$

with:

$$\begin{cases} p_\theta(y_m) & = \text{Cat}(y_m, \pi), \\ p_\theta(z_m|y_m) & = \mathcal{N}(z_m, \mu_\phi(y_m), \sigma^2_\phi(y_m) I_P), \\ p_\theta(x_m|y_m, z_m) & = p_\theta(x_m|z_m) = \prod_{n=1}^{N} p_\theta^{ZINB}(x_{m,n}|z_m), \end{cases}$$

where we use the ZINB likelihood and choose π as an equiprobable distribution.

Gaussian Mixture VAE II

- We also have the inference network:

$$q_\phi(z_m, y_m|x_m) = q_\phi(y_m|x_m)q_\phi(z_m|x_m, y_m)$$

where

$$\begin{cases} q_\phi(y_m|x_m) & = \text{Cat}(y_m, \pi_\phi(x_m)), \\ q_\phi(z_m|x_m, y_m) & = \mathcal{N}(z_m, \mu_\phi(x_m, y_m), \sigma^2_\phi(x_m, y_m)I_P). \end{cases}$$

- The GMVAE is trained by maximizing the ELBO (for a sample x_m):

$$\mathcal{E}_{\theta,\phi}(x_m) = \mathbb{E}_{q_\phi(z_m,y_m|x_m)}[\log p_\theta(x_m|z_m, y_m)] - D_{KL}(q_\phi(z_m, y_m|x_m)||p_\theta(z_m, y_m)).$$

Gaussian Mixture VAE III

- We can show that, with the hypotheses above:

$$\mathcal{E}_{\theta,\phi}(x_m) = \sum_{k=1}^{K} \pi_{\phi,k}(x_m) \Big[\mathbb{E}_{q_\phi(z_m|x_m,y_m=k)}[\log p_\theta(x_m|z_m)] - D_{KL}(q_\phi(z_m|x_m, y_m = k)||p_\theta(z_m|y_m = k)) - D_{KL}(q_\phi(y_m = k|x_m)||p_\theta(y_m = k)) \Big]$$

- The y_m is categorical $\rightarrow$ it should be treated with care: **a particular reparametrization trick** is needed!

VAEs for scRNA-seq data in the literature

Some major papers on the topic are

- (Wang et al. 2018): VAE + Binary cross entropy (data scaled to $[0, 1]$)
- (Dony et al. 2020): VAE + VAMP prior + Negative binomial likelihood
- (Grønbech et al. 2020): Gaussian Mixture VAE + Zero inflated likelihoods
- (Seninge et al. 2021): Semi-supervised VA

Data

Datasets

1. **Mouse Pancreas Single-cell RNA-Seq Dataset**
   822 cells with 14878 genes from 13 cell types (Baron et al. 2016). Also used with a Generative Adversarial Network in (Bahrami et al. 2021)

2. **Peripheral Blood Mononuclear Cells**
   9 datasets of different purified cell types from (Zheng et al. 2017)
   → 92043 cells with 32738 genes (before preprocessing and subsampling, see notebook) → Also studied in (Grønbech et al. 2020).

Code organization

Code organization

- `data` folder:
  - `files` folder: contains the scRNA-seq count matrix files
  - `datasets.py`: contains one class for each dataset to analyze
- `models` folder:
  - `distributions` folder: the probability distribution classes that we use as model likelihoods
  - `vae.py`: the VAE class
  - `gmvae.py`: the GMVAE class
  - `utils.py`: utility functions (the Multilayer Perceptron submodule)
- `saved_model_parameters` folder: contains the saved parameters of different models
- `(Part 1) Cell identification from scRNA-seq data with Variational Autoencoders.ipynb`
- `reparametrization_trick.qmd`
- `(Part 3) Gaussian Mixture VAE for scRNA-seq data.ipynb`

Time to experiment

Analysis of the results

Average Silhouette Width (ASW)

- Proposed in (Rousseeuw 1987) to evaluate the performance of a given result of a clustering method
- For a data point $\mu_{m,n} \in C_I$, let

$$a(\mu_{m,n}) = \frac{1}{|C_I| - 1} \sum_{\mu \in C_I, \mu \neq \mu_{m,n}} d(\mu_{m,n}, \mu) \text{ and } b(\mu_{m,n}) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{\mu \in C_J} d(\mu_{m,n}, \mu),$$

for some distance d. Then the ASW score is

$$s(\mu_{m,n}) = \frac{b(\mu_{m,n}) - a(\mu_{m,n})}{\max(a(\mu_{m,n}), b(\mu_{m,n}))}$$

- This metric is used in (Dony et al. 2020)

Activity of latent random variables

- Proposed in (Burda et al. 2015) to *evaluate whether latent dimensions encode useful information about the data* $\rightarrow$ *we would expect its distribution to change depending on the observations.*
- The activities score of the latent variable z_m is a real vector of size P such that

$$A_{z_m} = \text{diag}(\text{Cov}_{x_m}(\mathbb{E}_{z_m \sim q_\varphi(z_m|x_m)}[z_m]))$$

- A latent random variable $z_{m,p}$ is considered active if $A_{z_{m,p}} > 0.01$.
- We can also visualize, alternatively, the latent-space covariance matrix
- This metric is used in (Dony et al. 2020)

References I

[1] M. Bahrami, M. Maitra, C. Nagy, G. Turecki, H. R. Rabiee, and Y. Li. "Deep feature extraction of single-cell transcriptomes by generative adversarial network". In: *Bioinformatics* 37.10 (2021), pp. 1345–1351.

[2] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, et al. "A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure". In: *Cell systems* 3.4 (2016), pp. 346–360.

[3] Y. Burda, R. Grosse, and R. Salakhutdinov. "Importance weighted autoencoders". In: *arXiv preprint arXiv:1509.00519* (2015).

[4] L. Dony, M. König, D Fischer, and F. J. Theis. "Variational autoencoders with flexible priors enable robust distribution learning on single-cell RNA sequencing data". In: *ICML 2020 Workshop on Computational Biology (WCB) Proceedings Paper*. Vol. 37. 2020.

[5] C. H. Grønbech, M. F. Vording, P. N. Timshel, C. K. Sønderby, T. H. Pers, and O. Winther. "scVAE: variational auto-encoders for single-cell gene expression data". In: *Bioinformatics* 36.16 (2020), pp. 4415–4422.

[6] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[7] G. Loaiza-Ganem and J. P. Cunningham. "The continuous Bernoulli: fixing a pervasive error in variational autoencoders". In: *Advances in Neural Information Processing Systems* 32 (2019).

[8] P. J. Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[9] L. Seninge, I. Anastopoulos, H. Ding, and J. Stuart. "VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics". In: *Nature communications* 12.1 (2021), pp. 1–9.

[10] D. Wang and J. Gu. "VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder". In: *Genomics, proteomics & bioinformatics* 16.5 (2018), pp. 320–331.

References II

[11] H. Zhao, P. Rai, L. Du, W. Buntine, D. Phung, and M. Zhou. "Variational autoencoders for sparse and overdispersed discrete data". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 1684–1694.

[12] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, et al. "Massively parallel digital transcriptional profiling of single cells". In: *Nature communications* 8.1 (2017), pp. 1–12.