Deep generative models for biologists

Clayton W. Seitz

January 21, 2022

Outline

The logic of generative modeling

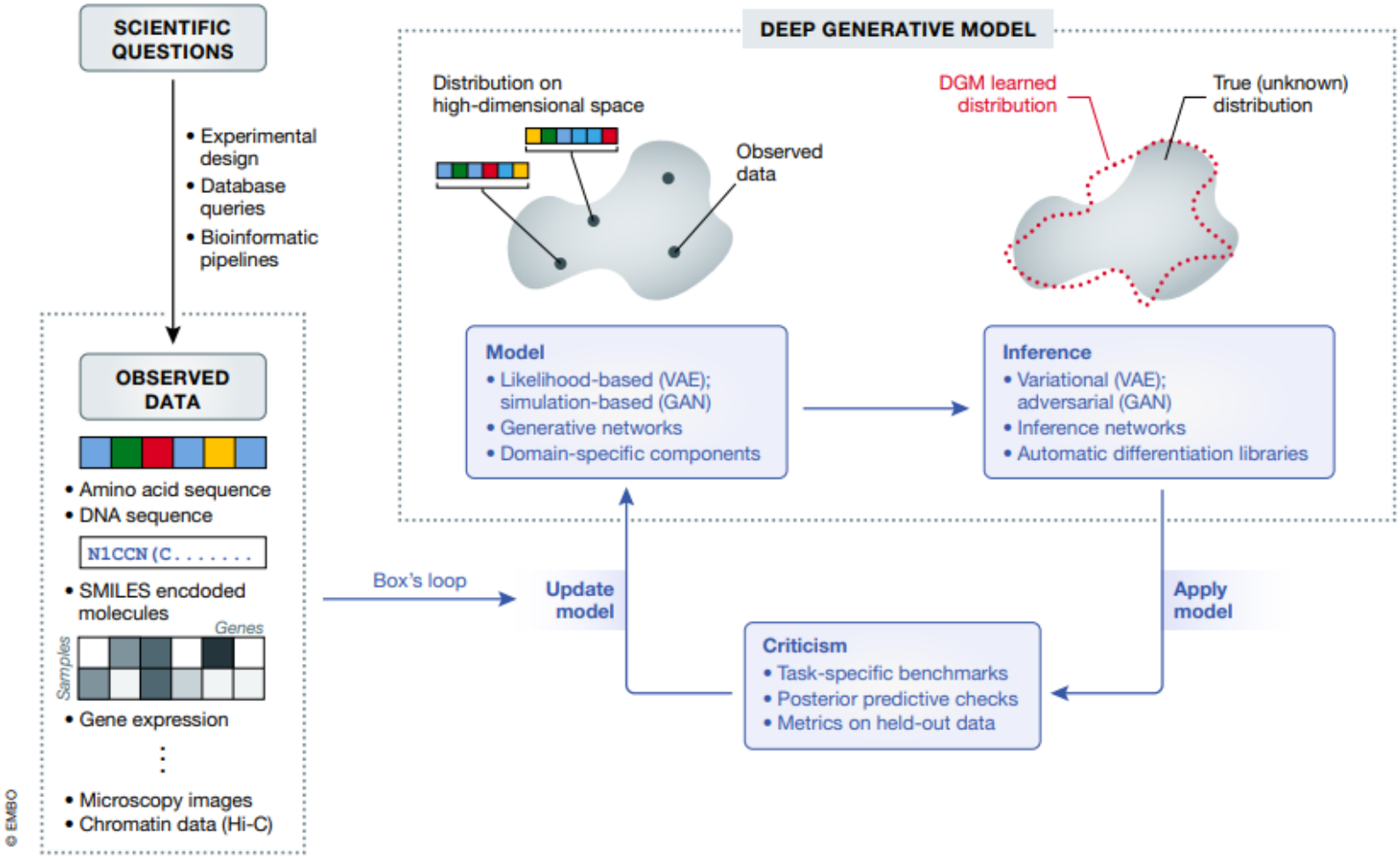
Say we have a set of variables $\mathbf{x} = (x_1, x_2, ..., x_n)$ which might have some statistical dependence

The variable $\mathbf{x}$ might be an amino acid sequence, gene expression data, microscopy image, etc.

- Often we are handed a batch of empirical samples $\{\mathbf{x}_i\}_{i=1}^N$
- We want to know the generating distribution $p(\mathbf{x})$

In supervised generative learning, we try to explicity learn the joint distribution $p(\mathbf{x}) = \prod_{i=1}^{N-1} p(x_i|x_{i+1:N})p(x_N)$, which is generally more difficult than discriminative learning.

Applying deep generative models to biological data



Perks of generative modeling

- Fitting complete multivariate distributions $p(\mathbf{x})$ goes beyond correlation-based or clustering approaches
- Correlations cannot discover partial correlation in the context of other neighbors
- Fitting $p(\mathbf{x})$ permits sampling based inference

Generative learning: probabilistic graphical models

PGMs may aid in the discovery of gene networks that drive complex diseases

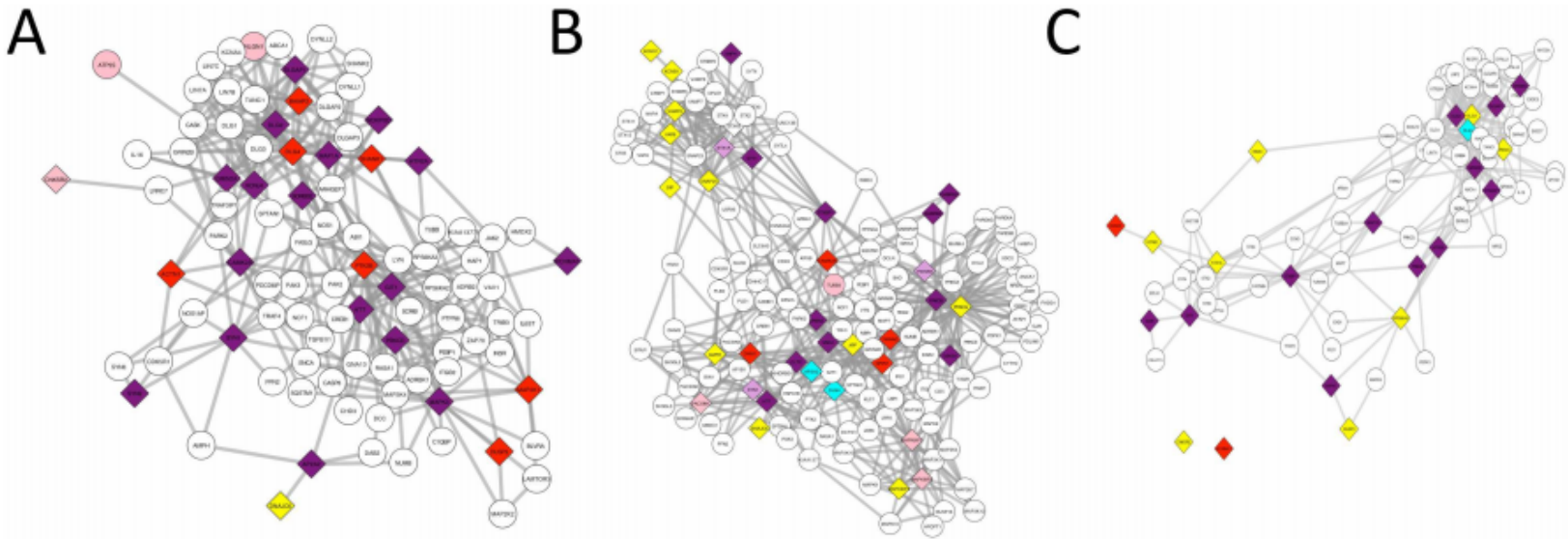


Figure 1: **Bayesian networks** for gene network discovery in (A) Schizophrenia (B) Epilepsy (C) Autism. Taken from Mezlini et al. 2017

Probabilistic graphical models (PGMs) are generative

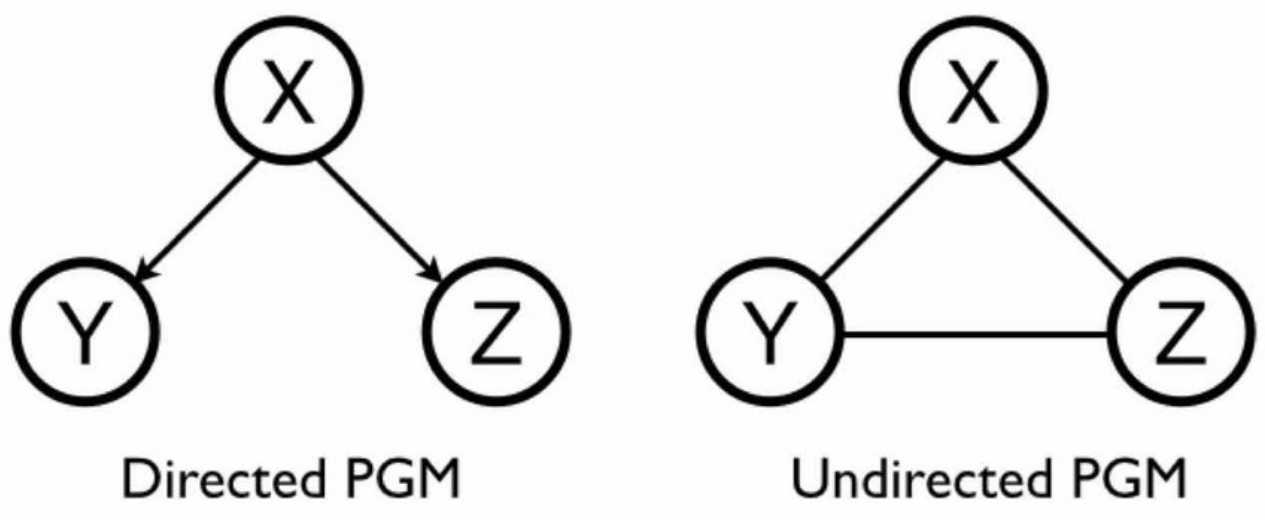
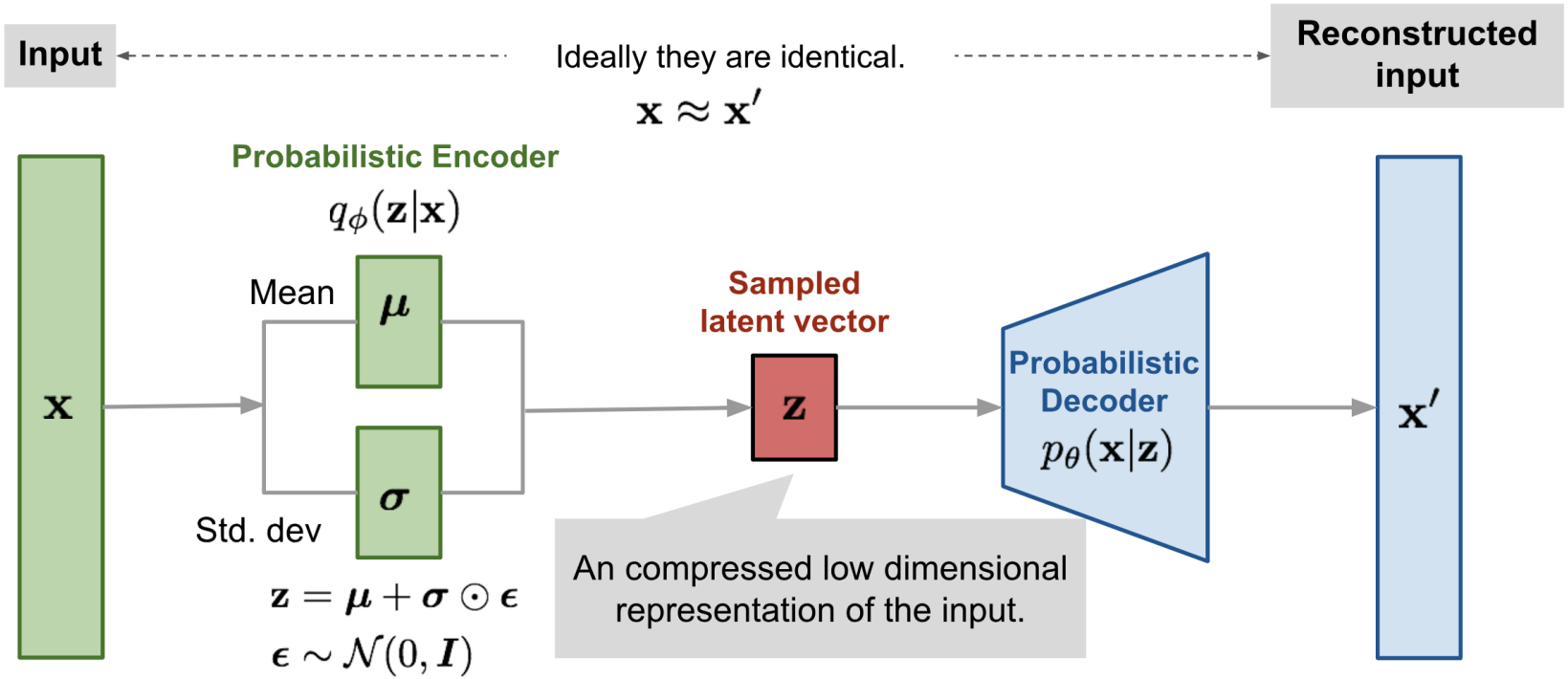


Figure 2: **PGMs** for the joint distribution $P(X, Y, Z)$

Two important classes are directed PGMs (Bayesian networks) and undirected PGMs (Markov random fields). Many applications: image processing, genomics, statistical mechanics

Generative learning: variational autoencoder (VAE)



Embedding the latent space of a VAE in 2D

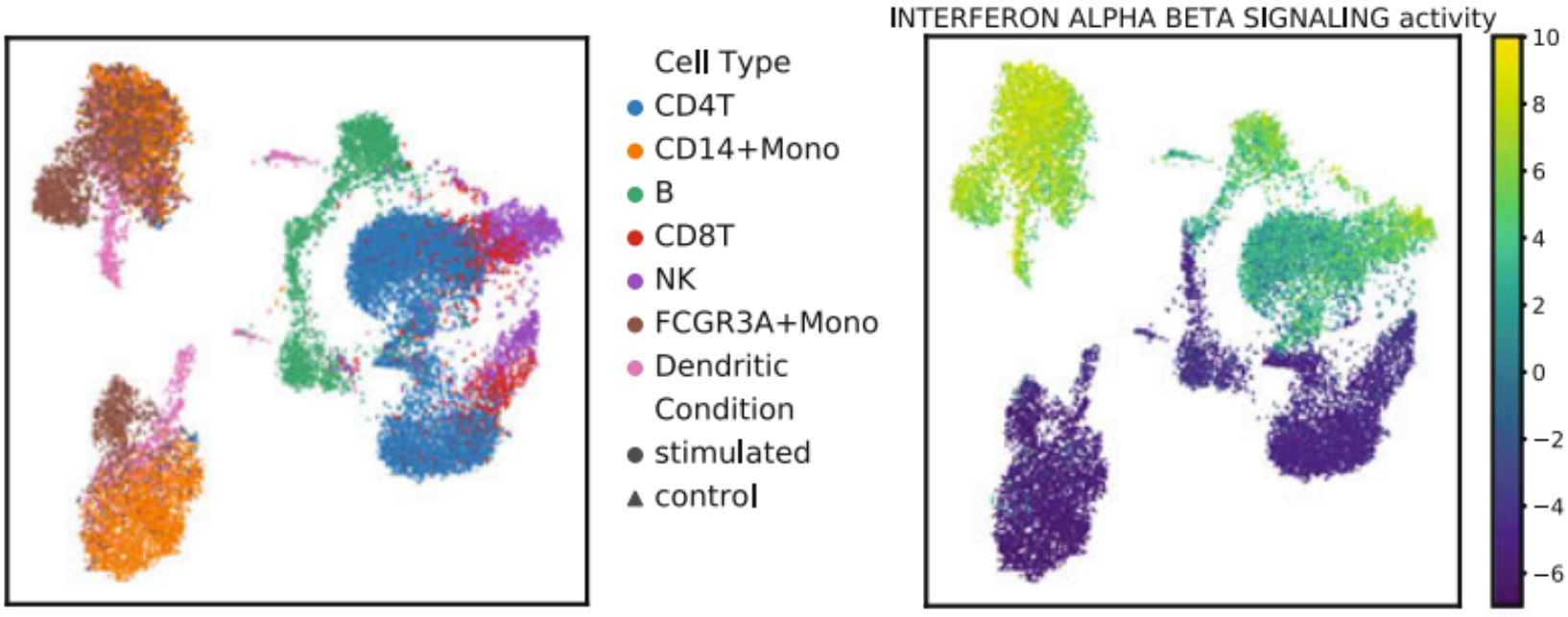


Figure 3: Phenotype segregation using a VAE on single-cell transcriptomics data. Taken from Seninge et al.

Probabilistic graphical models

+ structured representation
+ rigidity = efficiency
- rigid assumptions may not fit
- feature engineering

Deep learning

- Neural net "goo"
- Difficult parameterization
+ Flexible, high capacity
+ Feature learning

The sampling problem

We also may not know the proper normalization constant or partition function Z. Say we have

$$p(\mathbf{x}) = \frac{1}{Z}\tilde{p}(\mathbf{x})$$

where $p(\mathbf{x})$ is easy to compute but Z is (too) hard to compute.

This very important situation arises in several contexts:

1. In Bayesian models where $p(x_1, x_2) := p(x_1|x_2)p(x_2)$ is easy to compute but $Z = \int p(x_1|x_2)p(x_2)dx_2$ can be very difficult or impossible to compute.

2. In models from statistical physics, e.g. the Ising model, we only know $\tilde{p}(\mathbf{x}) = e^{-H(\mathbf{x})}$ where $H(\mathbf{x})$ is the Hamiltonian

Markov random fields

$$P(\mathbf{x}) = \frac{\exp\left(-H(\mathbf{x})\right)}{\sum_i \exp\left(-H(\mathbf{x}_i)\right)}$$

Suppose the energy function can be written as a sum over cliques:

$$H(\mathbf{x}) = \sum_n \tilde{\psi}_n(c_n)$$

Let $\psi_n = \log \tilde{\psi}_n$, which means $P(\mathbf{x})$ factors according to

$$P(\mathbf{x}) = \frac{\prod_n \psi_n(c_n)}{\sum_i \prod_n \psi_n(c_n)}$$

Rejection sampling with the uniform distribution

Let Ω be the state space or *support* of x. Let $U(\Omega)$ be the uniform distribution over Ω

Also notice that $p(x) \leq 1 \ \ \forall x \in \Omega$

The following procedure produces a sample $x \sim p(x)$.

1. Sample $u \sim U(\Omega)$
2. Sample $y \sim U([0, 1])$
3. If $y < p(u)$ return y as a sample of $p(x)$

This algorithm suffers from the curse of dimensionality. Generally, sampling becomes

References I