Neural Networks in Visualizing Structural Heterogeneity from Cryo-EM

Introduction

This paper is focused on single-particle cryo-electron microscopy (EM) and the visualization of the data that comes from it. Cryo-EM is a structural biology visualization method, which functions by taking a protein of interest and flash-freezing it so it can be visualized. By flash-freezing the proteins, the natural state of the sample would be preserved. When performing cryo-EM, a solution containing the specific protein is placed on a grid with many microscopic wells. After flash-freezing the grid of protein molecules, EM visualizes the entire grid and casts a two-dimensional shadow on every protein molecule. Since each instance of the protein is flash-frozen in a random orientation, the EM would produce many unique two-dimensional shadows of the same protein. By using computational methods to couple the 2D shadows, cryo-EM is meant to create a three-dimensional reconstruction of the protein.

The primary difficulty with cryo-EM is the computational conversion from the 2D shadows to the 3D protein structure. When working with rigid proteins, cryo-EM is effective at connecting the projection images to form a 3D reconstruction of the molecule. However, when modeling proteins with natural flexibility, each projection image generated would visualize a unique conformation, making the 3D reconstruction much more difficult. The current primary methods for analyzing structural heterogeneity in proteins utilize assumptions to classify the protein into multiple discrete states rather than a continuous spectrum of conformational changes. Using a discrete classification typically doesn't sufficiently describe proteins with continuous conformational changes.

To take advantage of single-particle cryo-EM's ability to visualize structural heterogeneity, Zhong et al. developed a neural network architecture called cryoDRGN to take the dataset of cryo-EM-generated unlabeled projection images and develop a 3D reconstruction of the protein that can represent continuous and discrete structural heterogeneity. CryoDRGN uses an unsupervised approach that takes inputs of images and encodes them into a variable in some continuous vector space, and then takes this variable and decodes them into a 3D volume. This neural network structure is similar to a variational autoencoder (VAE), so cryoDRGN was trained using a similar method as in standard VAEs.

To fully understand cryoDRGN, I believe it is first necessary to explore what VAEs are and how they function.

Overview of Autoencoders and VAEs

To begin exploring VAEs, we must begin with autoencoders. Autoencoders are an unsupervised learning model made from a pair of connected neural networks: the encoder and the decoder network. The encoder converts its inputs into a compressed (lower dimensional) representation, and then the decoder attempts to reconstruct the original input using this

compressed representation. Even with an unlabeled dataset, autoencoders can be viewed as a supervised learning algorithm by comparing the reconstruction of the original input $\hat{x}$ to the original input x. The reconstruction error, $\mathcal{L}(x, \hat{x})$, measures the difference between the original input and the reconstruction, so the model could be optimized by minimizing the reconstruction error. Since original input is compressed to a lower dimensional form, the compressed representation innately holds less information. As a result, the encoder must recognize important patterns in the inputs and preserve them while discarding irrelevant parts. In other words, the encoder must compress the data so that it contains the input's "latent" attributes. Hence, the compressed input data is often termed the latent state representation. However, the primary drawback with standard autoencoders is their latent space because it may not be continuous. For example, if one were to train an autoencoder and visualize the encodings onto a 2D latent space, they'd probably see distinct clusters. This is an issue because, if one were to choose a point in the latent space that was in the gaps between the clusters, the decoder would produce a very unrealistic reconstruction.

Variational autoencoders (VAEs) are able to resolve standard autoencoders' drawbacks because they uniquely possess continuous latent spaces. In standard autoencoder models, the encoder converts input values into an encoding vector with each entry containing a value describing its respective latent attribute. Contrastly, in VAEs, the encoder creates two encoding vectors, one vector containing means and the other vector containing standard deviations. By pairing up the ith mean with the ith standard deviation, the two encoding vectors construct a set of Gaussian distributions to describe each latent attribute. The decoder would then randomly sample from each normal distribution to get a stochastically generated encoding vector. By using Gaussian distributions instead of individual values to describe the latent space, the latent space becomes continuous and the decoder can be trained on various different latent state representations within the distribution.

One potential issue with VAEs is that, while we technically make the latent space continuous, it is possible for the encoder to learn to generate mean values for different classes to be very far apart and then minimize the standard deviation values. This would, in essence, make the latent space discrete-like. As previously encountered with vanilla autoencoders, this can lead to inaccuracy and lack of generalizability. To prevent this, one can introduce a regularization term: the Kullback-Leibler (KL) divergence. KL divergence is a function that measures how far two probability distributions are from each other. For VAEs, KL divergence is used to compare each latent distribution with the standard Gaussian distribution, $\mu = 0$ and $\sigma = 1$. By summing up each latent distribution's KL divergence, one can obtain the KL loss. KL loss rewards the encoder for distributing its encodings evenly around the origin of the latent space. Thus, by making a VAE's loss function a sum of the reconstruction loss and KL loss, the VAE is encouraged to form clusters of similar data points but put all encodings close to the origin so that each cluster is not too far from each other.

CryoDRGN's Neural Network Architecture

For cryoDRGN, Zhong et al. use a neural network architecture based on a VAE. They use an image-encoder-volume-decoder structure where they take cryo-EM images as input to the encoder and generate the latent variable $z \in \mathbb{R}^n$. Similar to a VAE, the encoder takes an image input and produces a mean vector μ and a variance vector Σ that define Gaussian distributions for z. The decoder is a bit more complex: a positionally encoded multilayer perceptron was used to approximate the function $V : \mathbb{R}^{3+n} \rightarrow \mathbb{R}$. V takes the n-dimensional latent variable z and a 3D Cartesian coordinate as input, which is why it takes a (n+3)-dimensional input. The 3D Cartesian coordinate they use is on the domain (-0.5, 0.5)^3, and represents a voxel of the 3D reconstruction. The output of V is the predicted electron density at the 3D Cartesian coordinate. Because the neural network uses the coordinate along with the latent variable, they call this architecture a "coordinate-based neural network."

To train the neural network, a loss function is calculated similarly to a standard VAE, except that cryoDRGN must regenerate the 2D images. After producing the function V and generating an electron density volume, an image can be reconstructed and compared to the input image in order to calculate a loss function. This results in a loss function that look similar to a standard VAE's loss function:

$$\mathcal{L}(X;\xi,\theta) = E_{q_\xi(z|X)}(\log p(X|z)) - \beta KL(q_\xi(z|X)||p(z))$$

The first term is the reconstruction error found from comparing the reconstructed image and the input image, and the second term is the KL divergence term that acts as a regularization term. Given an image X and the model's hyperparameters, one can calculate the loss. By minimizing this loss function through training the model on many 2D images, a 3D volume can be learned. When training the cryoDRGN model, Zhong et al. jointly trained the encoder and decoder networks with random initialization of weights and stochastic gradient descent. Some hyperparameters used in this model include β, ξ, θ, and the number of dimensions of z.

Applications to Structural Biology Datasets

To test cryoDRGN, Zhong et al. tested it on three datasets to empirically show its efficacy for modeling structural heterogeneity. They first tested the RAG complex and *Pf*80S ribosome, then they analyzed a data set of the *E. coli* large ribosomal subunit (LSU), and lastly they looked at the pre-catalytic spliceosome.

In testing the cryo-EM datasets of the RAG1-RAG2 signal-end complex (RAG complex) and the *Plasmodium falciparum* 80S ribosome (*Pf*80S), Zhong et al. found that cryoDRGN could resolve residual heterogeneity. A previously published paper reported two datasets of the RAG complex: the "signal-end complex" dataset, which fails to visualize certain distal elements, and

the “paired complex” dataset, which does resolve these distal elements. After testing the cryoDRGN model on only the “signal-end complex” with a 10D latent variable, Zhong et al. found that they could reveal heterogeneity in the distal elements and generate structures that match with canonical conformations found in the “paired complex” dataset. Furthermore, for the *Pf*80S ribosome, a previously published paper saw flexibility in the small subunit (SSU) and an unresolved density. Using a 10D latent variable model, Zhong et al. found that cryoDRGN generated structures that matched with homogeneous reconstruction values as well as additional heterogeneity in the SSU that could explain the unresolved density. In essence, cryoDRGN was able to explain heterogeneity that could not be directly observed.

The next dataset they analyzed was the cryo-EM dataset of *E. coli* LSU undergoing assembly, where they seeked to distinguish different assembly states. Previously, with user-guided 3D classification, four major assembly states were identified for the LSU. Zhong et al. sought to determine if cryoDRGN could determine these discrete structures without the need of user-guided 3D classification. Training with a 10D latent variable, cryoDRGN was able to construct electron density maps that matched the assembly states identified of the LSU, and even identified an unreported assembly intermediate. This empirical test demonstrates cryoDRGN’s ability to model highly heterogeneous datasets without the need for user intervention and also shows how it can identify novel structural classes.

The final dataset Zhong et al. tested was a cryo-EM dataset of the pre-catalytic spliceosome. They first trained a 10D variable model on downsampled images and found multiple clusters in the latent space encodings. Some of the clusters corresponded to undesired structures, so they filtered the latent space to only include desired clusters. With latent space filtered, they trained a 10D model on higher resolution images. Using principal component analysis to visualize the latent encodings in 2D space, they constructed structures along the first principal component, which created a trajectory of the spliceosome complex in motion. By traversing along the latent encodings using a nearest neighbor graph, cryoDRGN was able to construct a possible trajectory of the conformations. This application demonstrates cryoDRGN’s ability to reveal conformational dynamics found in cryo-EM.

Conclusion

This paper applies many of the machine learning concepts we’ve learned, such as its loss function and the addition of a regularization term, the optimization of a model through stochastic gradient descent, and the use of neural networks. That being said, this paper also covers a lot of information not covered in the course, such as the use of the variational autoencoder neural network architecture. By exploring all of these topics, this paper cohesively explains their methodology for visualizing structural heterogeneity from cryo-EM datasets.

Citation of Paper:

Zhong, E.D., Bepler, T., Berger, B. *et al.* CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks. *Nat Methods* **18**, 176–185 (2021). https://doi.org/10.1038/s41592-020-01049-4

Some other websites I used to learn about topics I didn't understand:

- https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf
- https://www.jeremyjordan.me/variational-autoencoders/
- https://arxiv.org/pdf/1909.05215
- https://www.countbayesie.com/blog/2017/5/9/kullback-leibler-divergence-explained
- https://arxiv.org/pdf/1601.00670
- https://proceedings.neurips.cc/paper_files/paper/2019/file/5a38a1eb24d99699159da10e71c45577-Paper.pdf
- https://www.technologynetworks.com/analysis/articles/cryo-electron-microscopy-principle-strengths-limitations-and-applications-377080