Exploration of Autoencoder Efficacy in Mechanism of Action (MoA) Data

Aman Sharma

Quantitative and Computational Biology Department

Dornsife College of Letters Arts and Sciences

University of Southern California

(aasharma@usc.edu)

Abstract:

This study investigates the application and potential contribution of autoencoders to the Mechanisms of Action (MoA) dataset, with a focus on different pathways for autoencoding this data. Namely, the study involved training the autoencoder on the whole dataset versus segregated cell viability feature sets versus segregated gene feature sets. Overall, the purpose was to evaluate how capable the autoencoders are in capturing and reconstructing biological data patterns, which could in turn contribute to algorithms that predict MoA classification of drugs. The findings of this analysis definitely show promise in the context of feature compression and reconstruction, but variability in performance across feature sets suggest a need for customized model architectures or training strategies. The main findings hold that autoencoders hold a lot of potential for feature extraction in MoA prediction, but with more optimization needed for handling biological data types. More specifically, the autoencoders seemed to work better with segregated features rather than the combined dataset as a whole.

Introduction:

In the content of drug discovery, understanding mechanisms of action is fundamental for the development of new therapeutics. The MoA essentially highlights how a molecule, usually a drug, affects a cell at a biochemical level. Since extensive, costly, and time-consuming biological experiments are typically needed to learn more about a molecule's MoA, computational approaches to predict MoA from high-dimensionality screening data could play an important role as an efficient and scalable alternative.

The MoA prediction challenge, hosted on Kaggle, provides an incredibly detailed dataset combining gene expression data and cell viability data from human cells exposed to various

compounds, each with a unique ‘sig_id.’ While the dataset poses an opportunity to apply machine learning techniques to predict MoA based on cellular responses, the primary objective of this study was to explore the potential of autoencoders - a type of unsupervised neural network - for feature reduction and reconstruction in the context of MoA data. Essentially, autoencoders compress or encode the data into a lower-dimensional space and subsequently reconstruct or decode the data back into its original dimensions. Overall, the hypothesis states that autoencoders could pave the way for a robust internal representation, uncovering inherent patterns in the data that may correlate with specific mechanisms of action.

The report will follow the development of three autoencoder architectures:

1. Whole dataset autoencoder: This will utilize the full set of available features (gene expression and cell viabilities) to learn comprehensive combined data representations.
2. Cell feature autoencoder: This will focus specifically on cell viability features to determine if isolated feature sets yield better or comparable performance to the whole dataset.
3. Gene feature autoencoder: This will concentrate specifically on gene expression data to determine if isolated feature sets yield better or comparable performance to the whole dataset.

Methods:

The dataset for this study was obtained from the Mechanisms of Action (MoA) prediction challenge on Kaggle, containing multiple features derived from human cells treated with various biological compounds. The primary features include gene expression profiles and cell viability

metrics, including 772 gene expression features and 100 cell viability features per sample, along with categorical features for type of treatment (compound or control perturbation), treatment duration (24, 48, or 72 hours), and dose level (high or low).

To pre-process the data, standard scaling normalization techniques were utilized to scale gene expression and cell viability features to ensure model input standardization. Effectively, this ensured that each feature was scaled to have zero mean and unit variance using the StandardScaler from the 'scikit-learn' library. This process is extremely important for any neural network, especially autoencoders which are sensitive to the scale of input data. Additionally, the categorical features in the dataset - 'cp_type', 'cp_time', and 'cp_dose' - were encoded to facilitate their use in the neural networks, which require numerical inputs. In more detail, one-hot encoding was used for 'cp_time' to remove any ordinal assumptions the model might infer. The cp_time was transformed into three binary columns, each representing one of the time points. On the other hand, binary label encoding was used for 'cp_type' and 'cp_dose', which correspond to treatment type and treatment dose respectively. For 'cp_type', the category corresponding to 'compound' was assigned a 1 while 'control' was assigned to 0. For 'cp_dose', 'high' and 'low' were assigned values 1 and 0 respectively.

After the conclusion of preprocessing, three separate autoencoder architectures were developed via TensorFlow and Keras to handle different forms of the dataset:

- For the whole dataset in particular, this received a total of 872 features (772 gene expressions and 100 cell viabilities). Two dense layers with 2100 and 1700 neurons were utilized in tandem with 'ReLU' activation functions and batch normalizations. The decoding layers mirrored the architecture of the encoding layers using the same activation

and normalization. The output layer consisted of 872 neurons, reconstructing the original input dimensions.

When creating autoencoder architectures for the segregated cell viability and gene expression data, a list of feature names was created with column names starting with “g-” corresponding to gene expression data and column names with “c-” corresponding to cell feature data. These features were then normalized through scaling and separated into two separate datasets.

- For the cell viability feature autoencoder, the input layer received a total of 100 cell viability features. Two dense layers with 90 and 75 neurons, respectively, using ReLU activation functions and batch normalizations were utilized. The decoding layer also mirrored the architecture of the encoding layers, using the same activation and normalization. The output layer consisted of 100 neurons, reconstructing the original input dimensions.
- For the gene expression feature autoencoder, the input layer received a total of 772 gene expression features. Two dense layers with 512 and 420 neurons, respectively, using ReLU activation functions and batch normalizations were utilized. The decoding layer also mirrored the architecture of the encoding layers, using the same activation and normalization. The output layer consisted of 772 neurons, reconstructing the original input dimensions.

Each model architecture was trained with an Adam optimizer with an exponential decay learning rate schedule. The learning rate was 0.001 with a decay rate of 0.96 every 1000 steps. Staircase decay was enabled to reduce the learning rate at discrete intervals to stabilize the training as it progresses. With this architecture, each model used a mean squared error as the loss function in order to quantify the difference between original and reconstructed outputs. After creation, each model was trained separately on their respective set of features for 200 epochs with a batch size of 512. Model performance was monitored using a 20% validation split and early stopping was used to halt training if the validation loss did not improve for 10 consecutive epochs, ensuring that overfitting on training data did not happen.

To assess and visualize the performance and characteristics t-SNE analysis and reconstruction comparison plots were used. t-Distributed Stochastic Neighbor Embedding reduced the dimensionality of the encoded data into two dimensions, allowing for the visualization of the data distribution and clustering tendencies. Reconstruction comparison plots allowed for qualitative assessment through comparing the original and reconstructed features of randomly selected samples through plots.

Results and Discussion:

Model	Loss	Validation Loss
Autoencoder-whole dataset	0.2250	0.1971
Autoencoder-cell viability	0.2839	0.2727
Autoencoder-gene expression	0.8517	0.9634

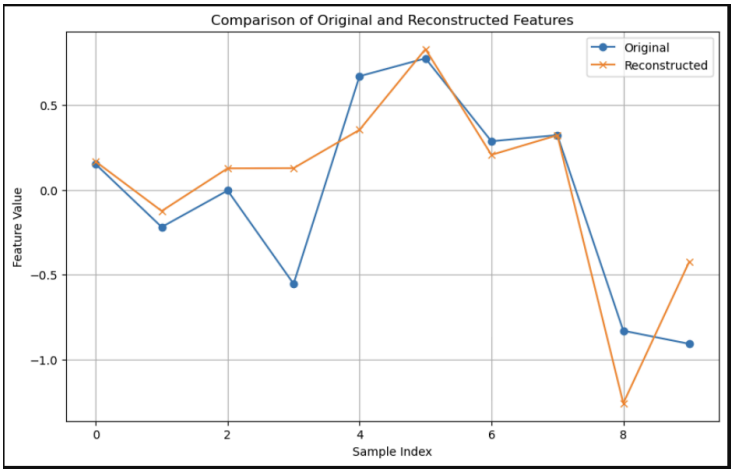
Table 1: Loss and validation loss of each autoencoder model

The results indicate varying levels of performance across the three models, similar to studies using deep learning for gene identification in cancer detection (Danaee et. al, 2017), indicating that the models exhibit potential to capture complex biological data patterns effectively. The autoencoder used on the whole dataset exhibited the best performance with the lowest validation loss, suggesting that it was the most effective in capturing and reconstructing the combined features of gene expression and cell viability. The lower validation loss compared to the training loss may indicate that the model has the potential to generalize well.

On the other hand, the cell viability autoencoder showed moderately good performance with both training and validation losses higher than those of the whole dataset, but still within generally good limits. Its performance, however, suggests that while the model was effective to an extent, cell viability features alone may not contain as rich information as the combined database, potentially limiting its learning capabilities on more complex patterns.

Lastly, the gene expression autoencoder demonstrated extremely high losses for both training and validation. This can be interpreted as a difficulty in modeling gene expressions ata accurate, potentially due to higher complexity and variability in gene expression features compared to cell viability features. Moreover, the higher validation loss suggests that there may

have been overfitting, indicating that the model may not have been tailored well to the data complexity and the potential to use more sophisticated regularization techniques in the future.



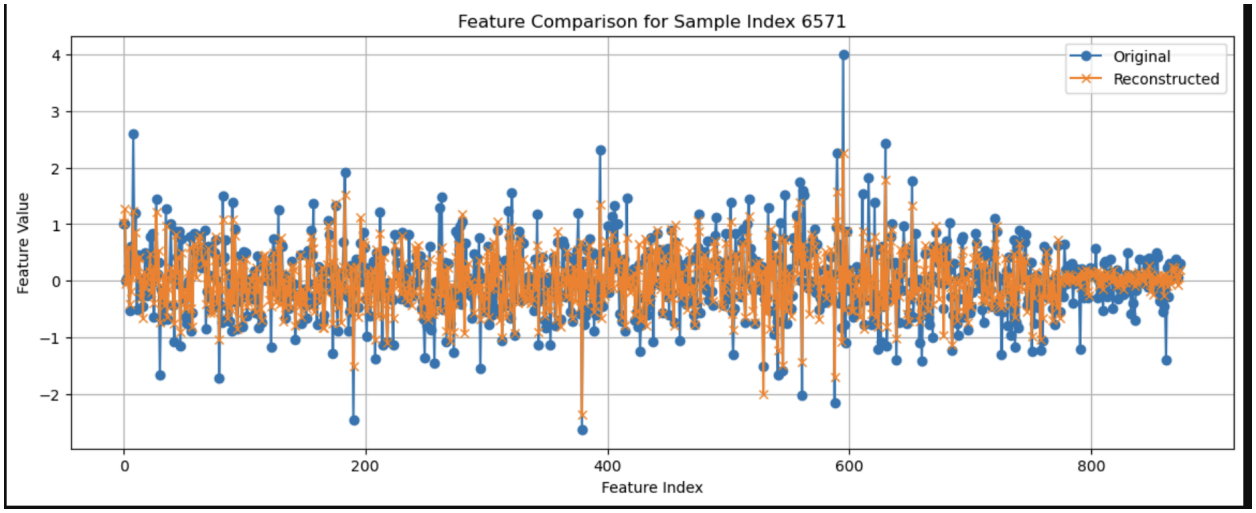


Figure 1: Feature comparison for random samples, comparing the original full dataset and the data reconstructed by the full-data autoencoder

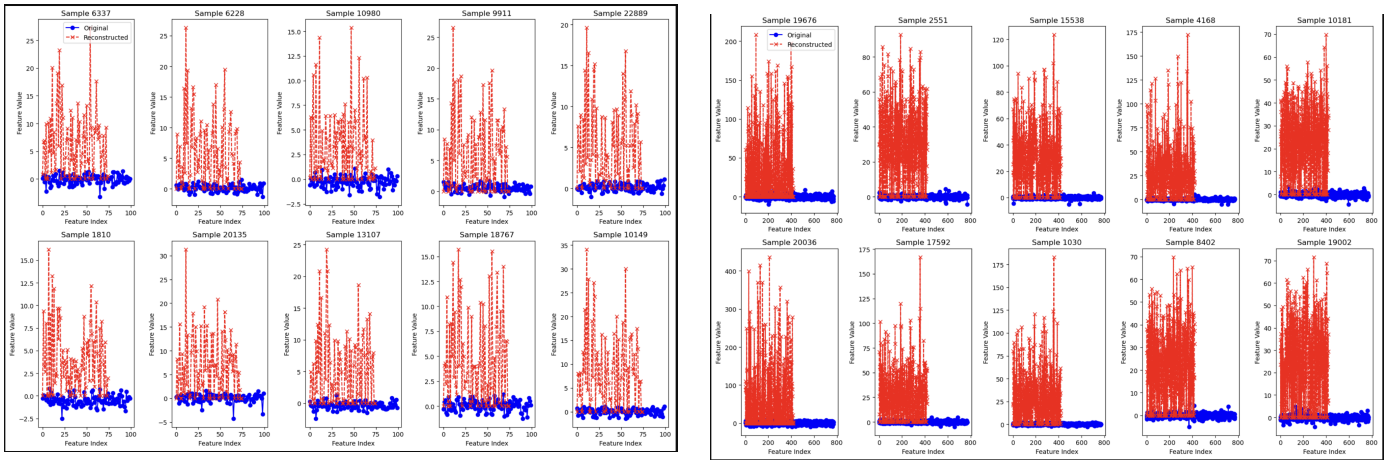


Figure 2+3: Feature comparison between original and reconstructed data for random samples of cell viability features (left) and gene expression features (right)

Based on the feature comparison plots, the full dataset autoencoder showed closer alignment between original and reconstructed data, suggesting that it learns and generalizes more effectively. On the other hand, cell viability autoencoder and gene expression autoencoders exhibited larger discrepancies between original and reconstructed data, indicating that the autoencoder experienced challenges with capturing the underlying patterns of these subsets. These results may tie into how the gene-expression data is high-dimensional with more complex biological interactions, making it challenging for an autoencoder to learn effective representations if not tailored correctly. Although the cell viability feature set is less complicated, the variable response to different treatments can cause noise and variability that an untailored autoencoder may struggle to model accurately. Differences in reconstruction accuracy may also indicate the model's tendency to overfit or underfit, raising the need for tailoring epochs, batch sizes, regularization techniques, and learning rates to improve model performance. At the same

time, the results also indicate that the full dataset autoencoder can benefit from a richer set of features that effectively provide more context for generalizable patterns.

Overall, the better performance of the full dataset autoencoder suggests that the combined features provide a more comprehensive view of the data, aiding in the model's ability to reconstruct the data. This suggests that gene expression and cell viability features are not necessarily independent, but rather work hand in hand in a beneficial manner to provide context for the model to generalize on. On the other hand, the distinct nature of gene expression features and cell viability features may require specialized neural network architectures or training regimes for their specific characteristics.

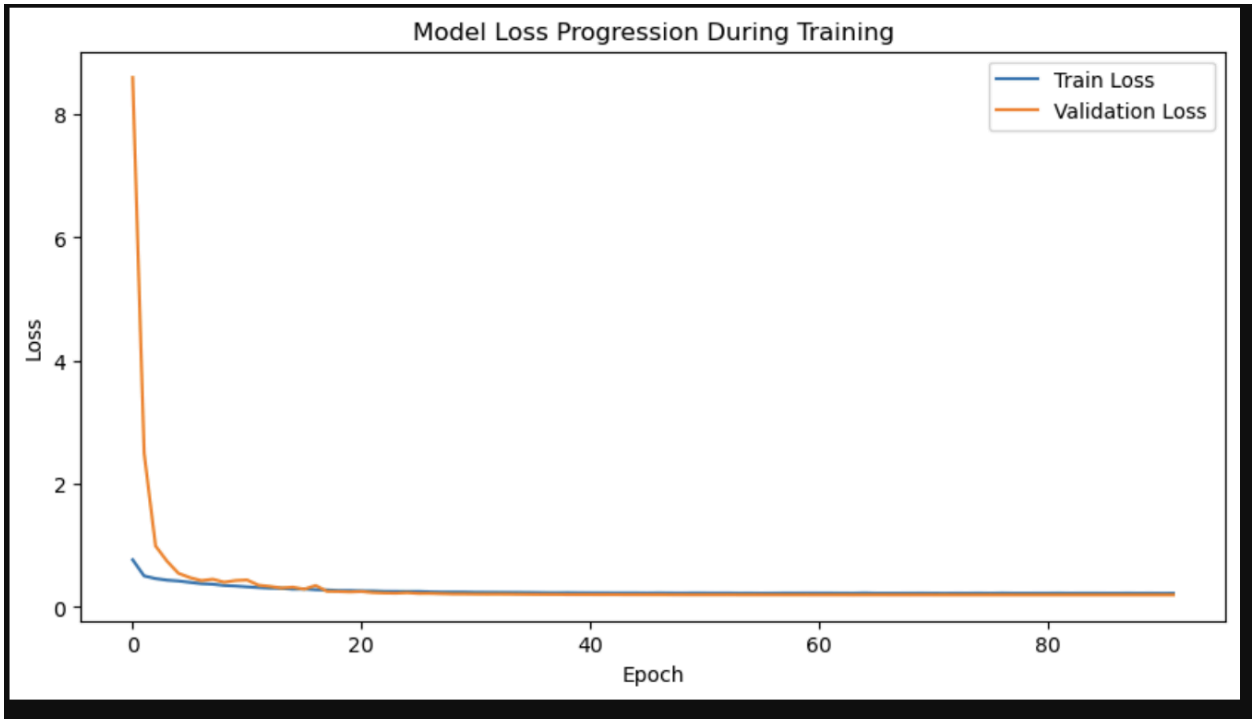


Figure 4: Graph of loss over epoch for autoencoders on the whole dataset

Specifically, the autoencoder on the full dataset was able to quickly capture a majority of the underlying patterns in the data, indicating that the architecture and learning rate was adequate and effective in minimizing errors between reconstruction outputs and original inputs. Moreover, the validation and training loss curves stayed close, indicating that the model is able to generalize well on the full data rather than stay specific to the training set. The loss curve stabilized at around epoch 20 with minor fluctuations after, indicating that there are diminishing improvements while also calling on early stopping mechanisms in the future to improve performance and avoid fitting noise.

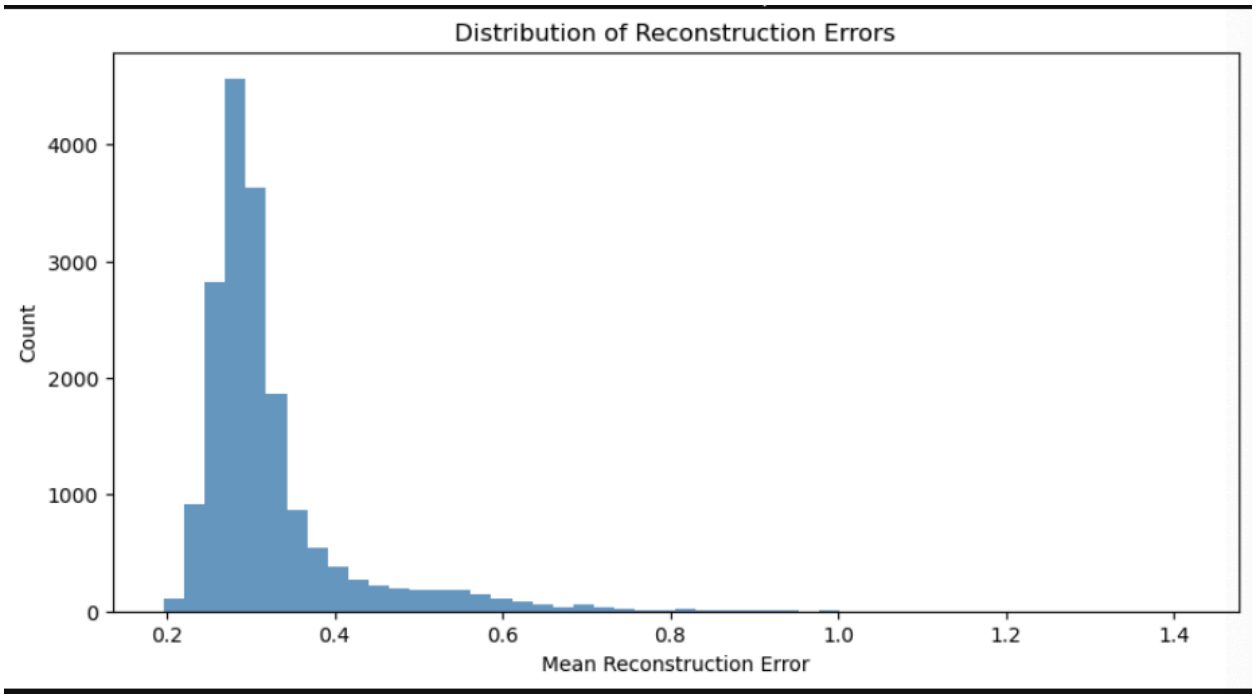


Figure 5: Histogram of reconstruction errors distributions for the full MoA data

In more detail, Figure 5 shows a left-skewed distribution, where the majority of the data points have a lower mean reconstruction error. This indicates that the autoencoder is effective in

capturing most characteristics of most of the data, with error rates clustering around 0.2 to 0.4. This indicates that reconstructed values are fairly close to original values, with room for improvement. However, this indicates that the autoencoder has a low error prevalence in its current state, showing that the model is capable of learning and representing features of the dataset. Higher error values could indicate the presence of outliers in the data or rather more complex patterns that the model struggled to learn, leaving room for model refinement and quality control.

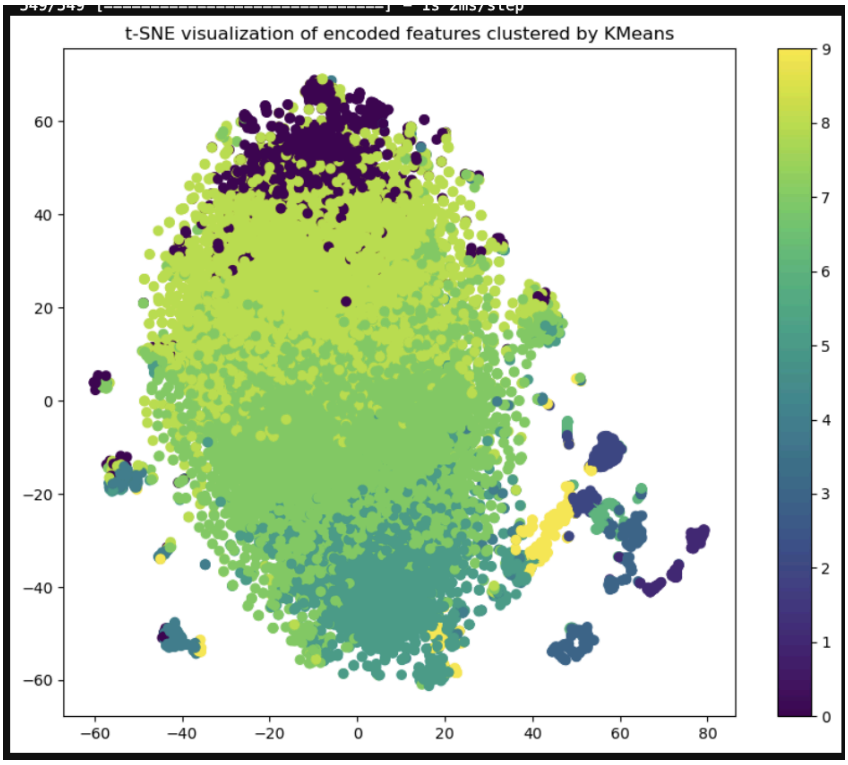


Figure 6: t-SNE visualization of encoded features of full MoA dataset clustered by KMeans

The t-SNE visualization provides an illustration of how the autoencoder along with KMeans clustering categorizes the encoded features into distinct groups, with each color indicating a different cluster identified by KMeans through variations that are captured by the

autoencoder. Overall, the data points are clustered distinctively in the t-SNE plot, indicating that the autoencoder is able to encode meaningful attributes of the data. This thereby allows the KMeans to segregate the data into clearly defined groups, indicative of underlying patterns and classifications as seen in Figure 6. Overall, the plot can offer insight into interactions and correlations between different features in the dataset, with tightly group clusters indicating that data points have similar properties and behaviors. On the other hand, the isolated clusters in the plot may suggest anomalies or unique characteristics.

Conclusions:

The study overall successfully utilized autoencoders to process and compress complex, high-dimensional MoA data with findings indicating that autoencoders truly are capable of dimensionality reduction and feature capturing of cell and gene data for classification, clustering, and visualization purposes. In this analysis, autoencoders showed high potential of extracting important patterns from both gene expression and cell viability data, but more so when combined into a single dataset. On the full MoA data in particular, the reconstructed features closely resembled the original data indicating successful learning and representation by the autoencoder models. Moreover, the progression of model loss during training showed a steady decline in both training and testing datasets, indicating that the model is able to generalize well on the data without overfitting. This is further supported by the distribution of reconstruction errors with most samples displaying low error rates. Moreover, the application of t-SNE visualization along with KMeans clustering provided insights into the actual MoA data structure, revealing unique subtypes or categories within the MoA data. Further studies on this could delve into these

clusters identified by the KMeans algorithm to get a better understanding of the biological significance of each cluster or to validate the clusters against known classifications of MoAs.

Performance-wise, there definitely is room for improvement through potentially integrating different forms of autoencoders. This could include variational autoencoders (VAEs) or generative adversarial networks (GANs), which could enhance the model's predictive ability and performance. Moreover, fine tuning the model's parameters and exploring different neural network architectures can result in improvements in accuracy and efficiency. If improved, models like these can be extended to more complex drug classification tasks through usage of cell viability and gene expression data. Studies have indicated that machine learning techniques in drug discovery align with autoencoder usage for exploration of MoA data (Rifaioglu et al. 2019).

In a more general sense, the techniques and methodologies used in this analysis can be adapted to other bioinformatics challenges that involve high-dimensional data, such as projects in personalized medicine and drug discovery where understanding action mechanisms is fundamental. Studies have shown that deep learning is increasingly being applied in the biomedical field, specifically in drug discovery, indicating that machine learning approaches are becoming more prevalent in the bioinformatics field (Mamoshina et al, 2016). Autoencoders, in particular, could be relevant for tertiary and quaternary protein structure prediction through learning efficient representations of amino acid sequences, thus allowing researchers to further understand protein function and interactions. They can additionally be applied to integrative genomics projects which involve genomic data derived from DNA methylation, gene expression, and histone modification. This can effectively uncover hidden factors that drive biological processes and disease progressions.

References:

Danaee, P., Ghaeini, R., & Hendrix, D. A. (2017). A deep learning approach for cancer detection and relevant gene identification. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 22, 219-229.

Mamoshina, P., Vieira, A., Putin, E., & Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Molecular Pharmaceutics*, 13(5), 1445-1454. https://doi.org/10.1021/acs.molpharmaceut.5b00982

Rifaioglu, A. S., Atas, H., Martin, M. J., Cetin-Atalay, R., Atalay, V., & Doğan, T. (2019). Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. *Briefings in Bioinformatics*, 20(5), 1878-1912. https://doi.org/10.1093/bib/bby061