Synthetic data generation for the optimization of strains in metabolic engineering using latent space representations derived from a Conditional Variational Autoencoder

Neil Alwani

Thesis committee
Prof. Dr. Thomas Abeel
Prof. Dr. Alan Hanjalic
MSc. Paul van Lent

Delft University of Technology
Faculty of electrical engineering, mathematics, and computer science

January 28, 2024

Abstract

This study investigates the application of generative models for synthetic data generation in pathway optimization experiments within the field of metabolic engineering. Conditional Variational Autoencoders (CVAEs) use neural networks and latent variable distributions to generate new, plausible data samples. We adapt this model by conditioning the training process on the target flux to acquire increased performance.

Additionally, a baseline model, namely Probabilistic Principal Component Analysis (PPCA), was selected for a comparative analysis to generate the underlying latent space to test the hypothesis that a type of Variational Autoencoder (VAE) can be used to learn a reduced-dimensional latent space for configurations of a kinetic pathway model. A dataset comprising 5000 hypothetical configurations of a kinetic pathway model was utilized to extract relationships between elements of a kinetic pathway.

The results indicate that PPCA can model the underlying distribution of the dataset when the latent space is large enough. However, the traditional CVAE might struggle to capture the underlying distribution, resulting in an entangled latent space. The study suggests that an implementation of β-CVAE could lead to a better balance between parts of the objective function during training, offering improved prospects for generating cost-efficient kinetic pathways for combinatorial pathway optimization experiments.

Introduction

Background

In metabolic engineering, the primary aim is to design aligned biological systems with predefined specifications with increasing precision using DNA technologies like CRISPR-CAS [1]. Combinatorial pathway optimization is employed to craft a proof-of-principle strain leading to the progress toward developing an industrially relevant variant of the strain. Within pathway engineering, this involves diversifying multiple pathway elements that directly or indirectly impact the pathway and its flux [2].

Several costs are associated with engineering a strain to ensure its production yield is economically feasible on an industrial scale. One crucial factor is the expense of generating data used to steer the engineering process. As proposed in [3], we can use kinetic models to predict flux using a system of ordinary differential equations (ODEs). While kinetic models have proven useful for modeling metabolic processes and solving the problem of acquiring genes from a physical host, they require deep expert insight into the modeled behavior of the reactions, which is often not available. Another reason for the increasing costs of combinatorial pathway optimization is the library sizes that are too large to process during the screening phase when conducting experiments [2].

We seek to employ strategies that allow us to reduce the dimensions of the data, by reducing the dependency between all the parameters of a kinetic model to a smaller subset of those parameters. Probabilistic Principle Component Analysis (PPCA) is one such strategy where data is reduced to the latent space, having fewer dimensions than the original data. The Variational Autoencoder (VAE) is a generative model, a type of latent space model, comprising an encoder for dimensional reduction and a decoder that generates synthetic data by sampling from the simple distribution, like a Gaussian.

Research Focus

We hypothesize that *a Conditional Variational Autoencoder (CVAE) can use a reduced-dimensional latent space model to learn characteristics of strains for combinatorial pathway optimization experiments.* To test this hypothesis, We explore the dimensional reduction aspect as well as the generative aspect of the PPCA and the CVAE.

To compare the two models, we quantitatively measure the underlying latent space representations and generate samples from this latent space. In this paper, we answer the question: *Can a Conditional Variational Autoencoder be used to learn the underlying distribution of the data such that a reduced-dimensional latent space representation of a strain can be used to generate a synthetic dataset for combinatorial pathway optimization experiments?*.

To do this, we give an overview of the data and dive into the experimental design in Chapter 2. In Chapter 3 we bring forward and analyze the results of the experiments. Finally, we conclude whether generative models can be used for combinatorial pathway optimization experiments. Additionally, we reflect on the reputability and ethical aspects of this study.

Experimental Design

In this study, we use a quantitative approach to evaluate how well generative models perform. We analyze and compare the models, focusing on their ability to create a latent space representation. This section gives an overview of the data used to train the generative models. We then discuss the implementation details of the Probabilistic Principal Component Analysis (PPCA), the Variational Autoencoder (VAE), and the Conditional Variational Autoencoder (CVAE). Subsequent sections cover the experimental variables, specifically the hyperparameters, and the research objective.

Our Dataset

Parameters of a kinetic model describe the quantities of metabolites and enzymes within a biological system or describe reactions within those systems. The level of quantities relates to a pathway and directly influences the flux[4]. The data used to train the models in this study includes parameters employed to create synthetic pathways and the predicted flux, which can be integrated into a kinetic model of strains resembling *E.coli (Escherichia coli)*.

This dataset comprises 5000 simulated pathways generated using 19 kinetic parameters with random settings within specified ranges, based on an initial parameter configuration. The columns of our dataset represent the kinetic parameters, referred to as the features, and the flux, considered as the target values. Initially, the dataset included all pathway configurations. However, retaining failed configurations would disrupt the model's ability to learn valid pathways. Consequently, unsuccessful configurations were filtered out from the data

Generative Models

For this study, we employ generative models that utilize the latent variable model to capture the structure and relations between data points [5]. Points from this latent variable model can generate new data points with similar characteristics to the original dataset [6]. In our case, the original dataset has 19 dimensions, corresponding to the number of kinetic parameters. We aimed to reduce the number of features required to describe the essential characteristics of our dataset, particularly focusing on its distribution.

A Pure Statistical Model

The first latent space model that was evaluated in this study was the Probabilistic Principal Component Analysis (PPCA), which was initially chosen as a baseline model for this study. The benefit of the PPCA model is that it enables us to make comparisons with other models more easily because it is a statistical model[7].

First, we had to derive the principal components of the dataset and project the dataset onto those principal components. This allowed us to produce the projection matrix using the predefined methods of the PCA[8] implementation in the Matrix decomposition module from the latest stable version(1.3.2) of the scikit-learn library.

Continuing, we calculated the mean of the data and the likelihood estimate for the variance and used the projection matrix to generate a synthetic dataset. The reference implementation from a medium article [1] was used to guide further the implementation and analysis of the model.

Our Machine Learning Model

The second model is the Variational Autoencoder (VAE). It is formally based on the autoencoder architecture and employs machine learning strategies to derive the latent space representation and subsequently transform samples from the latent distribution to generate a dataset with a distribution similar to the original dataset.

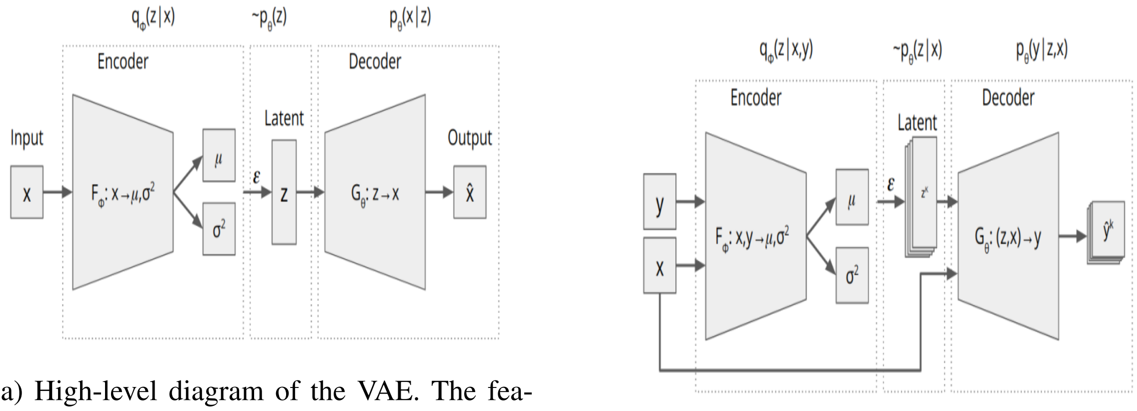
The implementation of the VAE in this study is created using modules from the latest stable version (2.1) of PyTorch. The model utilizes the unsupervised learning approach, eliminating the need for the target values column during training. Functions from torch.nn are used to build the neural networks for the encoder and decoder. To efficiently navigate the optimization landscape, we applied an Adam optimizer[2], which employs adaptive learning with an initial learning rate. The number of epochs was determined based on convergence towards better model performance. To address overfitting issues, the L2 regularization technique, namely weight decay, was applied. The VAE can also be modified to condition the generative model on additional information to solve more complex problems. This adapted model is called the Conditional Variational Autoencoder (CVAE) [5].

To implement the CVAE, an extra node in the input layer of the encoder and decoder was added, which we used to forward the target values through the neural network. A reference implementation[3] was used to guide the implementation step and subsequent analysis of the model.

[1] O. Ernst, ”The Simplest Generative Model You Probably Missed,” Accessed: November 20, 2023. Available: https://medium.com/practical-coding/the-simplest-generative-model-you-probably-missed-c840d68b704.

[2] ”PyTorch documentation for torch.optim.adam,” https://pytorch.org/docs/stable/generated/torch.optim.Adam.html, Accessed: November 26, 2023.

[3] A. van de Kleut, ”Variational Autoencoders,” https://avandekleut.github.io/vae/, Accessed: November 27, 2023.



(a) High-level diagram of the VAE. The features are passed forward to the encoder to learn the parameters of the underlying distributions. Using the parameters a latent variable is sampled and passed to the decoder. The decoder produces similar features as the input but with different values.

(b) High-level diagram of the CVAE. The difference between this model and the VAE is the same conditional information is passed to the encoder and decoder additionally to the features and latent variable.

Figure 1: Diagrams of the VAE and CVAE architecture. Adapted from [6].

Generative Model Parameters

To evaluate the PPCA model's performance, we need to determine the number of principal components that effectively describe the original data. This number should be lower than the number of features, as it governs the size of the latent space. However, opting for a lower number of components could result in a loss of variance[9] captured by the PPCA model from the original dataset. The results obtained from applying the general Principal Component Analysis (PCA) to our original dataset led to the choices for the number of principal components (n_components) (Table 1) used to test the model.

On the other hand, evaluating the VAE model's performance involves several selected hyperparameters, categorized into two groups. The first category includes architectural parameters, such as the number of hidden layers in the encoder and decoder, the nodes in these layers, and the latent layer. The second category encompasses learning parameters: learning rate (LR), regularization techniques, epochs, and mini-batch size. Decisions regarding architectural parameters were informed by the autoencoder architecture, utilizing the reparameterization trick to segregate the latent layer into distinct layers to learn the parameters of the posterior distribution.

Both the encoder and the decoder were designed with one input layer with as many nodes as features in the dataset (19) and a hidden layer consisting of 15 nodes. The learning parameters were set to values that yielded optimal performance within a computationally feasible number of epochs to make it possible to run on a personal computer. The findings from employing hyperparameter optimization using random search [10] also influenced the decision on which values (Table 1) to use in developing the models and conducting the experiments in this study.

Quantitative Metrics

"The objective functions used by the model to capture the distribution and characteristics of the dataset are different; however, we used the same quantitative metric, making comparisons

PPCA	n_components: 3, 5, 7, 9, **11**, 13, 15
VAE & CVAE	latent_dim: 3, 5, 7, 9, **11**, 13, 15 epoch: 100, 200, 300, **400**, mini-batch: 16, 32, **64**, 128 lr: 0.01, 0.001, **0.0001**, 0.00001 weight_decay: 0.1, 0.01, **0.001**, 0.0001

Table 1: In bold are the hyperparameters chosen when evaluating the PCA-reduced plots of the synthetic data for each model.

easier. The PPCA model relies on maximum likelihood estimation to derive the parameters of the original dataset[7]. In contrast, the VAE and the CVAE model the two neural networks constituting the encoder and decoder are simultaneously trained by optimizing the Evidence Lower Bound (ELBO) (3).

The Kullback-Leibler divergence (KL-div) is calculated (1) for each model in this study to quantitatively assess the difference between the distribution of the original dataset and that of the synthetic dataset. Additionally, we use Mean Squared Error (MSE) as the reconstruction loss (2) to measure how different the samples in the synthetic dataset are compared to that of the original dataset[6]. For the underlying structure, we aimed to make a visual comparison by plotting a PCA-reduction of the original dataset against that of the synthetic dataset.

$$\mathrm{KL}[q(z|X)\,||\,p(z)] = -\frac{1}{2}\sum_i \left(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\right) \tag{1}$$

$$\mathrm{MSE}(X, \hat{X}) = \frac{1}{N}\sum_{i=1}^{N}(X_i - \hat{X}_i)^2 \tag{2}$$

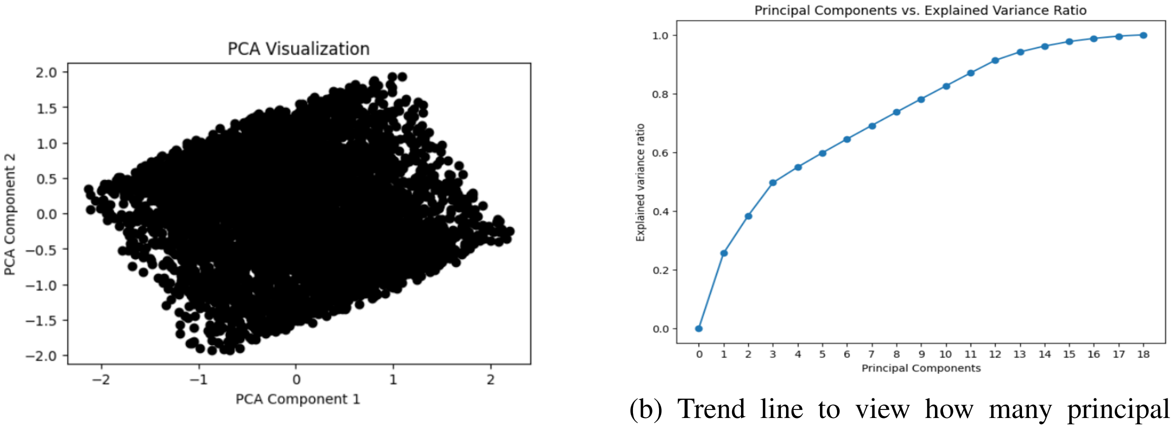
$$\text{ELBO} = \text{MSE}(X, \hat{X}) - \text{KL}\left[Q(z \mid X) \,\|\, P(z)\right] \tag{3}$$

Result

Insights from Principal Component Analysis Reduction

The PCA was useful for evaluating how many principal components are needed to capture the variance of the dataset. This is important to determine in order to create a latent variable space with fewer dimensions than that of the original dataset. To assess this, we plotted the total explained variance against the principal components of the dataset. Our plot (Figure 2, panel B) indicates that we could use more than 11 components to capture at least 90% of the variance in our data. This suggests that adding more dimensions would not lead to a significant increase in the accuracy of the models.

The principal components were also employed to create a PCA-reduced plot of the dataset, visualizing the structure that we aim for the latent space model to capture. The structure of the dataset resembles a slightly rotated square (Figure 2, panel A). For a comparative analysis, we generated a synthetic dataset of 5000 samples, similar to the original dataset.



(a) Scatter plot of the original dataset after applying PCA-reduction.

(b) Trend line to view how many principal components are essential to capture all the variance in the dataset.

Figure 2: Diagrams of the VAE and CVAE architecture. Adapted from [6]

Comparisons between captured distributions

For our models, we aimed to determine whether the latent space models could accurately capture the distribution of the dataset. To investigate this, we calculated the KL-divergence between the synthetic and the original dataset.

In the case of PPCA, we observed (Figure 3) a downtrend in KL-divergence for larger values of the latent space size. By adding more principal components, we were able to capture the distribution of the original dataset more accurately. However, as we increased the number of principal components, we approached the same number of features as in the original dataset. This indicates that the PPCA model requires as many features as our original dataset to accurately capture the distribution.

For the VAE, we observed (Figure 3) that the trend remains constant after a latent space size of 11, with values for PPCA also declining after using 11 principal components. This suggests that the VAE effectively captures the underlying distribution. In the case of the CVAE, we observed (Figure 3) noticeable differences in the values compared to the VAE values when measured with a precision of four decimal places.

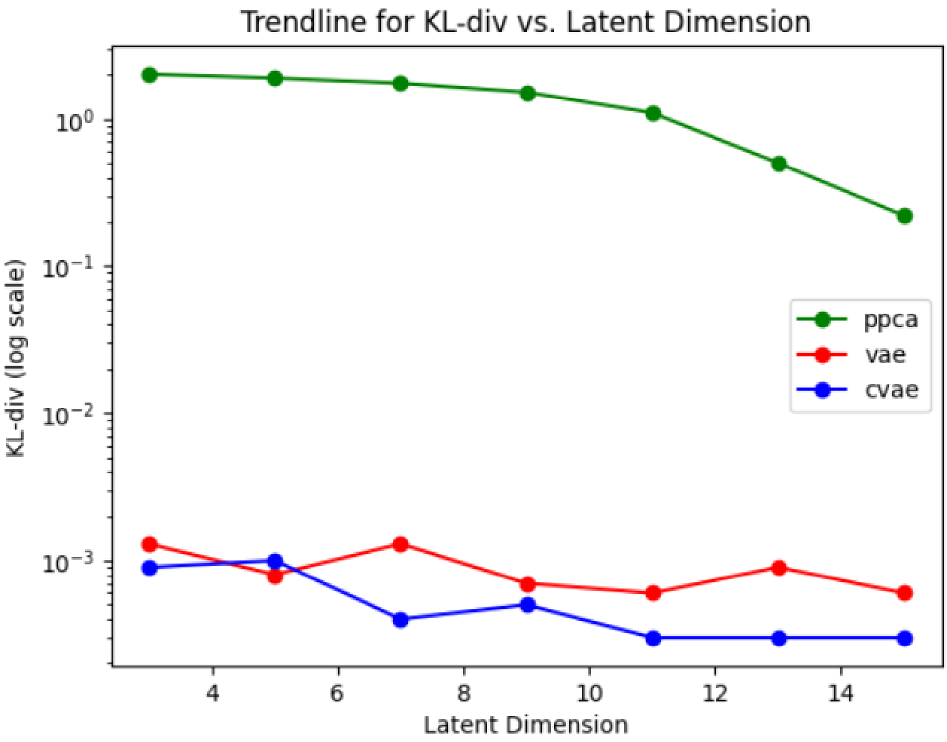


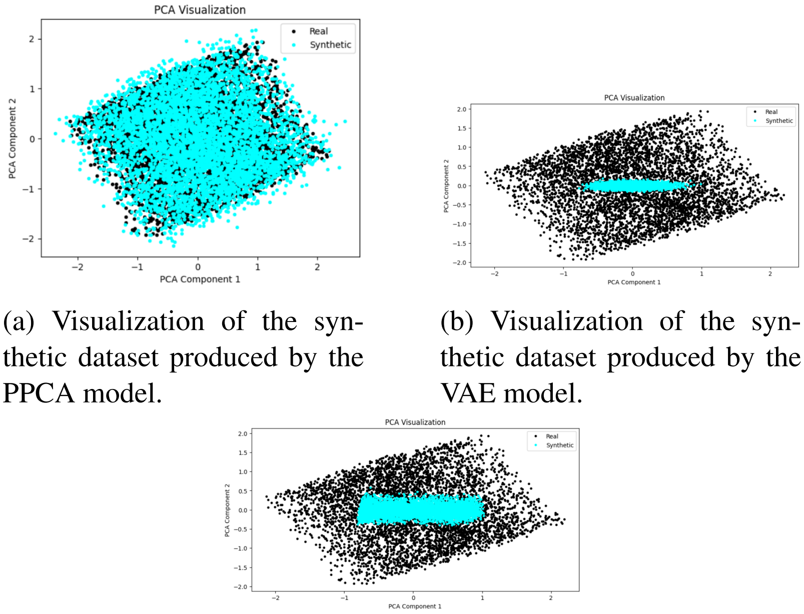
Figure 3: Trend line of computed KL-div on the logarithmic scale (y-axis) of the PPCA (green), VAE (red) and the CVAE (blue) model with latent space sizes (x-axis) with intervals of two steps on the.

Comparisons between learned structure

Aside from the underlying distribution, we were also interested in the underlying structure of the original dataset. We wanted to know if the synthetic dataset produced by our models retained this underlying structure. To investigate if the structure of the original dataset was learned, we used a PCA-reduced plot to compare the synthetic and the original dataset. The latent space size of 11 was chosen for this analysis based on the insights derived from the PCA reduction ().

For the PPCA model, we observed (Figure 4, panel A) that the synthetic dataset managed to model the underlying structure with softer boundaries, likely due to outliers. This means that the PPCA does well at generalizing the structure but cannot create precise boundaries present in the underlying structure of the dataset. For the VAE, we observed (Figure 4, panel B) that the dataset forms a single clustering along a certain line, indicating that the VAE was not able to capture the underlying structure.

In the case of the CVAE, we noticed (Figure 4, panel C) that the synthetic dataset has a rectangular structure along the same line. In both the VAE and the CVAE, we observed that the synthetic dataset is centered around 0.0 as the origin while applying some variance along the first and second principal components. This suggests that the models are not able to capture the underlying structure effectively. These results prompt a discussion about the performance of the CVAE and VAE models in comparison to that of the PPCA model.



(a) Visualization of the synthetic dataset produced by the PPCA model.

(b) Visualization of the synthetic dataset produced by the VAE model.

(c) Visualization of the synthetic dataset produced by the CVAE model.

Figure 4: PCA-reduction of the original dataset(black) against the synthetic dataset(cyan) generated by the PPCA, VAE and CVAE. The synthetic dataset has the same amount (50000) of data points as the original dataset.

Discussions

The measured KL-div for the VAE and the CVAE is low compared to the PPCA (Table 2). The VAE and CVAE models heavily minimize the KL-div part of the objective function. The CVAE and VAE don't require the latent space size to be large to learn the parameters needed to model the underlying distribution of the original dataset.

However, the structure of the original dataset isn't captured well by the VAE and CVAE models; the PPCA does this effectively. When comparing the MSE measurements between the models, we observe that the VAE and the CVAE might be prioritizing the modeling of the underlying distribution more than producing a dataset that is similar to the original dataset (Table 2). This is because the VAE and CVAE do not prioritize the MSE part of the objective function as much as the KL-div part.

Balancing the MSE and KL-div parts of the objective function can improve the implementation of the VAE and the CVAE models. This can be achieved by adding a β parameter to the objective function, which is adjusted based on the generated synthetic dataset to find the right value for the original dataset[6].

Latent	3	5	7	9	11	13	15
PPCA							
MSE	0.2177	0.1754	0.1366	0.0974	0.0598	0.0291	0.0131
KL-div	**2.0153**	**1.8956**	**1.7468**	**1.5181**	**1.0991**	**0.5015**	**0.2201**
VAE							
MSE	0.0586	0.0591	0.0586	0.0590	0.0594	0.0589	0.0593
KL-div	**0.0013**	**0.0008**	**0.0013**	**0.0007**	**0.0006**	**0.0009**	**0.0006**
CVAE							
MSE	0.0545	0.0541	0.0548	0.0548	0.0549	0.0550	0.0548
KL-div	**0.0009**	**0.0010**	**0.0004**	**0.0005**	**0.0003**	**0.0003**	**0.0003**

Table 2: MSE and KL-div computed between synthetic data generated by the PPCA, VAE, and the CVAE model with latent space sizes with intervals of two steps. These KL-div values were used to derive the trendline (Figure 3)

Responsible Research

In the field of computer science, massive innovations are taking place, and it might not be possible to keep up with them in real time. That is why research needs to be conducted in a way that allows anyone who decides to approach a particular study to enhance their understanding, and above all else, it can be replicated and used as a basis for any idea to further innovation in this field.

Data: Data was provided to us and generated using a kinetic model, with a workflow described in [4], and is openly available. The data can be challenging to understand and use; however, being open about this provides the possibility for further interdisciplinary study in the future. The original dataset, the models implemented to perform the experiments, and the results analysis of the study can be found by following the link here: https://github.com/NeilAlwani/RP_23-24_models.git

Reproducibility: The use of Jupyter notebooks makes it easy to reproduce this study. After importing the notebook, the data needs to be uploaded to the runtime of the notebook. Ensure that the path of the data file is given correctly in the notebook. After the preparation, all cells can be run sequentially to produce the results presented in this study.

Conclusion

TTo conclude, this study highlights the current problems that metabolic engineers face when conducting experiments for combinatorial pathway optimization. Namely, the costs related to processing synthetic pathways within biological systems and the domain knowledge required to develop novel pathways. We can approach this problem by deriving a generative model that uses a latent space model to learn the underlying distributions of a dataset and subsequently use this to generate a novel dataset with similar characteristics as the original dataset. In this study, we perform a comparative analysis of Probabilistic Principal Component Analysis (PPCA), Variational Autoencoder (VAE), and Conditional Variational Autoencoder (CVAE) to identify which of these models can be employed during experiments for combinatorial pathway optimization. A purely statistical model like the PPCA can generalize the underlying structure and model the distribution of the dataset; however, this requires the latent space size to remain as large as the original dataset. Models that employ neural networks like the VAE and the CVAE can model the underlying distribution with a latent space size using a fraction of the features of the original size. This is possible due to the model heavily minimizing regularization instead of the reconstruction. This leads to the CVAE and VAE not being able to capture the characteristics of the dataset. A suggestion to improve this is the use of an extra parameter to balance the regularization and reconstruction during the training process of the VAE and the CVAE [6]. To further investigate this, a colleague has conducted a separate study [11] for the CSE3000 Research Project. This way of working together on projects, like the CSE3000 Research Project, leads us to insights that can be useful to make an impact on experiments for combinatorial pathway optimization and the field of metabolic engineering as a whole.

References

[1] Q. Liu, T. Yu, X. Li, Y. Chen, K. Campbell, J. Nielsen, and Y. Chen, "Rewiring carbon metabolism in yeast for high level production of aromatic chemicals," *Nature Communications*, vol. 10, no. 1, 2019.

[2] M. Jeschek, D. Gerngross, and S. Panke, "Combinatorial pathway optimization for streamlined metabolic engineering," *Current Opinion in Biotechnology*, vol. 47, pp. 142–151, 2017.

[3] D. Weilandt *et al.*, "Symbolic kinetic models in python (skimpy): intuitive modeling of large-scale biological kinetic models," *Bioinformatics*, vol. 39, December 2022.

[4] P. V. Lent, J. P. J. Schmitz, and T. Abeel, "Simulated design–build–test–learn cycles for consistent comparison of machine learning methods in metabolic engineering," *ACS Synthetic Biology*, vol. 12, pp. 2588–2599, Aug 2023.

[5] C. Doersch, "Tutorial on variational autoencoders," *arXiv.org*, 2016.

[6] M. Debbagh, "Learning structured output representations from attributes using deep conditional generative models," *arXiv*, April 2023.

[7] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 61, pp. 611–622, 9 1999.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python." Accessed: November 25, 2023.

[9] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Factor analysis, probabilistic principal component analysis, variational inference, and variational autoencoder: Tutorial and survey," Jan. 2021. Online.

[10] S. Mishra, "Computational strategies for metabolic modeling of yeast." `https://www.ideals.illinois.edu/items/127308`, 2023. Accessed December 15, 2023.

[11] D. Kirbeyi, "Optimizingstrainsinmetabolicengineering:comparativeanalysisof - conditionalvariationalauto-encoderandprobabilisticpcaforsyntheticdata generation," February 2024.