Comprehensive Project Scope Document

Elysia Livia Gao s222445

Technical University of Denmark
July 31, 2023

1 Introduction

1.1 Background and Context

The project aims to bridge biological and physical models to study the heterogeneity behavior in bioreactors. It involves simulating cell performance inside a reactor with high substrate variability using biological/metabolic modeling. The project is based at the Novo Nordisk Center for Biosustainability at DTU, known for its expertise in industrial biotechnology and the design of cell factories. The Quantitative Modelling of Cell Metabolism group focuses on understanding cellular behavior in different environments through mathematical simulations applied to biology.

1.2 Overview of Objectives and Proposed Methods

The project objectives include analyzing big and complex datasets from advanced biological/CFD simulations, identifying reaction clusters in dynamic metabolic networks, utilizing dimensionality reduction techniques to plot and represent multi-dimensional metabolic networks, and learning to use an HPC cluster for computing complicated tasks.

2 Scope of the Project

2.1 Description of the Project's Focus

The project focuses on studying the behavior of cells inside a bioreactor with high substrate variability in a production industrial environment. It involves the analysis of internal cell data obtained from tracking cells moving inside the reactor for a certain time.

2.2 Objective 1: Problems to be Solved

The project aims to address the following problems:

- Identifying relevant information to extract from the internal cell data
- Exploring the possibility of dataset reduction
- Analyzing similarities and trends within a single cell lifeline (e.g., cycles inside the reactor)
- Identifying similarities and trends within lifeline groups (e.g., clusters of cells)
- Determining meaningful metabolic states of the cells (e.g., cycles) and the reactions that define metabolic clusters
- Exploring innovative ways to represent the data

2.3 Importance and Relevance

Understanding the behavior of cells in a bioreactor with substrate variability is crucial for optimizing bioprocesses in industrial settings. By studying the internal cell data and identifying meaningful metabolic states, the project aims to contribute to the development of strategies for improved bioprocess performance and efficiency.

3 Literature Review

3.1 Overview of Relevant Literature

3.1.1 Auto-encoders Based Dimensionality Reduction

The paper [**wang2016autoencoder**] explores the dimensionality reduction ability of auto-encoder, a three-layered neural network commonly used in deep learning. The authors aim to understand the unique properties of auto-encoder compared to state-of-the-art dimensionality reduction methods and its potential contribution to the success of deep learning.

The study involves experiments conducted on both synthesized and real datasets, including two and three-dimensional spaces for better visualization, as well as the MNIST and Olivetti face datasets. The results demonstrate that auto-encoder can indeed learn distinct representations compared to other methods. It exhibits the capability to capture complex structures, adaptively extract features, compress data, and provide accurate reconstructions. The experiments also investigate the influence of the number of hidden layer nodes on the performance of auto-encoder, revealing a possible relationship between the number of nodes and the intrinsic dimensionality of the input data.

Overall, the findings highlight the effectiveness of auto-encoder in dimensionality reduction and its potential as a building block in deep learning architectures. The unique properties of auto-encoder, such as its ability to detect repetitive structures, make it a valuable tool for various applications. Moreover, the study suggests that choosing the appropriate number of hidden layer nodes can enhance the performance of auto-encoder in reducing dimensionality.

3.1.2 A Folded Neural Network Autoencoder for Dimensionality Reduction

The paper [**wang2012folded**] introduces a new structure called the folded autoencoder for dimensionality reduction. It addresses the challenge of handling high-dimensional data by reducing computational costs while improving performance. The folded autoencoder is designed based on the symmetric property of the conventional autoencoder, reducing the number of weights that need to be tuned. The authors demonstrate the effectiveness of the folded autoencoder by comparing it with the conventional autoencoder and principal component analysis (PCA) on the MNIST dataset. The results show that the folded autoencoder outperforms PCA and achieves similar performance to the conventional autoencoder in reconstructing images. Furthermore, the training time of the folded autoencoder is reduced by 17.2% compared to the unfolded autoencoder. The study opens up possibilities for real-time applications of autoencoder-based dimensionality reduction.

In conclusion, the proposed folded autoencoder structure offers a promising approach to dimensionality reduction. It not only reduces computational costs but also improves the generalization performance compared to conventional approaches such as PCA. The results obtained on the MNIST dataset demonstrate the effectiveness of the folded autoencoder in reconstructing images. Further research directions include theoretical analysis of the framework, evaluation on larger and more complex datasets, and integration with other advanced techniques such as convolutional networks and incremental learning. The folded autoencoder presents a significant contribution to the field of dimensionality reduction and holds potential for various applications in handling high-dimensional data.

3.1.3 An Autoencoder-based Deep Learning Approach for Clustering Time Series Data

The paper [**tavakoli2020autoencoder**] introduces a two-stage approach for time series clustering to address the challenges posed by high dimensionality, hidden features, and unlabeled data. It emphasizes the importance of understanding dataset characteristics before conducting data analysis. The proposed methodology transforms the unsupervised clustering problem into a supervised learning one by creating cluster labels based on time series data characteristics. In the first stage, a conventional K-means

algorithm is applied to the feature vector data. The second stage utilizes an autoencoder-based deep learning algorithm to capture hidden features and reduce dimensionality. A case study on clustering time series data of over 70 stock indices demonstrates the methodology's effectiveness, achieving an accuracy of 87.5% in predicting cluster labels. The paper also compares the results obtained by the proposed approach and conventional K-means, highlighting that the deep learning-based model considers additional hidden features. The paper is structured with sections discussing key characteristics of time series data, an overview of artificial neural networks, the two-stage model, the encoder-decoder deep learning model, algorithms, evaluation, and future research directions.

In conclusion, this paper presents a novel approach to time series clustering that combines conventional clustering with deep learning techniques. By leveraging the power of deep neural networks, the methodology offers better performance and the ability to identify hidden patterns in the data. The case study on financial time series demonstrates the potential of the proposed approach in improving clustering accuracy and discovering latent features that may influence the clustering results. Future research can further explore the application of other deep learning techniques, such as LSTM and GANs, for various time series analysis tasks, including anomaly detection, seasonal effects, and prediction, to gain a deeper understanding of the advantages and limitations of deep learning in time series data analysis. Additionally, comparative studies between conventional methods and deep learning-based models in different application domains will provide valuable insights into the effectiveness of these approaches in specific scenarios.

3.2 Key Studies and Findings

1. **Auto-encoders Based Dimensionality Reduction** [**wang2016autoencoder**]:
    - Autoencoders learn distinct representations, capturing complex structures and compressing data effectively.
    - The number of hidden layer nodes affects performance, indicating a relationship with intrinsic data dimensionality.
    - Autoencoders show promise as building blocks in deep learning architectures.
2. **A Folded Neural Network Autoencoder** [**wang2012folded**]:
    - The folded autoencoder reduces computational costs while outperforming PCA and matching conventional autoencoders.
    - It holds potential for high-dimensional data handling and improves dimensionality reduction.
3. **Autoencoder-based Deep Learning for Time Series Clustering** [**tavakoli2020autoencoder**]:
    - A two-stage approach combining K-means and autoencoders achieves high clustering accuracy for time series data.
    - Deep learning models capture hidden features, enhancing clustering performance.

In conclusion, autoencoders are valuable for dimensionality reduction in large datasets. The folded autoencoder offers an efficient alternative, while deep learning approaches improve clustering accuracy in time series data analysis. These innovative strategies hold promise for various applications and further research.

3.3 Previous Work on Metabolic Analysis

4 Methodology

4.1 Starting Point: From Lifeline to Internal Cell Data

In this section, we describe the starting point of our analysis, where we transform the approximately 150000 lifeline of cells inside the reactor into internal cell data suitable for further analysis. The process involves several key steps and data transformations to obtain valuable time series data representing the behavior of cells.

The data includes time series readings of substrate concentration seen by the cell at each time point, computation of substrate uptake rate (qS) at each time point (350 s with steps of 0.1 s), metabolic analysis using dynamic Flux Balance Analysis (dFBA), and the generation of a pd.df with the internal fluxes of the cell at each time point. Fluxes are for basic E.Coli and approximately 150 reactions.

4.2 Objective 1: Data Analysis of the Internal Cell Data

Background: Autoencoders

Autoencoders play a crucial role in this project, as they are neural network models designed for unsupervised learning and dimensionality reduction. They consist of two main components: an encoder and a decoder. Working together, the encoder compresses the input data into a latent space, while the decoder reconstructs it. By minimizing the reconstruction error between the original and reconstructed data, autoencoders capture salient features and extract meaningful representations. The compressed latent space obtained from the autoencoder can be used for subsequent analysis and interpretation. By leveraging the reduced-dimensional representation, it becomes easier to explore patterns, extract meaningful features, and perform various analysis techniques. The dimensionality reduction aspect of autoencoders helps to focus on the essential characteristics of the data, making it more amenable for further analysis tasks such as clustering, visualization, or supervised learning. Objectives 4.2.1 and 4.2.2 focus on building an autoencoder model where as subsequent objecives focus on analysis and use for latent space from the autoencoder models for further processing.

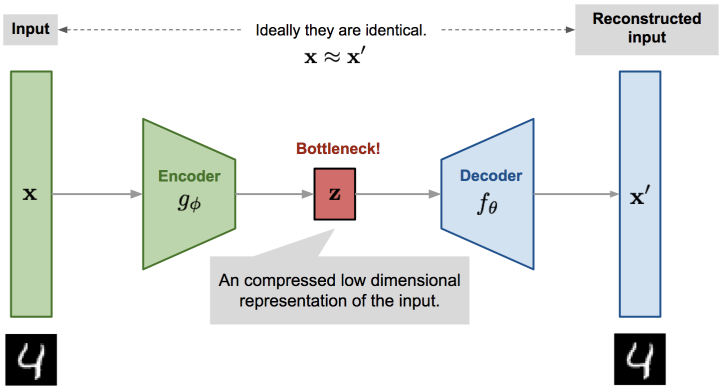

Figure 1: Basic Autoencoder Architecture

Autoencoder for 3-Dimensional Input: Cell Lifeline, Timepoints, and Reaction Fluxes

In this section, we describe the specifics of utilizing an autoencoder for processing 3-dimensional input data, where each dimension represents critical information about the cell behavior within the bioreactor.

The input consists of three dimensions:

1. **Cell Lifeline:** This dimension represents the lifeline of a single cell within the bioreactor, capturing the sequential movement and behavior of the cell over time.
2. **Timepoints:** The timepoints dimension corresponds to the time series data, providing a series of observations recorded at regular intervals (e.g., 0.1 seconds) over a specific duration (e.g., 100 seconds).
3. **Reaction Fluxes:** This dimension contains the internal fluxes of the cell at each timepoint, representing the metabolic activity and reactions occurring inside the cell as it interacts with the environment.

The goal of the autoencoder is to reduce the dimensionality of this 3D input data while preserving meaningful patterns and characteristics. By doing so, the autoencoder captures the salient features and essential temporal dynamics of the cell behavior, facilitating subsequent analysis and interpretation.

The provided code showcases an example of constructing and training an autoencoder using Keras. The model architecture consists of an encoder and a decoder, connected through a latent space. The encoder processes the 3D input data, compressing it into a lower-dimensional latent space. The decoder then reconstructs the original data from this latent representation.

The architecture includes the following layers:

1. **Input Layer:** The input layer is defined to accept the 3D input data.
2. **Reshape Layer:** Since the autoencoder operates on flattened data, the reshape layer converts the 3D input into a 2D format.
3. **Encoder Layers:** The encoder is composed of dense layers with a variable number of hidden units and activation functions. The number of hidden units and the activation function are hyperparameters that can be optimized using techniques like Optuna.
4. **Latent Space:** The output of the encoder is the latent space, which is a reduced-dimensional representation of the original data.
5. **Decoder Layers:** The decoder part of the autoencoder mirrors the encoder's architecture. It reconstructs the original data from the latent space.
6. **Output Layer:** The output layer of the autoencoder restores the 3D format of the data through reshaping.

The model is then trained on the input data using the mean squared error (MSE) loss function and the Adam optimizer. The latent space representation obtained from the trained autoencoder can be used for various purposes, such as clustering similar lifelines or visualizing the cell behavior in a lower-dimensional space.

4.2.1 Identifying relevant information to extract from the internal cell data

Input data from the domain can then be provided to the autoencoder model, and the output of the model at the bottleneck, which represents the compressed latent space, can be used as a feature vector. This feature vector can be utilized in various ways, such as feeding it into a supervised learning model for classification or regression tasks, enabling visualization of the data in a lower-dimensional space, or more generally, facilitating dimensionality reduction. By leveraging the capabilities of autoencoders, particularly LSTM autoencoders in this project, the goal is to extract the most informative and relevant features that capture the essential characteristics of the internal cell data. This process effectively reduces the complexity of the data while preserving its temporal nature, enabling further analysis and interpretation of the cell behavior within the bioreactor.

4.2.2 Exploring the possibility of dataset reduction

This objective aims to reduce the dimensionality of the internal cell data using autoencoders, specifically leveraging the latent space representation. Autoencoders are neural network models for unsupervised learning and dimensionality reduction. They compress the input into a latent space through an encoder and reconstruct it using a decoder. By minimizing the reconstruction error between the original and reconstructed data, autoencoders capture salient features and extract meaningful representations.

Traditional methods like Principal Component Analysis (PCA) are not well-suited for temporal data due to its temporal dependencies and sequential nature. PCA fails to capture relevant temporal patterns, interactions between reactions, and important information related to temporal dynamics. In contrast, autoencoders are better equipped to preserve temporal dynamics and capture important features. They are specifically designed for time series analysis and feature selection in order to address these limitations.

LSTM (Long Short-Term Memory) autoencoders, a variant of autoencoders utilizing LSTM layers, are particularly appropriate for handling temporal data. LSTM networks are capable of capturing long-term dependencies and temporal relationships in sequential data. In the context of this project, LSTM autoencoders can effectively reduce the number of reaction fluxes while preserving the key temporal dynamics. By leveraging the advantages of LSTM autoencoders, the dataset can be reduced in dimensionality while retaining important information regarding the behavior of cells inside the bioreactor.

The goal of this objective is to employ LSTM autoencoders to reduce the number of reaction fluxes in the internal cell data. By doing so, the project aims to extract the most informative and relevant features, effectively reducing the complexity of the data while preserving its temporal nature. This reduction in dimensionality facilitates subsequent analysis and interpretation of the cell behavior within the bioreactor.

4.2.3 Analyzing similarities and trends within a single cell lifeline (e.g., cycles inside the reactor)

One of the objectives of this project is to analyze the similarities and trends within a single cell lifeline, such as the cycles inside the reactor. To achieve this, the latent space or low-dimensional representation obtained from the autoencoder can be leveraged. The reduced-dimensional representation captures the essential characteristics of the internal cell data while preserving the temporal dynamics.

Time Series Analysis:

1. Seasonal Decomposition:
    - Seasonal decomposition of time series (STL): Separate the time series into its seasonal, trend, and residual components, aiding in identifying cyclic patterns and variations.
2. Autocorrelation Analysis:
    - Measure autocorrelation between a time series and its lagged versions to reveal repeating patterns and cycles.
3. Fourier Transform:
    - Decompose the time series into its constituent frequencies, helpful in identifying periodic trends.
4. Cluster Analysis:
    - Group similar segments of the time series together based on patterns and trends, facilitating the identification of distinct stages or cycles within the cell lifeline.
5. Dynamic Time Warping (DTW):

- Measure similarity between two time series with different lengths or temporal distortions, useful for identifying similar cycles within the reactor's lifeline.

To complement traditional methods, machine learning techniques can be applied to the time series analysis. Some relevant approaches include:

1. Time Series Clustering with k-means:
   - Apply k-means clustering to group similar segments of the time series and identify distinct stages or cycles within the cell lifeline without predefined labels.
2. Time Series Classification with Machine Learning Algorithms:
   - Utilize various classification algorithms (e.g., Random Forest, Support Vector Machines) if labeled data is available to identify and classify different stages or cycles in the time series.

4.2.4 Identifying similarities and trends within lifeline groups (e.g., clusters of cells)

Another objective of this project is to identify similarities and trends within lifeline groups, which refer to clusters of cells that exhibit similar behavior. By leveraging the latent space or low-dimensional representation obtained from the autoencoder trained on all lifelines, it becomes possible to analyze and characterize these groups.

Time Series Analysis for Lifeline Groups: To achieve this, various time series analysis methods can be applied to compare and identify similarities and trends within lifeline groups:

1. Dynamic Time Warping (DTW) for Multiple Time Series:
   - DTW can be extended to compare multiple time series simultaneously, measuring the similarity between each time series and a reference series. This approach can reveal common patterns and cycles shared among cells in a lifeline group.
2. Cross-Correlation Analysis:
   - Cross-correlation measures the similarity between two time series as a function of the time lag. By applying cross-correlation to multiple time series within a lifeline group, shared patterns and temporal relationships can be identified.
3. Cluster Analysis of Lifeline Groups:
   - Cluster analysis can be applied to group cells with similar time series patterns and behaviors into lifeline groups. This unsupervised approach helps identify distinct clusters of cells sharing common characteristics.
4. Time Series Alignment and Averaging:
   - Align multiple time series within a lifeline group to a common time frame and compute the average time series. This process can reveal the collective behavior and trends shared by the cells in the group.

Machine Learning Techniques for Lifeline Groups: In addition to the above methods, machine learning techniques can be utilized to identify and explore similarities and trends within lifeline groups:

1. Time Series Clustering with Hierarchical Clustering:
   - Apply hierarchical clustering to group cells based on the similarity of their time series data. This approach can reveal hierarchies of similarities within the lifeline groups.
2. Time Series Classification with LSTM for Lifeline Grouping:

- Use LSTM networks for time series classification to group cells with similar patterns into lifeline clusters. LSTM can capture long-term dependencies and complex relationships among time series data.

4.2.5 Determining meaningful metabolic states of the cells (e.g., cycles) and the reactions that define metabolic clusters

To identify meaningful metabolic states based on the reaction fluxes within the cell lifelines, we can employ clustering techniques tailored for time series data. These methods will group reactions with similar temporal behavior together, revealing distinct metabolic states and characteristic patterns exhibited during different stages of the lifeline.

1. **Density-Based Clustering**: Density-based clustering methods, like DBSCAN, can identify regions of high density in the reaction flux data, representing sets of reactions with similar behaviors that define specific metabolic states.
2. **Agglomerative Hierarchical Clustering**: Hierarchical clustering can be applied to group reactions based on similarity in their temporal behavior. This approach constructs a tree-like hierarchy of clusters, which can be cut at different levels to identify different metabolic states.
3. **Partitioning Around Medoids (PAM)**: PAM is a variation of K-Medoids clustering that can handle time series data by using medoids (representative time series) to cluster reactions based on similarity in temporal profiles.
4. **Time Series Shapelets**: Time series shapelets are small, informative subsequences that can be used for clustering time series data. By identifying shapelets in the reaction flux time series, we can cluster reactions based on their shape similarities.

By applying these time series clustering techniques to the reaction flux data, you can identify meaningful metabolic states by grouping together reactions with similar temporal patterns. The identified clusters of reactions will provide insights into the dynamic behavior of the cellular metabolic network, aiding in the optimization and control of bioprocesses in industrial settings. It is recommended to start with simple methods like K-Means and then explore more complex techniques to find the most suitable approach for your specific dataset.

4.2.6 Exploring Innovative Ways to Represent the Data

t-SNE (t-Distributed Stochastic Neighbor Embedding) is an algorithm widely used for visualizing high-dimensional data in a lower-dimensional space, typically 2D or 3D. In the context of this project, t-SNE can be applied to visualize the latent space obtained from the autoencoder, providing valuable insights into the underlying structure and relationships of the internal cell data.

The visualization process involves the following steps:

1. **Projection to Latent Space:** The encoder layers of the autoencoder are used to project a set of validation images into the latent space. This step yields a set of latent vectors representing the essential characteristics of the internal cell data.
2. **t-SNE Embedding:** The obtained latent vectors are then embedded into a two-dimensional space using t-SNE. The algorithm aims to preserve the pairwise distances between the data points, allowing the data to be visualized in a lower-dimensional space while retaining meaningful relationships and patterns.
3. **Visualization of Latent Space for Sample Images:**

   The t-SNE visualization of the latent space reveals how similar and dissimilar objects are distributed in this reduced space. In this context, objects correspond to different cellular states or metabolic

patterns within the bioreactor. Similar metabolic states, such as cycles or distinct cellular behaviors, form clusters, while dissimilar states are located farther apart in the latent space.

4.3 Objective 2: Try More Complex Metabolic Models (Second Phase)

4.4 Innovative Ideas Strategies

To address the issue of memory constraints and efficiently input a very large dataset of 150k lifelines into the autoencoder, several strategies and ideas can be considered. These strategies aim to optimize the training process, reduce memory consumption, and enable the autoencoder to effectively handle the large-scale dataset. Some potential approaches include:

1. **Folded Autoencoder Structure:** Inspired by the folded autoencoder structure proposed in [**wang2012folded**], explore the potential of using a folded autoencoder for dimensionality reduction. The folded autoencoder leverages the symmetric property of the conventional autoencoder to reduce the number of weights, improving performance while reducing computational costs
2. **Mini-Batch Training:** Implement mini-batch training, where the large dataset is divided into smaller batches. This approach allows the autoencoder to be trained iteratively on these batches, reducing memory consumption and enabling efficient updates to the model parameters.
3. **Data Sampling:** Consider using data sampling techniques to create a representative subset of the lifelines. Random or stratified sampling can significantly reduce the data size while preserving meaningful representations.
4. **Incremental Learning:** Explore incremental learning techniques where the autoencoder is updated with batches of new data as it becomes available. This approach enables the model to handle large datasets over time without retraining on the entire dataset.
5. **Rolling Fixed Window Approach:** Explore the use of a rolling fixed window approach to generate training samples from unbounded time series data. This method can summarize the change of time series features over time as a smooth trajectory path, providing insights into the dynamics of the data.

5 References

5.1 Comprehensive List of Cited Literature and References

Sairam Sundaresan, Ayush Thakur. “An Introduction to Adversarial Latent Autoencoders.” An Introduction to Adversarial Latent Autoencoders, 22 Nov. 2022, wandb.ai/authors/alae/reports/An-Introduction-to-Adversarial-Latent-Autoencoders–VmlldzoxNDA2MDY.

Tavakoli, Neda, et al. “An Autoencoder-Based Deep Learning Approach for Clustering Time Series Data.” SN Applied Sciences, vol. 2, no. 5, 2020, https://doi.org/10.1007/s42452-020-2584-8.

Wang, Jing, et al. “A Folded Neural Network Autoencoder for Dimensionality Reduction.” Procedia Computer Science, vol. 13, 2012, pp. 120–127, https://doi.org/10.1016/j.procs.2012.09.120.

Wang, Yasi, et al. “Auto-Encoder Based Dimensionality Reduction.” Neurocomputing, vol. 184, 2016, pp. 232–242, https://doi.org/10.1016/j.neucom.2015.08.104.

https://hackernoon.com/latent-space-visualization-deep-learning-bits-2-bd09a46920df

https://arxiv.org/abs/1809.00999 sources