Analysis of Omics Data using OmiEmbed

Urvish Pujara - 2020101032
Syed Imami - 2020113012

Introduction

Multi-omics analysis is an emerging field that aims to integrate multiple types of biological data, such as genomics, transcriptomics, proteomics, and metabolomics, to gain a more comprehensive understanding of biological systems.

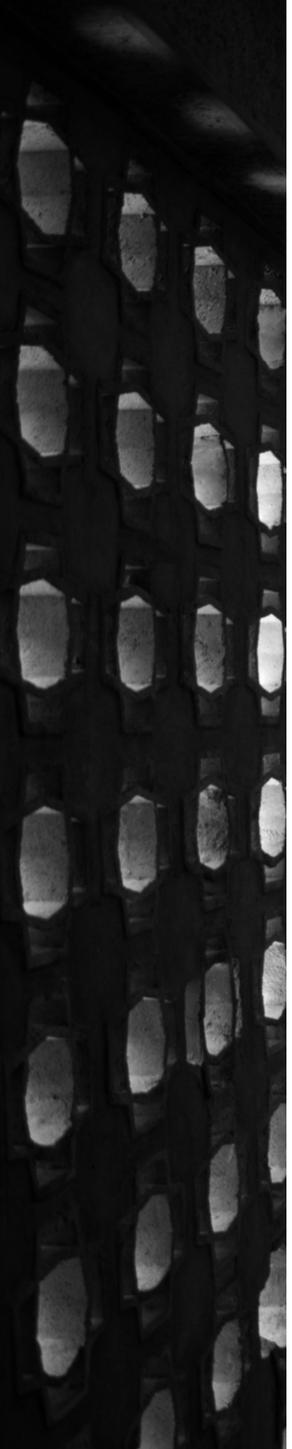
VAE learns to encode omics data into a lower-dimensional representation and then decodes this representation back into the original input data.VAE can be trained to generate new samples of gene expression data that are consistent with other types of biological data, such as protein expression data.

By learning joint representations of multiple types of biological data, VAEs can help to identify new biological insights and biomarkers that would be difficult to detect using single-omics analysis like cancer classification.

These techniques will help us understanding the complex interactions between different types of biological data.

Problem Statement

OmiEmbed supports multiple tasks for omics data including dimensionality reduction, tumour type classification, multi-omics integration, demographic and clinical feature reconstruction, and survival prediction. This can further be extended to providing feature importance for each type of cancer. Overall, the new model aims to reduce the number of features and thereby focusing only on the important features reducing the training time of the model. By incorporating clinical features, the model aims to provide a more comprehensive analysis that takes into account factors that can affect patient outcomes.



Goals

PART 1

Relating and Analyzing properties for multi-omics

PART 2

Generate a lower dimensional Representation of omics data

PART 3

Providing feature importance, and training a model with reduced number of features

Dataset

GSE109381 BTM DATASET

for DNA methylation
86+6 normal tumour types
features : 485577
samples : 3905

TGCA PANCANCER DATASET

for Gene expression , DNA methylation , miRNA expression
33+ 3 tumour samples

Gene expression : 60483 features , 11538 samples
DNA methylation : 485577 features, 9736 samples
miRNA expression : 1881 features, 3905 samples

Link to Datasets

- BTM datasets - link
- GDC pancancer datasets - link

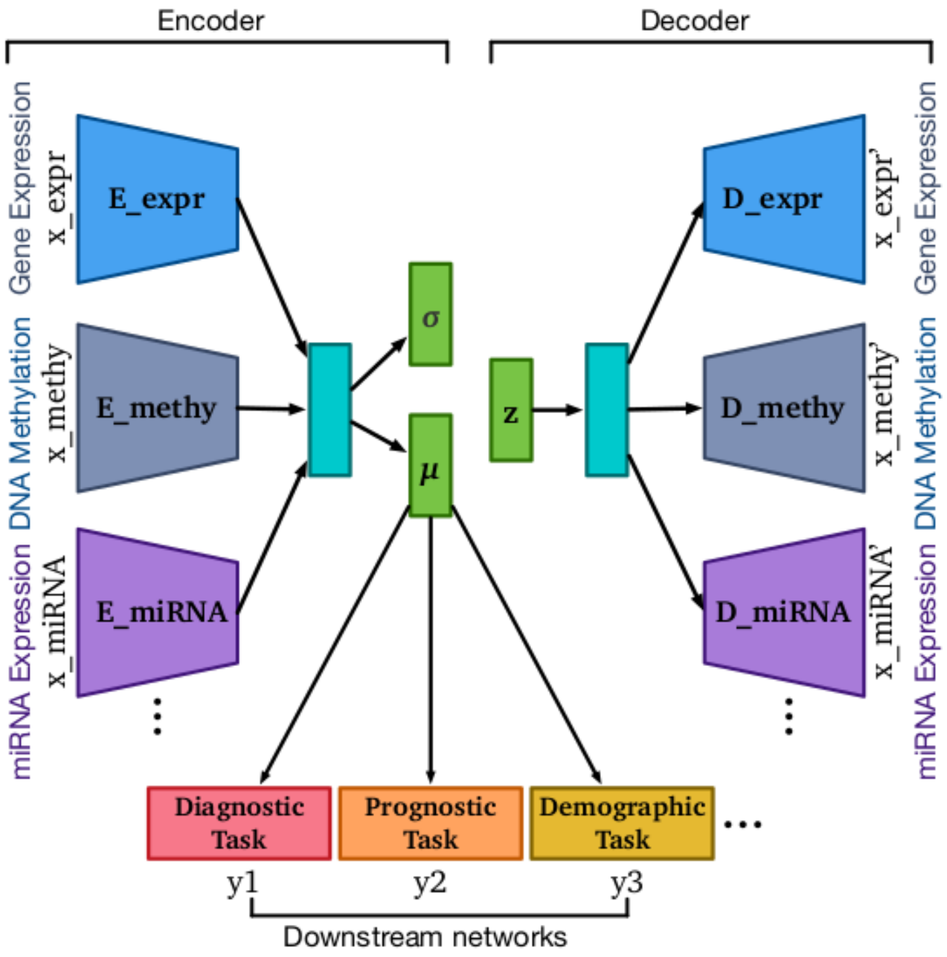
Exploratory data analysis

The baseline model was trained and tested on gene expression and DNA methylation data from the GDC dataset, using a dataset size of 1000 and a batch size of 32.

The training was done over three phases, with 50 epochs each for the first two phases and 100 epochs for the third phase.

For the project the training and testing for 5 specific tumour types and 1000 samples of data because of computational limit.

Baseline Model



Implementation of Baseline Model

- Data preprocessing
  - Data is divided into 80:20 train:test ratio.
  - Processed by the Bioconductor R package minfi to obtain the beta value of each CpG probe.
- Feature selection.
  - probes were filtered out in DNA methylation and gene expression data. (ex. the ones targeting y chromosome 46k features filtered out)
- Dimensionality reduction using auto-encoders.
  - one-dimensional convolutional neural network (CNN) and the fully connected neural network (FC) for encoder and decoder in deep embedding module
- Downstream networks
  - diagnostic task
  - prognostic task
  - demographic task

Training Strategy & Loss functions

$$\mathcal{L}_{embed} = \frac{1}{M}\sum_{j=1}^{M} BCE\left(\mathbf{x}_j, \mathbf{x}'_j\right) + D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \| \mathcal{N}(\mathbf{0}, \mathbf{I}))$$

$$\mathcal{L}_{classification} = CE(y, y')$$

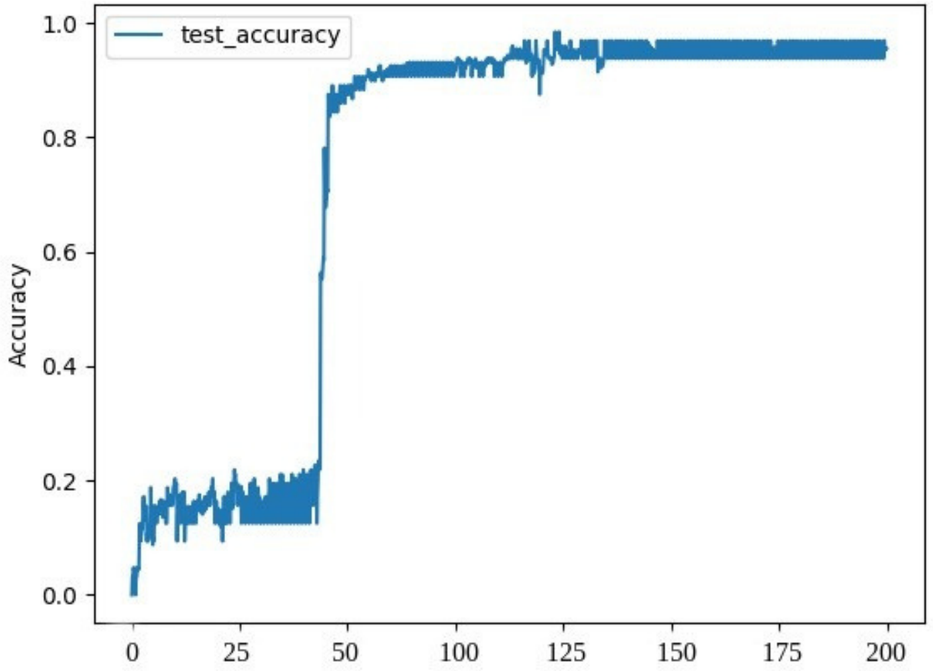
$$\mathcal{L}_{regression} = MSE(y, y')$$

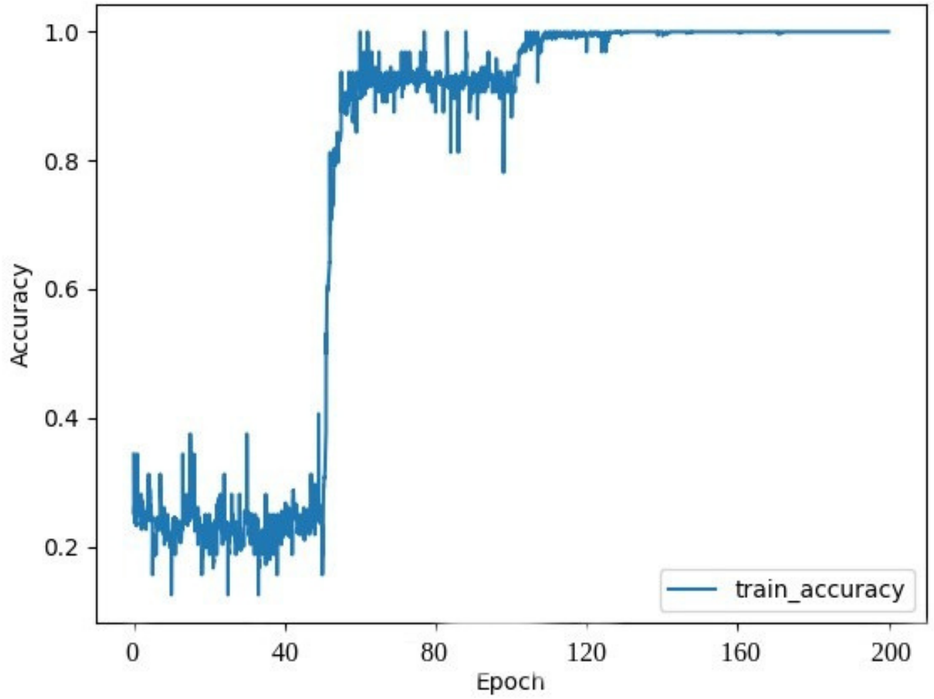
$$\mathcal{L}_{total} = \lambda \mathcal{L}_{embed} + \mathcal{L}_{down}$$

Baseline Results

- Testing accuracy obtained from the baseline for the classification of these 5 tumour types and the normal control using gene expression and DNA methylation data from the GDC dataset is 97.5%
- Testing framework for 5 classes randomly selected in 33 types and 1000 as the dataset size initially
- Presented accuracy for the multi-omics classification is 0.9771 ± 0.0027
- Obtained as baseline result is 97.5 for the train_test.py in testing

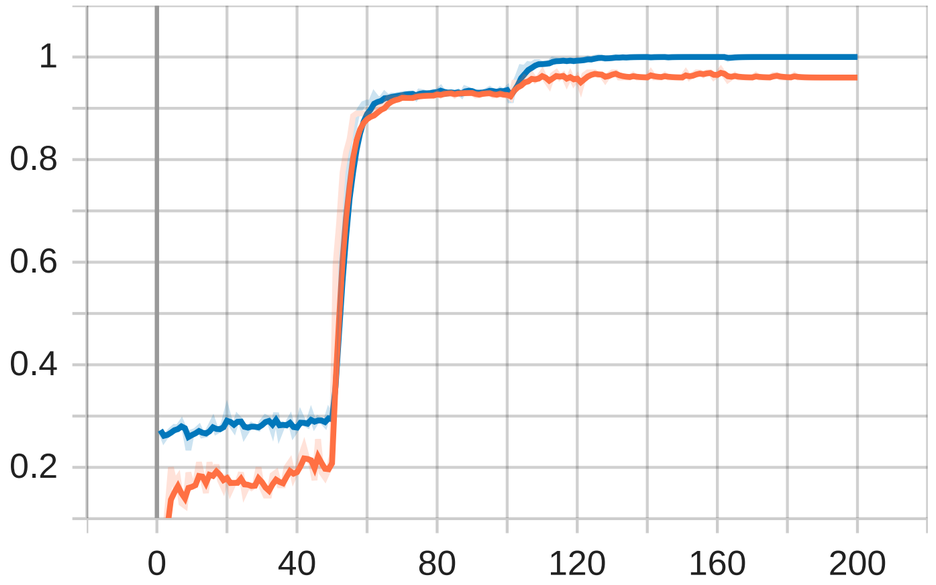
Baseline Results - Accuracy





Output

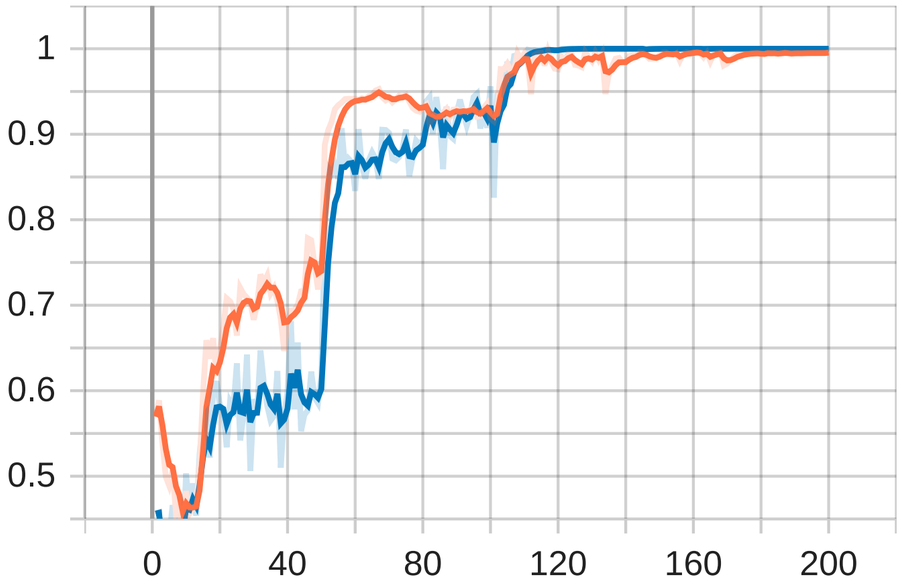
TSV files are obtained and 5 metrics are measured to ensure the classification is done properly



Metric : Accuracy
Testing : 0.96
Training : 1

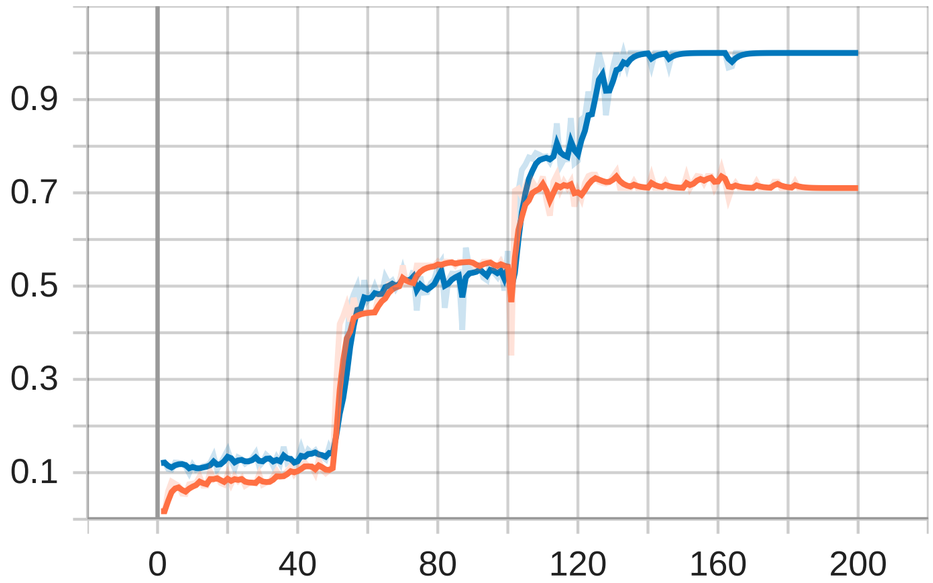
test
train

13



Metric : Auc
Testing : 0.9954
Training : 1

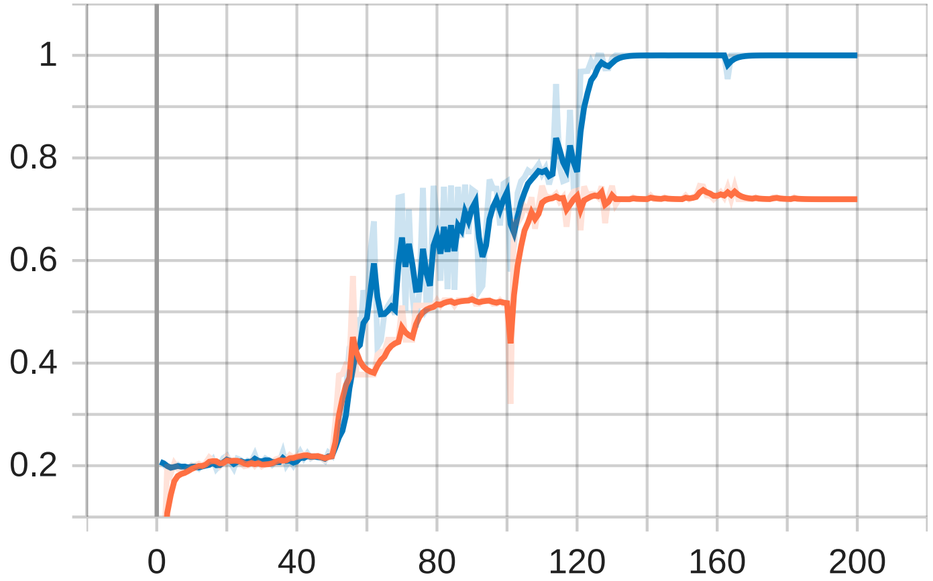
test
train



Metric : F1
Testing : 0.7103
Training : 1

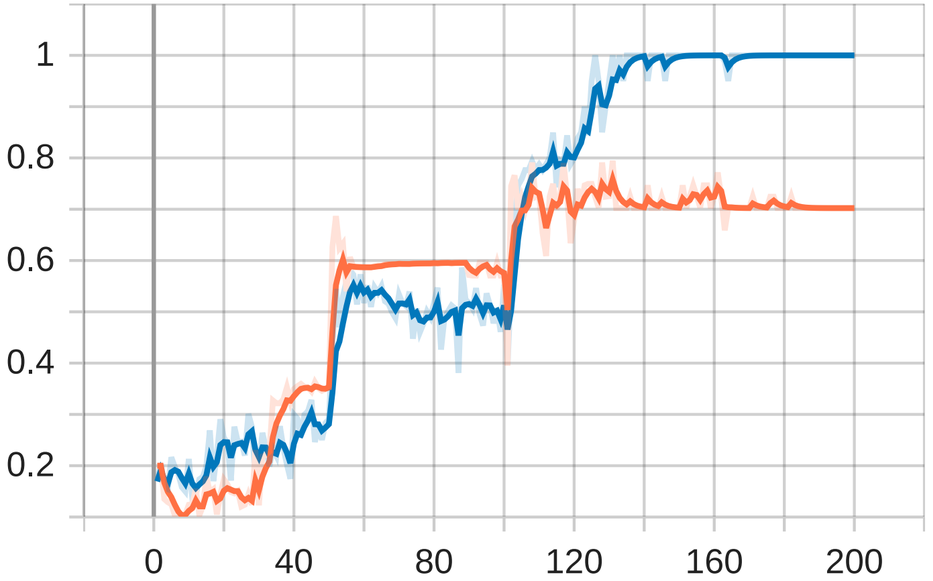
test
train

14



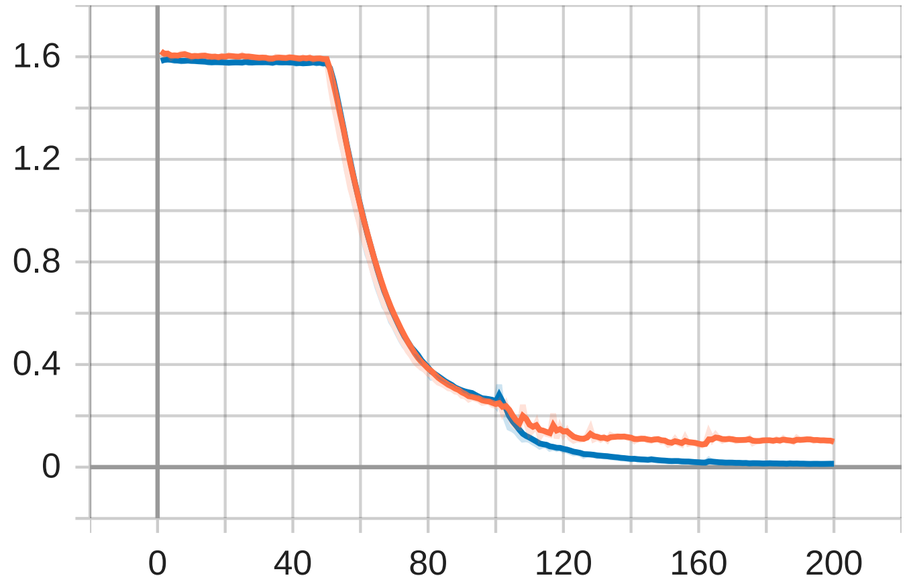
Metric 1 : precision
Testing : 0.7195
Training : 1

test
train



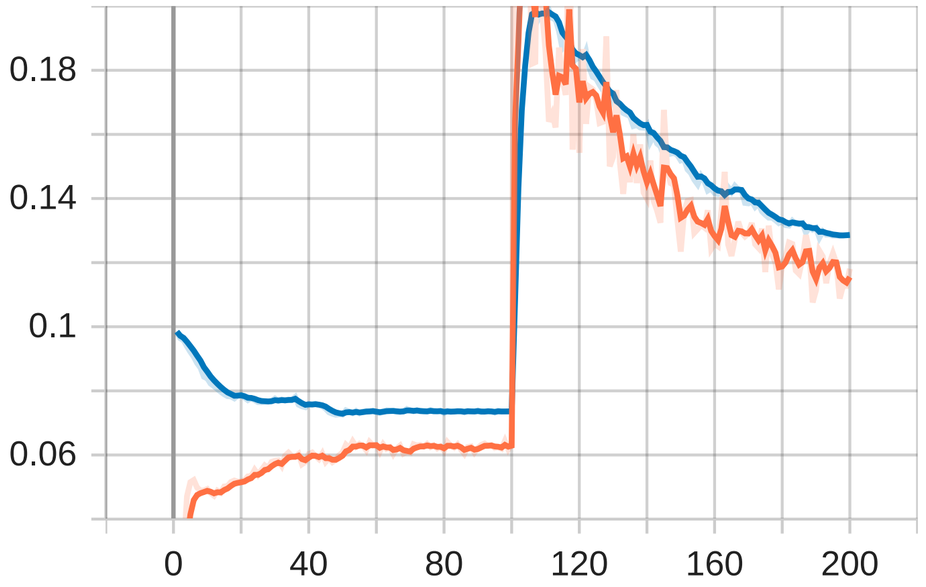
Metric 1 : recall
Testing : 0.7022
Training : 1

test
train



Classifier Loss
Testing : 0.09333
Training : 0.01282

test
train



KL loss
Testing : 0.1181
Training : 0.1287

test
train

Areas for Improvement

1. Dimensionality reduction: Omics datasets are high dimensional, which makes it difficult to visualize and analyze. By reducing the number of features, we can reduce the dimensionality of the dataset and make it easier to analyze.
2. Reducing noise: Omics datasets can have a lot of noise and redundant information, which can lead to overfitting and poor performance of the model. By reducing the number of features, we can remove the noise and redundant information, which can improve the performance of the model.
3. Faster training: Omics datasets can be very large, which can make training the model time-consuming. By reducing the number of features, we can reduce the training time and make the model more efficient.

Improvement in the Baseline Model

The main improvement in the baseline model is reducing the number of features. By reducing the number of features, the model can be trained more efficiently and can reduce the overfitting problem. Additionally, reducing the number of features can lead to better interpretability of the model, which can help in understanding the biological processes underlying the data.

In our project, we focused on feature extraction using a deep learning autoencoder to learn a latent representation of the data. By using the latent representation as input to downstream tasks, we were able to reduce the number of features while still maintaining high accuracy in cancer subtype classification.

Predicting feature Importance

The VAE is trained using a dataset of input data with known features, and the encoder and decoder parameters are optimized to minimize the reconstruction error. After training, the learned latent representation can be analyzed to identify the dimensions with higher variance or magnitude, which may correspond to more important features. Linear regression can then be used to model the relationship between the learned latent representation and the original features, and the slope of the linear regression can indicate feature importance.

Predicting feature Importance

By calculating the reconstruction error for each feature using a trained VAE, we were able to determine the relative importance of features in our dataset. Higher reconstruction error values represented more important features, while lower reconstruction error values represented less important features. Our analysis showed that features with higher reconstruction error were more difficult to reconstruct accurately, suggesting they may be more important for capturing variability in the data. Conversely, features with lower reconstruction error were easier to reconstruct, indicating they may be less important for capturing variability in the data.

Integration

We combined the latent representation and reconstruction loss analyses by taking their weighted sum, where the weight is determined by the hyperparameter lambda. By experimenting with different lambda values and evaluating the estimates against ground truth labels, we found that combining the two techniques led to more accurate estimates of feature importance. The optimal lambda value may depend on the specific dataset and task at hand, so it's important to tune lambda for each dataset to obtain the most accurate estimates. This approach improved computational efficiency for our large dataset.

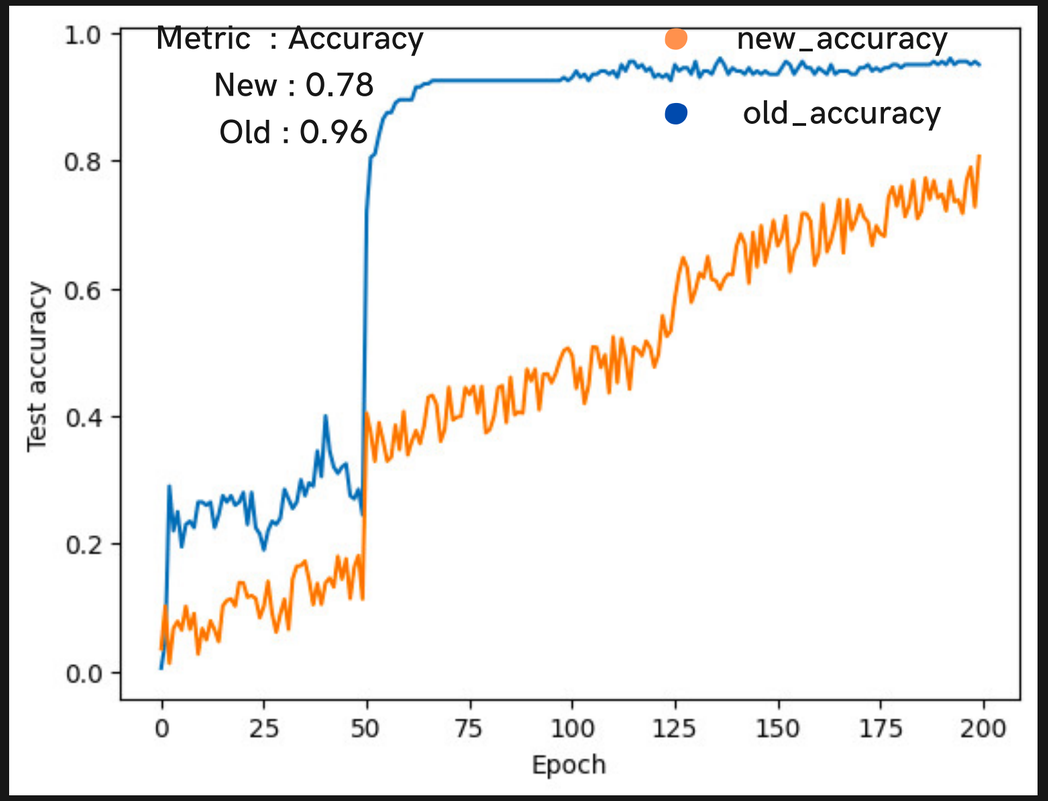
Challenges

1. Huge data: Working with large amounts of data can be a challenge in terms of storage and processing power. This can lead to slower training times, longer computation times, and higher costs for computing resources.
2. Huge training time: Deep learning models often require a lot of time to train, especially when working with large datasets. This can lead to longer development cycles and slower iteration times.
3. High number of features: Omics data often contain a large number of features, which can make it challenging to train accurate models. This can lead to overfitting, poor performance, and longer training times.

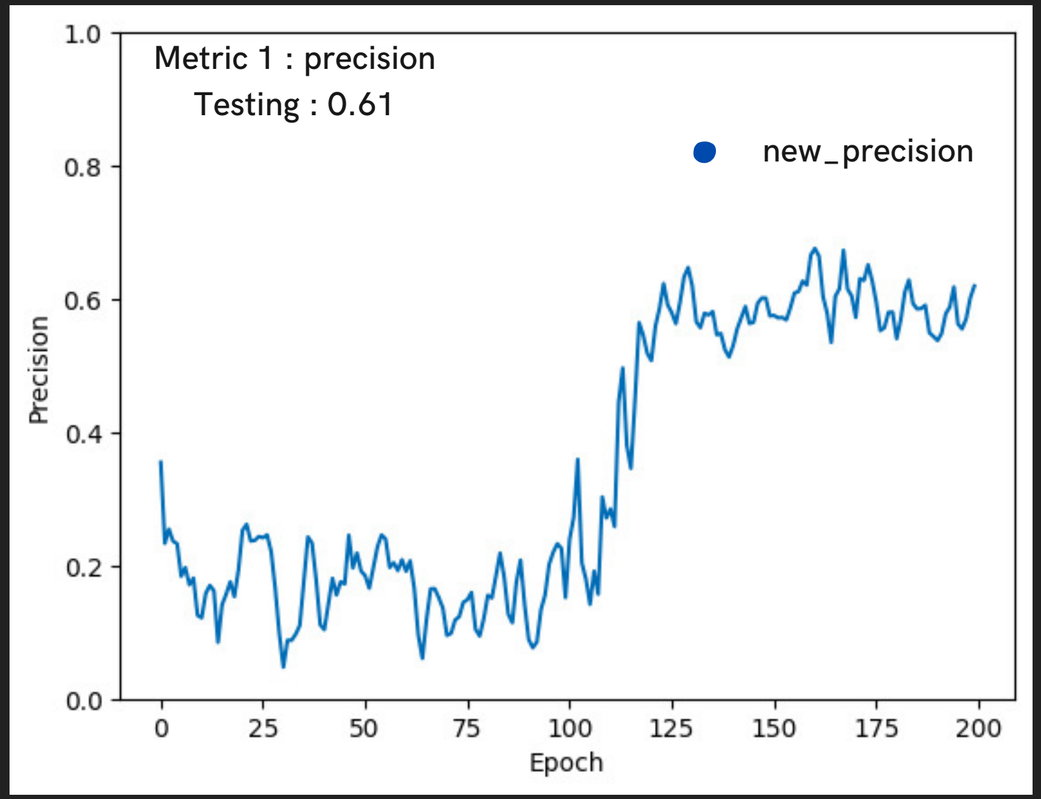
Results

We modified the OmiEmbed framework to reduce the number of features for analyzing omics data, using feature selection and feature extraction techniques. We evaluated the performance on benchmark datasets and found that the feature extraction technique outperformed feature selection. The training time and efficiency was checked for 7 values of hyperparameters ranging from 0,1,0.25,0.5,1,2,4,10 and best results were achieved for the value of 0.25 with training time being reduced from ~13 minutes to 49 seconds and accuracy reducing from 97.5% to 78.2% for a set of 1000 samples of 5 tumour types by incorporating only 50000 features instead of 4.5 lakhs. Our modifications show promising results in improving training time and efficiency, and future work could involve exploring other techniques and optimizing hyperparameters.

Results


Metric : Accuracy
New : 0.78
Old : 0.96
new_accuracy
old_accuracy
Test accuracy
Epoch
1.0
0.8
0.6
0.4
0.2
0.0
0
25
50
75
100
125
150
175
200

Results


Metric 1 : precision
Testing : 0.61
new_precision
Precision
Epoch
1.0
0.8
0.6
0.4
0.2
0.0
0
25
50
75
100
125
150
175
200

Our Team

Urvish Pujara - 2020101032

Contribution:

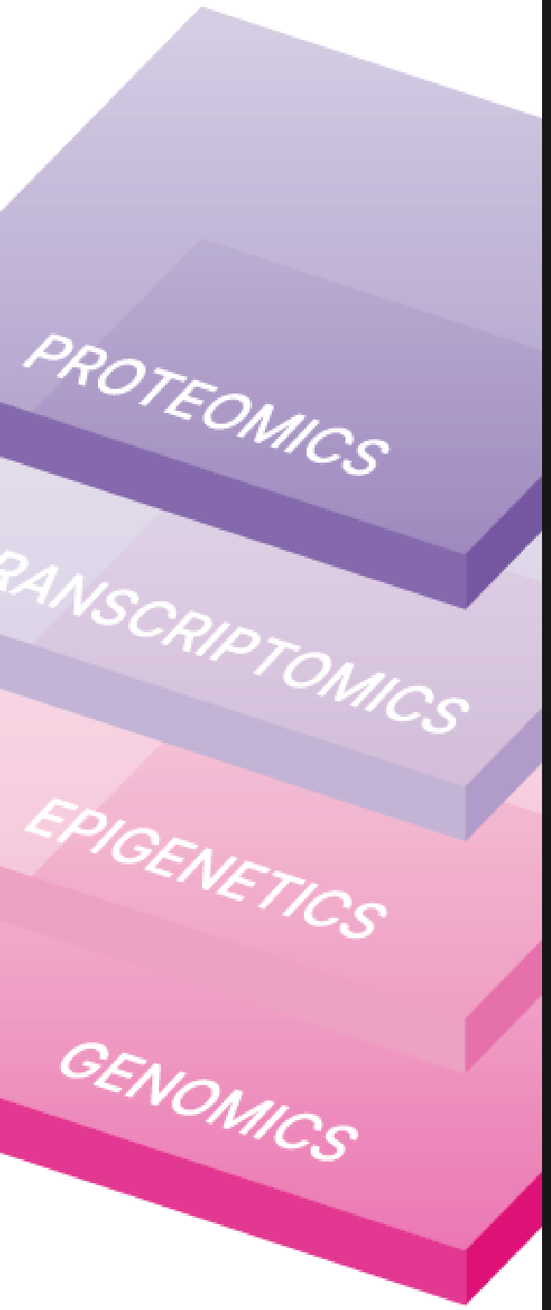
- Feature Selection (both methods)
- Documentation
- Data Analysis

Syed Imami- 2020113012

Contribution:

- Baseline Implementation
- Integration of both methods
- Generating plots

Resources



LINK

OmiEmbed: A Unified Multi-Task Deep Learning Framework for Multi-Omics Data
Published on:18 June 2021
Journal: Cancers - Volume 13, Issue 12

LINK

Performance Comparison of Deep Learning Autoencoders for Cancer Subtype Detection Using Multi-Omics Data
Published on: 22, April 2021
Journal: Cancers (Basel)

LINK

Multi-omics Data Integration, Interpretation, and Its Application
Published in: 2020
Journal: Bioinform Biol Insights , v.14; 2020

Thank you