INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 07 ISSUE: 06 | JUNE - 2023 SJIF RATING: 8.176 ISSN: 2582-3930

Study of Single Cell Integration Using Machine Learning

[1]Anamika, [2]Ajay K Kaushik
[1]Maharaja Agrasen Institute of Technology, [2] Maharaja Agrasen Institute of Technology

Abstract *:* Single-cell genomics has revolutionized our understanding of biology by enabling the measurement of DNA, RNA, and proteins in individual cells. However, analyzing single-cell data presents several challenges due to sparse and noisy measurements, molecular sampling depths, and batch effects. Additionally, current pipelines for single-cell data analysis treat cells as static snapshots, disregarding underlying dynamical biological processes. Incorporating temporal dynamics alongside state changes over time is a crucial and ongoing challenge in single-cell data science.
This paper presents a methodology for analyzing single-cell multiomics data collected from mobilized peripheral CD34+ hematopoietic stem and progenitor cells (HSPCs) isolated from four healthy human donors. The data comprises five time points over a ten-day period, during which cells were cultured with StemSpan SFEM media supplemented with CC100 and thrombopoietin (TPO) and incubated at 37ºC. Two single-cell assays were used to measure two modalities each: chromatin accessibility (DNA) and gene expression (RNA) for the Multiome kit, and gene expression (RNA) and surface protein levels for the CITEseq kit.

The task is to predict gene expression from chromatin accessibility for the Multiome samples, and protein levels from gene expression for the CITEseq samples. The cell types include Mast Cell Progenitor, Megakaryocyte Progenitor, Neutrophil Progenitor, Monocyte Progenitor, Erythrocyte Progenitor, Hematoploetic Stem Cell, and B-Cell Progenitor.

The methodology includes exploratory data analysis, data pre-processing and feature engineering, and a model architecture comprising LightGBM and a neural network. The data pre-processing includes normalization, transformation, standardization, and batch-effect correction. Feature engineering involves decomposition methods such as Principal Component Analysis, Incremental PCA, and Factor Analysis, as well as feature selection based on stable correlations within each group. Cell-type encoding is done using a one-hot encoding scheme.
Cross-validation is performed using GroupK fold validation. We have been able to get an accuracy of 87.63% on the test dataset.

INTRODUCTION

In the last decade, single-cell genomics has revolutionized our understanding of biology by enabling the measurement of DNA, RNA, and proteins in individual cells. These technologies have provided unparalleled insights into cellular processes at an unprecedented scale and resolution. The outcomes have included intricate maps of early human embryonic development, the discovery of novel disease-associated cell types, and targeted therapeutic interventions. Recent advancements now allow the simultaneous measurement of multiple genomic modalities in the same cell, leading to multimodal single-cell data. However, despite the availability of such data, analysis methods remain limited. The analysis of single-cell data presents several challenges. Sparse and noisy measurements due to the small volume of a single cell, coupled with variations in molecular sampling depths and batch effects, often overshadow true biological differences. Furthermore, the current pipelines for single-cell data analysis treat cells as static snapshots, disregarding underlying dynamical biological processes. Incorporating temporal dynamics alongside state changes over time is a crucial and ongoing challenge in single-cell data science.
Traditionally, genetic information flows from DNA to RNA to proteins, with each step regulated by feedback mechanisms. In single-cell data science, dynamic processes have been modeled through pseudotime algorithms, capturing the progression of biological processes. However, generalizing these algorithms to account for both pseudotime and real time remains an open problem.
Understanding how a single genome gives rise to the diverse range of cellular states is crucial for unraveling the mechanistic insights behind tissue function and dysfunction in health and disease. Addressing the challenges of predicting cellular behaviors over time could potentially yield groundbreaking insights into how gene regulation influences the differentiation of blood and immune cells.
In recent years, machine learning techniques have played a pivotal role in advancing research in various domains, including bioinformatics and data science. Two powerful algorithms that have gained significant attention are LightGBM and Multilayer Perceptron (MLP).

LightGBM, a gradient boosting framework developed by Microsoft, has become a popular choice for modeling complex datasets. It is known for its efficiency, scalability, and accuracy in handling large-scale datasets. LightGBM utilizes a tree-based ensemble method, where each tree is grown in a leaf-wise fashion rather than the traditional level-wise approach. This unique approach allows for faster convergence and improved performance, especially in scenarios with a large number of features. Moreover, LightGBM supports various optimization techniques, such as bagging and boosting, making it a versatile tool for predictive modeling and feature selection.

On the other hand, MLP, a type of artificial neural network, has been widely used for solving complex problems due to its ability to capture non-linear relationships between features. MLP consists of multiple layers of interconnected nodes (neurons), each performing a weighted sum of inputs followed by an activation function. This architecture enables the model to learn complex patterns and make accurate predictions. MLPs can be trained using backpropagation, an algorithm that adjusts the weights of the connections between neurons to minimize the error between predicted and actual values. With advancements in hardware and the availability of deep learning frameworks, training deep MLPs with multiple hidden layers has become feasible, further improving their modeling capabilities.

In this research, we leverage the strengths of LightGBM and MLP to develop a robust predictive model. LightGBM harnesses its efficiency and accuracy in handling diverse input features, while MLP excels in capturing complex relationships and making accurate predictions. By combining these two algorithms, we aim to leverage the advantages of both techniques and enhance the predictive performance for our specific research problem.

The remainder of this paper is organized as follows: In the next section, we describe the dataset, methodology and data preprocessing steps, including feature engineering techniques used to prepare the data for modeling. Subsequently, we present the experimental setup, including model configurations and evaluation metrics. Finally, we discuss the results, draw conclusions, and provide insights for future research.

DATASET

The datase comprises single-cell multiomics data collected from mobilized peripheral CD34+ hematopoietic stem and progenitor cells (HSPCs) isolated from four healthy human donors.
Measurements were taken at five time points over a ten-day period. During this time, cells were cultured with StemSpan SFEM media supplemented with CC100 and thrombopoietin (TPO) and incubated at 37ºC. Media was changed every 2-3 days. No additional media supplements were added to the cell culture conditions.

From each culture plate at each sampling time point, cells were collected for measurement with two single-cell assays. The first is the 10x Chromium Single Cell Multiome ATAC + Gene Expression technology (Multiome) and the second is the 10x Genomics Single Cell Gene Expression with Feature Barcoding technology technology using the TotalSeq™-B Human Universal Cocktail, V1.0 (CITEseq).
Each assay technology measures two modalities. The Multiome kit measures chromatin accessibility (DNA) and gene expression (RNA), while the CITEseq kit measures gene expression (RNA) and surface protein levels.
Following the central dogma of molecular biology: DNA --> RNA-->Protein, the task :
For the Multiome samples: given chromatin accessibility, predict gene expression.
For the CITEseq samples: given gene expression, predict protein levels.

Cell Types

MasP = Mast Cell Progenitor
MkP = Megakaryocyte Progenitor
NeuP = Neutrophil Progenitor
MoP = Monocyte Progenitor
EryP = Erythrocyte Progenitor
HSC = Hematoploetic Stem Cell
BP = B-Cell Progenitor
File and Field Descriptions
metadata.csv
cell_id - A unique identifier for each observed cell.
donor - An identifier for the four cell donors.

day - The day of the experiment the observation was made.
technology - Either citeseq or multiome.
cell_type - One of the above cell types or else hidden.
The experimental observations are contained in several large arrays.

Multiome

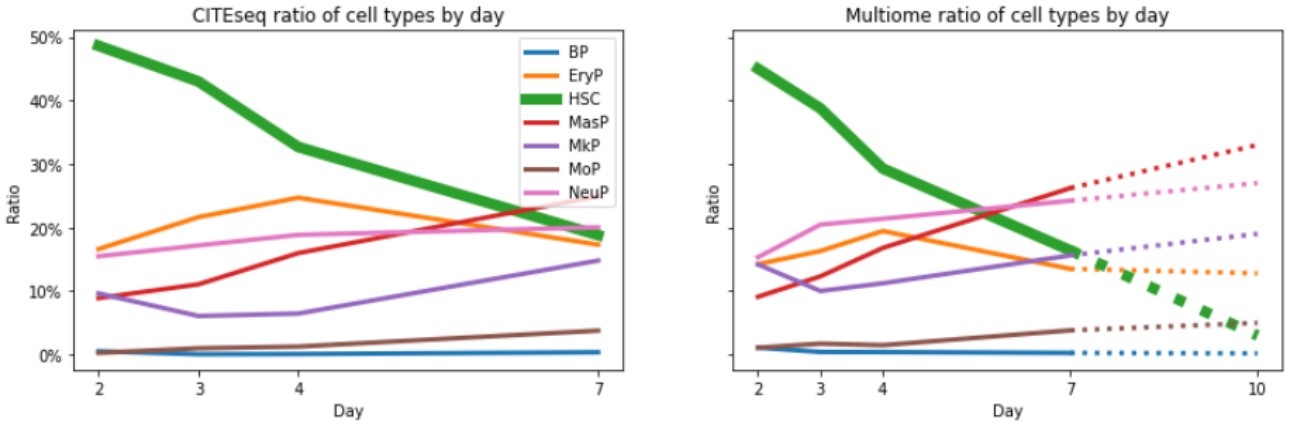
train/test_multi_inputs.h5 - ATAC-seq peak counts transformed with TF-IDF using the default log(TF) * log(IDF) output (chromatin accessibility), with rows corresponding to cells and columns corresponding to the location of the genome whose level of accessibility is measured, here identified by the genomic coordinates on reference genome GRCh38 provided in the 10x References - 2020-A (July 7, 2020).
train_multi_targets.h5 - RNA gene expression levels as library-size normalized and log1p transformed counts for the same cells.
CITEseq
train/test_cite_inputs.h5 - RNA library-size normalized and log1p transformed counts (gene expression levels), with rows corresponding to cells and columns corresponding to genes given by {gene_name}_{gene_ensemble-ids}.
train_cite_targets.h5 - Surface protein levels for the same cells that have been dsb normalized.

RESEARCH METHODOLOGY

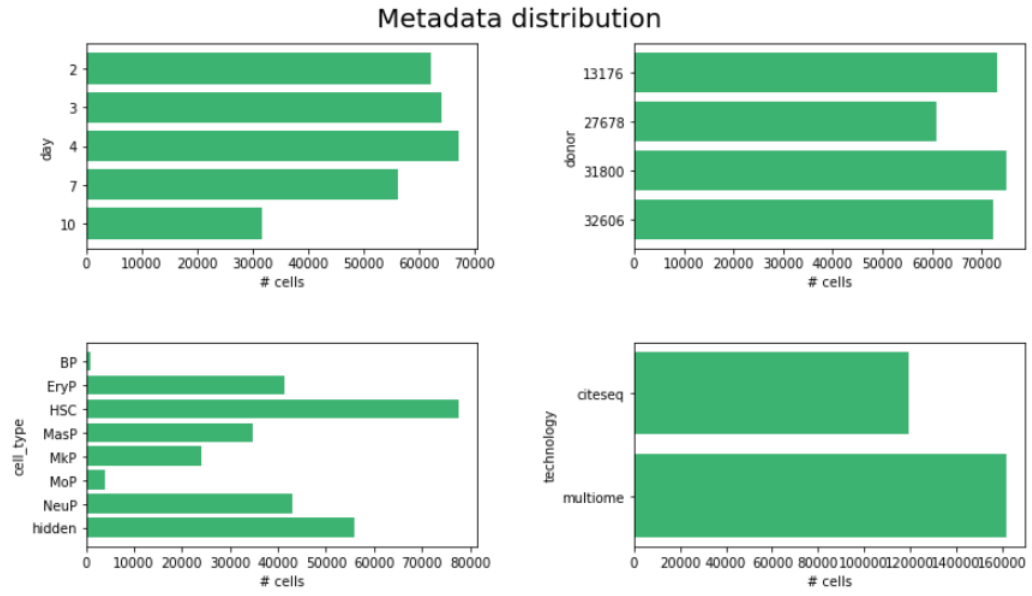
Exploratory Data Analysis

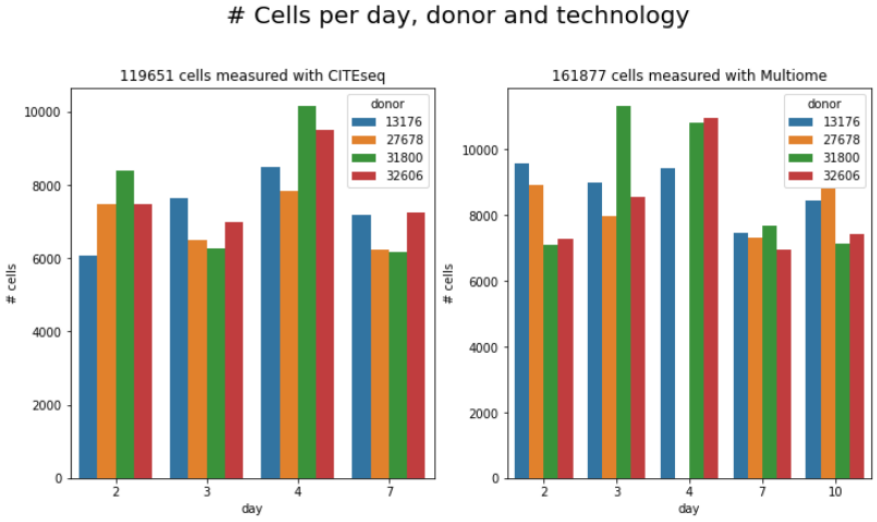
The metadata table provides insight into the training and test data, showing that there are 281528 unique cells belonging to five days, four donors, eight cell types (including one type named 'hidden'), and two technologies. There are no missing values in the metadata table.

Further analysis reveals that each cell is used only on a single day and then discarded, indicating that there are no time series over single cells. Additionally, the two technologies do not share cells, meaning that two completely independent models can be created, one per technology, even if they share the same four donors. Therefore, it's recommended to work with two separate notebooks, one for CITEseq and the other one for Multiome.



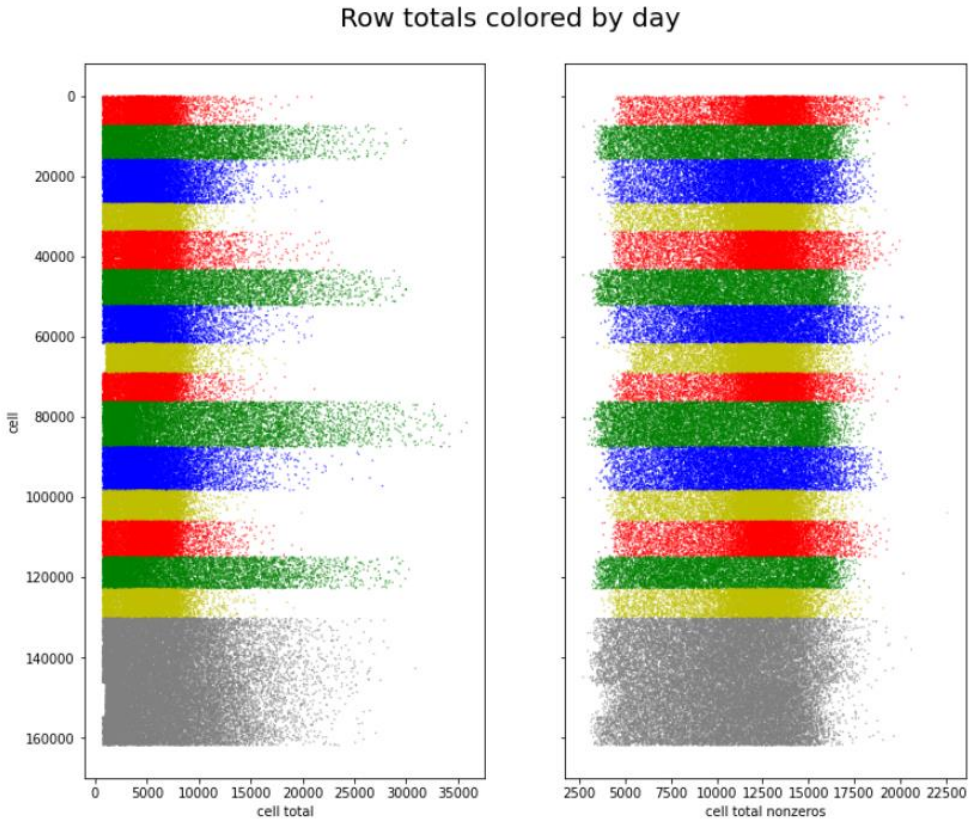
The CITEseq measurements took place on four days, the Multiome measurements on five (except that there are no measurements for donor 27678 on day 4. For every combination of day, donor and technology, there are around 8000 cells:

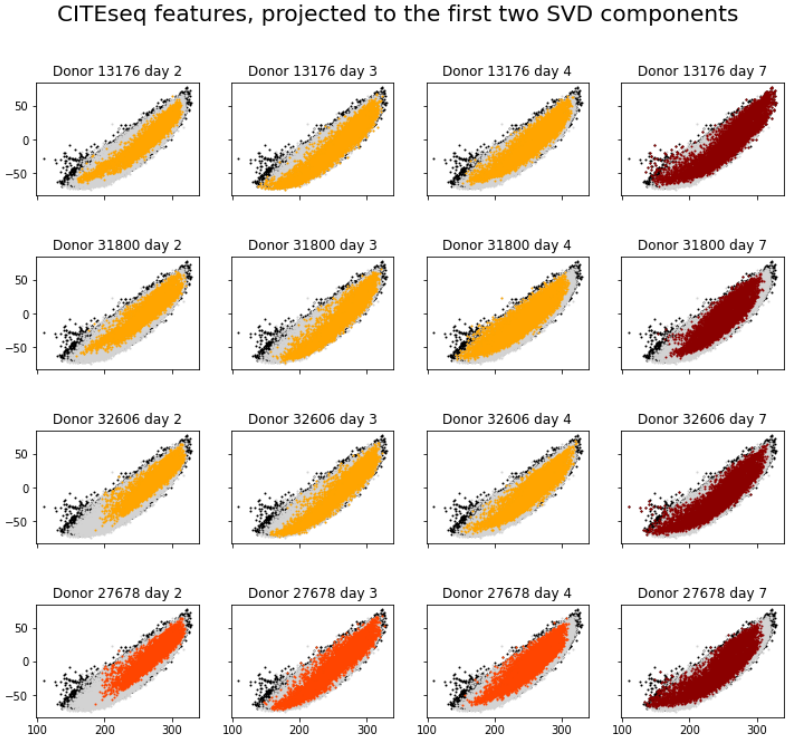




Biological experiments often suffer from batch effects, which lead to systematic differences in measurements when the same experiment is repeated multiple times. In this case, we observed batch effects in ATACseq measurements, which we visualized by plotting the row sums of the train and test datasets and colored the dots by the day of the experiment. The diagram clearly showed that day 3 gave the highest measurements, while day 7 gave the lowest, with some variations among donors.

Based on this insight, we recommend normalizing the data in a preprocessing step to mitigate the effects of batch differences on predictions. Additionally, we suggest using GroupKFold on days for cross-validation to ensure that the model remains robust against differences between experimental days.


Row totals colored by day
cell
0
20000
40000
60000
80000
100000
120000
140000
160000
0 5000 10000 15000 20000 25000 30000 35000
cell total
2500 5000 7500 10000 12500 15000 17500 20000 22500
cell total nonzeros


CITEseq features, projected to the first two SVD components
Donor 13176 day 2
Donor 13176 day 3
Donor 13176 day 4
Donor 13176 day 7
Donor 31800 day 2
Donor 31800 day 3
Donor 31800 day 4
Donor 31800 day 7
Donor 32606 day 2
Donor 32606 day 3
Donor 32606 day 4
Donor 32606 day 7
Donor 27678 day 2
Donor 27678 day 3
Donor 27678 day 4
Donor 27678 day 7
50
0
−50
100 200 300

The gray area marks the complete training data (union of the nine orange diagrams), and the black area behind marks the test data.The three diagrams with orange-red dots are the validation set and the four dark red diagrams at the right are the test set. The diagrams confirm the data leak described above: The lower two diagrams of day 2 are identical.

Data Pre-processing and Feature Engineering

Pre-processing :

1. Normalization: Perform sample normalization by mean values over features. This ensures that the expression values of each sample are adjusted based on the mean expression across all features (genes).
2. Transformation: Apply a square root transformation to the normalized data. This transformation helps stabilize the variance and mitigate the impact of high expression values, which can dominate the analysis.
3. Standardization: Conduct feature z-score standardization. This involves calculating the z-score for each feature across all samples, which enables comparison of expression levels relative to the mean and standard deviation of each feature.
4. Batch-effect correction: Address batch effects by treating the "day" variable as the batch indicator. For each batch, compute the column-wise median, representing a "median-sample" for that batch. Subtract this median-sample from each sample within the batch. This approach helps minimize the influence of batch effects while keeping the correction process simple and low-risk.
5. Row-wise Z-score transformation: Before inputting the preprocessed data into a neural network, perform row-wise z-score transformation. This involves calculating the z-score for each individual gene across all samples, ensuring that each gene's expression values are standardized relative to the mean and standard deviation within that gene.

Feature Engineering

Decomposition Methods:

a) Principal Component Analysis (PCA): Perform PCA with 64 components to extract linear combinations of features that capture the maximum variance in the data.

b) Incremental PCA (IPCA): Utilize IPCA with 128 components to incrementally calculate the principal components, which can be advantageous for handling large datasets efficiently.

c) Factor Analysis: Apply factor analysis with 64 components to identify latent factors that explain the correlations among observed variables.

It is important to note that relying solely on PCA might not be sufficient, as utilizing additional decomposition features has shown improved performance in cross-validation and leaderboard metrics.

Feature Selection:

Instead of selecting features solely based on high correlation with the target in the entire dataset, it is preferable to select features that exhibit stable correlations with the target within each group (donor-day). Even if the Pearson correlation coefficient (PCC) is slightly lower, features demonstrating consistent associations are more likely to be reliable indicators.

Additionally, consider including features with the same name as the target by relaxing the PCC threshold. This can capture potential relationships between these specific features and the target variable.

Cell-Type Encoding (One-Hot):

Encode cell types using a one-hot encoding scheme. This allows the representation of categorical cell types as binary vectors, where each cell type is represented by a unique combination of 0s and 1s.

Model Architecture

Lightgbm+NN

For the LightGBM models, we trained four different models with distinct input features. The predictions from these models were transformed using Truncated Singular Value Decomposition (TSVD) and used as meta-features for the neural network (NN) model.

Feature Set 1: Library-size Normalized and Log1p Transformed Counts
Input: Library-size normalized and log1p transformed counts
Preprocessing: Apply TSVD to the predictions from the LightGBM model trained on this feature set

Feature Set 2: Raw Counts with Centered Log Ratio (CLR) Transformation
Input: Raw counts transformed using the CLR method
Preprocessing: Apply TSVD to the predictions from the LightGBM model trained on this feature set
Feature Set 3: Raw Counts
Input: Raw counts without any transformation

Preprocessing: None

Feature Set 4: Raw Counts with Raw Target
Input: Raw counts with the raw target
Preprocessing: None

For the NN model, a basic three-layer Multi-Layer Perceptron (MLP) architecture was employed. However, a notable modification was made by either replacing the first dense layer with a Gated Recurrent Unit (GRU) layer or adding a GRU layer after the final dense layer. This modification was found to improve the performance of the model.

Cross Validation strategy
For cross validation we use GroupK fold validation.
Model Performance
F1 score is calculated to test performance of the model. The equations used are

$$\text{Precision} = \frac{T_\text{p}}{T_\text{p} + F_\text{p}}$$

$$\text{Recall} = \frac{T_\text{p}}{T_\text{p} + F_\text{n}}$$

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where Tp denotes the number of images that are true positive, and Fn, Fp denote the false negative and false positive, respectively.

RESULTS AND DISCUSSION

Results of Machine Learning Architecture and inputs
Table 4.1: Accuracy of model with different input feature

LightGBM+ NN	Feature set 1	Feature set 2	Feature set 3	Feature set 4
Accuracy	87.63	80.91	80.66	61.20
Train Loss	0.134	0.142	0.131	0.131
Validation Loss	0.134	0.134	0.142	0.162

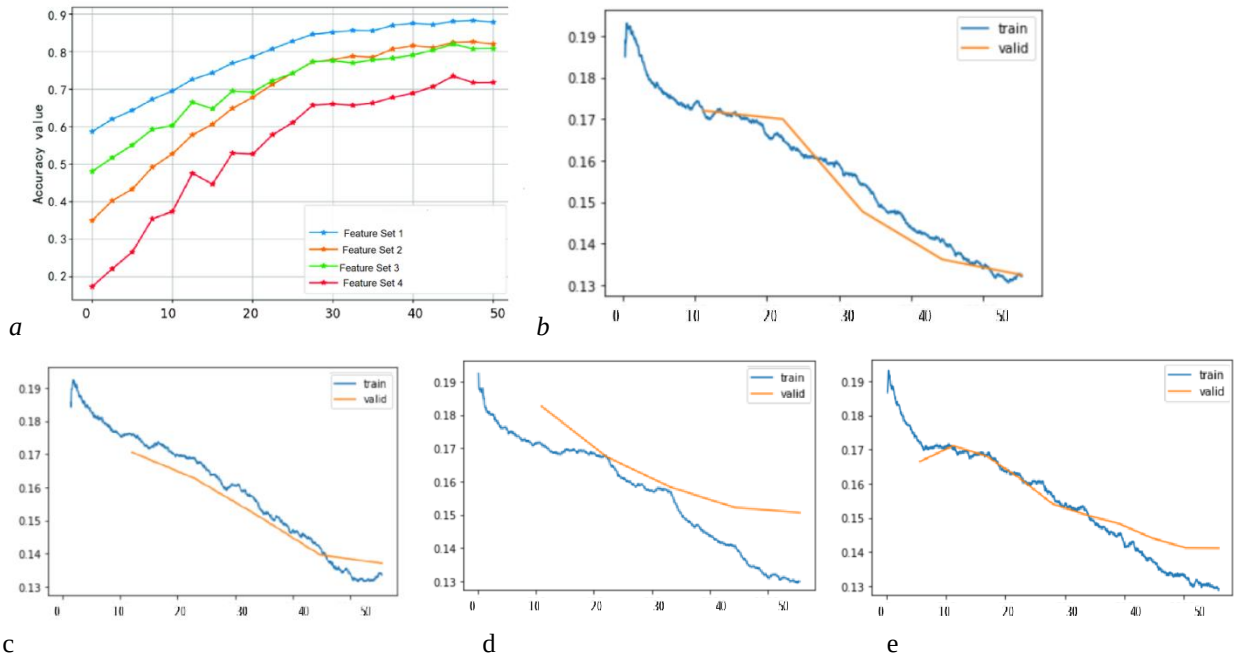


fig a : Accuracy vs epochs *fig b : train loss, validation loss vs epochs on feature set 1* f*ig c : train loss, validation loss vs epochs on feature set 2* f*ig d : train loss, validation loss vs epochs on feature set 3* f*ig e : train loss, validation loss vs epochs on feature set 4*

Discussion

The predictive modeling approach combines the strengths of LightGBM, a gradient boosting framework known for its efficiency and accuracy, with a neural network architecture designed to capture complex relationships. This hybrid model is used to predict gene expression from chromatin accessibility for the Multiome samples and protein levels from gene expression for the CITEseq samples.

Evaluation of the models using GroupK fold validation demonstrates promising results, achieving an accuracy of 87.63% on the test dataset. This suggests that the integration of multiomics data and the utilization of advanced machine learning techniques can effectively predict gene expression and protein levels in single cells.

This research contributes to the ongoing challenge of incorporating temporal dynamics and capturing the dynamic biological processes within single-cell data. By leveraging the power of single-cell genomics, the study provides valuable insights into the dynamics and functional characteristics of hematopoietic stem and progenitor cells.

The outcomes of this project have implications beyond the specific context of hematopoietic cells, highlighting the potential of single-cell multiomics analysis in enhancing our understanding of diverse biological systems. The methodology developed here serves as a foundation for further studies exploring the temporal dynamics of single-cell processes and the elucidation of underlying mechanisms in cellular function and dysfunction

ACKNOWLEDGMENT

I would like to express my sincere thanks to Mr. Ajay K. Kaushik, for his valuable guidance and support in completing my project. I would also like to express my gratitude towards our HOD Dr. M. L. Sharma for giving me this great opportunity to do a project on Study Of Single Cell Integration Using Machine Learning. Without their support and suggestions, this project would not have been completed.

REFERENCES

[1] Ricard Argelaguet, Damien Arnol, Danila Bredikhin, Yonatan Deloro, Britta Velten, John C Marioni, and Oliver Stegle. 2020. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome biology 21, 1 (2020), 1–17.
[2] Fatima Batool and Christian Hennig. 2021. Clustering with the average silhouette width. Computational Statistics & Data Analysis 158 (2021), 107190.

[3] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. ArXiv preprint (2018).
[4] Junyue Cao, Darren A Cusanovich, Vijay Ramani, Delasa Aghamirzaie, Hannah A Pliner, Andrew J Hill, Riza M Daza, Jose L McFaline-Figueroa, Jonathan S Packer, Lena Christiansen, et al. 2018. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. Science 361, 6409 (2018), 1380–1385.
[5] Zhi-Jie Cao and Ge Gao. 2022. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. Nature Biotechnology (2022), 1–9.
[6] Song Chen, Blue B Lake, and Kun Zhang. 2019. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. Nature biotechnology 37, 12 (2019), 1452–1457.
[7] Madalina Ciortan and Matthieu Defrance. 2022. GNN-based embedding for clustering scRNA-seq data. Bioinformatics 38, 4 (2022), 1037–1044.
[8] Jiayuan Ding, Hongzhi Wen, Wenzhuo Tang, Renming Liu, Zhaoheng Li, Julian Venegas, Runze Su, Dylan Molho, Wei Jin, Wangyang Zuo, et al. 2022. DANCE: A Deep Learning Library and Benchmark for Single-Cell Analysis. bioRxiv (2022).