Robust Multi-Omics Prediction (RoMOP) for RNA Expression & Protein Surface Levels

Zachary Abessera (zca2001) and Quentin Chappat* (qjc2002)*

Department of Biomedical Engineering
Columbia University
New York, USA

**indicates equal contribution*

Abstract - This report describes an approach to predict how DNA, RNA, and protein measurements co-vary in single cells as bone marrow stem cells develop into more mature blood cells. With the increasing availability of experimental techniques to measure multiple modalities within a single cell, the demand for these measurements is driven by the promise to provide a deeper insight into the state of a cell. The modalities are intrinsically linked; DNA must be accessible to produce mRNA, and mRNA in turn is used as a template to produce protein. The challenge in this task is to predict a paired modality measured in the same cell from a later time point in the dataset, which has not been seen during training. This challenge may yield new insights into how gene regulation influences differentiation and how tissues function or malfunction in health and disease. The ability to predict one modality from another may expand our understanding of the rules governing complex regulatory processes, and accelerate innovation in methods of mapping genetic information across layers of cellular state.

1. Introduction & Datasets

In the last decade, single-cell genomics has revolutionized the study of biology by enabling the measurement of DNA, RNA, and proteins in individual cells [1]. These technologies have produced detailed maps of early human embryonic development, new disease-associated cell types, and cell-targeted therapeutic interventions. Recent advances in experimental techniques have made it possible to measure multiple genomic modalities in the same cell [2]. Despite this, data analysis methods for multimodal single-cell data are still limited. The small volume of a single cell leads to sparse and noisy measurements, with the additional biological confound of differences in sequencing depth and batch effects [3]. Moreover, current analysis pipelines treat cells as static snapshots, even though underlying biological processes are known to be dynamic.

In genetic regulation, information flows from DNA to RNA to proteins through a feedback mechanism. For example, a protein can bind DNA to prevent the production of more RNA. Dynamic cellular processes rely on this genetic regulation, which allows organisms to develop and adapt to changing environments. Modeling these processes in single-cell data science has been accomplished through pseudotime algorithms that capture the progression of the biological process [4], [5], [6]. However, extending these algorithms to account for both pseudotime and real time remains an open problem. With approximately 37 trillion cells in the human body, understanding how a single genome gives rise to diverse cellular states is crucial for gaining mechanistic insights into tissue function and dysfunction. Solving the prediction problems over time may provide new insights into how gene regulation influences differentiation as blood and immune cells mature. Therefore, this report aims to predict how DNA, RNA, and protein measurements co-vary in single cells as bone marrow stem cells develop into more mature blood cells.

The dataset we are using is made available by the Open Problems in Single-Cell Analysis team [7]. It consists of single-cell multiomics data obtained from mobilized peripheral CD34+ hematopoietic stem and progenitor cells (HSPCs) isolated from four healthy human donors. The cells were cultured for a period of ten days, during which measurements were taken at five time points. We got access to four out of five time point measurements. At each sampling time point, two single-cell assays were conducted: the 10x Chromium Single Cell Multiome ATAC + Gene Expression technology (Multiome) [8] and the 10x Genomics Single Cell Gene Expression with Feature Barcoding technology [9] using the TotalSeq™-B Human Universal Cocktail, V1.0 (CITE-seq). The Multiome kit measures chromatin accessibility (DNA) and gene expression (RNA), while the CITE-seq kit measures gene expression (RNA) and surface protein levels.

The objective of the current work is to predict one modality from another based on the central dogma of molecular biology. Specifically, for the Multiome samples, the goal is to predict gene expression from chromatin accessibility, while for the CITE-seq samples, the goal is to predict protein levels from gene expression. The dataset provides measurements from an unseen later time point, making the prediction task more challenging.

2. Methods & Algorithms

A. Multiome Data and Model

As mentioned, the Multiome dataset contains chromatin accessibility as inputs and gene expressions as outputs. In other words, our goal with this dataset was to predict each gene's expression given chromatin accessibility for a specific cell. Importantly, we incorporated real time into our model by making the prediction with respect to an unseen later time point than any of the time points in the training dataset. Therefore, we divided the Multiome data into training (days 2, 3, 4) and testing set (day 7). Our evaluation metrics were Pearson correlation coefficient and MSE. The tested machine learning algorithms were: Elastic Net, Linear Regression, Lasso, Ridge Regression, Bayesian Ridge Regression, and Automatic Relevance Determination [10].

The Multiome data preprocessing pipeline can be found in Figure 1. The input data was first converted to sparse matrices [13] (around 98% of the data contained zeros). The variance explained for the highest 100 singular values for both the inputs and outputs (Figure 2A) had clear elbows before 40 features. As a follow up pre-processing, we calculated and visualized the correlation between input features and output features (Figure 2B). Due to this result, we opted to keep the 10 most correlated singular features to each singular target. A singular feature here is referred to as a singular vector that was the output of the SVD on the inputs, whereas a singular target is referred to as a singular vector that was the output of the SVD on the targets.

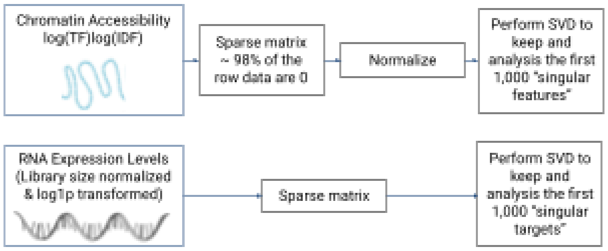

Fig. 1. Multiome preprocessing pipeline of inputs (top) and outputs or targets (bottom).

Additionally, each discrete cell type was used as a hot encoded variable. The methods for clustering cell types into seven categories can be found in Velten et al. [14].

RandomizedSearchCV [10] was used to validate the models and optimize the hyperparameters of each model. The validation split was done using KFold [10] (4 days & 3 patients = 12 folds).

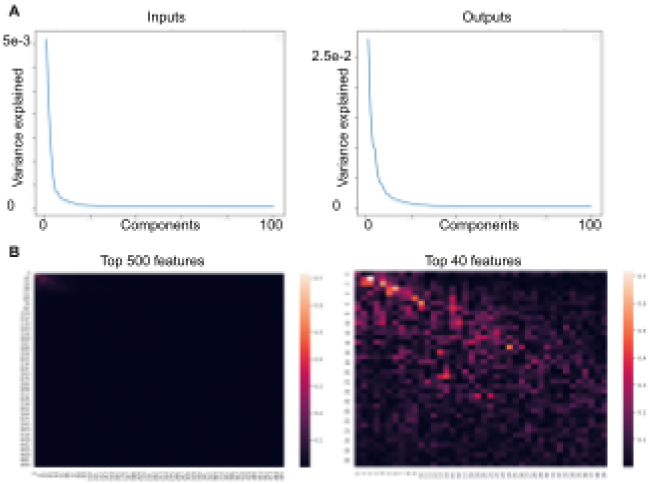

Fig. 2. Dimensionality reduction pre-processing. (A) Variance explained for the first 100 singular values for inputs (left) and targets or outputs (right). (B) Correlation between inputs (x-axis) and targets (y-axis) for top 500 features (left) and top 40 features (right).

B. CITE-seq Data and Model

As described earlier, the CITE-seq dataset contains gene expressions as inputs and surface protein levels as outputs. Similarly as what we did with the Multiome dataset, our goal with the CITE-seq dataset was to predict surface protein levels given gene expressions for a specific cell at an unseen later time point than any of the time points in the training dataset. However, the CITE-seq dataset did not contain data for day 7. Thus, we divided our dataset into a training set containing data from days 2 and 3, and a testing set with data from day 4. Our evaluation metrics were Pearson correlation coefficient and MSE. We attempted to model the relationship between these two sets of variables using three different machine learning algorithms: a basic multi-output linear regression, a LightGBM model [15], and a sequential neural network with four hidden layers.

The input data had already undergone preprocessing using a basic pipeline consisting of library size normalization and a counts log1p transformation. We ran the multi-output linear regression and the LightGBM models two times: once with default preprocessing and principal component analysis (PCA) as dimensionality reduction, and another time working with sparse matrices for memory efficiency, an added preprocessing and truncated singular value decomposition (SVD) as dimensionality reduction. The added preprocessing involved deleting constant features always equal to zero that did not add any information to the data and storing 'important features' apart so that it would not undergo dimensionality reduction. To define the 'important features', we chose genes whose name matches the name of one of the target proteins. For instance it is clear that the gene 'ENSG00000114013_CD86' as an input is related to the target protein 'CD86'. Thus, we kept this set of genes apart, we ran dimensionality reduction over all other features, and we finally concatenated the 'important features' with the reduced features before training the models (Figure 3).

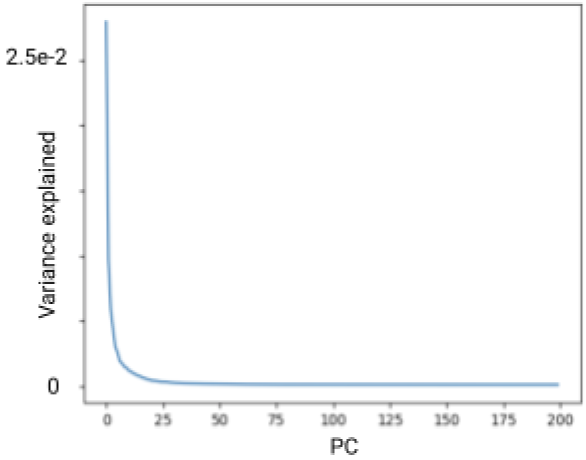

Fig. 3. CITE-seq variance explained (y-axis) by principal component (x-axis).

Truncated SVD is a popular technique for dimensionality reduction that computes only the top k singular vectors where k is a user-defined parameter determining the desired dimensionality of the reduced data. It is well-suited for working with sparse matrices. Unlike PCA, truncated SVD does not require the data to be centered.

LightGBM is a popular gradient boosting framework that has gained attention in recent years for its impressive performance on large datasets. Unlike traditional gradient boosting frameworks that use a depth-first approach to build trees, LightGBM uses a leaf-wise approach that can reduce memory usage and provide faster training times. Additionally, LightGBM is designed to handle large datasets by supporting parallel and distributed computing. It uses histogram-based algorithms to discretize continuous features, which can greatly reduce memory usage without sacrificing accuracy. LightGBM also includes several other features, such as early stopping and regularization, to prevent overfitting and improve model performance. These characteristics make LightGBM a valuable tool for applications that involve large and complex datasets.

The last point concerning the models we want to highlight is that we used Bayesian optimization to tune the hyperparameters of our sequential neural network, including the sizes of the hidden layers and the regularization factors. Unlike grid search and random search, which explore the hyperparameter space exhaustively or randomly, respectively, Bayesian optimization uses a probabilistic model to predict the performance of different hyperparameters and guide the search towards promising regions of the space. Bayesian optimization can be more efficient than other methods when the search space is large or when evaluating the performance of each set of hyperparameters is computationally expensive.

3. Detailed Implementation

We implemented all machine learning models using Python and the scikit-learn [10], Keras [11], and LightGBM [15] libraries.

For both implementations, we used the mean_squared_error function from sklearn.metrics and a correlation_score function we coded to evaluate each model.

A. Multiome Implementation

The Multiome dataset we used consisted of 105,942 cells with each 228,942 genomes' chromatin accessibility and 23,418 RNA gene expression levels.

To implement our models for this dataset, we utilized the scikit-learn library.. The RandomizedSearchCV class was used for cross validation, and the GroupKFold method was employed to separate the different folds of our pre-processed dataset by patient per day. In order to ensure reproducibility, a random state of 42 was set for cross-validation. For hyperparameter tuning, we created a grid dictionary for each model's hyperparameters, covering logarithmic intervals, with 20 iterations to achieve high precision.

Prior to modeling, we preprocessed our data by converting our two raw datasets to sparse matrices using the scipy.sparse.csr_matrix method. We then normalized our features by cells using the normalize function from scikit-learn. Subsequently, we applied the TruncatedSVD method from the sklearn.decomposition package to extract the 1,000 first components, along with the transformation matrix and explained variance. For hyperparameter tuning of TruncatedSVD, 10 iterations were performed, with a random state of 42 to ensure reproducibility. To improve the efficiency of our computations, we designed our own correlation function and computed the correlation between our singular features and singular targets.

B. CITE-seq Implementation

The CITE-seq dataset we used consisted of 70,988 samples, with 42,843 samples in the training set and 28,145 samples in the testing set.

For the multi-output linear regression model, we used the LinearRegression class from the scikit-learn library with default parameters. For the LightGBM model, we used the

LGBMRegressor class from the LightGBM library and we selected hyperparameters that have been shown to be effective in previous studies. For the sequential neural network, we used the Keras library to build a model with four hidden layers. The sizes of the hidden layers and regularization factors have been determined by the BayesianOptimization method from the KerasTuner package. The four hidden layers consisted of 64, 256, 256 and 32 neurons respectively with a 'Swich' activation function. The 'Swish' activation function is a new activation function similar to ReLU but with a smoothly varying shape that may lead to better performance. It is defined as the product of the input and the sigmoid function of the input and has been shown to improve the performance of deep neural networks on various tasks [16]. We used the Adam optimizer and a negative correlation loss function to train the model. The learning rate started at 0.01 and was decreased by a factor of 0.5 if the validation loss did not improve for 4 epochs. This was done to ensure the model doesn't get stuck in a local minima and can continue to optimize. An early stopping condition stopped the training process if the validation loss did not improve for 12 epochs in a row, thereby preventing overfitting. Finally, a TerminateOnNaN() callback ensured that training was terminated if any NaN value was encountered in the model's output during training.

For the preprocessing steps, we first used the scikit-learn library to standardize the data before applying PCA. Then, we used the TruncatedSVD method from the sklearn.decomposition package and converted data to sparse matrices using the scipy.sparse.csr_matrix method. We also deleted constant features that were always equal to zero and stored the 'important features' apart before running dimensionality reduction, as described in the previous section.

We compared the performance of the linear regression and LightGBM models with the two different preprocessing techniques.

4. Results

A. Multiome Results

A summary of the regression Multiome results is presented in Table I, showing the performance of each model on the test data based on correlation and MSE scores. The two leftmost columns of Table I show the results for training data, whereas the two rightmost columns show the results for testing data.

Table I
Multiome Performance per Model

	MSE (Train)	Pearson (Train)	MSE (Test)	Pearson (Test)
Linear Regression	2.019	0.676	2.071	0.643
ElasticNet	2.044	0.672	2.117	0.635
Ridge Regression	2.019	0.676	2.072	0.643
Lasso	2.020	0.676	2.075	0.642
BayesianRidge	2.019	0.676	2.071	0.643
ARDRegression	2.019	0.676	2.071	0.643
Maximum	1.925	0.689	1.925	0.666

B. CITE-seq Results

A summary of the CITE-seq results is presented in Table II, showing the performance of each model on the test data based on correlation and MSE scores. The training dataset is composed of data from days 2 and 3 and the test dataset includes data from day 4 only.
We can see that the incorporation of a gene importance highlighting preprocessing step led to a significant improvement in the performance of the model (Table II).

Table II
CITE-seq Multi-output Linear Regression performance with default preprocessing + PCA and added preprocessing + SVD

	Lin. Reg. PCA	Lin. Reg. SVD & prep.
Correlation	0.74	0.88
MSE	6.88	2.74

Again, we can see the value of the added preprocessing. Comparing LightGBM and linear regression, the LightGBM model performs slightly better especially regarding the accuracy of the model evaluated by the MSE (Table III).

Table III
LightGBM performance with default preprocessing + PCA and added preprocessing + SVD

	LightGBM PCA	LightGBM SVD & prep.
Correlation	0.76	0.89

MSE	6.21	2.70

Although the neural network outperformed the LightGBM model slightly with a higher correlation score of 0.90, the limited improvement in overall performance can be attributed to the nature of the multi-output regression task, which may not fully benefit from the additional complexity and flexibility of the neural network architecture (Table IV).

Table IV

Neural network performance with bayesian optimization for hyperparameter-tuning, added preprocessing + SVD

	Neural Net Bayesian Optimisation SVD & prep.
Correlation	0.90

5. Discussion

The results presented in this study demonstrate the successful prediction of gene expression using chromatin accessibility data and successful prediction of protein levels from gene expression data. The use of a dataset that contains overlapping cells for both Multiome and CITE-seq makes the current results further relevant because the combination of both data for the same cells provides several advantages for investigating cellular function and disease

However, there are some limitations and future steps that must be taken into account. In the Multiome data analysis step, the usage of discrete cell types as features may be a potential limitation, since literature has shown that cell types are actually a continuum [17]. A potential future step that may tackle this issue is the re-evaluation of the current methods using this continuum rather than the current discrete categories as a feature. Another important future step would be to pre-process the Multiome data in a fashion more similar to the approach used in the CITE-seq data. In specific, it may be possible to use SVD only on the Multiome chromatin accessibility data that has not been traced back to a specific gene expression [12].

Even though it is crucial to identify and tackle the downsides of this research, overall the current work has a positive outcome. The integration of the Multiomics datasets allows for a more comprehensive and holistic understanding of cellular function. By analyzing multiple omics layers simultaneously, researchers can identify molecular pathways and regulatory mechanisms that would be difficult to identify using a single omics layer alone. Thus, the combination of CITE-seq and transcriptomic data allows for the identification of protein expression patterns in individual cells, which can be used to validate transcriptomic findings or provide additional insights into cellular function. CITE-seq uses oligonucleotide-labeled antibodies to simultaneously measure gene expression and protein levels in single cells, providing a powerful tool for investigating the relationship between gene expression and protein synthesis. In other words, the integration of Multiome and CITE-seq data can provide a more accurate and detailed understanding of cellular heterogeneity. By analyzing multiple molecular features simultaneously, it is possible to identify subpopulations of cells with distinct molecular profiles, which may have important implications for disease diagnosis, prognosis, and treatment.

In the current work, not only do we use both Multiome and CITE-seq data, but we also leverage this data availability to demonstrate each dataset's predictive potential.

The ability to predict gene expression from chromatin accessibility data has important implications for understanding the regulation of gene expression. The chromatin accessibility data provides information on the structure of DNA and the ability of regulatory proteins to access specific regions, which can influence gene expression. The current successful prediction of gene expression from chromatin accessibility data suggests that these structural features play a critical role in regulating gene expression.

Furthermore, the ability to predict protein levels from gene expression data is an important step towards understanding the complex relationship between gene expression and protein synthesis. This is particularly relevant as gene expression is often used as a proxy for protein expression, despite the fact that many factors can influence protein levels, including post-transcriptional modifications, degradation, and protein-protein interactions. The ability to predict protein levels from gene expression data provides a more accurate and comprehensive understanding of gene regulation and protein expression.

Overall, the findings presented in this study highlight the potential for integrative analysis of Multiomics datasets to reveal novel insights into the regulation of gene expression and protein synthesis. The ability to predict gene expression and protein levels from chromatin accessibility and gene expression data, respectively, provides a powerful tool for investigating the complex relationships between these molecular features and their impact on cellular function and disease.

6. Contributions

Both authors contributed equally.

The contributions of our team members to this study were critical in achieving the results presented in this paper.

Zachary Abessera, with his Computer Science and Data Science degree, played a key role in data analysis and building models for the CITE-seq dataset. He devoted significant effort towards understanding the intricacies of the dataset and developing the necessary pre-processing steps to ensure that our models were robust and reliable. Zachary's contributions were instrumental in advancing our understanding of the CITE-seq dataset and its potential applications.

Quentin Chappat, also with a Computer Science and Data Science degree, made important contributions to this study by focusing on building models for the Multiome dataset. Additionally, he devoted significant effort towards optimizing the Random Access Memory usage to run our code efficiently and with our technical constraints, ensuring that we could process the data in a timely manner. Quentin's contributions were instrumental in advancing our understanding of the Multiome dataset and developing new approaches to analyze and interpret genomic data.

7. References

[1] Cameron, D., Bashor, C. & Collins, J. “A brief history of synthetic biology”. *Nat Rev Microbiol* 12, 381–390 (2014). https://doi.org/10.1038/nrmicro3239

[2] Zhu, C., Preissl, S. & Ren, B. Single-cell multimodal omics: the power of many. Nat Methods 17, 11–14 (2020). https://doi.org/10.1038/s41592-019-0691-5

[3] Tung, PY., Blischak, J., Hsiao, C. et al. Batch effects and the effective design of single-cell gene expression studies. Sci Rep 7, 39921 (2017). https://doi.org/10.1038/srep39921

[4] Van den Berge, K., Roux de Bézieux, H., Street, K. et al. “Trajectory-based differential expression analysis for single-cell sequencing data.” *Nat Commun* 11, 1201 (2020). https://doi.org/10.1038/s41467-020-14766-3

[5] Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. “A comparison of single-cell trajectory inference methods.” *Nat. Biotechnol.* 37, 547–554 (2019).

[6] Cannoodt, R., Saelens, W. & Saeys, Y. Computational methods for trajectory inference from single-cell transcriptomics. Eur. J. Immunol. 46, 2496–2506 (2016).

[7] *“Open problems in single cell analysis”*, website: https://openproblems.bio/

[8] *“10X Genomics - Chromium Single Cell Multiome ATAC + Gene Expression”*, website: https://www.10xgenomics.com/products/single-cell-multiome-atac-plus-gene-expression

[9] “ 10X Genomics - Single Cell Gene Expression with Feature Barcoding technology”, website: https://support.10xgenomics.com/permalink/getting-started-single-cell-gene-expression-with-feature-barcoding-technology

[10] Pedregosa, Fabian & Varoquaux, Gael & Gramfort, Alexandre & Michel, Vincent & Thirion, Bertrand & Grisel, Olivier & Blondel, Mathieu & Prettenhofer, Peter & Weiss, Ron & Dubourg, Vincent & Vanderplas, Jake & Passos, Alexandre & Cournapeau, David & Brucher, Matthieu & Perrot, Matthieu & Duchesnay, Edouard & Louppe, Gilles. (2012). “Scikit-learn: Machine Learning in Python”. *Journal of Machine Learning Research*. 12.

[11] Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

[12] Schmidt, F., Kern, F. & Schulz, M.H. Integrative prediction of gene expression with chromatin accessibility and conformation data. Epigenetics & Chromatin 13, 4 (2020). https://doi.org/10.1186/s13072-020-0327-0

[13] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. *Nature Methods*, 17(3), 261-272.

[14] Velten, L., Haas, S., Raffel, S. et al. Human haematopoietic stem cell lineage commitment is a continuous process. Nat Cell Biol 19, 271–281 (2017). https://doi.org/10.1038/ncb3493

[15] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu; “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”

[16] Prajit Ramachandran, Barret Zoph, Quoc V. Le; 10/16/17; "Searching for Activation Functions" *Neural and Evolutionary Computing (cs.NE); Computer Vision and Pattern Recognition (cs.CV); Machine Learning (cs.LG)*: arXiv:1710.05941

[17] Becker, W.R., Nevins, S.A., Chen, D.C. et al. Single-cell analyses define a continuum of cell state and composition changes in the malignant transformation of polyps to colorectal cancer. Nat Genet 54, 985–995 (2022). https://doi.org/10.1038/s41588-022-01088-x