Multi-Target Integration and Annotation of Single-Cell RNA-Sequencing Data

Sapan Bhandari, Nathan P. Whitener, Konghao Zhao and Natalia Khuri
Wake Forest University
Winston-Salem, North Carolina, USA
natalia.khuri@wfu.edu

ABSTRACT

Cells are the building blocks of human tissues and organs, and the distributions of different cell-types change due to environmental or disease conditions and treatments. Single-cell RNA sequencing is used to study heterogeneity of cells in biological samples. To date, computational approaches aided in the discovery of dominant and rare cell-types and facilitated the construction of cell atlases. Integration of new data with the existing reference atlases is an emerging computational problem, and this paper proposes to frame it as a multi-target prediction task, solvable using supervised machine learning. We systematically and rigorously test 63 different predictors on synthetic benchmarks with different properties. The best performing predictor has high Cohen's Kappa scores and low mean absolute errors in single-batch and multi-batch integration experiments.

CCS CONCEPTS

• **Applied computing** → **Bioinformatics**.

KEYWORDS

data integration, classification, regression, single-cell sequencing

ACM Reference Format:
Sapan Bhandari, Nathan P. Whitener, Konghao Zhao and Natalia Khuri. 2022. Multi-Target Integration and Annotation of Single-Cell RNA-Sequencing Data. In *13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '22), August 7–10, 2022, Northbrook, IL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3535508.3545511

1 INTRODUCTION

Cells are the building blocks of all multi-cellular organisms, including humans. Each cell can be characterized on the basis of its spatial, structural, functional, molecular or malignant properties. RNA-sequencing (RNA-seq) is a high-throughput experimental approach that measures the quantities and determines the sequences of active gene transcripts in a sample [17]. There are two major types of RNA-seq experiments, namely, bulk RNA-seq and single-cell RNA-seq (scRNA-seq). Bulk RNA-seq experiments study average gene expression patterns of an entire population of cells in a sample, while scRNA-seq experiments study gene expression in each cell individually [2].

To date, scRNA-seq has been performed in samples derived from different organisms, tissues, and organs [5, 9, 13]. A great number of clinically-relevant scRNA-seq datasets have been also released, including scRNA-seq data of patients with COVID-19 disease, cancer, and many others. These data are large, high-dimensional and zero-inflated due to the detection limits of scRNA-seq instruments. Therefore, to derive biological and clinical insights from these data, machine learning (ML) has been used to study differential gene expression of individual cells, and to discover dominant and rare cell-types that correlate with different conditions, diseases and treatments.

Computational methods for scRNA-seq data analysis are abundant and rapidly evolving. Nevertheless, unsupervised methods, such as cluster analysis, dominate the field. Clustering methods partition scRNA-seq data into distinct subsets with the purpose of the discovery of cells of similar types and states. Ideally, each cluster should contain cells with similar expression profiles. To discern the identity of each cell-type in a cluster, marker genes must be found that are highly expressed in that cluster compared to all of the other clusters. These gene markers are then used to assign the cell-type for the entire cluster. The challenges and limitations of cluster-based approaches are well known [8].

First, each clustering algorithm makes specific assumptions about the distribution of data and comes with user-defined parameters, such as the number of clusters or the number of nearest neighbors. Because true data distribution is unknown, cluster-based techniques may generate partitions that are not pure, in which one cluster contains multiple cell-types or in which one cell-type could be split into multiple clusters. Moreover, cell-types and cell-states may be similar to each other, which presents a challenge to separate them into distinct groups. Second, the use of gene markers to assign cell-types requires expert knowledge of these specific markers, and sometimes they are not well characterized or they may be difficult to find in the literature. Some markers may be also expressed by more than one cell-type, and some cell-types may have no known markers.

Robust differential gene expression analysis of clusters requires sufficient numbers of cells to obtain statistically meaningful results. Therefore, it is common to increase the power of such analyses by combining multiple datasets into one and performing cell-type detection and differential gene expression analyses on the integrated data. Integration is a process that combines multiple datasets into

BCB '22, August 7–10, 2022, Northbrook, IL, USA

ACM ISBN 978-1-4503-9386-7/22/08…$15.00
https://doi.org/10.1145/3535508.3545511

one and derives low-dimensional representation of the aggregated data. Reference-free or unsupervised data integration is prone to its own set of challenges and limitations [8]. For example, technical batch effects may bias the results because cells tend to cluster based on the specific experimental conditions rather than their types [6]. Therefore, specialized integration tools have been developed [7, 10].

The number of integrated and annotated scRNA-seq datasets has been increasing. Known as cell atlases, they can serve as references for the integration and annotation of newly sequenced scRNA-seq data. Reference-based integration can reduce computational and labor costs of unsupervised methods. Integration of new data with an existing cell atlas can be framed as a supervised ML problem, where outcomes for new data are predicted using a mapping function learned from past data. Prediction of categorical outcomes or labels, such as cell-types, is known as the classification task, while the prediction of numeric outcomes is considered a regression task, respectively. Therefore, unlike reference-free integration, where the low-dimensional representations and cell-type assignments are discovered without prior knowledge, the aim of reference-based data integration is to accurately embed new high-dimensional data onto an existing low-dimensional representation and annotate them by predicting categorical labels of new cells.

In this work, we make three main contributions. Our first contribution is methodological. We formulate the reference-based scRNA-seq data integration problem as the multi-target prediction problem. We design and implement 63 different predictors to solve the three-target problem, and test them systematically and rigorously on synthetic datasets with different properties. Second, we demonstrate empirically that the feed-forward neural network (FFNN), coupled with the Ridge regression, can accurately reconstruct two-dimensional representations of synthetic datasets and predict cell-types. We show that our three-target predictor outperforms baseline distance-based methods implemented in the existing tools and it requires significantly less training and inference times. Finally, our third contribution highlights broader considerations of practical reference-based integration of scRNA-seq data, such as the required minimum number of cells and batches.

2 METHODS AND DATA

Given a matrix M of size $n \times d$, where n is the number of cells, d is the number of genes, and m_{ij} is the count of active transcripts of gene j in cell i. Assume that each cell has a unique cell-type drawn from $C = \{c_1, c_2, \ldots, c_k\}$, such that $k \geq 2$. Also, assume that each cell can be embedded onto a p-dimensional space, and its embedding is $x = (x_1, x_2, \ldots, x_p)$, where $p \geq 1$.

The prediction task is to automatically predict multiple targets, namely embedding values, x_i, and a cell-type label, c_i, for each cell in a given count matrix M. Recall, that values of x are numeric and values of c are categorical. In this work, we focus on a three-target prediction problem, which can be easily extended to a greater number of targets. In this work, we focus on a three-target prediction problem, which can be easily extended to a greater number of targets.

The input to the predictor is a count matrix of gene expression values and the outputs are cell's embeddings, $x = (x_1, x_2)$, and a cell-type label, c_i. In this work, we choose to use the Uniform Manifold Approximation and Projection (UMAP) to compute cell's embeddings, which is a flexible and fast non-linear dimensionality reduction method [12] and a standard tool in scRNA-seq workflows and visualizations.

2.1 Three-Target Predictor

We solved the problem by constructing a model for every target independently and combining them into the multi-target prediction [16]. Our best performing predictor was a hybrid, comprising Ridge regression for the prediction of x values and a FFNN for the prediction of cell-types, respectively.

Ridge regression is a method for fitting multiple-regression models. Ridge regression minimizes the sum of the error term along with sum of squares of coefficients, which is called the regularization term. In our implementation, the regularization was given by the L2-norm.

To predict cell-types, we used a FFNN, comprising 3 hidden layers. The first hidden layer had 256 hidden nodes, the second and third hidden layers had 128 and 1 hidden nodes, respectively. The nodes of the input layer accept raw expression values of M. All nodes in the input layer were connected to all nodes in the first hidden layer, and all nodes in the last hidden layer were connected to all nodes in the output layer. Likewise, all hidden layers were interconnected, forming a dense network. We added a dropout layer between the first and the second layer, with the dropout rate of 0.6. We used a rectified linear unit (ReLU) activation function in the hidden layers, and an L2 kernel regularizer with the regularization factor of 0.01. The number of nodes in the output layer was set to the number of cell-types, and a softmax activation function was used to output a score for each cell-type. These scores were converted into categorical cell-type labels using an argmax function. We used RMSprop optimizer with a learning rate of 1E-4, and a categorical cross-entropy as a loss function, respectively. Fifty epochs were used for training, and batch sizes were set to 128.

2.2 Alternative Multi-Target Predictors

We considered several alternative multi-target predictors. These alternative methods were inspired by some of the prior works on cell-type prediction in scRNA-seq and bulk RNA-seq. As an alternative to Ridge regression, we considered k-Nearest Neighbors (kNN) and extreme gradient boosted tree (XGB) regressions for the prediction of x targets, and for the prediction of cell-types, we considered six neural networks and two classifiers, kNN and XGB classifiers. Therefore, in total, we considered 63 (3×3×7) alternative combinations of predictors for the three-target prediction problem. In lieu of the available code in prior works, we either adopted published parameters as defaults, or tuned hyper-parameters via grid search.

In kNN, the similarity was measured using the Euclidean distance, and we set the number of neighbors to 15, which was determined by experimentation.

XGB algorithm builds its model in a greedy fashion; trees that provide the most gain to the model are chosen first [3]. We set the number of iterations to 50. We used a multi-class logarithmic loss function to predict cell-types. The output was a vector of scores for all unique cell-types, which were converted into categorical

cell-type labels using an argmax function. In XGB regression, L2 regularization was used.

In addition, we evaluated convolutional neural networks (CNN), autoencoders (AE) and residual networks. CNN and AE models were implemented with 2 different input dimensions. In the one-dimensional (1D) encoding, networks accepted normalized and scaled gene expression values. In the two-dimensional (2D) encoding, we converted 1D expression vectors into matrices using an approach developed for bulk RNA-seq data [11, 14]. Specifically, genes were ordered based on their chromosomal locations, which placed genes with similar expression values near each other. Then, each expression vector was reshaped into 102×102 matrices by padding with zeros at the last row, if needed.

3 DATA SETS

Twelve simulated benchmarks were created using the Splatter package (version 1.18.1) [18]. To simulate benchmarks SIM1 to SIM5, we created single batches of 100,000 cells, with the number of genes set to 720. Each benchmark had a different number of cell-types, ranging from 4 (SIM1) to 64 (SIM5). To simulate benchmarks SIM6 to SIM9, we set the total number of cells to 50,000 and the number of genes to 720. We varied the number of batches from 5 to 25, keeping the number of cell-types in each batch at 2. These benchmarks were representative of the scRNA-seq experiments, where samples were collected at different time-points, in different laboratories or using different sequencing platforms, for example. For a more realistic integration, we created benchmarks SIM10 to SIM12, each comprising 5 batches. The number of cell-types in each benchmark differed, ranging from 4 (SIM10) to 16 (SIM12). Moreover, we simulated 4,000 cells of each cell-type and hence, the total number of cells in these benchmarks ranged from 16,000 (SIM10) to 64,000 (SIM12).

In preprocessing, we followed the standard workflow implemented in the Seurat package (version 4.0.6) [15], followed by the harmonization using Harmony package and dimensionality reduction using UMAP.

3.1 Computing Resources

All models were implemented using the Tensorflow library [1] and Keras Application Programming Interface [4]. The code was written in the Python programming language, and all experiments were performed on a High Performance Computing cluster.

4 RESULTS

We considered 63 different predictors and tested them in 5-fold cross-validation. To evaluate the predictions of target x, we used mean absolute error (MAE) as the criterion of performance, with smaller values indicating better performance. To evaluate cell-type predictions, we focused on Cohen's Kappa, which ranges between -1 and 1. Higher values imply better performance.

The training and inference times of cell-type predictors varied significantly, with ResNet and XGB being the slowest. The average training time of ResNet was between 63 to 76,583 seconds (sec) and testing time was between 0 to 165 sec in all benchmarks. XGB's training time ranged between 6 to 11,922 sec and testing time between 0 to 23 sec, respectively. Interestingly, 2D classifiers CNN2D and AE2D were fast, requiring only 0 to 7 sec of testing time per benchmark. Training times of AE1D and AE2D were between 0 to 442 sec and between 1 to 255 sec, respectively. Likewise, CNN2D needed between 2 to 542 sec to train and CNN1D required between 3 to 1092 sec of training time. Finally, the average training time of FFNN was between 3 to 402 sec and testing time was between 0 to 20 sec for all benchmarks. We note that all time estimates were collected from training and testing of models on exactly the same folds, to avoid biasing them. Also, as expected, greater training and test times were observed for benchmarks with larger number of cells. Finally, ResNet, XGB, AE1D and AE2D had larger memory consumption than the other methods.

All 1D models and XGB had better results than the 2D models. FFNN was the best performing neural network and the second best performing classifiers were trained using CNN1D. AE1D, ResNet and XGB failed to train good models for some of the datasets.

With respect to the prediction of target x, Ridge regression outperformed kNN and XGB regression models and had significantly faster training and inference times. Therefore, based on the three criteria, such as performance, compute time and memory requirements, we finalized our three-target predictor as a hybrid, comprising two Ridge regression predictors for UMAP coordinates and a FFNN classifier for the cell-type prediction.

To better understand predictor's sensitivity to the training size, in particular to the number of cells per cell-type, we simulated 5 benchmarks (SIM1 to SIM5) with the number of cell-types ranging from 2 to 64. By keeping the total number of cells in each benchmark constant, we studied how the results changed when the number of cells per cell-type was reduced from 25,000 to about 1,500. We observed a steady increase in the absolute prediction errors of the two values of x (Table 1). Moreover, we saw a clear trend of decreased performance in cell-type predictions. Kappa scores dropped from 1.0 to 0.68 when models were trained with fewer samples per each cell-type. Therefore, we conclude that at least 6,500 samples per cell-type were needed in single-batch integration and greater training sample sizes were needed for a more accurate prediction of embeddings.

Table 1: Performance of three-target predictor on simulated benchmarks. Shown are benchmark ID, number of batches, cells, and cell-types. The number of genes was 720 in all benchmarks. Predicted performance as measured by Cohen's Kappa and two MAE scores (MAE-1 and MAE-2).

ID	Batches	Cells	Types	Kappa	MAE-1	MAE-2
SIM1	1	100,000	4	1.00	1.11	1.37
SIM2	1	100,000	8	0.99	0.97	1.97
SIM3	1	100,000	16	0.97	1.86	2.94
SIM4	1	100,000	32	0.84	1.71	1.54
SIM5	1	100,000	64	0.68	1.92	0.99
SIM6	5	50,000	2	1.00	1.78	5.95
SIM7	10	50,000	2	1.00	3.29	5.15
SIM8	20	50,000	2	0.99	4.29	6.50
SIM9	25	50,000	2	0.99	4.39	5.75
SIM10	5	16,000	4	1.00	3.62	9.06
SIM11	5	32,000	8	0.97	7.78	7.93
SIM12	5	64,000	16	0.82	2.48	8.64

We tested the performance of the proposed predictor in the integration of individual batches. Using SIM7 to SIM12, we repeatedly trained models on all batches, except one, and predicted targets for the withheld batch. In these experiments, we observed higher prediction errors than in the experiments with the single-batch integration (Table 1). Average MAE values for the first target increased from 1.51 (SIM1-SIM5) to 3.98 (SIM6-SIM13). Interestingly, average MAE values for the second target were higher and they also increased from 1.76 to 6.71 when we moved from single-batch to multi-batch integration. We note that both, the number of batches and the number of cell-types within each batch impacted the accuracy of the predictions.

Therefore, we confirmed our findings from the single-batch experiments, about the sensitivity to the number of cells of different cell-types, available for training. For instance, Kappa scores steadily decreased from 1 to 0.82 in the integration of 5 batches comprising 4 to 16 cell-types. Alongside, MAE values were higher than MAE values of SIM6 to SIM9, where the batches comprised only 2 cell-types (Table 1).

Having single and multi-batch simulated benchmarks allowed us to compare systematically the proposed method with the baseline kNN predictors, which are commonly used in reference-free integration. Overall, FFNN and Ridge regression outperformed baseline kNN methods. Interestingly, however, we noticed sensitivity to batch effects. For example, while the FFNN predictor had slightly lower NMI scores compared with the kNN predictor on SIM1-SIM5 benchmarks, it had significantly higher NMI scores in multi-batch benchmarks SIM6-SIM12. On average, NMI scores of SIM1-SIM5 were 0.91 for kNN and 0.88 for FNN, while NMI of multi-batch benchmarks were 0.85 for kNN and 0.96 for FFNN, respectively.

Similarly, Ridge regression lagged slightly behind kNN and XGB methods in single-batch integration on SIM1 to SIM5 benchmarks, yet it significantly outperformed the other 2 methods in the multi-batch integration of the remaining benchmarks. Overall, MAE scores of Ridge regression were 3.03 and 4.81 for UMAP-1 and UMAP-2, respectively. These values were 3.81 and 4.98 for kNN predictions and 2.84 and 4.86 for XGB, respectively.

Based on these results, we recommend to use FFNN and Ridge regression over kNN-based methods in reference-based integration. Predictors trained with these methods perform better in multi-batch settings and they are also significantly faster than their kNN baselines.

5 CONCLUSION

Advances in the next-generation sequencing technologies made it possible to examine gene expression at a single-cell resolution. As the result, large high-dimensional datasets became available for computational studies into the heterogeneity of cells' populations. This work addressed two grand computational challenges of scRNA-seq data science, namely reference-based data integration and systematic comparison of computational approaches, methods and tools. We formulated the problem of reference-based data integration and annotation as a supervised multi-target prediction task. Rigorous computational testing on synthetic datasets showed that a hybrid predictor, comprising feed-forward neural network and Ridge regression, outperformed alternative predictors and the baseline k-Nearest Neighbors algorithms. In addition, this work provided practical insights about the utility of the proposed approach and its sensitivity to properties of the datasets. Future work will focus on the parallelization of the proposed approach.

ACKNOWLEDGMENTS

This research was partially supported by the Pilot Award from the Wake Forest Center for Biomedical Informatics. The authors thank Stanley Thomas and Victor Paul Pauca for helpful comments and Cody Stevens for technical assistance. The authors acknowledge the Distributed Environment for Academic Computing (DEAC) at Wake Forest University for providing HPC resources that have contributed to the research results reported within this paper.

REFERENCES

[1] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/
[2] Geng Chen, Baitang Ning, and Tieliu Shi. 2019. Single-cell RNA-seq technologies and related computational data analysis. *Frontiers in genetics* 10 (2019), 317.
[3] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), 785–794.
[4] Francois Chollet et al. 2015. *Keras.* https://github.com/fchollet/keras
[5] Tabula Muris Consortium et al. 2018. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* 562, 7727 (2018), 367–372.
[6] Eva Hedlund and Qiaolin Deng. 2018. Single-cell RNA sequencing: technical advancements and biological applications. *Molecular aspects of medicine* 59 (2018), 36–46.
[7] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. 2019. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature methods* 16, 12 (2019), 1289–1296.
[8] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J McCarthy, Stephanie C Hicks, Mark D Robinson, Catalina A Vallejos, Kieran R Campbell, Niko Beerenwinkel, Ahmed Mahfouz, et al. 2020. Eleven grand challenges in single-cell data science. *Genome biology* 21, 1 (2020), 1–35.
[9] Rik GH Lindeboom, Aviv Regev, and Sarah A Teichmann. 2021. Towards a Human Cell Atlas: Taking Notes from the Past. *Trends in Genetics* (2021), 625–630.
[10] Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. 2022. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods* 19, 1 (2022), 41–50.
[11] Boyu Lyu and Anamul Haque. 2018. Deep learning based tumor type classification using gene expression data. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*. 89–96.
[12] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* 3, 29 (2018), 861.
[13] Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, et al. 2017. Science forum: the human cell atlas. *elife* 6 (2017), e27041.
[14] Alok Sharma, Edwin Vans, Daichi Shigemizu, Keith A Boroevich, and Tatsuhiko Tsunoda. 2019. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Scientific reports* 9, 1 (2019), 1–7.
[15] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. 2019. Comprehensive Integration of Single-Cell Data. *Cell* 177 (2019), 1888–1902. https://doi.org/10.1016/j.cell.2019.05.031
[16] Willem Waegeman, Krzysztof Dembczyński, and Eyke Hüllermeier. 2019. Multi-target prediction: a unifying view on problems and methods. *Data Mining and Knowledge Discovery* 33, 2 (2019), 293–324.
[17] Zhong Wang, Mark Gerstein, and Michael Snyder. 2009. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics* 10, 1 (2009), 57–63.
[18] Luke Zappia, Belinda Phipson, and Alicia Oshlack. 2017. Splatter: simulation of single-cell RNA sequencing data. *Genome biology* 18, 1 (2017), 1–15.