Prediction of functional markers of mass cytometry data via deep learning

Claudia Solís-Lemus, Xin Ma, Maxwell Hostetter II, Suprateek Kundu, Peng Qiu, Daniel Pimentel-Alarcón

Abstract Recently, there has been an increasing interest in the analysis of flow cytometry data, which involves measurements of a set of surface and functional markers across hundreds and thousands of cells. These measurements can often be used to differentiate various cell types and there has been a rapid development of analytic approaches for achieving this. However, in spite of the fact that measurements are available on such a large number of cells, there have been very limited advances in deep learning approaches for the analysis of flow cytometry data. Some preliminary work has focused on using deep learning techniques to classify cell types based on the cell protein measurements. In a first of its' kind study, we propose a novel deep learning architecture for predicting functional markers in the cells given data on surface markers. Such an approach is expected to automate the measurement of functional markers across cell samples, provided data on the surface markers are available, that has important practical advantages. We validate and compare our approach with competing machine learning methods using a real flow cytometry dataset, and showcase the improved prediction performance of the deep learning architecture.

Claudia Solís-Lemus
Emory University, Atlanta, GA, e-mail: csolisl@emory.edu

Xin Ma
Emory University, Atlanta, GA, e-mail: xin.ma@emory.edu

Maxwell Hostetter II
Georgia State University, Atlanta, GA, e-mail: mhostetter1@student.gsu.edu

Suprateek Kundu
Emory University, Atlanta, GA, e-mail: suprateek.kundu@emory.edu

Peng Qiu
Georgia Institute of Technology and Emory University, Atlanta, GA, e-mail: peng.qiu@bme.gatech.edu

Daniel Pimentel-Alarcón
Georgia State University, Atlanta, GA, e-mail: pimentel@gsu.edu

1 Introduction

Multiparametric single-cell analysis has advanced our understanding of diverse biological and pathological processes, providing insights into cellular differentiation, intracellular signaling cascades and clinical immunophenotyping. Modern flow cytometers typically provide simultaneous single-cell measurements of up to 12 fluorescent parameters in routine cases, and analysis of up to 30 protein parameters has been recently made commercially available. In addition, a next-generation mass cytometry platform (CyTOF) has become commercially available, which allows routine measurement of 50 or more protein markers.

Despite the technological advances in acquiring an increasing number of parameters per single cell, approaches for analyzing such complex data lag behind. The existing approaches are often subjective and labor-intensive. For example, the widely used gating approach identifies cell types by user-defined sequences of nested 2-D plots. There have been efforts to develop clustering algorithms (e.g., flowMeans[1], flowSOM[2], X-shift[3], and dimension reduction algorithms (e.g., SPADE [4], tSNE[5], Scaffold[6]. However, there is still huge space for developing new methods to ask new questions in this field.

Recently, deep learning models are revolutionizing the fields of precision medicine, data mining, astronomy, human-computer interactions, among many others, by becoming a major discovery force in science due to the unprecedented accuracy in prediction. Moreover, deep learning approaches have shown accurate performance on genomics and biomedical applications[7, 8, 9, 10, 11, 12, 13].

Furthermore, CyTOF data is perfectly suited for deep learning methods. On one side, identify markers define a cell type (e.g., B cell, T cell, monocytes, MSC), and on the other side, expressions of functional markers identify the cell's activity (e.g., quiescent, secreting cytokines, proliferating, apoptosis). Since CyTOF technology allows for the simultaneous measurement of a large number of protein markers, most CyTOF studies measure both identity markers and functional markers, providing data for supervised learning tools, like neural networks. In addition, each CyTOF run typically collects data on 10^6 cells, creating an ideal large dataset in which the number of samples (cells) is orders of magnitude larger than the number of variables (markers). Deep learning methods are particularly suited for this type of big data.

In terms of motivation, there are two main reasons to predict the functional markers from surface markers in CyTOF data: 1) monetary and time cost, and 2) technical limit of the total number of markers CyTOF can measure, which is currently around 50 protein markers. That is, if we can accurately predict some functional markers based on surface markers, there is no longer the need to include those functional markers in the staining panel (experimental design), and thereby freeing up channels to measure more surface markers or additional functional markers that cannot be predicted.

Here, we explore neural network models to predict functional markers (internal phosphoproteins) with identify markers (cell surface proteins), and compare its performance in terms of accuracy and speed to other standard statistical approaches like regression, and random forests. We show that neural networks improve prediction

of functional markers, making them a powerful alternative to the usual regression techniques.

2 Data

2.1 Pre-processing

The CyTOF dataset has been previously published in [14, 4]. It contains single-cell data for 5 bone marrow samples from healthy donors. The data for each sample contains measurements for 31 protein markers for individual cells, including 13 cell surface markers which are conventionally used to define cell types, as well as 18 functional markers which reflect the signaling activities of cells. The number of cells per sample is roughly 250,000, and the total number of cells across all 5 samples is 1,223,228. Thus, the data can be expressed in a 1223228×31 matrix.

The data was transformed with inverse hyperbolic sine function (*arcsinh* with co-factor of 5), which is the standard transformation for CyTOF data [14].

2.2 Exploratory analysis

We will compare the performance of different methods (explained next section) to predict the functional markers with the surface markers. The data is highly complex and correlated, violating some of the fundamental assumptions of standard statistical approaches (like regression). For example, the data is highly skewed and the pattern between response and predictor is not linear (see figure 1).

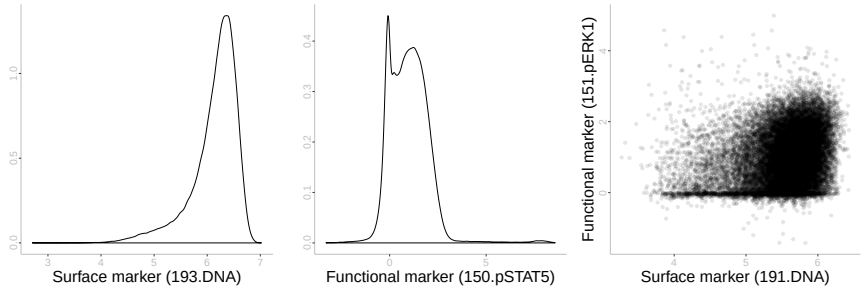

Fig. 1 Exploratory plots of surface and functional markers. The histograms show a biased pattern, and the scatterplot shows non linearity, both violations of crucial assumptions in standard regression models.

3 Materials and Methods

3.1 Background on neural networks

A neural network model is formed by several *neurons*. Each neuron receives an input vector x, then weights its components according to the neuron's weight vector w, adds a *bias* constant b, and passes the result through a non-linear *activation function* σ. This way, the output of a neuron is given by $\sigma(w^T x+b)$. There are several options for the activation function σ. Common choices include the *sigmoid* function $\sigma(z) = \frac{1}{1+e^{-z}}$ or the *rectified linear unit* (ReLU) $\sigma(z) = \max(0,z)$. For the CyTOF data, we use the hyperbolic tangent as activation function, as it showed better performance than the sigmoid or ReLU functions (more details on the specific neural network fit in subsection 3.2).

The final output of the network is given by $\hat{f}(x)$ with parameters $\mathbf{W}_1, \cdots, \mathbf{W}_L$ for the weight matrices and $b_1, \cdots, b_L$ for the bias vectors for each layer.

The estimation of the parameters is done through the following optimization

$$\min_{\{W_l, b_l\}_{l=1}^{L}} \sum_{i=1}^{n} \|y_i - \hat{f}(x_i)\|^2. \tag{1}$$

The most widely used technique to solve this optimization is through stochastic gradient descent (SGD) and back-propagation, but we discovered that Adam[15], an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments had better performance for our data (more details in subsection 3.2).

3.2 Methods comparison

We fit a neural network model to predict functional markers from surface markers, and compare its performance to three classical statistical methods: 1) linear regression (unpenalized and penalized), 2) decision trees, and 3) random trees. Due to computational time constraints, we could not fit a support vector regression (SVR) model. We compared the performance of the four approaches by computing the mean square error (MSE) of the predicted responses.

To fit the models, we divided the data into training set, validation set and testing set. The training set was used to fit each of the four models. The validation set was used to determine the best setup (tuning parameters) of each model in terms of MSE. Finally, the test set was used to compare the MSE across methods.

The complete data consisted of 1,223,228 cells in 18 functional markers (responses) and 15 surface markers (predictors), which we divided as follows: 750,000 rows as training set, 250,000 rows as validation set, and 223,228 rows as testing set.

We used two separate measure of performance: a vector MSE (equation 2) and individual MSE (equation 3), one per predictor (so, 18 in total).

The vector MSE is defined as

$$MSE_{vec} = \frac{1}{2n}\sum_{i=1}^{n} ||\hat{Y}_i - Y_i||_2^2 \tag{2}$$

where $\hat{Y}_i \in \mathbf{R}^{18}$ is the predicted vector of responses for individual i, and $Y_i \in \mathbf{R}^{18}$ is the observed vector of responses for individual i.

The individual MSE for predictor k is defined as

$$MSE_{(k)} = \frac{1}{2n}\sum_{i=1}^{n} (\hat{Y}_{k,i} - Y_{k,i})^2 \tag{3}$$

where $\hat{Y}_{k,i} \in \mathbf{R}$ is the k^{th} predicted response ($k = 1, \cdots, 18$) for individual i, and $Y_{k,i} \in \mathbf{R}$ is the k^{th} observed response for individual i.

Neural network model: We tested different network architectures, activation functions, regularization coefficients, solver methods, momentum policies, and learning rates with 50,000 maximum epochs. The best network has four layers (see figure 2) with 90, 90, 45 and 45 nodes. The network uses hyperbolic tangent as activation function, regularization coefficient of 0.0001, momentum policy fixed at 0.8, inverse-decay learning rate policy with base learning rate, gamma and power parameters at 0.01, 0.0001, 0.75. We used Adam solver[15], an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments, instead of Stochastic Gradient Descent (SGD) as the former showed increased accuracy. All networks were trained using the julia package `Mocha`[16, 17].

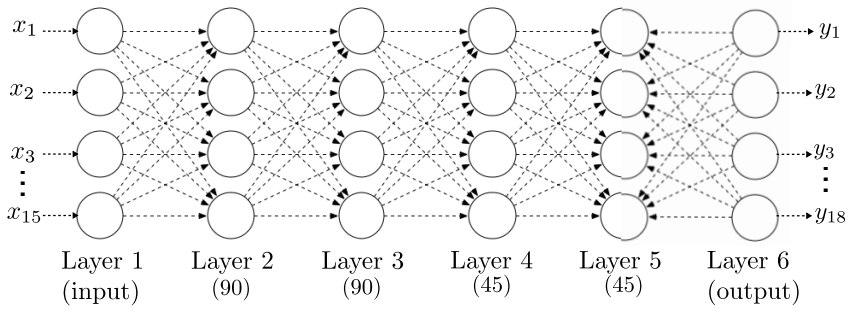

Fig. 2 Neural network for predicting functional markers (18 responses) from surface markers (15 predictors) with 4 hidden layers with 90, 90, 45 and 45 nodes each.

Linear regression (unpenalized/penalized): We fit standard linear regression, as well as the penalized version with LASSO penalty under different penalization pa-

rameters. We used the `ScikitLearn`[18] julia wrapper, with default settings. We noted that the penalized version performed worse than the unpenalized version for all the predictors (regardless of penalty parameter), so we only present results below for the unpenalized linear regression model.

Decision tree and random forest regressions: We fit one decision tree regression per response with `ScikitLearn` julia wrapper, with default settings. We compared the performance of the "mse" criterion and the Friedman's improvement score, deciding on the former ("mse") which is the default setting. We did not constraint the maximum depth of the tree, and set as 2 the minimum number of samples required to split an internal node. In addition, we did not constraint the maximum number of features to consider when looking for the best split. Later, we fit 20 trees into a random forest regression. We could not explore more than 20 trees due to computational time constraints.

4 Results

Figure 3 (left) shows the vector MSE (equation 2) across all four different methods, being decision tree the least accurate and neural network the most accurate. Figure 3 (right) shows a comparison on computation time (in seconds) among the four methods, being linear regression the fastest and random forest the slowest. To sum up, the neural network approach outperforms the other three methods in terms of prediction accuracy, without sacrificing too much computational speed.

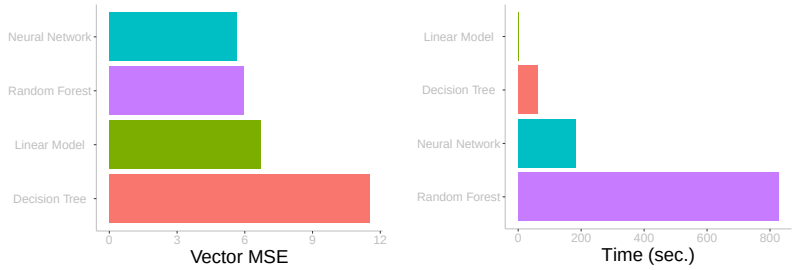

Fig. 3 Left: Vector MSE (equation 2) for all four methods sorted from most accurate (neural network) to least accurate (decision tree). Right: Running time (in seconds) for the training and validation sets (sample 750,000 rows) for all four methods, sorted from fastest (linear regression) to slowest (random forest).

Figure 4 shows the individual MSE (equation 3) per response (18 responses in x-axis) for each of the four methods. Again, the prediction accuracy of neural network is better than the other three methods for all the 18 predictors.

The MSE performance varies across responses. For example, the first response (functional marker 141.pPLCgamma2) has an overall MSE lower than other re-

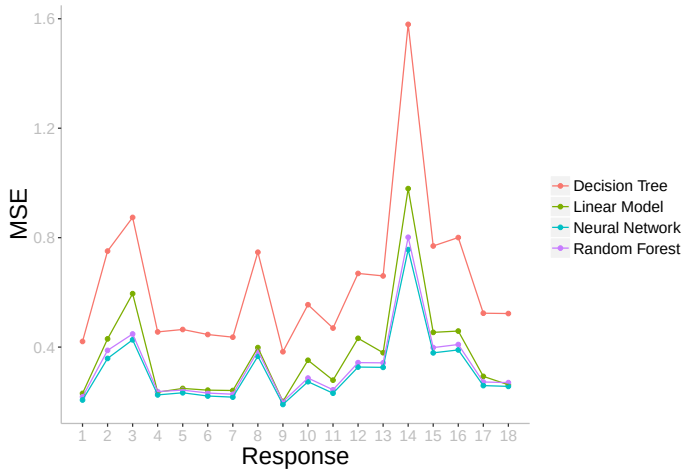

Fig. 4 Individual MSE (equation 3) for each 18 responses. The neural network outperforms all other methods across all responses. Lines are drawn simply for visual effect.

sponses like the third (functional marker 152.Ki67), the 8th (functional marker 159.pSTAT3) or the 14th (functional marker 171.pBtk.Itk).

Figure 5 (left) shows the violin plots for these 4 functional markers. We observe that the 14th response has a wider range and heavier tails than the other responses, which is confirmed in the scatterplots on the center and right (figure 5). It appears that the wider spread and higher variability of the 14th response (functional marker 171.pBtk.Itk) causes the lower prediction accuracy compared to other responses, like the first one (functional marker 141.pPLCgamma2).

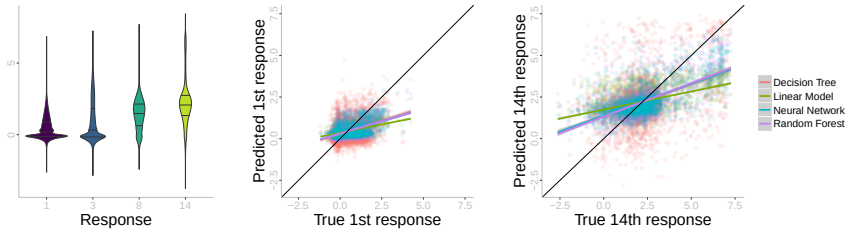

Fig. 5 Left: Violin plot for four functional markers (responses). Horizontal lines represent the 25th quantile, median and 75th quantile. Center: Predicted vs observed responses on the first functional marker (141.pPLCgamma2) across all four methods. Left: Predicted vs observed responses on the 14th functional marker (171.pBtk.Itk) across all four methods. The closer the slope to 1 (black line), the better.

Finally, we present selected scatterplots of surface markers as predictors for the responses 1,3,8 and 14 (figure 6). We can appreciate in these plots the non-linear

relationship between the predictors and responses, which justifies the use of a neural network approach.

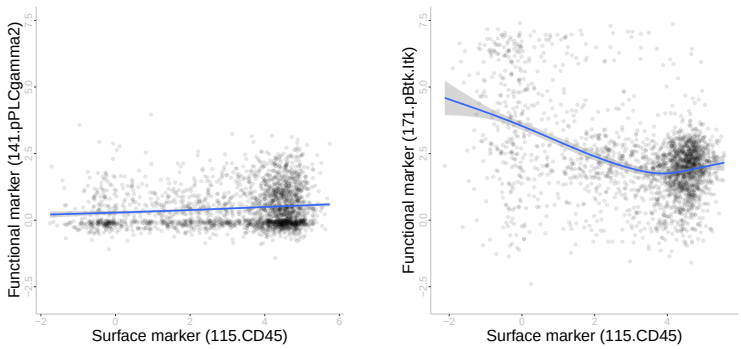

Fig. 6 Scatterplot of selected surface markers (predictors) and selected functional markers (responses). Left: the first response (functional marker 141.pPLCgamma2) shows a linear relationship to the predictor (surface marker 115.CD45), which partially explains the better MSE in figure 4. Right: the 14th response (functional marker 171.pBtk.Itk) shows a non-linear relationship to the predictor (surface marker 115.CD45), which partially explains the worse MSE in figure 4.

5 Discussion

In this work, we showed that a neural network model outperforms standard statistical approaches like linear regression and random forest in the prediction of functional markers from surface markers for CyTOF data. Neural networks were also faster and more efficient than random forests, which make them a more viable choice for big datasets.

The improved prediction accuracy of neural networks can be explained by their flexibility to account for non-linearity or skewness. Unlike regression models, neural networks do not have linearity or normality assumptions, and they take advantage of the correlation structure among responses by fitting a network for the whole response vector.

As mentioned already, CyTOF data is perfectly suited for deep learning methods given the simultaneous measurement of a large number of protein markers, including both identity markers and functional markers. Both measurements allow for the implementation of highly accurate supervised methods, like neural networks. In addition, the structure of CyTOF data is ideal for deep learning: number of samples orders of magnitude greater than the number of variables.

The accuracy in the prediction of functional markers from surface markers has economic and computational advantages, for example, considering the limitation to the total number of markers CyTOF can measure, which is currently around 50 pro-

tein markers. Being able to predict functional markers from surface markers could allow for different types of staining panels which could measure more surface markers, or focus on functional markers not so easily predicted.

For future work, we can include an extended version of the dataset[14, 4] that includes 24 healthy sample of bone marrow treated by 24 different drugs. In this setting, we are interested in predicting the functional markers under different drug scenarios, using information at baseline (no treatment) and surface markers at different treatment levels. Furthermore, based on the trained deep learning model, we are interested in the question of whether we can identify cell clusters, and whether these cell clusters agree with well-accepted cell types in literature. Finally, if we focus on cells belonging to the same known cell type, and examine the distribution of functional markers and the correlation with the subtle variations of the identity markers among cells of this type, we can explore whether there is evidence that the specific cell type could be further divided into subtypes.

References

1. N. Aghaeepour, R. Nikolic, H. H. Hoos, and R. R. Brinkman. Rapid cell population identification in flow cytometry data. *Cytometry Part A*, 79A(1):6–13, 2010.
2. S. Van Gassen, B. Callebaut, M. J. Van Helden, B. N. Lambrecht, P. Demeester, T. Dhaene, and Y. Saeys. FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. *Cytometry Part A*, 87(7):636–645, 2015.
3. N. Samusik, Z. Good, M. H. Spitzer, K. L. Davis, and G. P. Nolan. Automated mapping of phenotype space with single-cell data. *Nature Methods*, 13:493, may 2016.
4. P. Qiu, E. F. Simonds, S. C. Bendall, K. D. Gibbs Jr, R. V. Bruggner, M. D. Linderman, K. Sachs, G. P. Nolan, and S. K. Plevritis. Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature Biotechnology*, 29:886, oct 2011.
5. L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
6. M. H. Spitzer, P. F. Gherardini, G. K. Fragiadakis, N. Bhattacharya, R. T. Yuan, A. N. Hotson, R. Finck, Y. Carmi, E. R. Zunder, W. J. Fantl, S. C. Bendall, E. G. Engleman, and G. P. Nolan. An interactive reference framework for modeling a dynamic immune system. *Science*, 349(6244), 2015.
7. D. Cire¸san, A. Giusti, L. Gambardella, and J. Schmidhuber. Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks. In K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab, editors, *Medical Image Computing and Computer-Assisted Intervention*. Springer, Berlin, 2013.
8. O. Denas and J. Taylor. Deep modeling of gene expression regulation in an Erythropoiesis model. In *Representation learning, ICML Workshop*, 2013.
9. R. Fakoor, F. Ladhak, A. Nazi, and M. Huber. Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.
10. M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129, jun 2014.
11. A. Cruz-Roa, A. Basavanhally, F. González, H. Gilmore, M. Feldman, S. Ganesan, N. Shih, J. Tomaszewski, and A. Madabhushi. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In *SPIE Medical Imaging*, volume 9041, pages 904103–904103–15, 03/2014 2014.

12. H. Li, U. Shaham, K. P. Stanton, Y. Yao, R. R. Montgomery, and Y. Kluger. Gating mass cytometry data by deep learning. *Bioinformatics*, 33(21):3423–3430, 2017.
13. P. Mobadersany, S. Yousefi, M. Amgad, D. A. Gutman, J. S. Barnholtz-Sloan, J. E. Velázquez Vega, D. J. Brat, and L. A. D. Cooper. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proceedings of the National Academy of Sciences*, 2018. PMID: 29531073.
14. S. C. Bendall, E. F. Simonds, P. Qiu, E.-a. D. Amir, P. O. Krutzik, R. Finck, R. V. Bruggner, R. Melamed, A. Trejo, O. I. Ornatsky, R. S. Balderas, S. K. Plevritis, K. Sachs, D. Pe'er, S. D. Tanner, and G. P. Nolan. Single-Cell Mass Cytometry of Differential Immune and Drug Responses Across a Human Hematopoietic Continuum. *Science (New York, N.y.)*, 332(6030):687–696, may 2011.
15. D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014.
16. J. Bezanson, A. Edelman, S. Karpinski, and V. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98, 2017.
17. Mocha: julia package. https://mochajl.readthedocs.io/en/latest/. Accessed: 2018-10-22.
18. ScikitLearn: julia package. https://scikitlearnjl.readthedocs.io/en/latest/. Accessed: 2018-10-22.