CYTOSET: PREDICTING CLINICAL OUTCOMES VIA SET-MODELING OF CYTOMETRY DATA

Haidong Yi[1], **Natalie Stanley**[1,2]
[1]Department of Computer Science
[2]Computational Medicine Program
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514, USA
{haidyi,natalies}@cs.unc.edu

ABSTRACT

Single-cell flow and mass cytometry technologies are being increasingly applied in clinical settings, as they enable the simultaneous measurement of multiple proteins across millions of cells within a multi-patient cohort. In this work, we introduce CytoSet, a deep learning model that can directly predict a patient's clinical outcome from a collection of cells obtained through a blood or tissue sample. Unlike previous work, CytoSet explicitly models the cells profiled in each patient sample as a set, allowing for the use of recently developed permutation invariant architectures. We show that CytoSet achieves state-of-the-art classification performance across a variety of flow and mass cytometry benchmark datasets. Specifically, CytoSet greatly outperforms two baseline models by 20.6% on a large multi-sample clinical flow cytometry dataset. The strong classification performance is further complemented by demonstrated robustness to the number of sub-sampled cells per patient, enabling CytoSet to scale to hundreds of patient samples. Furthermore, we also conducted an ablation study with networks of varying depths to demonstrate that much of the representation power of CytoSet comes from the permutation-equivalent architectures. The superior performance achieved by the set-based architectures used in CytoSet suggests that clinical cytometry data can be appropriately interpreted and studied as sets. The code is publicly available at https://github.com/CompCy-lab/cytoset.

1 INTRODUCTION

High-throughput single-cell technologies, such as flow and mass cytometry, have shown particular promise in numerous translational applications in systems immunology (Davis et al., 2017), such as, pregnancy (Aghaeepour et al., 2017), aging (Alpert et al., 2019), and recovery from surgical trauma (Ganio et al., 2020). In clinical settings, these technologies can simultaneously measure between 20-40 markers at single-cell resolution (Spitzer & Nolan, 2016), across multiple patient samples. The protein measurements collected for each cell facilitate the phenotyping and functional characterization of diverse immune cell-types. Such information can be further used to predict a patient's clinical outcome, or classification (e.g. healthy or sick).

Based on the financial and time expenses of modern single-cell assays, it is increasingly important to develop robust machine learning methods that enable clinically-meaningful predictions from a set of profiled cells. Such tasks therefore defines a computational challenge of linking the set of multiple single-cell measurements per patient to their respective clinical outcomes. To address this, two major classes of techniques have been proposed. The so-called 'gating-based' methods (Bruggner et al., 2014; Lun et al., 2017; Weber et al., 2019; Stanley et al., 2020) first cluster cells across all patient samples into homogeneous subsets according to the expression of the measured proteins. From the resulting clusters, a simple feature vector is typically engineered for each sample, reflecting the relative distribution (or frequency) of cells across each cluster. Such engineered feature vectors can therefore be used for downstream tasks, such as regression (Aghaeepour et al., 2017) and classification (Stanley et al., 2020; Bruggner et al., 2014; Ganio et al., 2020).

The feature vector defined in this ‘gating’ step has two critical properties: (1) it has a user-specified length, despite the fact that the number of cells across different samples is often quite variable; (2) the order of cells in each sample does not affect the corresponding feature vector. Although manual or automatic gating generates features that can be input to a standard machine learning model (e.g. logistic regression), these approaches also present several problems. First, unsupervised clustering requires specifying the number of clusters in advance and most methods do not scale well to millions of cells. Second, clustering often suffers from high variance by different random seeds (Stanley et al., 2020). Finally, the gating step builds on the assumption that all of the useful and clinically-relevant information can be represented by computing the distribution of cells across clusters in each sample. In summary, the ‘gating’ step imposed through clustering introduces a strong inductive bias in the feature extraction step.

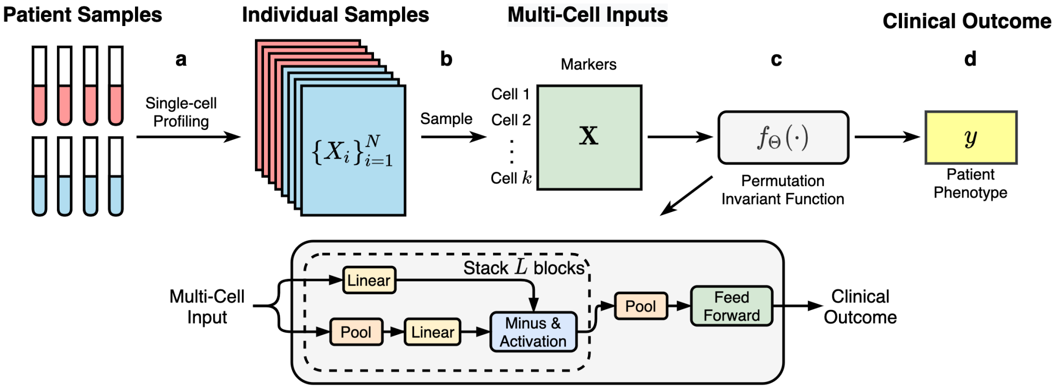

Figure 1: An illustration of the model architecture of CytoSet. (**a**) The collected patient samples are profiled with multiple proteins (markers) at a single-cell level. (**b**) A subsample of all profiled cells across samples is used for training the model. Each multi-cell input is sampled with replacement from an individual sample and has a fixed number of cells, k. (**c**) The details of the permutation invariant network architecture used in the CytoSet model. (**d**) The learned feature vectors of multi-cell inputs are used to classify the clinical outcome of patients (e.g. patient phenotype).

To address these limitations, two gating-free methods (Arvaniti & Claassen, 2017; Hu et al., 2019) were recently proposed. Unlike the clustering, or primarily ‘gating’-based approaches, these techniques operate on the single-cell level and try to aggregate information across all single cells in a learnable way. For example, CellCNN (Arvaniti & Claassen, 2017) uses convolutional neural networks (CNNs) as an end-to-end model to learn the associated phenotype from multi-cell input. Specifically, CellCNN uses a 1d-convolution layer to project the measurements of each cell to an embedding space and then applies a pooling layer to aggregate information across multiple different cells. CytoDx (Hu et al., 2019) uses a two-level linear model to predict clinical outcome across individual cells. On the individual cell level, CytoDx operates on each cell and generates the predictor for the sample. At the sample level, CytoDx uses logistic regression to link the predictor and the corresponding clinical outcome.

Although these techniques can successfully identify and achieve strong classification accuracy in small datasets, these methods often do not adapt well to larger datasets, which are likely to contain bias from batch effects and noisy measurements. In this paper, we propose a new deep learning model called CytoSet to predict clinical outcomes from single-cell flow and mass cytometry data. We demonstrate that CytoSet can robustly predict clinical outcome on several benchmark flow and mass cytometry datasets even with only a limited subset of cells (on the order of thousands per sample). The contributions of CytoSet can be summarized as follows:

1. CytoSet is an end-to-end model that can predict clinical outcomes directly on a set of cells profiled in each patient sample. The prediction task with cells as input contrasts the gating-based approaches, which train a model based on the engineered feature vectors;

2. CytoSet uses the permutation invariant network architecture inspired by the seminal Deep Sets work introduced in Ref. (Zaheer et al., 2017) to extract information from set-structured cytometry data.
3. CytoSet achieves state-of-the-art classification performance on several benchmark flow and mass cytometry datasets from the FlowCap-II challenge (Aghaeepour et al., 2013). CytoSet further outperforms two baseline methods (CellCNN and CytoDX) by a large margin.

2 METHODS

In this section, we introduce the CytoSet model. We begin by introducing the notation used in this paper. Next, we describe general set modeling and why we need it for classification tasks on cytometry data. We will then provide a comprehensive description of CytoSet including the network architecture and the loss function. Finally, we compare the CytoSet model to two existing methods, CellCNN (Arvaniti & Claassen, 2017) and CytoDx (Hu et al., 2019).

2.1 NOTATION

In this paper, $\boldsymbol{X}$ denotes a set, $\mathbf{X}$ denotes a matrix, $\boldsymbol{x}$ denotes a column vector, x or X denotes a scalar. Given a matrix $\mathbf{X}$, we use $\mathbf{X}(i,:)$ and $\mathbf{X}(:,j)$ to represent the ith row and jth column of $\mathbf{X}$, respectively. The (i,j) element in $\mathbf{X}$ is denoted by $\mathbf{X}(i,j)$. Given a vector $\boldsymbol{x}$, we use $\boldsymbol{x}(i)$ to denote the ith element of $\boldsymbol{x}$. We use $\mathbf{X}^\top$ and $\boldsymbol{x}^\top$ to represent the transpose of matrix $\mathbf{X}$ and vector $\boldsymbol{x}$, respectively.

2.2 SET MODELING

For set-structured data, each sample is a collection of unordered data points denoted by $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\} \in \mathcal{X}^m$, where m is the cardinality of the set $\boldsymbol{X}$ and $\mathcal{X}$ denotes the domain of each element $\boldsymbol{x}_i$. For example, a set of multiple dimensional points, such as d-dimensional single-cell measurements can be viewed as a set. Since different sets are equal only if they have the same elements, models that map a set to some target values must preserve the following permutation invariant property:

Definition 1 (Permutation Invariant) *Let $f : \mathcal{X}^m \rightarrow \mathcal{Y}$ be a function, then f is permutation invariant iff for any permutation $\pi(\cdot)$, $f(\boldsymbol{X}) = f(\pi(\boldsymbol{X}))$.*

In addition, we also introduce another kind of mapping that is important for building permutation invariant functions in set modeling:

Definition 2 (Permutation Equivalent) *Let $f : \mathcal{X}^m \rightarrow \mathcal{Y}^m$ be a function, then f is permutation equivalent iff for any permutation $\pi(\cdot)$, $f(\pi(\boldsymbol{X})) = \pi(f(\boldsymbol{X}))$.*

One natural way to encourage permutation invariance is to cluster the elements of a set and output some summary statistics, such as, the proportion of cells in each cluster, as the feature vector. Although clustering can preserve permutation invariance, it cannot be trained in an end-to-end fashion using back-propagation because the operation in clustering is non-differentiable. To solve this problem, Refs. Zaheer et al. (2017) and Edwards & Storkey (2017) propose permutation invariant neural network architectures by incorporating permutation equivalent layers and set pooling layers. The set-pooling layers play a key role in preserving permutation invariance and in aggregating information over the elements of a set. Additionally, Ref. Zaheer et al. (2017) also showed the general form of functions that have permutation invariant properties. Recently, Ref. Lee et al. (2019) proposed a new transformer-based architecture that uses attention mechanisms for both encoding and aggregating features in a set.

2.3 PROBLEM FORMULATION

Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ denote a collection of multiple cells profiled with a high-throughput single-cell technology, such as flow or mass cytometry. Here, m is the number of cells and each $\boldsymbol{x}_i \in \mathbb{R}^p$ represents the vector of expression measurements for each of the p proteins in cell i. In flow and mass cytometry experiments, the order in which cells are profiled has no biological relevance, so such a collection of cells, $\boldsymbol{X}$, can be viewed as a set rather than as a data matrix. Given a training cytometry dataset with N example cells $\{(\boldsymbol{X}_i, y_i)\}_{i=1}^{N}$ where $\boldsymbol{X}_i$ is the i-th sample that has m_i profiled single cells and $y_i \in \{0, 1\}$ denotes the clinical outcome of $\boldsymbol{X}_i$, our objective is to learn a permutation invariant function $f_\Theta(\cdot)$ that can assign the corresponding label y_i to each set instance, $\boldsymbol{X}_i$.

2.4 Network architecture

Here, we introduce our CytoSet model, with the corresponding architecture illustrated in Figure 1. Like CellCNN, CytoSet also takes a set of cells from each individual sample as the input. Let $\mathbf{X}$ denote the multi-cell input matrix of k measured cells, which is randomly drawn with replacement across all of the individual samples $\boldsymbol{X}$, i.e. $\mathbf{X} \subseteq \boldsymbol{X}$. This sampling step will generate multiple instances of each sample in the model's training. This is necessary because (1) the number of cells typically varies across the input samples; (2) the collective number of cells across the original input samples is too large to fit into a GPU. In addition, the diversity across sampled subsets further increases the number of cells involved in training and help the model scale to larger datasets (Amores, 2013). Our experimental results demonstrate that our CytoSet model is robust to the total number of sampled cells, k (see section 3.3).

Inspired by Ref. Zaheer et al. (2017), the network $f_\Theta(\cdot)$ starts by stacking several permutation equivalent blocks that transform the representation of each element in the set. We denote the function of each block as $\mathrm{Block}(\cdot)$. The set-based transformation layer will update the set representation iteratively with L blocks, and the output by the lth block can be denoted as

$$\begin{cases} \mathbf{H}^{(0)} = \mathbf{X} \\ \mathbf{H}^{(l)} = \mathrm{Block}(\mathbf{H}^{(l-1)}), \forall l \in \{1, 2, \cdots, L\} \end{cases} \tag{1}$$

where $\mathbf{H}^{(l-1)}$ is the output of the $(l-1)$th block. In each block, we also add a residual connection to stabilize the network training. After L permutation equivalent blocks, a pooling operation is then performed within the rows of $\mathbf{H}^L \in \mathbb{R}^{k\times d}$, and yields a fixed-length embedding vector for the set $\boldsymbol{X}$. We denote the embedding vector as $\boldsymbol{v} \in \mathbb{R}^d$,

$$\boldsymbol{v} = \mathrm{Pool}(\mathbf{H}^L). \tag{2}$$

In the implementation of (2), we can either use max or mean pooling. The max pooling will return the maximum value of each column in $\mathbf{H}^L$:

$$\boldsymbol{v}(j) = \max_{1\le i\le k}(\mathbf{H}(i, j)), \forall j \in \{1, 2, \cdots, d\}, \tag{3}$$

whereas the mean pooling will return the mean vale of each column in $\mathbf{H}^L$:

$$\boldsymbol{v}(j) = \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}(i, j), \forall j \in \{1, 2, \cdots, d\}. \tag{4}$$

As was previously described in CellCNN (Arvaniti & Claassen, 2017), max pooling can measure the presence of cells with high response, while mean pooling can approximate the frequency of particular cell subsets. In our experiments, we consistently used max pooling to aggregate information among cells. Finally, the embedding vector $\boldsymbol{v}$ is connected to a feed forward network (fully connected layers) to predict the clinical outcome y, which refers to the clinical outcome of the sample (e.g. the binary phenotype).

2.5 Loss Function

Let $\boldsymbol{h}$ denote the input of the final classification layer in $f_\Theta(\cdot)$. We can write the conditional log likelihood of the label y_i, given the set $\boldsymbol{X}_i$ and model parameters Θ as

$$
\begin{aligned}
\log p(y_i \mid \boldsymbol{X}_i, \Theta) = {} & y_i \log \sigma(\boldsymbol{\theta}_c^\top \boldsymbol{h}_i + \theta_0) \\
& + (1 - y_i) \log(1 - \sigma(\boldsymbol{\theta}_c^\top \boldsymbol{h}_i + \theta_0)).
\end{aligned} \tag{5}
$$

Here, $\boldsymbol{\theta}_c$ and θ_0 are the classification parameters in Θ and $\sigma(z) = 1/(1 + \exp(-z))$ is the *sigmoid* function. The presented permutation invariant deep neural network is trained by minimizing the following binary cross-entropy loss function:

$$
\ell = -\frac{1}{N} \sum_{i=1}^{N} \log p(y_i \mid \boldsymbol{X}_i, \Theta) \tag{6}
$$

We have thus far introduced all of the components used in our CytoSet model (Figure 1). The details of the training algorithm used for CytoSet is given in Algorithm 1.

```
**Algorithm 1** CytoSet Algorithm
    **Input**:
      • Labeled cytometry dataset: $\mathcal{X} = \{(\boldsymbol{X}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{X}_i$ is a collection of cells with $p$
        measured markers and $y_i \in \{0, 1\}$ is the clinical outcome of $\boldsymbol{X}_i$;
      • model: $f_\Theta(\cdot)$ with initialized parameters $\Theta^{(0)}$;
      • Batch size: $B$ and learning rate: $\alpha$.
    **Output**: Trained model $f_\Theta(\cdot)$.
    **repeat until** $\Theta$ **convergence**
    1. Sample a batch of data $\{(\mathbf{X}_b, y_b) : b \in (1, \ldots, B)\}$ from $\mathcal{X}$.
    2. $\ell \leftarrow \frac{1}{B} \sum_{i=1}^{B} \texttt{BCELoss}(f_\Theta(\mathbf{X}_i), y_i)$ (Binary cross-entropy loss).
    3. $\boldsymbol{g}_\Theta \leftarrow \nabla_\Theta \ell$ (Compute the gradient).
    4. $\Theta \leftarrow \Theta - \alpha \cdot \mathrm{Adam}(\Theta, \boldsymbol{g}_\Theta)$ (Update parameters).
```

2.6 Model Comparison

CytoSet and the two baseline methods, CellCNN (Arvaniti & Claassen, 2017) and CytoDx(Hu et al., 2019), all belong to the class of end-to-end gating-free methods. These methods all combine permutation equivalent and pooling functions to build a permutation invariant model on cytometry data. For the pooling function, CytoSet and CellCNN can either use max or mean pooling, while CytoDx only uses mean pooling. For the permutation equivalent function, both CellCNN and CytoDx use only one permutation equivalent layer while CytoSet can stack multiple layers. Thus, CellCNN and CytoDx can be viewed as special cases of CytoSet by limiting the model depth. The deeper architecture of CytoSet makes it more flexible for learning the complicated relationships between the raw input and the associated clinical outcome. This is because the model can reuse more information and increasingly generate abstract features from previous layers (Krizhevsky et al., 2012; He et al., 2016).

3 Results and Analysis

We ran a series of experiments to compare CytoSet to CellCNN and CytoDx on several benchmark flow and mass cytometry datasets introduced in the FlowCAP-II challenge (Aghaeepour et al., 2013). In section 3.1, we give an introduction to the datasets used for the classification tasks. Next, in section 3.2, we provide details about the model training procedure, including the selection of hyper-parameters and model. Following in section 3.3, we evaluated our method and compare the classification performance with CellCNN and CytoDx. Finally, to ultimately make our model more interpretable, we visualized the embedding vector learned by CytoSet in section 3.4.

3.1 Description of Datasets

In our experiments, we tested the classification performance of CytoSet along with two baseline methods, CellCNN (Arvaniti & Claassen, 2017) and CytoDx (Hu et al., 2019), on the benchmark

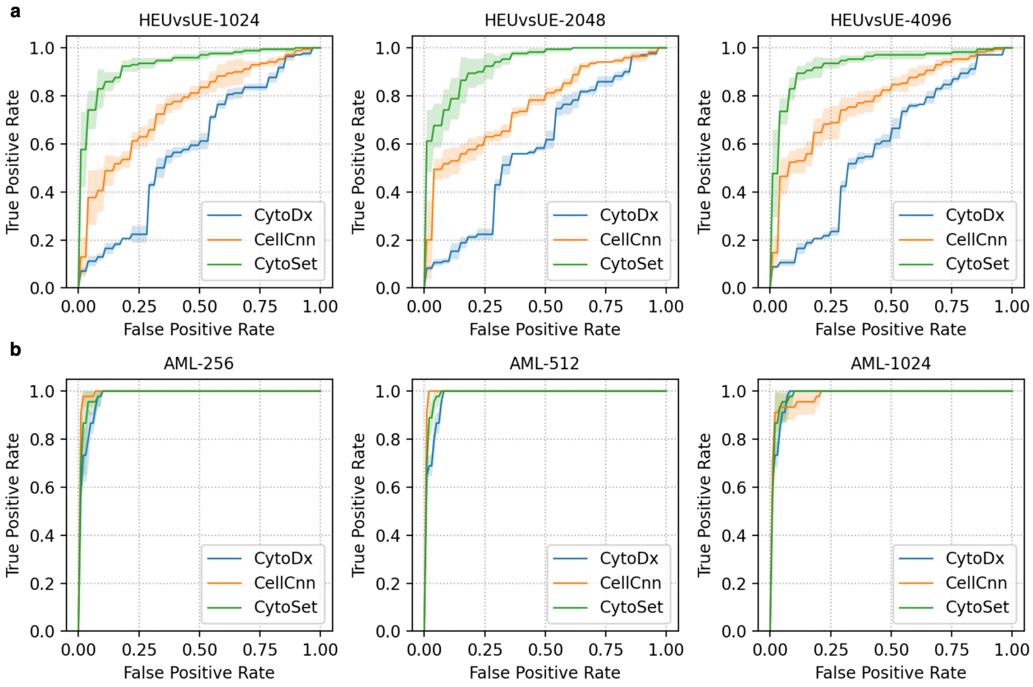

Figure 2: The test ROC curves of CytoDx, CellCNN and CytoSet on the HEUvsUE (**a**) and AML (**b**) datasets from subsampling {1024, 2048, 4096} and {256, 512, 1024} cells across patient samples, respectively. The shaded regions represent the standard deviation of 5 runs with different random seeds. All three methods perform well on the AML dataset, but CytoSet shows considerable improvement on HEUvsUE.

flow cytometry datasets from the FlowCAP-II challenge (Aghaeepour et al., 2013). The FlowCAP-II challenge consists of three different datasets, denoted as HEUvsUE, AML, and HVTN, respectively. The HEUvsUE dataset consists of 308 blood samples from African infants who were either exposed to HIV in *utero* but remain uninfected (HEU) or who were unexposed (UE). From the results previously reported in the FlowCap-II challenge (Aghaeepour et al., 2013), the ability to achieve high prediction quality in the HEUvsUE dataset was shown to be quite challenging. The AML dataset has totally 2872 samples collected from 359 AML (acute myeloid leukemia) and non-AML individuals. The HVTN dataset consists of 96 samples of two antigen stimulation groups of post-HIV vaccination T cells (Gag versus Env stimulated) from the HIV Vaccine Trials Network (HVTN). In addition, we also use a smaller mass cytometry dataset called NK cell (Horowitz et al., 2013), which was used as an example in CellCNN. This dataset was pre-processed to include only cells across 20 samples that were characterized as NK cells according to a biological expert. Moreover, the interest in this dataset is to characterize human natural killer cell diversity. For the statistics of the datasets, please refer to Table 5 in the Appendix.

Availability The HEUvsUE, AML and HVTN datasets are all available through FlowRepository[1] (Spidlen et al., 2012) with the following experiment IDs: FR-FCM-ZZZU (HEUvsUE), FR-FCM-ZZYA (AML), and FR-FCM-ZZZV (HVTN). The NK cell dataset is available with CellCNN (`https://github.com/eiriniar/CellCnn`).

3.2 Experimental Setup

In the HEUvsUE and AML datasets, we randomly chose 80% of the samples from each class for training and validation and left out the remaining 20% for testing. For the analysis of the HVTN

[1] `https://flowrepository.org`

dataset, we used the 50%-50% train-test split as indicated in the associated metadata (Aghaeepour et al., 2013; Spearman et al., 2011). We further adopted the same train-test split as CellCNN to facilitate objective comparison in the NK cell dataset. Before training, all of the raw measurements of cells across different datasets were transformed using an `arcsinh` normalization, $f(x) = \text{arcsinh}(x/5)$ (Azad et al., 2016). We used both the classification accuracy (ACC) and the area under the receiver operator curve (AUC) to quantify the classification performance. We trained our deep learning model using Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ across different datasets. The learning rate and batch size are set to 0.0001 and 200 across different experiments, respectively. In all the experiments, we selected the best model based on the performance (interms of AUC) on the validation dataset. To prevent overfitting, we also applied early stopping to the AML and NK cell datasets. As a result, the training stopped after five epochs of unimproved AUC on the validation dataset.

3.3 Results on Benchmark Flow and Mass Cyometry Datasets

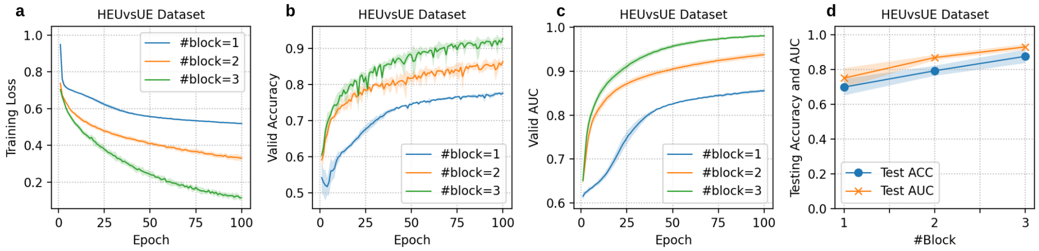

Figure 3: Training loss (**a**), validation ACC (**b**) and validation AUC (**c**), versus training epochs of CytoSet model on HEUvsUE dataset using 1, 2 and 3 blocks. (**d**) The test ACC and AUC of CytoSet model on HEUvsUE dataset using 1, 2 and 3 blocks. The shaded regions in (**a**)-(**d**) represent the standard deviation of 5 runs with different random seeds.

3.3.1 HEUvsUE Flow Cytometry Dataset

We trained our model on the HEUvsUE dataset for 100 epochs using three permutation equivalent blocks. Considering the difficulty of fitting an appropriate model to this dataset, we used larger ks (e.g. the number of subsampled cells) in training. We report the test classification results in Table 1 and the corresponding ROC curves in Figure 2(**a**). For varying values of k, our CytoSet model consistently outperforms other baselines by a large margin (20.6% on average). Varying k, or the number of cells sampled from each patient's set of cells, the performance across different models is only slightly changed. Furthermore, we find that CellCNN, as a simpler deep learning method in comparison to CytoSet, also outperforms the logistic regression method, CytoDx. This may be explained by the fact that CytoDx has less parameters than CellCNN and CytoSet. By comparing model complexity and classification accuracy for different models, we hypothesize that the classification performance on the HEUvsUE dataset is limited by the number of model parameters in previous work.

Model	$k = 1024$		$k = 2048$		$k = 4096$	
	ACC	AUC	ACC	AUC	ACC	AUC
CellCNN (Arvaniti & Claassen, 2017)	0.630	0.778	0.674	0.763	0.689	0.785
CytoDx (Hu et al., 2019)	0.587	0.587	0.600	0.588	0.597	0.590
CytoSet	**0.858**	**0.936**	**0.842**	**0.935**	**0.881**	**0.933**

Table 1: The testing ACC and AUC obtained from varying, k, the number of subsampled cells per patient sample in the HEUvsUE dataset. The numbers reported are averaged over 5 runs with different random seeds.

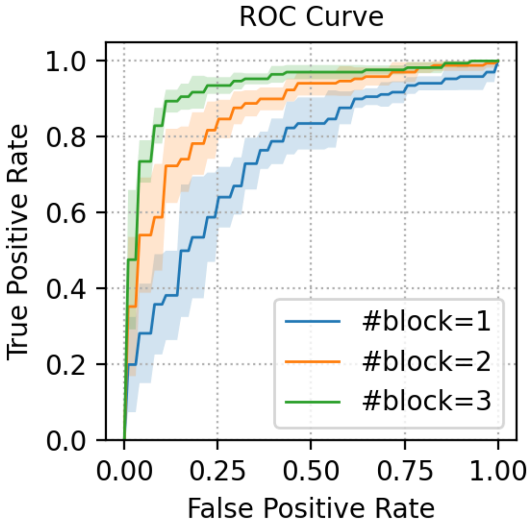

Figure 4: The test ROC curves of CytoSet on the HEUvsUE dataset, trained using 1, 2, and 3 blocks (blue, orange, and green curves, respectively). The shaded region represents the standard deviation of five runs with different random seeds.

To verify the importance of the model's depth, we carried out the experiments on the HEUvsUE dataset using various numbers of blocks. Here, we trained our model using 1, 2 and 3 permutation equivalent blocks (blue, orange, and green curves, respectively). The number of parameters used in different models is given in Table 6 of Appendix. The training loss curve is shown in Figure 3(**a**). Here in 3(**a**), we observe that the loss drops faster and the final convergence loss is also lower when using more blocks. We also report the validation ACC in Figure 3(**b**) and validation AUC in Figure 3(**c**). The validation ACC and AUC demonstrate that adding more blocks does not cause over-fitting. The test ACC and AUC of CytoSet with different training blocks are illustrated in Figure 3(**d**). In addition, we also give the test ROC of models with different blocks in Figure 4. The test performance shows that models with more parameters can achieve better performance on HEUvsUE dataset.

3.3.2 AML Mass Cytometry dataset

We trained our model on the AML dataset for 20 epochs using only one permutation equivalent block. We report the performance of the two baseline methods along with CyoSet in Table 2. The corresponding receiver operator curve (ROC) is illustrated in Figure 2(**b**). All of the methods perform reasonably well and exhibit comparable performance. This indicates that the AML dataset is relatively simple for models to fit. This point is also verified in the evaluation results of Aghaeepour et al. (2013). Again, the number of subsampled cells per patient sample, k, only slightly affects the classification performance of different models.

Model	$k = 256$		$k = 512$		$k = 1024$	
	ACC	AUC	ACC	AUC	ACC	AUC
CellCNN (Arvaniti & Claassen, 2017)	**0.981**	**0.998**	**0.967**	**0.999**	**0.956**	0.985
CytoDx (Hu et al., 2019)	0.940	0.983	0.943	0.984	0.948	0.986
CytoSet	0.970	0.991	0.959	0.993	0.954	**0.988**

Table 2: The testing ACC and AUC obtained from varying, k, the number of subsampled cells per patient sample in the AML dataset. The numbers reported are averaged over 5 runs with different random seeds.

3.3.3 HVTN Flow Cytometry dataset

We trained our model on the HVTN dataset for 100 epochs using three permutation equivalent blocks. The test classification results are reported in Table 3. Similar to the results observed in the HEUvsUE dataset, our CytoSet model achieves the best performance across the three different methods, and CellCNN consistently outperforms CytoDx. The test results on the HVTN dataset also demonstrate that more trainable parameters can help CytoSet generalize well to the testing samples and bypass potential underfitting in the model training.

Model	$k = 256$		$k = 512$		$k = 1024$	
	ACC	AUC	ACC	AUC	ACC	AUC
CellCNN (Arvaniti & Claassen, 2017)	0.829	0.934	0.796	0.927	0.837	0.933
CytoDx (Hu et al., 2019)	0.658	0.746	0.629	0.738	0.654	0.741
CytoSet	**0.913**	**0.963**	**0.892**	**0.964**	**0.883**	**0.957**

Table 3: The testing ACC and AUC obtained from varying, k, the number of subsampled cells per patient sample in the HVTN dataset. The numbers reported are averaged over 5 runs with different random seeds.

3.3.4 NK-Cell Mass Cytometry dataset

In addition to the large datasets used in the FlowCap-II challenge, we also tested our model along with two baselines on the smaller NK-cell mass cytometry dataset. We trained our model for only 10 epochs using one permutation equivalent block, since there were only 14 training samples. The test classification results is shown in Table 4. Unlike other datasets, there is no consistent winner on this dataset. CytoSet and CytoDx outperform CellCNN on this dataset. The performance of CytoDx indicates that logistic regression is also good enough when there are only a few of samples available for training the classification model. The performance of CytoSet demonstrates our model are applicable for both large and small cytometry datasets.

Model	$k = 256$		$k = 512$		$k = 1024$	
	ACC	AUC	ACC	AUC	ACC	AUC
CellCNN (Arvaniti & Claassen, 2017)	**0.833**	0.950	0.767	0.975	0.667	0.825
CytoDx (Hu et al., 2019)	**0.833**	0.925	**0.833**	**1.000**	0.833	0.850
CytoSet	0.800	**0.975**	**0.833**	0.975	**0.867**	**0.950**

Table 4: The testing ACC and AUC obtained from varying, k, the number of subsampled cells per patient sample in the NK cell mass cytometry dataset. The numbers reported are averaged over 5 runs with different random seeds.

3.4 Visualization of Learned Sample Embeddings

To better understand the information learned by CytoSet for each test patient sample, we visualized the set embedding vector $\boldsymbol{h}$ in 2-dimensions with t-SNE (Van der Maaten & Hinton, 2008). As representative examples, we show these visualizations for the AML and HEUvsUE datasets in Figure 5. The corresponding t-SNE visualizations were omitted for HVTN and NK cell datasets since the number of testing samples was too small. The t-SNE visualization shows that the AML and normal test samples are clustered together independently (Figure 5(**a**)) and the HEU and UE test samples only have a small number of misclassified samples 5(**b**)). These observations qualitatively demonstrate that CytoSet can learn the effective representations of the sets for discriminating the associated clinical outcomes.

4 Conclusion

In this paper, we proposed a new deep learning model called CytoSet for predicting clinical outcomes from single-cell flow and mass cytometry data. In the problem setup, CytoSet innovatively

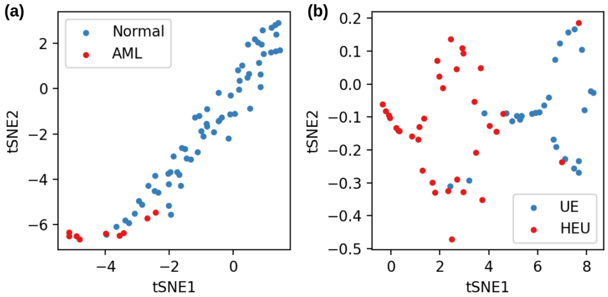

Figure 5: t-SNE visualizations of the test samples of the patients based on the set embedding vector $\boldsymbol{h}$ learned by CytoSet. The color of each point (patient sample) indicates the phenotype (clinical outcome). (a) AML dataset, (b) HEUvsUE dataset.

formulates the prediction of clinical outcomes as a classification task on sets. For the architecture design, CytoSet generalizes two gating-free methods CellCNN and CytoDx using stackable permutation invariant network architectures, which improves the model's expressive power.

In the experiments, we demonstrate that CytoSet greatly outperforms two other gating-free methods on the classification task of two large cytometry datasets, HEUvsUE and HVTN, and is also robust to the k, or the number of subsampled cells. We also show that it is beneficial to incorporate deeper architectures while training the model on large and challenging datasets such as HEUvsUE. We believe that the presented deep learning method CytoSet successfully advances the ability to link data generated through single-cell assays, such as flow and mass cytometry, to external clinical variables of interest.

References

Nima Aghaeepour, Greg Finak, Holger Hoos, Tim R Mosmann, Ryan Brinkman, Raphael Gottardo, and Richard H Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nature methods*, 10(3):228–238, 2013.

Nima Aghaeepour, Edward A Ganio, David Mcilwain, Amy S Tsai, Martha Tingle, Sofie Van Gassen, Dyani K Gaudilliere, Quentin Baca, Leslie McNeil, Robin Okada, et al. An immune clock of human pregnancy. *Science immunology*, 2(15), 2017.

Ayelet Alpert, Yishai Pickman, Michael Leipold, Yael Rosenberg-Hasson, Xuhuai Ji, Renaud Gaujoux, Hadas Rabani, Elina Starosvetsky, Ksenya Kveler, Steven Schaffert, et al. A clinically meaningful metric of immune age derived from high-dimensional longitudinal monitoring. *Nature medicine*, 25(3):487–495, 2019.

Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial intelligence*, 201:81–105, 2013.

Eirini Arvaniti and Manfred Claassen. Sensitive detection of rare disease-associated cell subsets via representation learning. *Nature communications*, 8(1):1–10, 2017.

Ariful Azad, Bartek Rajwa, and Alex Pothen. flowvs: channel-specific variance stabilization in flow cytometry. *BMC bioinformatics*, 17(1):1–14, 2016.

Robert V Bruggner, Bernd Bodenmiller, David L Dill, Robert J Tibshirani, and Garry P Nolan. Automated identification of stratifying signatures in cellular subpopulations. *Proceedings of the National Academy of Sciences*, 111(26):E2770–E2777, 2014.

Mark M Davis, Cristina M Tato, and David Furman. Systems immunology: just getting started. *Nature immunology*, 18(7):725, 2017.

Harrison Edwards and Amos Storkey. Towards a neural statistician. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Edward A Ganio, Natalie Stanley, Viktoria Lindberg-Larsen, Jakob Einhaus, Amy S Tsai, Franck Verdonk, Anthony Culos, Sajjad Ghaemi, Kristen K Rumer, Ina A Stelzer, et al. Preferential inhibition of adaptive immune system dynamics by glucocorticoids in patients after acute surgical trauma. *Nature communications*, 11(1):1–12, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Amir Horowitz, Dara M Strauss-Albee, Michael Leipold, Jessica Kubo, Neda Nemat-Gorgani, Ozge C Dogan, Cornelia L Dekker, Sally Mackey, Holden Maecker, Gary E Swan, et al. Genetic and environmental determinants of human nk cell diversity revealed by mass cytometry. *Science translational medicine*, 5(208):208ra145–208ra145, 2013.

Zicheng Hu, Benjamin S Glicksberg, and Atul J Butte. Robust prediction of clinical outcomes using cytometry data. *Bioinformatics*, 35(7):1197–1203, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753. PMLR, 2019.

Aaron TL Lun, Arianne C Richard, and John C Marioni. Testing for differential abundance in mass cytometry data. *Nature methods*, 14(7):707, 2017.

Paul Spearman, Michelle A Lally, Marnie Elizaga, David Montefiori, Georgia D Tomaras, M Juliana McElrath, John Hural, Stephen C De Rosa, Alicia Sato, Yunda Huang, et al. A trimeric, v2-deleted hiv-1 envelope glycoprotein vaccine elicits potent neutralizing antibodies but limited breadth of neutralization in human volunteers. *Journal of Infectious Diseases*, 203(8):1165–1173, 2011.

Josef Spidlen, Karin Breuer, Chad Rosenberg, Nikesh Kotecha, and Ryan R Brinkman. Flowrepository: A resource of annotated flow cytometry datasets associated with peer-reviewed publications. *Cytometry Part A*, 81(9):727–731, 2012.

Matthew H Spitzer and Garry P Nolan. Mass cytometry: single cells, many features. *Cell*, 165(4): 780–791, 2016.

Natalie Stanley, Ina A Stelzer, Amy S Tsai, Ramin Fallahzadeh, Edward Ganio, Martin Becker, Thanaphong Phongpreecha, Huda Nassar, Sajjad Ghaemi, Ivana Maric, et al. Vopo leverages cellular heterogeneity for predictive modeling of single-cell data. *Nature communications*, 11(1): 1–9, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Lukas M Weber, Malgorzata Nowicka, Charlotte Soneson, and Mark D Robinson. diffcyt: Differential discovery in high-dimensional cytometry via high-resolution clustering. *Communications biology*, 2(1):1–11, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, 2017.

A Appendix

Scale	Name	Cytometry	#Sample	#Individual	Train/Test Split
Large	AML	Mass	2872	359	4:1
Large	HEUvsUE	Flow	308	44	4:1
Medium	HVTN	Flow	96	96	1:1
Small	NK Cell	Mass	20	20	7:3

Table 5: The details of the datasets used in the experiments.

	#block=1	#block=2	#block=3
#Params($\times 10^3$)	71.4	202.8	334.1

Table 6: The number of parameters of the CytoSet model on the HEUvsUE dataset with various numbers of blocks.