Exploring Single-Cell Data with Deep Multitasking Neural Networks

Matthew Amodio[*,1] Krishnan Srinivasan[*,1]
David van Dijk[*,1] Hussein Mohsen[4] Kristina Yim[2]
Rebecca Muhle[2] Kevin R. Moon[2,3] Susan Kaech[5]
Ryan Sowell[5] Ruth Montgomery[5] James Noonan[2]

Guy Wolf[†,3]

Smita Krishnaswamy[†,**,2]

[1]Department of Computer Science;
[2]Department of Genetics;
[3]Applied Mathematics Program;
[4]Computational Biology and Bioinformatics
[5]Department of Immunobiology
Yale University, New Haven, CT, USA
[**]Corresponding author. E-mail: smita.krishnaswamy@yale.edu
Address: 333 Cedar St, New Haven, CT 06510, USA
[*] These authors contributed equally. [†] These authors contributed equally.

Abstract

Handling the vast amounts of single-cell RNA-sequencing and CyTOF data, which are now being generated in patient cohorts, presents a computational challenge due to the noise, complexity, sparsity and batch effects present. Here, we propose a unified deep neural network-based approach to automatically process and extract structure from these massive datasets. Our unsupervised architecture, called SAUCIE (Sparse Autoencoder for Unsupervised Clustering, Imputation, and Embedding), simultaneously performs several key tasks for single-cell data analysis including 1) clustering, 2) batch correction, 3) visualization, and 4) denoising/imputation. SAUCIE is trained to recreate its own input after reducing its dimensionality in a 2-D embedding layer which can be used to visualize the data. Additionally, it uses two novel regularizations: (1) an information dimension regularization to penalize entropy as computed on normalized activation values of the layer, and thereby encourage binary-like encodings that are amenable to clustering and (2) a Maximal Mean Discrepancy penalty to correct batch effects. Thus SAUCIE has a single architecture that denoises, batch-corrects, visualizes and clusters data using a unified

representation. We show results on artificial data where ground truth is known, as well as mass cytometry data from dengue patients, and single-cell RNA-sequencing data from embryonic mouse brain.

1 Introduction

Vast amounts of high-dimensional, high-throughput, single-cell data measuring various aspects of cells including mRNA molecules, proteins, epigenetic marks and histone modifications are being generated via new technologies. Furthermore, the number of patients included in large-scale studies of single-cell data for comparing across populations or disease conditions is rapidly increasing. Processing data of this dimensionality and scale is an inherently difficult prospect, especially considering the degree of noise, batch effects, artifacts, sparsity and heterogeneity in the data. Here, we propose a deep learning approach to process and analyze this type of data (single-cell data from a cohort of patients).

While traditional deep learning methods aim to automate predictive and generative tasks, we utilize deep learning in an exploratory fashion, to reveal the structure of multi-sample data without supervision. We base our approach on the *autoencoder*. An autoencoder is a neural network that learns to recreate its own input via a low-dimensional bottleneck layer that learns meaningful representations of the data and enables a denoised recreation of the input [1, 2, 3, 4]. If the low-dimensional bottleneck is chosen to be 2-D, then it naturally serves as a visualization of these meaningful representations as well. Since neural networks learn their own features, they can reveal structure that other 2-D visualization methods that require a definition of distance in the original data space cannot [5].

SAUCIE leverages the ability of an autoencoder to denoise, impute, and visualize, and adds carefully-designed regularizations to perform batch correction and clustering, which are essential tasks in single-cell data analysis. We introduce two novel regularizations: 1) *information dimension regularization* which allows us to recover cluster structure in a hidden layer of the autoencoder, and 2) a maximal mean discrepancy regularization that constrains the embedding layers such that different samples (of the same experimental condition) overlap with each other to perform batch correction. SAUCIE, like other modern neural networks, is also scalable to large datasets due to its massive parallelizability [6].

2 Results

2.1 SAUCIE architecture

SAUCIE is based on the autoencoder neural network framework for unsupervised learning, which learns to reconstruct its input after passing it through a low-dimensional bottleneck layer. The bottleneck layer forces the autoencoder to learn compressed representations of the input and find high-level relationships between parts of the input space. The overall structure of an autoencoder

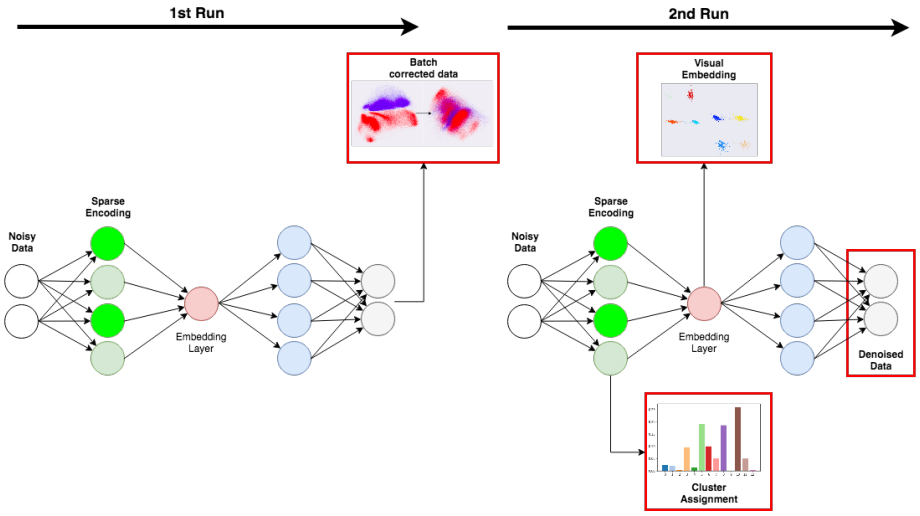

Figure 1: SAUCIE's neural network framework.

can be divided into two parts: an encoder network that maps the input space to the low-dimensional representation, and a decoder network that maps the low-dimensional representation back to the original input space. The general architecture of SAUCIE (Figure 1) is as follows:

1. An input layer
2. A variable number of encoder layers
3. A sparse encoder layer for clustering
4. An embedding layer for visualization
5. A corresponding decoder symmetric to the encoder
6. An output layer for the reconstruction.

Different layers of SAUCIE are used for different analysis tasks. A sparse encoder layer for unsupervised clustering, an embedding layer for visualization and batch normalization, and the output layer for reconstruction of denoised and imputed input values. Thus SAUCIE uses a unified representation and framework for many tasks, thereby increasing the coherence between tasks. For instance, the visualization and clustering in SAUCIE correspond with each other as they are in subsequent layers. We developed and implemented novel regularizations designed to restrict the representations in particular layers to achieve particular tasks. These regularizations used together form a pipeline for using neural networks to analyze biological data.

2.2 Multitask training by sequential optimization

To perform multiple tasks, SAUCIE uses the single architecture as described above, but is run and optimized twice sequentially. The first run imputes noisy values and corrects batch effects in the original data, while also providing two-dimensional coordinates for visualization. This preprocessed data is then run through SAUCIE again to pick out clusters. The two different runs are done by optimizing two different objective functions. In the following, we describe the optimization of each run over a single batch of n data points. However, the full optimization of each run independently utilizes multiple (mini-)batches in order to converge and minimize the described loss functions.

For the first run, formally let X be an $n \times d$ input batch, where each row is a single data point, and d is the number of features in the data. It is passed through a cascade of encoding linear and nonlinear transformations before reaching the first visualization layer that computes visualizable 2D coordinates, denoted $V \in \mathbb{R}^{n \times 2}$, for the data point in the batch. Then, a cascade of decoding transformations reconstruct the denoised batch $\hat{X}$, which has the same dimensions as the input X and is optimized to reconstruct it.

For the second run, the cleaned batch $\hat{X}$ is passed through encoding transformations until it reaches a visualization layer. However, in this case, we also consider an intermediate clustering layer that outputs near-binary activations $B \in \mathbb{R}^{n \times d_B}$, where d_B is the number of hidden nodes in the layer, which will be used to encode cluster assignments, as described below. The activations in B then pass to the second visualization layer, whose output will be denoted by $V' \in \mathbb{R}^{n \times 2}$. Finally, the second decoding transformation leads to a reconstruction $\tilde{X}$ that has the same dimensions as $\hat{X}$ (and X) and is optimized to reconstruct the cleaned batch.

The loss function of both runs starts with a reconstruction loss L_r forcing the autoencoder to learn to reconstruct its input at the end. SAUCIE uses the standard mean-squared error loss (e.g., $L_r(X, \hat{X}) = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2$, where x_i and $\hat{x}_i$ are the i-th row of X and $\hat{X}$ correspondingly), but this can be substituted with any other loss function that may be appropriate for the data. For the first run, we add to this loss a regularization term L_b that enables SAUCIE to perform batch correction. This regularization is computed from the visualization layer to ensure consistency across subsampled batches. The resulting total loss is then

$$L = L_r(X, \hat{X}) + \lambda_b \cdot L_b(V).$$

The loss function of the second run then optimizes L_r along with two regularization terms L_c and L_d that together enable SAUCIE to learn clusters:

$$L = L_r(\hat{X}, \tilde{X}) + \lambda_c \cdot L_c(B) + \lambda_d \cdot L_d(B, \hat{X}).$$

The first term L_c guides SAUCIE to learn binary representations via the activations in B using a novel information dimensionality penalty that we introduce in this paper. The second term L_d encourages interpretable clusters that contain

similar points by penalizing intra-cluster distances in the cleaned batch $\hat{X}$, which is fixed for this second run.

2.2.1 MMD regularization for batch correction

A major challenge in the analysis of single-cell data is dealing with so-called batch effects, which result from technical variability between replicates of an experiment. Combining replicates often results in technical and experimental artifacts being the dominant source of variability in the data, even though this variability is entirely artificial. This experimental noise can come in the form of dropout, changes of scale, changes of location, or even more complicated differences in the distributions of each batch. It is infeasible to parametrically address all of the potential differences explicitly, such as assuming measurements are drawn from a Gaussian distribution. Instead of addressing specific models of noise, SAUCIE minimizes a distance metric between distributions. The batch correction term L_b calculates the Maximal Mean Discrepancy [7] (MMD) between batches:

$$L_b = \Sigma_{i \neq ref} MMD(V_{ref}, V_i),$$

where V_{ref} is the visualization layer of one of the replicates arbitrarily chosen to be the reference batch. MMD compares the average distance of each point to other points in its own batch with the distance to points in the other batch. MMD is zero only when two distributions are equal. Thus minimizing this metric encourages SAUCIE to align each batch. MMD has been used effectively to remedy batch effects in residual networks, but here SAUCIE uses it in a feedforward autoencoder and combines it with other tasks of interest in biological exploratory data analysis [8].

The choice of reference does not affect the degree to which two distributions can be aligned, but a reference batch is necessary because the encoding layers of a standard network will be encouraged to embed different batches in different places in the visualization layer. It does this because the decoder is required to make its reconstruction $\hat{X}$ match the original data in X (which includes the batch effects). To remedy this, SAUCIE's decoder is required to reconstruct the reference batch exactly as usual, but all other batches must only be reconstructed to preserve the pairwise distances between points in the batch. Consequently, the MMD regularization term will be minimized when batches are aligned, and the decoder need only be able to reconstruct the exact values of the reference batch and the *relative values* of the non-reference batches. The non-reference batches will be aligned to the reference batch in such a way as to preserve its internal structure as best as possible.

2.2.2 Information dimension regularization for clustering

We consider the task of clustering data points by interpreting the sparse layer B in the network as encoding cluster assignments. We note that common activation functions that are used to introduce nonlinearities in neural networks (including

SAUCIE), such as hyperbolic tangent or sigmoid, provide a natural threshold for binarizing neuron activation to be either zero or one. Indeed, such functions are monotone and asymptotically tend to constant upper and lower bounds away from the origin. Therefore, we use their inflection point, or the midpoint between the upper and lower bounds, as a threshold to binarize the activations in B. This results in an interpretable clustering layer that creates 'digital' cluster codes out of an 'analog' encoder layer, thus providing a binary code for each input point of the network. These binary codes are in turn used as cluster identifiers in order to group data points with the same code into a single cluster.

In order to automatically learn an appropriate granularity of clusters, we developed a novel regularization that encourages near-binary activations and minimizes the information (i.e., number of clusters) in the clustering layer. Our regularization is inspired by the von Neumann (or spectral) entropy of a linear operator [9], which is computed as the Shannon entropy of their normalized eigenvalues [10, 11]. This entropy serves as a proxy for the numerical rank of the operator [12], and thus provides an estimation of the essential dimensionality of its range. In our case, we extend this notion to the nonlinear transformation of the neural network by treating neurons as our equivalent of eigenvalues, and computing the entropy of their total activation over a batch. We call this entropy 'information dimension' (ID) and the corresponding ID regularization aims to minimize this entropy while still encoding sufficient information to allow reconstruction of the input data points.

The ID regularization is computed from the clustering layer activations in B by first computing the activation of each neuron j as $a_j = \sum_{i=1}^{n} B_{ij}$, then normalizing these activations to form an activation distribution $\vec{p} = \vec{a}/\|\vec{a}\|_1$, and finally computing the entropy of this activation distribution as

$$L_c(B) = -\sum_{j=1}^{k} p_j \log p_j.$$

By penalizing the entropy of neuron activations, this regularization encourages a sparse and binary encoding. This counters the natural tendency of neural networks to maximize the amount of captured (i.e., encoded) information by spreading activations out across a layer evenly. By forcing the activations to be concentrated in just a few distinct neurons, different inputs end up being represented with rather similar activation patterns, and thus naturally clustered. When combined with the reconstruction loss, the network will retain enough information in the sparse layer for the decoder to reconstruct the input, keeping similar points in the same cluster.

2.2.3 Intracluster distance regularization

SAUCIE learning digital codes creates an opportunity to interpret them as clusters, but these clusters wouldn't necessarily be comprised of only similar points. To emphasize that inputs only be represented by the same digital code if they are similar to each other, SAUCIE also penalizes intracluster pairwise

distances. Beyond suffering reconstruction loss, using the same code for points that are far away from each other will now incur an even greater loss.

This loss is calculated as the euclidean distance between points with the same binary code:

$$L_d(B, \hat{X}) = \sum_{i,j:b_i=b_j} \|\hat{x}_i - \hat{x}_j\|^2$$

where $\hat{x}_i, \hat{x}_j$ and b_i, b_j are the i-th and j-th rows of $\hat{X}$ and B, respectively.

Since ID regularization is minimized by using the same code to represent all inputs, this term acts as an opposing balance. Intracluster distances are minimized when all points are in a cluster by themselves. Together with the reconstruction penalty, these terms encourage SAUCIE to learn clusters that are composed of as many points as possible that are near to each other.

An additional benefit of clustering via regularization is that not only does the number of clusters not needed be set *a priori*, but by changing the value of λ_c the level of granularity of the clustering can be controlled, so both coarse clustering and fine clustering can be obtained to further add insight into the underlying structure of the data.

2.3 Experimental setting

We test SAUCIE on an artificial dataset where the ground truth is known, and on two mass cytometry datasets and two single-cell RNA-sequencing datasets. The mass cytometry datasets come from patients with the dengue and a healthy control patient while the single-cell RNA-sequencing datasets are two mouse brain embryo datasets, one from Yale University and the other the openly available 1.3 million cell 10xGenomics Megacell dataset [1]. Details on the training process can be found in the methods section.

2.3.1 Validation on artificial data

The artificial data tests both SAUCIE's ability to correct batch effects and cluster. It is comprised of a set of 10 separable 100-dimensional multivariate Gaussians divided into batches. For the first experiment, these 10 Gaussians were divided into 5 batches. This tests the case where there are two underlying Gaussian populations, but batch effects have shifted the measurements so that the two populations look different in each batch. SAUCIE's goal would be to align all 5 batches so that the true underlying two populations emerge via clustering. Figure 2 shows the two-step process of first aligning the batches by running SAUCIE with MMD regularization and then using the reconstructed values of that run as the input to another run through SAUCIE with ID regularization.

After successfully aligning and clustering a simple underlying population of only two Gaussians obfuscated by batch effects, the next test creates a more

[1] https://support.10xgenomics.com/single-cell-gene-expression/datasets

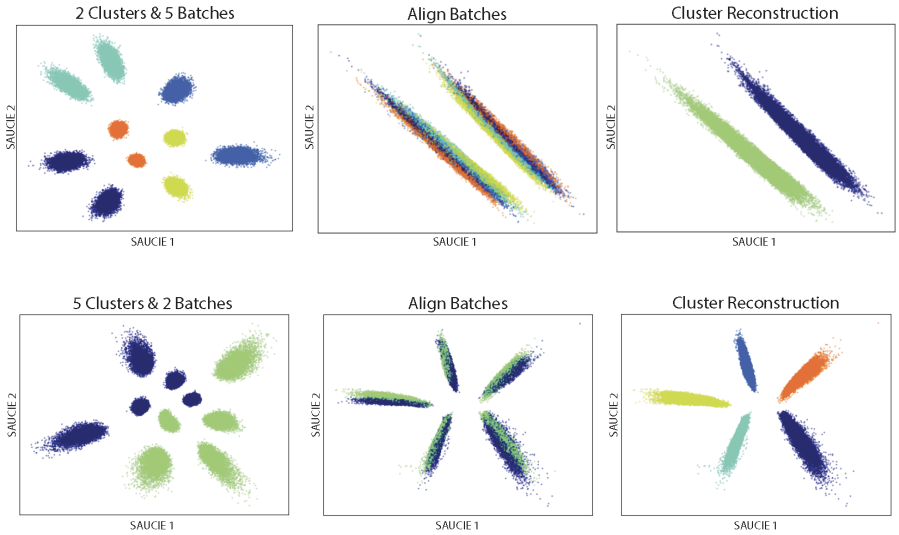

Figure 2: SAUCIE first aligns the clusters of each batch with MMD regularization, and then successfully recovers the underlying clusters from the reconstruction with ID regularization.

complex underlying population by dividing the 10 Gaussians into 2 batches each with 5 Gaussians. Figure 2 shows SAUCIE correct the batch effects and successfully cluster this more complex scenario, as well.

The ID regularization is encouraging SAUCIE to organize the activations of its neurons as intended, as the histogram of the activations in the clustering layer shows in Figure 3. After transforming the hyperbolic tangent's activations from [-1,1] to [0,1] by a shift of positive 1 and a division by 2 and using a threshold of .5, all activations are either near 0 or near 1. ID regularization achieves this while neither no regularization nor penalizing the L1-norm of the activations [11] does.

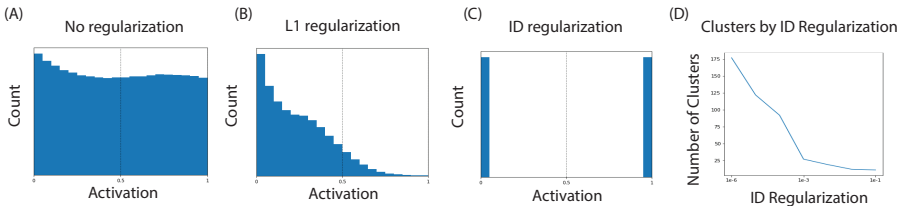

Figure 3: ID regularizations encourage activations near 0 or near 1 more than no regularization or L1-regularization.

	Before SAUCIE	After ComBat	After SAUCIE
$MMD^2(Batch_1, Batch_2)$	1.21467	0.02742	0.00004
10-kNN in same batch	99.58%	81.44%	50.51%
30-kNN in same batch	98.48%	81.88%	50.18%
50-kNN in same batch	98.42%	70.36%	50.49%

Table 1: SAUCIE corrects the batch effects more thoroughly than ComBat.

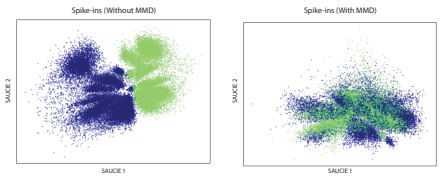

Figure 4: Left: batch effects separate the two batches. Right: SAUCIE learns to align them.

2.3.2 Cytometry: dengue patients

We examine SAUCIE's batch correction on real biological data from mass cytometry of peripheral blood mononuclear cells (PBMC) from dengue patients that were introduced (in vitro) to strains of the Zika virus obtained from the lab of Dr. Ruth Montgomery at Yale University. The measurements were done in multiple runs both in the same day and across multiple days and have large batch effects (Figure 4). Table 1 quantifies the batch effect between two sets of 20,000 cells from distinct batches. One metric is MMD while the other is measuring on average how many of each cell's nearest neighbors belong to the same batch, for different values of k. When the batches are perfectly mixed, 50% of each cell's nearest neighbors should belong to the same batch, while when the batches are completely separate, nearly 100% will be. SAUCIE is compared to a leading tool used for batch correction, ComBat [13]. SAUCIE outperforms ComBat at aligning these samples by all metrics.

SAUCIE learned a mapping that aligns the data from different runs. As a preprocessing step, the output of this batch correction can be sent through other models, including SAUCIE with ID regularization to obtain clusters.

As an initial exploratory step, SAUCIE's embedding can also be visualized. The ID regularization has an impact on the embedding, creating more distinct and separate clusters as it increases (Figure 5).

For the clustering considered here, we use a coarse-grained clustering obtained with a high coefficient for ID regularization (0.5). If instead other granularities were desired, lower coefficients could be used and the impact of this parameter on the number of clusters is shown in Figure 3(D).

Figure 6(A) details the mean expression of each channel z-scored by cluster, which can then be used for further biological analysis. For example, clusters

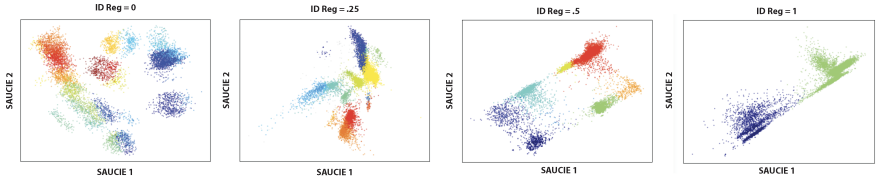

Figure 5: The effect on the embedding of increasing ID reg.

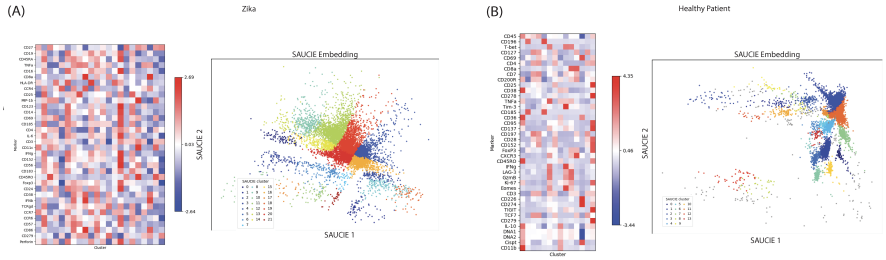

Figure 6: Heatmaps of SAUCIE's clusters on cytometry data.

have markers associated with different kinds of cells. Cluster 2, 6, and 14 are CD14+, consistent with monocytes. Clusters 3 and 18, being CD45RO+, are consistent with memory Tregs. The highly HLA-DR+ of cluster 10 point to activated Tregs, meanwhile. These cluster associations can help with identifying cells of interest for more investigation.

2.3.3 Cytometry: healthy patients

Another cytometry dataset comes from a healthy patient used as part the control process for clinical trials from Dr. Susan Kaech at Yale University. SAUCIE picks out clusters that look coherent as viewed on a TSNE embedding, similar to those of another clustering method, Louvain. One heuristic for measuring the general quality of a clustering is modularity. Louvain directly optimizes for this, which may not lead to the most desirable result. However, any reasonable clustering would have relatively high modularity, albeit not necessarily the highest. It is notable that SAUCIE's clusters have high modularity without explicitly targeting this in its task.

The heatmap showing mean expression profiles of each cluster can then be used for further exploratory analysis, for example seeing which markers are correlated, leading to the next steps of biological inquiry.

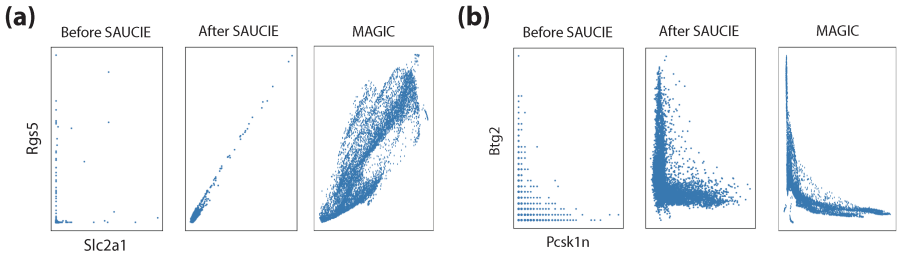

Figure 7: Relationships between variables lost to dropout are restored in SAUCIE's reconstruction.

Modularity	SAUCIE	Louvain
Dengue Patients	.85	.96
Healthy Patients	.77	.92
Mouse Brain Embryo	.78	.93
10xGenomics Megacell	.90	.92

Table 2: Louvain directly optimizes modularity, a heuristic that may not lead to the best clustering. SAUCIE's clusters are still reasonable and coherent by this metric despite a completely independent optimization procedure.

2.3.4 scRNA-seq: Mouse brain embryo

SAUCIE's ability to restore variable relationships lost to noise is explored in a dataset of cells from mouse brain embryos collected by Dr. James Noonan at Yale University. Single-cell RNA-seq data is afflicted with abundant dropout, where variables are erroneously measured at 0 due to instrument-induced noise. Dropout obscures true relationships between variables, but because SAUCIE learns the low-dimensional manifold and reconstructs smoothed data, these relationships can be recovered. Figure 7 illustrates that SAUCIE projects the anomalous points back to the manifold and restores the relationships that can be seen from non-zero points.

2.3.5 scRNA-seq: 10xGenomics Megacell

Finally, we consider the 1.3 million cells from the 10xGenomics Megacell dataset. The size of this publicly available dataset does not present a problem for SAUCIE, and as with the previous data, the clustering results look coherent as visualized on a TSNE embedding. As measured by modularity, SAUCIE almost matches Louvain's value despite not explicitly targeting the heuristic in any way.

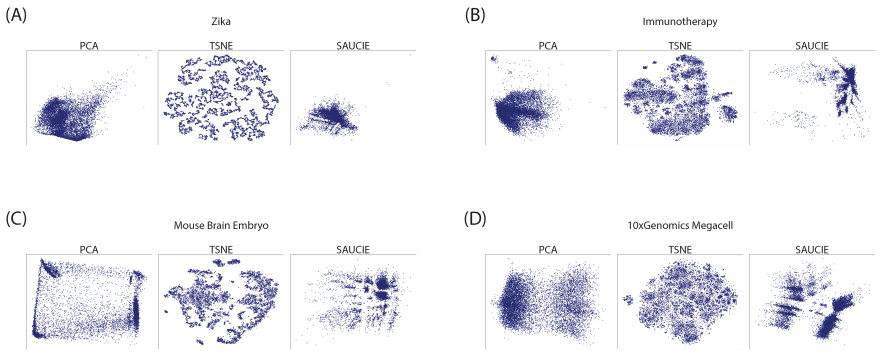

Figure 8: A comparison of three different visualization methods on the cytometry datasets (A) and (B) and the single cell RNA-seq datasets (C) and (D).

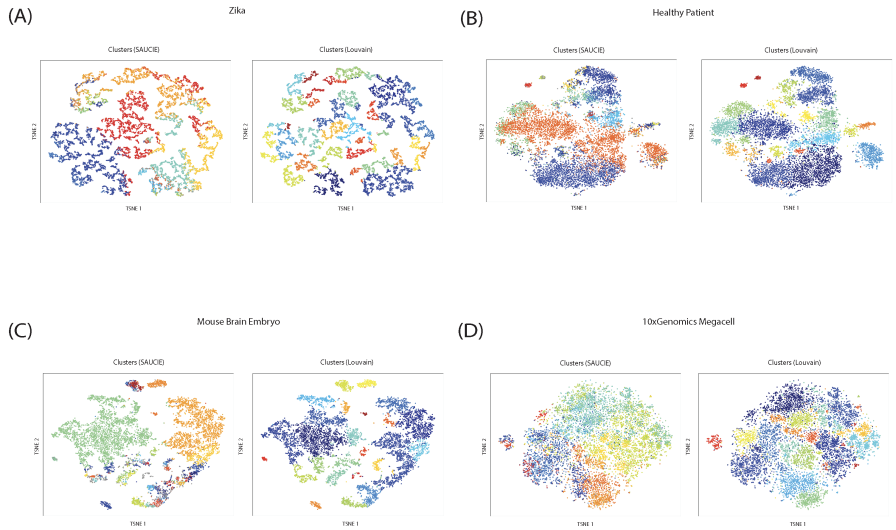

Figure 9: A comparison of visualizations of SAUCIE's clusters and Louvain's clusters on TSNE's embedding.

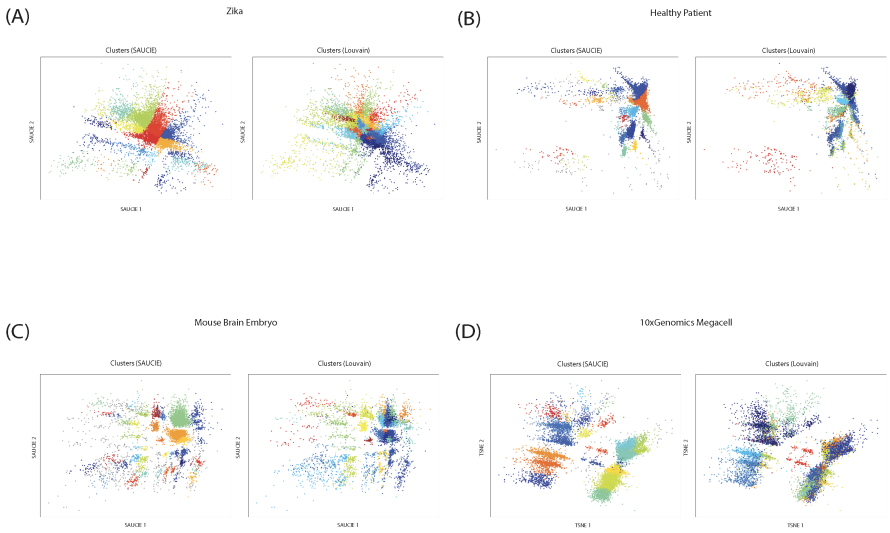

Figure 10: A comparison of visualizations of SAUCIE's clusters and Louvain's clusters on SAUCIE's embedding.

3 Discussion

Deep neural networks have been shown to be effective in processing massive datasets such as in the context of images and search engines. Here we apply, for the first time, a deep learning approach in an unsupervised fashion to large-scale single-cell data. As single-cell datasets become larger both in terms of cells and experimental samples (batches), scalability of analysis methods becomes key. The GPU-based parallelizable approach that training of neural networks offer makes deep learning especially suited for finding structure in single-cell data. We have presented SAUCIE, a new autoencoder framework, that performs four key tasks in single-cell data analysis: 1) data imputation, 2) clustering, 3) batch correction, 4) visualization. These tasks are normally performed by separate algorithms which often entail re-representation as graphs or diffusion operators which may not scale. Additionally, different methods chosen for the different tasks may make incompatible assumptions as a result of the different internal representations. SAUCIE shows the versatility of neural networks in being able to perform these varied tasks and detect emergent patterns in biological data based on novel regularizations simultaneously, thus operating on a single representation of the data. Each of the four analyses outcomes are thus guaranteed to reveal structure that is compatible between the tasks, e.g. the 2D visualization will show the same clusters that the clustering method finds. While these four tasks present the main types of analysis currently performed on single-cell data, we

believe that using our framework, and neural networks in general, additional tasks can be performed on the data, such as the inference of gene logic, artificial generation of data, and others.

4 Methods

4.1 Autoencoders

In vanilla autoencoders, the network only minimizes reconstruction loss. Minimization is done by backpropagating gradients stochastically on minibatches of input examples one at a time. Unlike SAUCIE's regularizations, typical autoencoders only use information from one point at a time. SAUCIE leverages that minibatches act as samples from the whole data space, and thus sample statistics can be calculated across individual inputs in a minibatch and included in the loss. This gives access to a richer set of functions than regularizations which are calculated from each input point independently.

4.2 Training

Training was performed with minibatches of 256, mean-squared-error for the reconstruction error function, and the optimizer chosen is ADAM with learning rate .001. All regularizations are applied to the 256-dimensional encoder layer, which uses a hypberbolic tangent activation. The 2-dimensional embedding layer uses a linear activation, while all other layers use a leaky rectified linear activation.

5 Software

SAUCIE is written in Python using the Tensorflow library for deep learning. The source code is available at `https://github.com/KrishnaswamyLab/SAUCIE/`.

References

[1] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828, 2013.

[2] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[3] Wei Wang, Yan Huang, Yizhou Wang, and Liang Wang. Generalized autoencoder: A neural network framework for dimensionality reduction. In *CVPR Workshops*, 2014.

[4] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[5] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICML*, 2014.

[7] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015.

[8] Uri Shaham, Kelly P. Stanton, Jun Zhao, Huamin Li, Khadir Raddassi, Ruth Montgomery, and Yuval Kluger. Removal of batch effects using distribution-matching residual networks. *Bioinformatics*, 33(16):2539–2546, 2017.

[9] Kartik Anand, Ginestra Bianconi, and Simone Severini. Shannon and von neumann entropy of random networks with heterogeneous expected degree. *Physical Review E*, 83(3):036109, 2011.

[10] Devansh Arpit, Yingbo Zhou, Hung Q. Ngo, and Venu Govindaraju. Why regularized auto-encoders learn sparse representation? In *ICML*, 2016.

[11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011.

[12] Kevin R Moon, David van Dijk, Zheng Wang, William Chen, Matthew J Hirn, Ronald R Coifman, Natalia B Ivanova, Guy Wolf, and Smita Krishnaswamy. Phate: A dimensionality reduction method for visualizing trajectory structures in high-dimensional biological data. *bioRxiv*, page 120378, 2017.

[13] W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007.