

Exploring Single-Cell Data with Deep Multitasking Neural Networks

Matthew Amodio^{†,1}, David van Dijk^{†,1,2}, Krishnan Srinivasan^{†1},
William S Chen⁶, Hussein Mohsen⁴, Kevin R. Moon^{2,3},
Allison Campbell⁶, Yujiao Zhao⁵, Xiaomei Wang⁵,
Manjunatha Venkataswamy⁸, Anita Desai⁸, Ravi V.⁸,
Priti Kumar⁷, Ruth Montgomery⁵, Guy Wolf^{†,3,2,1},
Smita Krishnaswamy^{†,*,2,1}

¹Department of Computer Science

²Department of Genetics

³Applied Mathematics Program

⁴Computational Biology and Bioinformatics

⁵Department of Rheumatology

⁶School of Medicine

⁷Department of Microbial Pathogenesis

Yale University, New Haven, CT, USA

⁸Department of Neurovirology, NIMHANS, Bangalore, India

*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

Biomedical researchers are generating high-throughput, high-dimensional single-cell data at a staggering rate. As costs of data generation decrease, experimental design is moving towards measurement of many different single-cell samples in the same dataset. These samples can correspond to different patients, conditions, or treatments. While scalability of methods to datasets of these sizes is a challenge on its own, dealing with large-scale experimental design presents a whole new set of problems, including batch effects and sample

11 comparison issues. Currently, there are no computational tools that can both handle large
12 amounts of data in a scalable manner (many cells) and at the same time deal with many
13 samples (many patients or conditions). Moreover, data analysis currently involves the use
14 of different tools that each operate on their own data representation, not guaranteeing a
15 synchronized analysis pipeline. For instance, data visualization methods can be disjoint
16 and mismatched with the clustering method. For this purpose, we present SAUCIE, a deep
17 neural network that leverages the high degree of parallelization and scalability offered by
18 neural networks, as well as the deep representation of data that can be learned by them to
19 perform many single-cell data analysis tasks, all on a unified representation.

20 A well-known limitation of neural networks is their interpretability. Our key contribu-
21 tion here are newly formulated regularizations (penalties) that render features learned in
22 hidden layers of the neural network interpretable. When large multi-patient datasets are fed
23 into SAUCIE, the various hidden layers contain denoised and batch-corrected data, a low
24 dimensional visualization, unsupervised clustering, as well as other information that can
25 be used to explore the data. We show this capability by analyzing a newly generated 180-
26 sample dataset consisting of T cells from dengue patients in India, measured with mass
27 cytometry. We show that SAUCIE, for the first time, can batch correct and process this
28 11-million cell data to identify cluster-based signatures of acute dengue infection and cre-
29 ate a patient manifold, stratifying immune response to dengue on the basis of single-cell
30 measurements.

31 1 Introduction

32 Vast amounts of high-dimensional, high-throughput, single-cell data measuring various aspects
33 of cells including mRNA molecules, proteins, epigenetic marks and histone modifications are
34 being generated via new technologies. Furthermore, the number of samples included in large-
35 scale studies of single-cell data for comparing across populations or disease conditions is rapidly
36 increasing. Processing data of this dimensionality and scale is an inherently difficult prospect,
37 especially considering the degree of noise, batch effects, artifacts, sparsity and heterogeneity
38 in the data [1, 2]. However, this effect becomes exacerbated as one tries to compare between
39 samples, which themselves contain noisy heterogeneous compositions of cellular populations.

40 Deep learning offers promise as a technique for handling the size and dimensionality of
41 modern biological datasets. However, while work has been done in training networks to per-
42 form certain supervised tasks such as predicting binding [3, 4] or classifying patients [5], deep
43 learning has been underutilized for unsupervised exploratory tasks. In this paper, we develop a
44 deep learning framework that focuses on unsupervised data exploration. Our key insight is that
45 the layers of a deep neural network form representations of the data, and that if those layers are
46 properly constrained (via architectural choices and regularization), they can be used to extract
47 task-oriented features of the data.

48 We base our approach on the *autoencoder* [6–8]. An autoencoder is a neural network that
49 learns to recreate its own input via a low-dimensional bottleneck layer that learns representa-

50 tions of the data and enables a denoised reconstruction of the input from them [9–13]. Since
51 autoencoders learn their own features, they can reveal structure in the data without defining or
52 explicitly learning a similarity or distance metric in the original data space as other dimension-
53 ality reduction methods do (for instance, PCA uses covariance and diffusion maps [14] utilize
54 affinities based on a kernel choice). We use this approach to construct SAUCIE, a Sparse Au-
55 toencoder for Unsupervised Clustering, Imputation, and Embedding, which is aimed to enable
56 exploratory tasks via its design choices.

57 SAUCIE is a multilayered deep neural network, whose input layer is fed single-cell mea-
58 surements, such as mass cytometry or single-cell RNA sequencing, of an individual cell. Then,
59 SAUCIE gradually reduces the dimensionality of the dataset by taking the data through nar-
60 rower and narrower hidden layers. We see that the output or reconstruction layer of SAUCIE
61 gives similarly denoised and imputed data as the manifold denoising method MAGIC [15] on a
62 1.3 million single-cell RNA sequencing dataset from embryonic mouse brain. In other words,
63 SAUCIE effectively learns the manifold of the data in a similar way to data diffusion [16]
64 methods. Thus, SAUCIE can leverage the power of manifold learning, which has shown to be
65 key for analyzing single-cell data [17] in a scalable fashion. Manifold learning methods are
66 traditionally difficult to scale due to the computational complexity of kernel computation and
67 eigendecomposition operations. Deep learning comes to the rescue here by being amenable to
68 GPU speedup and parallelization of matrix operations.

69 As SAUCIE reduces input dimensionality, regularizations on different layers reveal differ-
70 ent representations of the data: for visualization, batch correction, clustering, and denoising.
71 In order to achieve these representations we use customized regularizations in each layer. We
72 use the architectural choice of having a two-dimensional bottleneck layer to provide a *visual-*
73 *ization* of the data. We develop a novel batch-level maximal mean discrepancy (MMD)-based
74 penalty constraint to *remove batch effects* in the embedding layer. A customized sparse encod-
75 ing layer featuring our novel information-dimension (ID) regularization provides an *automated*
76 *clustering* of the data with no parametric assumptions on the shape or number of clusters. All
77 regularizations balance against reconstruction accuracy, which is the basic penalty in an au-
78 toencoder that steers the network convergence away from trivial solutions. Furthermore, this
79 penalty ensures that the final layer of the network provides reconstructed measurements that
80 are *denoised*; in the case of single-cell RNA sequencing data, this layer also naturally *imputes*
81 *missing values*.

82 Guiding the internal representations of the data to be effective at each of these disparate
83 tasks together fit SAUCIE into the field of multitask learning. Results in multitask learning have
84 generally shown that optimizing multiple tasks over the same latent representation is helpful in
85 increasing the reliability and consistency of various algorithms. We apply the same approach
86 here by having the representation (or data manifold) learned by SAUCIE be jointly optimized
87 for multiple tasks. Further, SAUCIE itself forms a near complete analysis of the data. The
88 clustering layer in SAUCIE for instance, actually performs clustering, and clusters are read out
89 from this layer. This is in contrast to other methods that simply use the autoencoder for coming
90 up with a reduced dimensional representation, which is then fed to other (generally unscalable)

91 algorithms, for example scVI which outputs a latent layer that then needs another clustering
92 algorithm [18].

93 We apply SAUCIE to a twenty-million cell mass cytometry dataset with 180 samples from
94 forty subjects in a study of the dengue flavivirus [19]. SAUCIE is the only method that is able to
95 batch correct 180 samples and then cluster them in such a way that subpopulation proportions
96 become comparable prima facia. This obviates the need for approaches such as first clustering
97 samples separately and then performing “meta-clustering” as with the Phenograph method, or
98 other methods that cannot operate uniformly on combined data of this size (the problems of
99 which are illustrated in Figure S10). We are also able to tune the granularity of clustering with
100 SAUCIE in order to get a clustering that is informative of the differences between conditions.
101 SAUCIE results show that acute subjects are characterized by enrichment in distinct subpopula-
102 tions of CD4-CD8- $\gamma\delta$ T cells and cells involved in Type I interferon signaling. When subjects
103 are measured in convalescence, there is an increase in CD4+Foxp3+ T reg cells.

104 Thus, SAUCIE provides a unified representation of data where different aspects or features
105 are emphasized in different layers, forming a one-step data analysis pipeline. This unified
106 analysis uncovers a cell-space manifold as well as a sample-space manifold, thus enabling a
107 multilevel analysis of complex experimental design where the samples are stratified on the basis
108 of their cell-level features. We additionally evaluate SAUCIE extensively on all of its designed
109 tasks using ten public single-cell datasets.

110 **2 Results**

111 **2.1 The SAUCIE Architecture and Layer Regularizations**

112 To enable unsupervised learning in a scalable manner, we base our method on the autoencoder.
113 Autoencoders learn to recreate their input at the output layer, but via a low-dimensional infor-
114 mational bottleneck layers which are forced to learn meaningful structure-preserving represen-
115 tations of the data. However, a key challenge is to extract meaning from this representation.
116 Specifically, we seek representations in hidden layers that are useful for performing the various
117 analysis tasks associated with single cell data. Here, we introduce several design decisions and
118 novel regularizations to our autoencoder architecture (Figure 1) in order to constrain the learned
119 representations for four key tasks:

- 120 1. visualization and dimensionality reduction,
- 121 2. batch correction,
- 122 3. clustering, and
- 123 4. denoising and imputation.

124 For each task, dedicated design decisions are used to produce the desirable result.

125 **Clustering:** First, to cluster the data, we introduce the *information dimension* regularization
126 that encourages activations of the neurons in a hidden layer of the network to be binarizable.
127 The idea is that if we can obtain a “digital” binary encoding, then we can easily turn these
128 codes into clusters. As Figure 2A shows, the network without regularizations tends to store its
129 information in a distributed, or “analog” way. With the ID regularization the activations are all
130 near 0 or 1, i.e., binary or “digital”, and thus amenable to clustering by simple thresholding-
131 based binarization. As seen in Figure 3A, this leads to a clustering of the cells that effectively
132 represents the data space. Thus, the ID regularization achieves an analog-to-digital conversion
133 that enables interpretation of the representation as data groups or clusters corresponding to each
134 binary code. A previous work in the same vein, Binary Connect, has shown the promise in en-
135 couraging networks to learn in ways that are easy to binarize. That work differs from SAUCIE
136 though, in that they learn binary weights rather than binary activations, along with the goal
137 being to improve computational efficiency rather than achieve a clustering of the data [20].

138 **Batch Correction:** Batch effects are generally systematic differences found in biological data
139 measured under different experimental runs, largely due to ambient conditions such as temper-
140 ature, machine calibration or day-to-day variation in measurement efficiency. Thus, measure-
141 ments even from very similar systems, such as blood cells of the same patient, appear to have
142 a shift or difference between two different experimental runs. To solve this problem, we in-
143 troduce a maximal mean discrepancy (MMD) correction that penalizes differences between the
144 probability distributions internal activations of samples. Previous work has attempted batch cor-
145 rection by minimizing MMD. However, those models assume that batch effects are minor and
146 simple shifts close to the identity function, which is often the case [21]. Moreover, minimizing
147 MMD alone only removes any and all differences between batches. In contrast, the additional
148 autoencoder reconstruction penalty in SAUCIE forces it to preserve the original structure in
149 each batch, balancing the goals of, on one hand, making the two batches alike while on the
150 other hand not changing them. We note that this notion of a biological batch (data measured or
151 run together) is distinct from the mini-batches used in stochastic gradient descent to train neural
152 networks and the two should not be confused. The term batch is exclusively used to describe
153 biological batches and when training with stochastic gradient descent the term mini-batches is
154 used.

155 Figure 4 shows that analyzing data before batch correction can lead to misleading results,
156 as artificial variation from batch effects can drown out the relevant variation within the biology
157 that we are interested in. Penalizing MMD directly on the input space would be a flawed way
158 of addressing batch effects because it would require making the assumption of (and thus being
159 sensitive to the choice of) meaningful distance and similarity measures on the input points.
160 Since the data is noisy and possibly sparse, by instead penalizing MMD on an internal layer
161 of the network, we can correct complex, highly nonlinear batch effects by aligning points on a
162 data manifold represented in these layers.

163 **Imputation and denoising** Next, we leverage the fact that an autoencoder does not recon-
164 struct its input exactly, but instead must learn a lower dimensional representation of the data,
165 and decode this representation for data reconstruction. This means the reconstructions are de-
166 noised versions of the input and are thus naturally solutions to the dropout and other noise
167 afflicting much real-world data, especially single cell RNA-sequencing data. The gene-gene
168 relationships plotted in Figure 3C illustrate the ability of SAUCIE to recover the meaningful
169 relationship between genes despite the noise in the data.

170 **Visualization** Finally, we design the informational bottleneck layer of the autoencoder to be
171 two dimensional, which lets it serve as a visualization and nonlinear embedding of the data.
172 Because the network must reconstruct the input accurately from this internal representation, it
173 must compress all the information about a cell into just these two dimensions, unlike methods
174 like PCA or Diffusion Maps, which explicitly leave some variation unmodeled. Consequently,
175 the information stored is also global, meaning points close together in the SAUCIE visualization
176 are more similar than points that are farther apart, which is not true beyond small neighborhoods
177 in a local method like tSNE. The ability to flexibly learn and accurately reflect the structure in
178 the data with SAUCIE is demonstrated in Figure 3B.

179 Considered together, these customized regularizations and architectural choices make SAUCIE
180 ideally suited for the exploratory data analysis when presented with single-cell biological data.
181 Further, SAUCIE is entirely self-contained and not require any external algorithms that may not
182 be able to process the scale of multisample single-cell data.

183 2.2 Comparison to other methods

184 We begin by offering an extensive comparison between SAUCIE and other (generally special-
185 ized) methods at each of these tasks in turn. We find that SAUCIE performs as well as, or
186 even better than, specialized algorithms, which are much less scalable, for each individual task.
187 Moreover, SAUCIE performs all tasks on a unified representation leading to visualizations that
188 are coherent with clusters and cluster expression.

189 Throughout the comparisons on each of the tasks, we use two artificial datasets (simulation
190 from mixtures of Gaussians and the canonical MNIST handwritten digit dataset), along with
191 ten different single-cell datasets. Five datasets are CyTOF: the dengue dataset we extensively
192 evaluate later in the manuscript, T cell development data from [22], renal cell carcinoma data
193 from [23], breast tumor data from [24], and iPSC data from [25]. Five datasets are scRNA-seq:
194 mouse cortex data, retinal bipolar cells from [26], hematopoiesis data from [27], mouse brain
195 data from [28], and the 10x mouse megacell demonstration from [29].

196 2.2.1 Clustering

197 To evaluate the ability of SAUCIE to find meaningful clusters in single-cell data, we compare
198 it to several alternative methods: minibatch kmeans [30], Phenograph [31], and another neural

199 network approach called Single-cell Variational Inference (scVI) [18]. While we compare to
200 scVI as it and SAUCIE are both neural networks, we emphasize a fundamental difference be-
201 tween the two: scVI only returns a latent space, which must then be visualized or clustered by
202 another outside method, while SAUCIE explicitly performs these tasks. Since kmeans needs to
203 be told how many clusters there are ahead of time (k), we use the number of clusters identified
204 by Phenograph as k. We look at the following datasets: MNIST handwritten digits for which
205 there are ground truth labels, artificially generated Gaussians rotated into high dimensions, and
206 public single-cell datasets for which we have curated cell clusters as presented by the authors:
207 [26], [23], [28], [27], and [22].

208 In addition to analyzing the clusters visually (Figure S2), we also quantitatively assess clus-
209 ter performance of the methods by computing modularity and silhouette scores [30] on the
210 generated clusters and ground truth labels (Table 1). For MNIST, we find that just as we would
211 expect given they are both non-Euclidean clustering methods that do not need a specified num-
212 ber of clusters, SAUCIE and Phenograph are the most comparable, with their having the high-
213 est modularities, similar silhouette scores, and very similar visual appearance. Next, we look at
214 an artificially generated dataset of four two-dimensional Gaussian point clouds with different
215 means rotated into 100 dimensions. We find that SAUCIE is the only method that automatically
216 identifies exactly four clusters, which was the underlying number of clusters in the generation
217 model. This illustrates why optimizing modularity, like Phenograph does, is not necessarily the
218 best heuristic to follow, as it adds additional complexity to the clustering in order to increase
219 the modularity score, resulting in too many clusters. Likewise, scVI did not identify the four
220 clusters, which is unsurprising as the data did not fit its parametric model appropriate for gene
221 counts.

222 We also examine clustering performance on five public single-cell datasets to evaluate the
223 ability of SAUCIE to cluster real biological data: from [26], [23], [28], [27], and [22].
224 Visual inspection reveals that SAUCIE produces clusters that are qualitatively coherent on the
225 embedding. Quantitatively, the modularity scores of its clusters corroborate this evaluation. As
226 shown in Table 1 the average modularity score across datasets is 0.8531. In a wide variety of
227 data from both CyTOF and scRNA-seq measurements, SAUCIE is able to produce clusters that
228 reasonably represent the data qualitatively, quantitatively, and by comparison to other methods.

229 2.2.2 Batch correction

230 We assess our ability to remove batch-related artifacts with SAUCIE by comparison to two
231 published batch correction methods that have been specifically designed to remove batch ef-
232 fects in single-cell data. The first, Mutual Nearest Neighbors (MNN) [32], uses mutual nearest
233 neighbors on a k-nearest neighbors graph to align two datasets, and the second, Canonical Cor-
234 relation Analysis [33], finds a latent space in which the two batches are aligned. To evaluate
235 the performance of these methods and SAUCIE, we use several different datasets with varying
236 degrees of batch artifacts. We note that SAUCIE is the only method capable of scaling batch
237 correction to hundreds of samples as we do in the next section. Nonetheless, here we compare

238 performance on datasets small enough for the alternative methods to handle.

To quantitatively assess the quality, we apply a test we term the *mixing score* (similar to that of [34]):

$$mixing\ score = \frac{N_{b_1}}{N_{b_2}} * \sum_{x_j \in KNN(x_i)} (\mathbb{1}_{batch(x_i)=batch(x_j)}) \quad (1)$$

239 where N_{b_1} and N_{b_2} are the number of points in the first and second batch respectively. This
240 score calculates for each point the number of nearest neighbors that are in the same batch as
241 that point, accounting for the difference in batch sizes. In perfectly mixed batches, this score
242 is 0.5, while in perfectly separated batches it is 1.0. As batch correction should not only mix the
243 batches but also preserve their shape as best as possible, we quantify the distortion between the
244 original and batch corrected data using Procrustes, which finds the error between the optimal
245 alignments of the two batches by linear transformation [35]. These numbers are reported in
246 Table 2. While the other methods each have some datasets that violate their assumptions and
247 thus they perform poorly, SAUCIE performs as well or better at each of the wide variety of
248 datasets.

249 First, we generated two batches, each consisting of two ten-dimensional Gaussian point
250 clouds with different means. We then rotated this into 1000 dimensions to simulate realistic
251 single-cell data. Visual inspection shows that CCA appears to align the batches (i.e., the batch
252 label is well mixed), however it distorts the original shape of the data, creating more distinct
253 clusters per batch than originally existed. MNN pulls the batches closer together but does not
254 fully mix them. SAUCIE appears to successfully align the two batches while at the same time
255 preserving the original data structure shape without distortion. SAUCIE scores as well as the
256 alternative methods at the mixing score, while only SAUCIE can easily scale this performance
257 to hundreds of batches.

258 Next, we look at the CyTOF measurements of spike-in data where the same blood sample
259 has been measured twice on different days. Since they are technical replicates, the difference
260 between them confirms that there are batch effects in this data that need to be corrected. We ex-
261 pect perfect alignment after batch correction. We can observe well-aligned batches for SAUCIE
262 and MNN, however CCA does not remove any batch effect. As before, SAUCIE scores well
263 both in the mixing score and the Procrustes score.

264 Then, we evaluate nontechnical replicates of scRNA-seq data from developing mouse cor-
265 tex. While the batch effect is the dominant signal in the data, we do not expect perfect align-
266 ment, as there are also possible differences between the time points that we expect to remain
267 (the two samples are from embryonic day 14.5 and 17, respectively). CCA partially aligns the
268 two batches. However, batch effect remains the strongest signal in the embedding and the shape
269 of the data has been distorted: there now appear to be more clusters than were present origi-
270 nally in the data. SAUCIE and MNN, however, well align the two batches, but like in previous
271 datasets, MNN appears to also remove much of the population structure of the data. SAUCIE
272 both preserves the original population structure of each sample and aligns them. This is also
273 reflected in the nearest neighbor values, which are 0.544, 0.689, and 0.902, respectively.

274 SAUCIE is also able to correct varying degrees of batch effect on public datasets from [23],
275 [24], and [22]. With an average mixing score of 0.518 across the datasets, SAUCIE effectively
276 aligns each different pair appropriately, due to its combination of reconstruction penalty and
277 batch correction term. Both on scRNA-seq and CyTOF data, SAUCIE can integrate different
278 samples for later downstream analysis.

279 **2.2.3 Visualization**

280 To evaluate the SAUCIE visualization and its ability to provide a faithful low-dimensional data
281 representation, we provide an extensive comparisons of this visualization to other frequently
282 used methods. We make use of artificial datasets where the underlying structure is known, as
283 well as real biological datasets that have been extensively characterized previously, so we have
284 prior understanding of the structure we expect to see in the visualization (Figure S4).

285 The first three datasets come from a continuous artificially-generated tree structure with
286 different amounts of Gaussian noise added to it. All seven of the branches are recovered by
287 SAUCIE, tSNE, and PHATE. However, without enough noise, tSNE shatters branches, mis-
288 leadingly showing them as different clusters. PCA, Monocle2, and Diffusion Maps correctly
289 display the continuous tree-like nature of the data. However, in the two dimensions that are
290 shown, they do not capture all of the branches.

291 In the tree generated using diffusion limited aggregation (DLA), we have a more compli-
292 cated tree than in the previous examples. Only SAUCIE and PHATE effectively illustrate this
293 branching structure, while PCA places spherical clouds with many branches overlapping, and
294 Monocle2 and Diffusion Maps collapse several of the branches together. tSNE shatters the
295 different branches into one or more clusters, losing the continuous nature.

296 Next, to evaluate the ability of the various embedding methods to handle intersecting mani-
297 folds, we generated a dataset of three intersecting half circles. Both SAUCIE and PCA preserve
298 the circular shape as well as the intersecting positions. The other methods either distort the
299 curvature of the data, shatter the trajectory, or remove the intersecting nature of the data.

300 To evaluate the ability of SAUCIE and the existing visualization methods to recover under-
301 lying structure we embed the MNIST dataset where there are true labels that correspond to the
302 digit each image represents. We find that these different digits are well represented by SAUCIE,
303 tSNE, and PHATE. In PCA, Monocle2, and Diffusion Maps, only some of the digits are distinct
304 in the two dimensions that are shown, with the others being erroneously blended.

305 Another dataset where we have ground truth is a synthetic Gaussian mixture model (GMM).
306 Here, four shifted Gaussians represented in the GMM dataset show the ability of each method
307 to capture the distinct clusters present in the data. Diffusion Maps collapses all of the data into a
308 single point in the two dimensions shown, while Monocle2 places the clusters closer or farther
309 to each other erroneously. Additionally, PCA, Monocle2, and Diffusion Maps do not capture the
310 spherical structure of the data. SAUCIE, tSNE, and PHATE all capture this structure effectively.

311 In [27], the authors performed an extensive characterization of hematopoiesis in mouse bone
312 marrow and identified different cell types as shown in the colors in the embedding. SAUCIE

313 produces a visualization that reflects branching structure that is consistent with PHATE. Mon-
314 ocle2 and Diffusion Maps collapse the trajectories into a single branch while tSNE shows them
315 as contiguous clusters.

316 The data from [22] describes a system of T cell development in the mouse thymus in which
317 T cells develop from CD4-CD8 double negative phenotype into double positive and then branch
318 out into CD4+/CD8- and CD4-/CD8+. We therefore expect the embedding to show a continuous
319 trajectory that then branches into two. This is the case for SAUCIE and PHATE. While tSNE
320 shows the two directions, it does not optimally show the continuous progression. PCA and
321 Monocle2 show a continuous progression but fail to show the branch point. Diffusion Maps
322 fails to accurately capture any meaningful structure at all.

323 Next we looked at the dataset of [25] with induced pluripotent stem cells that were measured
324 in CyTOF over the course of several days, denoted by different colors. We expect the time
325 points to correlate with the embedding as cells gradually change phenotype over time. We can
326 see that SAUCIE, PHATE, tSNE, and Diffusion Maps show this significant separation. PCA
327 and Monocle2 show the least separation across time.

328 In [26], we examine retinal bipolar cells, along with the different subtypes identified by the
329 authors. We expect the embedding to reflect these different populations that they identified. We
330 can see that PHATE, tSNE, and SAUCIE are able to show all of the different clusters within the
331 two dimensional embedding. PCA, Monocle2, and Diffusion Maps show some of the structure
332 but clearly do not show all of the distinctions between cell types.

333 In [28], we look at mouse neural cells, which were also accompanied by different neural
334 cell types that are reflected by different colors in the embeddings. Again we find that SAUCIE,
335 PHATE, and tSNE show all the expected cell types and that PCA, Monocle2, and Diffusion
336 Maps only capture some of the structure within the two dimensions that are shown.

337 In addition to the previous extensive qualitative evaluation, we also measure the quality of
338 the visualizations with a quantitative metric taken from [36]. In line with to their method's
339 precision and recall metrics, we compute a neighborhood around each point in both the original
340 data space and the embedding space, and compare the neighbors of each. An embedding with
341 high recall has most of a point's original-space neighbors in its embedding-space neighborhood.
342 Similarly, an embedding with high precision has most of the point's embedding-space neighbors
343 in its original-space neighborhood. As directed by the authors' algorithm, we gradually increase
344 the size of the neighborhood and report the area-under-the-curve (AUC) for the precision-recall
345 curve. These results are in Table 3, where SAUCIE has the highest average score of 0.9342,
346 averaged across all datasets.

347 **2.2.4 Imputation**

348 We analyze the SAUCIE imputation and its ability to recover missing values by implicitly in-
349 terpolating on a data manifold in several ways. First, Figure S5 shows several relationships
350 from the scRNA-seq data of the 10x mouse megacell dataset affected by severe dropout. This
351 dataset consists of 1.3 million cells, and SAUCIE was the only method in the comparison to

352 be able process the full dataset. Moreover, it was able to do this in just 44 minutes. Addition-
353 ally, because training a neural network only requires small minibatches in memory at one time,
354 we were able to do this without ever loading the entire large dataset into memory all at once.
355 Thus, to enable this comparison, we subsampled the data by taking one of the SAUCIE clusters
356 consisting of 4172 cells.

357 For this comparison, we measure against several popular imputation methods for scRNA-
358 seq data: MAGIC, which is a data diffusion based approach, scImpute, which is a parametric
359 statistical method for imputing dropouts in scRNA-seq data, and Nearest Neighbors Completion
360 (NN Completion), which is an established method for filling in missing values in a general
361 application of high-dimensional data processing.

362 In Figure S5, we show six relationships of the mouse megacell dataset for the original data
363 and the different imputation methods. We observe that the original raw data is highly sparse,
364 which can be seen by the large number of values on the axes where one of the variables is
365 exactly zero. Note that most cells have one or both genes missing. This is a problem because
366 this prevents us from identifying trends that exist between the genes. After imputation with
367 SAUCIE, we can observe that the sparse character of the data has been removed, with values
368 filled in that reveal underlying associations between the gene pairs. These associations are
369 corroborated by MAGIC, which imputes similar values to SAUCIE in each case. MAGIC
370 is a dedicated imputation tool that is widely used, so SAUCIE matching the relationships it
371 found gives confidence in the ability of SAUCIE to impute dropout effectively. The resulting
372 imputation in scImpute does not look significantly less sparse from the original and we do not
373 see continuous trends emerge. NN Completion appears to desparsify the data, but the resulting
374 trends all look similar to each other (i.e., positively correlated). This suggests that it does not
375 correctly identify the underlying trends, as we would expect different genes to have different
376 relationships. While scRNA-seq is highly sparse, the undersampling affects all entries in the
377 matrix, including the nonzero values. As such, manifold-based methods like SAUCIE and
378 MAGIC are more suited for finding these true relationships because they denoise the full dataset
379 as opposed to just filling in zeros.

380 Due to the fact that ground truth values for the missing counts in this single-cell data are
381 not known, we further test the accuracy of the imputation abilities of SAUCIE with an artifi-
382 cially constructed experiment. We first leverage the bulk RNA sequencing data of 1076 cells
383 from [37], because it accurately captures the relationships between genes due to it not being
384 sparse (as opposed to generating our own synthetic data from a parametric generating function
385 that we have the ability to choose, where we can create the relationships). We then simulate
386 increasing amounts of dropout and compare the imputed values returned by each method to the
387 true values we started with. To simulate dropout in a manner that reflects the underlying mech-
388 anisms of inefficient mRNA capture, we remove *molecules* instead of just setting values for
389 genes to zero. As a result, the level of dropout is conditional upon expression level, reflecting
390 the dropout structure of single-cell RNA sequencing data. The results are reported in Figure S6,
391 where SAUCIE compares favorably to other methods, recovering the true values accurately
392 even after as much as 99% dropout. The dataset for this experiment consisted of just 1076 cells,

393 which allowed us to compare to the methods that cannot process larger datasets, but even on
394 a dataset of this size SAUCIE gave a more than 100-times speedup over NN Completion and
395 600-times speedup over scImpute.

396 **2.2.5 Runtime Comparison**

397 In order to showcase the scalability of SAUCIE, we compare to a host of other methods on a
398 subset of our newly generated CyTOF dataset consisting of over 11 million cells existing in 35
399 dimensions. We display the runtimes of each method on a random sample of N points, with
400 $N = 100, 200, 400, 800, \dots, 11000000$ in Figure S1. For each step, the method was given a
401 timeout after 24 hours. Points where a method stopped scaling in Figure S1 are marked with
402 an ‘x’.

403 SAUCIE performs visualization, batch correction, imputation, and clustering in its run,
404 while each of the other methods only performs one of these tasks. Moreover, SAUCIE does
405 not just compute simple linear functions on the data, but instead performs complex non-linear
406 transformations in the process. Despite its complexity, it also scales very well with the ex-
407 tremely large dataset sizes, which can be further improved by simply adding more independent
408 GPUs for calculations. Each additional (relatively inexpensive) GPU can offer a near linear
409 increase in computation time, as opposed to more CPUs which offer diminishing returns in par-
410 allelizability. All experiments were run on a single machine with just one GPU, meaning these
411 results could still benefit even more from this potential for scalability. For further details on
412 how the runtime experiment was performed, see the Methods section.

413 Among the batch correction methods, there are no other methods that correct multiple
414 batches simultaneously. However even when we restrict to pairwise comparisons, SAUCIE
415 is the only method that comes close to handling this amount of data. CCA and MNN both stop
416 scaling in the tens of thousands of cells. In the group of imputation methods, scImpute and NN
417 completion also stop scaling in the tens of thousands, while MAGIC stops scaling in the hun-
418 dreds of thousands. For visualization, PCA was the only method faster than SAUCIE, which
419 is unsurprising because calculating it using fast randomized SVD is quick, but it gives a sim-
420 ple, strictly linear blurry views of the data, in contrast to SAUCIE’s nonlinear dimensionality
421 reduction. The other more complex visualization methods do not scale to these dataset sizes:
422 Diffusion Maps, PHATE, tSNE, and Monocle2 all stop scaling before even reaching the full
423 eleven million cells. For clustering, kmeans is the only one faster than SAUCIE, due to using
424 its minibatched version. However, it still assumes circular clusters in the Euclidean space and
425 comes with the intrinsic flaw that the number of clusters must be known ahead of time, which is
426 not possible in any realistic setting like ours where we are performing exploratory data analysis
427 on a large new dataset. Phenograph and scVI do not scale to the full dataset, either. Despite
428 being another neural network method, scVI cannot scale to these larger sizes because it only
429 produces a latent space that then must be clustered with another method. This requirement then
430 becomes its bottleneck, emphasizing the importance of SAUCIE performing all tasks directly
431 instead of acting as a pre-processing step for other methods.

432 SAUCIE is the only method that can efficiently batch correct, impute and denoise, visualize,
433 and cluster datasets of this size, while using a nonlinear manifold representation of the data.

434 **2.3 Analysis of immune response to dengue infection with SAUCIE**

435 Next, we demonstrate an application of SAUCIE as an important tool enabling exploratory anal-
436 ysis of a new “big” dataset that consists of single-cell CyTOF measurements of T cells from
437 45 subjects including a group acutely infected with the dengue virus and healthy controls from
438 the same endemic area [19]. While dengue is estimated to affect sixty million people yearly
439 and cause ten thousand deaths, like other tropical diseases, it remains understudied. Moreover,
440 dengue is especially challenging since there are several different serotypes with complex inter-
441 actions between them. Specifically, there are four strains that have very different characteristics.
442 While infection with a particular strain may provide some immunity towards reinfection with
443 that same strain, an antibody dependent enhancement results in faster uptake of another strain
444 upon reinfection [38]. Drugs have proven difficult to develop for dengue. Further, vaccine
445 development has also been challenging in the case of dengue. Recently, the WHO has ruled
446 that the dengue vaccine of Senofi Pasteur only be administered to patients who are infected
447 for the second (or subsequent) time [39]. This is because the vaccine itself is thought to leave
448 patients vulnerable to very severe reinfections. So unlike other viruses, the dengue virus appar-
449 ently leaves patients more vulnerable the second time. These types of complex effects require
450 deep and detailed analysis of both infected and convalescent patients at the single cell level to
451 understand the immune response.

452 We applied SAUCIE to the single-cell CyTOF data of T cells collected in an area endemic
453 for dengue virus infection [19] to study general T cell compartment composition, variability
454 and changes in the variability after convalescence. We believe that the dengue data is an ideal
455 test case for SAUCIE, because the samples are shipped from India and samples were collected
456 over a period of months and were assessed over different experiment days [19]. Thus, there is a
457 pressing need for batch correction and data cleaning as well as uniform processing, clustering
458 and meta-analysis of patient stratification. As part of the study, cells from additional patient
459 groups beyond the acutely infected were also measured: healthy people unrelated to the subjects
460 as a control and the same acute subjects at a later convalescent time point. Primary research
461 questions include understanding profile of the acute subjects and how they differ from the other
462 groups. Across all groups, there are 180 samples resulting in over twenty million cells with
463 results analyzed on 35 different protein markers, a massive amount of data that would cause
464 difficulties in most standard analytic frameworks.

465 **2.3.1 Batch correction**

466 Beyond the sheer size of the total dataset, due to the large number of distinct samples in the
467 experiment there are significant batch related artifacts effects, stemming from day-to-day dif-
468 ferences, instruments, handling and shipping of the samples. While there are true biological

469 differences between the individual samples, to identify those true differences in the samples we
470 have to remove differences that are caused by these technical variables.

471 Differences that are highly associated with the day they were run on the cytometry instru-
472 ment can be seen by grouping all of the samples together by run day and examining their
473 marker-by-marker abundances. Each run day has twelve samples chosen such that each day
474 has samples from each experimental condition, so any differences between the samples from
475 each day are batch effects. As shown in Figure S3, these difference exist in the spike-in controls
476 as well as the samples, confirming their identity as batch effect and not true variation.

477 Figure S7 shows four markers with extreme batch effects: TCRgd, IL-6, IFNg, and CD86.
478 These batch effects would normally mean only samples within each run day could be compared
479 to each other, as comparisons between samples from different run days would be dominated by
480 the differences in the run days. Instead, the SAUCIE batch correction removes these undesirable
481 effects by combining the samples from each day and aligning them to a reference batch, here
482 chosen to be Day 1. Figure S7 shows that after SAUCIE the differences between run days
483 disappear so that now what it means to be low or high in a marker is the same for each day.
484 Before, the cells with the lowest IFNg in samples from Day 3 would still be considered IFNg+
485 while the cells with the highest IFNg in samples from Day 1 would still be IFNg-. After batch
486 correction with SAUCIE, these can be directly compared.

487 The challenge of batch correction is to remove differences due to artifacts while preserving
488 biological differences. We reason that to prevent removing true biological variation, the ‘shape’
489 of the data (but not its position and scale) within each day must be preserved. We define the
490 shape of the data as any moment beyond the first two - mean and variance. We examine this in
491 detail by considering a run day with the most significant batch effects, Day 2. In Figure S3C,
492 the SAUCIE visualization shows that the reference and nonreference batches are completely
493 separated. When MMD regularization is added in SAUCIE, though, these two batches are fully
494 overlapped. In Figure S8, we examine the twelve individual samples that were run on Day 2.
495 Initially, we see that this confirms our idea that the differences between days are batch effects,
496 because each sample measures high in IL-6 and CD86. So the differences between samples
497 run on Day 1 and Day 2 in CD86 abundance is not dominated by having more of a certain
498 sample type in Day 2. Instead, all samples in Day 2 have been shifted higher. As desired, after
499 batch correction, the mean of each marker is reduced to the level of the reference-batch mean.
500 Crucially, the relationship of samples in Day 2 relative to each other is preserved. The samples
501 with the highest IL-6 in Day 2 are still Samples 3, 9, and 11 while the samples with the lowest
502 are still Samples 4, 5, and 6. SAUCIE has just changed what it means to be high or low for
503 samples in this day such that it reconciles what it means to be high or low for samples in the
504 reference day.

505 In conclusion, the batch correction and denoising ability of SAUCIE has transformed the
506 data into a form that is amenable to biological discovery. We investigate this in the next section.

507 2.3.2 Differential cluster proportions between subjects

508 We first obtain the clusters characteristic of each group and then further analyze them for marker
509 enrichments as single cell versions of blood biomarkers [40]. For the clustering considered here,
510 we use a coarse-grained clustering obtained with a coefficient for ID regularization of 0.1. This
511 was chosen by scanning across values of 0.01, 0.1, 0.2, 0.3, 0.4, and 0.5, and choosing the
512 clustering that yielded the best modularity. If other granularities are desired, lower coefficients
513 could be used and the impact of this parameter on the number of clusters is shown in Figure S9.
514 The two regularizations λ_d and λ_c affect the number of clusters that result. For a given value
515 of λ_d , as λ_c increases, the number of clusters decreases (coarser granularity). Higher values of
516 λ_d yield more clusters (finer granularity). Notably, these results are robust and yield reasonable
517 results for varying values of the two regularizations. These two together act as knobs that can
518 be tuned to get the desired granularity of clustering. The methods section further discusses how
519 these regularizations affect the number of clusters.

520 For the SAUCIE clustering, we focus on T cells as particularly relevant to the immune pro-
521 cess and an abundant subset of the data (eleven million total cells), looking for clusters that
522 are over- or under-represented in the cells of each group. We look for clusters that behave dif-
523 ferently in the acute compared to the convalescent time points. These would then represent a
524 population of cells that might have an important role in the process, which could be further in-
525 vestigated. To understand what cell population this is, we examine the marker abundance profile
526 for the cluster. The mean for each cluster and marker is shown in the heatmap in Figure 6B.

527 We find twenty total clusters within the T cell populations, five of which are CD8 T cells
528 and thirteen of which are CD4 T cells. In addition, interestingly, there are six clusters of CD4-
529 CD8- T cells, where four are $\gamma\delta$ T cells. These have been noted as a characteristic of reaction to
530 viral infections [41–45]. There are twelve clusters representing effector memory cells and nine
531 regulatory T cells that are CD4+Foxp3+. Two of the clusters are naive T cells.

532 Several of these populations are indicative of differences between acute, convalescent, and
533 healthy subjects, and can be used for characterizing the nature of the reaction of each of these
534 groups, as we do below.

- 535 1. $\gamma\delta$ T cells are a relatively rare type of T cells, but SAUCIE is still able to identify them.
536 Despite their rarity, they appear to have significance in identifying different populations,
537 which emphasizes the importance of this attribute of SAUCIE. These cells signal espe-
538 cially strong early in immune response, particularly skin and mucosal immunity. They
539 have less variable TCR sequences than $\alpha\beta$ T cells [46]. These cells are a bridge between
540 T cells and myeloid cells, as they have some innate immune activity, where they express
541 CD11c and CD86. They can bind to lipid antigens. Clusters 0 and 3 (consisting of 7% of
542 the total cells) shows upregulation of CD57. This is an indication of terminal differentia-
543 tion. CTLA-4 and CD38 are also high, so these are highly activated cells and potentially
544 dysfunctional. We see that these clusters are highest in the acute subjects and lowest in
545 the healthy subjects. Out of the fifteen subjects that were measured both as acute subjects
546 and later in convalescence, thirteen had more of these cells during their acute infection.

- 547 2. We find another group of $\gamma\delta$ T cells that are CD45RO and CD45RA positive (cluster 2,
548 consisting of 1% of the total cells), but not yet fully terminally differentiated, so these
549 could be transitional between naïve and effector memory. The effector memory cells
550 express less IFN β . As this cluster is more expressed in the healthy subjects, it indicates
551 that even these subjects may have had some exposure to dengue. There is a lack of an
552 inflammatory state, i.e., low in IFN β and Perforin, so we expect that these are actually
553 memory cells instead of effector cells. It makes more sense then that these populations
554 are more expressed in convalescent and healthy subjects.
- 555 3. We also find another population of CD4+ T cells (clusters 3-15, consisting of 45% of the
556 total population) that are not expressing any inflammatory markers or activation markers,
557 and these are higher in the convalescent and healthy subjects, while being very low in
558 the acute subjects. These look to be other memory cells that may characterize these
559 convalescent subjects. In fact, out of the fifteen subjects with acute-convalescent paired
560 measurements, eleven had more of these cells during convalescent measurement. These
561 have signs of recent activation as they do not have CD69, which is an early activation
562 marker, nor any of the cytokines like IFN γ , IFN β , or IL-6.
- 563 4. Additionally, we find a population of CD8+ effector cells (cluster 15, which consists of
564 3% of the total cells) that are highly expressed in the acute subjects. These cells also
565 express CD57 and CD38, but are not $\gamma\delta$ as the previous populations were. These appear
566 to be more differentiated and are likely not transitional, as the previous ones were, either.

567 We can also visualize the cell-level cluster proportions on a patient manifold (Figure 5B). There,
568 we see that cluster proportions arranged on this manifold reveal clusters that are changing across
569 the space. This analysis indicates clearly that cluster 1 is representative of acute subjects and
570 cluster 5 is representative of the healthy subjects. Furthermore, we can evaluate the same in-
571 dividual when measured after acute infection, and then later at a convalescent time point (Fig-
572 ure 5C). Viewed in this way, we see that cluster 11 is also more present in most subjects when
573 they came in with an acute infection than at the convalescent time point.

574 2.3.3 Visualization

575 SAUCIE can process all cells from all subjects to construct a cellular manifold and extract its
576 features. First, we visualize this manifold using the 2-D visualization layer. Figure 6A is divided
577 into two embeddings that show the cell manifolds for acute and healthy subjects separately.
578 As can be seen, there is a characteristic change in the manifold that becomes apparent when
579 comparing the embeddings side-by-side. The acute subjects have cell populations distinctly
580 missing that are present in the healthy subjects.

581 After characterizing the nature of the cellular space in the aggregate, we can additionally
582 analyze manifolds formed by the distributions of T lymphocytes within each patient separately.
583 As each patient has a heterogeneous population of cells, including with different total numbers

584 of cells, it becomes a challenge to define a meaningful measure of similarity between the indi-
585 viduals. Here we are able to leverage the manifold constructed by the SAUCIE embedding and
586 calculate MMD (a distribution distance) between the distribution of cells in the latent space for
587 each pair of subjects. With a measure of similarity between each pair of patients, we can now
588 construct a manifold not of the cells but also of the *subjects* (Figure 5A).

589 **2.3.4 Comparison to existing method**

590 We next compare the SAUCIE pipeline of batch correcting, clustering, and visualizing single-
591 cell data from a cohort of subjects to an alternative approach called metaclustering [47]. We
592 first cluster each sample individually with Phenograph. Then, we represent each cluster as
593 its centroid and use Phenograph again on the clusters to obtain metaclusters. We examine
594 the pipelines on ten of the 180 samples here, where the metaclustering approach took forty
595 minutes. We note that the SAUCIE pipeline took *45 minutes* to process all 180 samples, while
596 the metaclustering approach would take *12 hours* to process all of them. Figure S10 shows
597 tSNE embeddings of the cluster centroids where the size of the cluster is proportional to the
598 size of the point. Coloring by sample, we see that the metaclusters have identified batch effects.
599 Metacluster 0 is only composed of samples 1, 3, 4, and 5. These samples have no clusters in any
600 other metacluster, and none of the other samples have any cluster in this metacluster. Examining
601 the gene expression heatmap, we see that metacluster 0 has separated cells with high CD86
602 values, which were shown earlier to be batch effects. Moreover, the metaclusters are very
603 heterogeneous internally with respect to gene expression. This is a results of metaclustering
604 the cluster centroids, as the metaclusters then have no information about the individual cells
605 comprising that centroid.

606 In contrast, Figure S11 shows the SAUCIE pipeline on these ten samples. The cluster
607 proportions show that each cluster is fully mixed with respect to the samples, as opposed to
608 the sample-segregated metaclusters of the previous approach. Similarly, the clusters are more
609 homogeneous internally, meaning they actually keep similar cells together, as opposed to the
610 metaclusters, which lost this information when each cluster was represented by only its centroid.
611 Finally, we find that SAUCIE effectively compares cells across subjects, while the metacluster-
612 ing approach still fails at patient-to-patient comparisons, instead only identifying batch effect
613 variation. This emphasizes the importance of multitask learning using a unified representation
614 in SAUCIE.

615 **3 Discussion**

616 We presented SAUCIE, a neural network framework that streamlines exploratory analysis of
617 datasets that contain a multitude of samples and a large volume of single cells measured in each
618 sample. The key advantage in SAUCIE is its ability to perform a variety of crucial tasks on
619 single-cell datasets in a highly scalable fashion (utilizing the parallelizability of deep learning
620 with GPUs) without needing to call external algorithms or processing methods. As a result,

621 SAUCIE is able to process multisample data in a unified way using a single underlying repre-
622 sentation learned by a deep autoencoder. Thus, different samples can be visualized in the same
623 coordinates without batch effects via the embedding layer of the neural network, and cluster
624 proportions can be directly compared, since the whole dataset is decomposed into a single set
625 of clusters without requiring cluster matching or metaclustering. These unified representations
626 can be readily used for inter-sample comparisons and stratification, on the basis of their under-
627 lying cell-to-cell heterogeneity.

628 Mathematically, SAUCIE presents a new way of utilizing deep learning in the analysis of
629 biological and biomedical data by directly reading and interpreting hidden layers that are regu-
630 larized in novel ways to understand and correct different aspects of data. Thus far, deep learning
631 has primarily been used in biology and medicine as a black-box model designed to train clas-
632 sifiers that often mimic human classifications of disease or pathology. However, the network
633 internal layers themselves are typically not examined for mechanistic understanding. SAUCIE
634 is leading a new wave of deep learning models that obtain information from internal layers
635 of a deep network. Deep autoencoding neural networks essentially perform nonlinear dimen-
636 sionality reduction on the data. As such they could be used “off-the-shelf” for obtaining new
637 coordinates for data in a reduced-dimension space, to which other algorithms can be applied.
638 However, in SAUCIE we aim to go further to structure the reduced dimensions in specifically
639 interpretable ways using novel regularizations. Our information-theoretic regularization en-
640 courages near-binary activations of an internal layer, thus making the layer amenable to directly
641 output encoded cluster identifications. We believe that this is just the first foray into what could
642 be a vast number of such regularizations that can offer interpretability of specialized layers in
643 neural networks, thus turning these “black boxes” into “glass boxes.”

644 The ability to stratify patients on the basis of their single-cell subpopulations, which can
645 emerge as features in deep neural networks, can be key to a new generations of biomarkers that
646 can be used in diagnosis and treatment. Traditionally, biomarkers are proteins or antibodies that
647 are circulating in blood, which signals the presence of infection or other conditions. However,
648 immune cells are highly plastic and can evolve or activate in specific ways in response to dis-
649 ease conditions in different patients. Here, we showcase the heterogeneity of immune cells in
650 response to acute dengue infection in a large patient cohort. We see that specific subpopula-
651 tions are enriched in the acute conditions, as opposed to convalescent or healthy controls. We
652 showed with our dengue dataset it is possible discover cell populations, even rare ones that are
653 indicative of patient and experimental conditions. Other datasets comprising of large patient
654 cohorts measured at single-cell resolution are underway already in many hospitals and clinical
655 trials. In the future, we are confident that this capability will be useful in many studies includ-
656 ing immunotherapy, autoimmunity, and cancer, where there are immune subsets that emerge in
657 response.

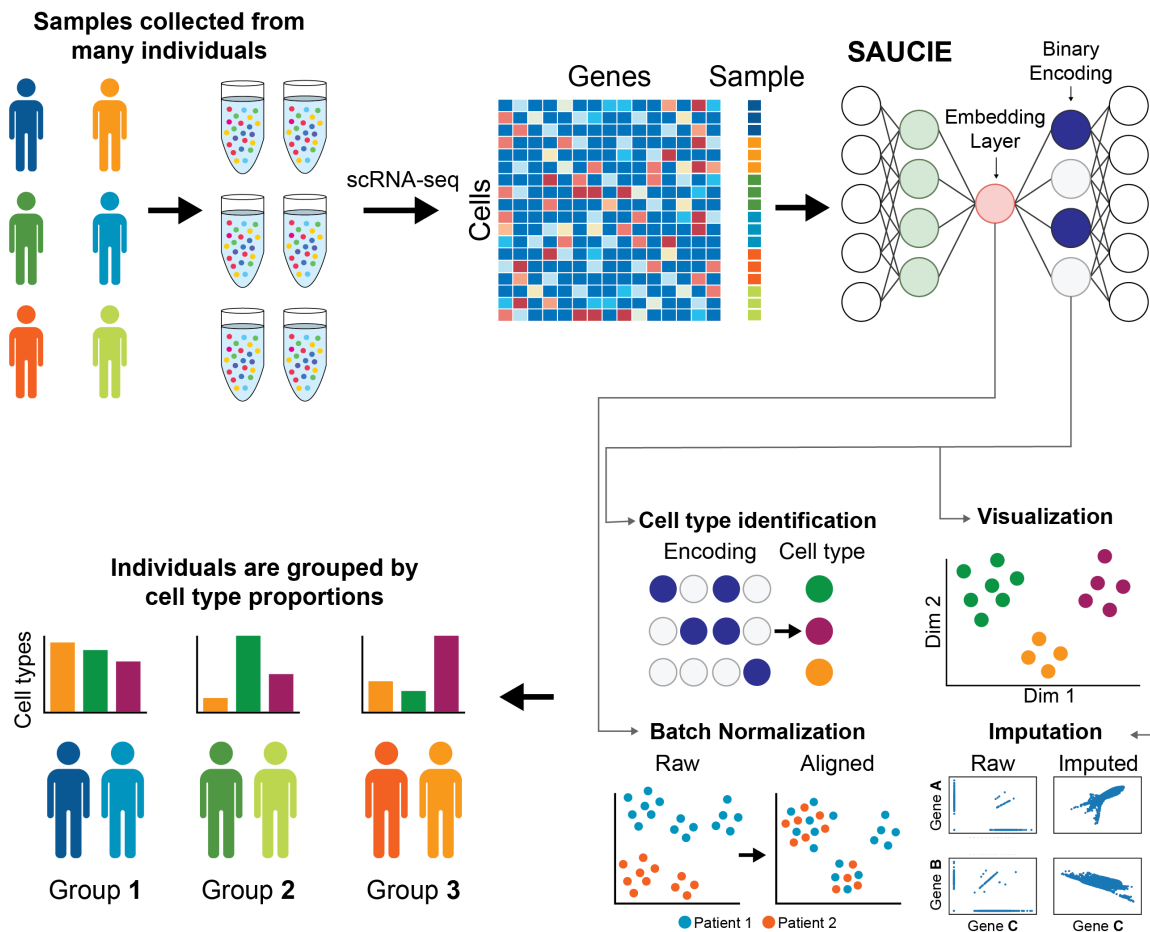


Figure 1: The pipeline for analyzing single-cell data in large cohorts with SAUCIE. Many individual patients are separately measured with a single-cell technology such as CyTOF or scRNA-seq, producing distinct datasets for each patient. SAUCIE performs imputation and denoising, batch effect removal, clustering, and visualization on the entire cohort with a unified model and is able to provide interpretable, quantifiable metrics on each subject or group of subjects.

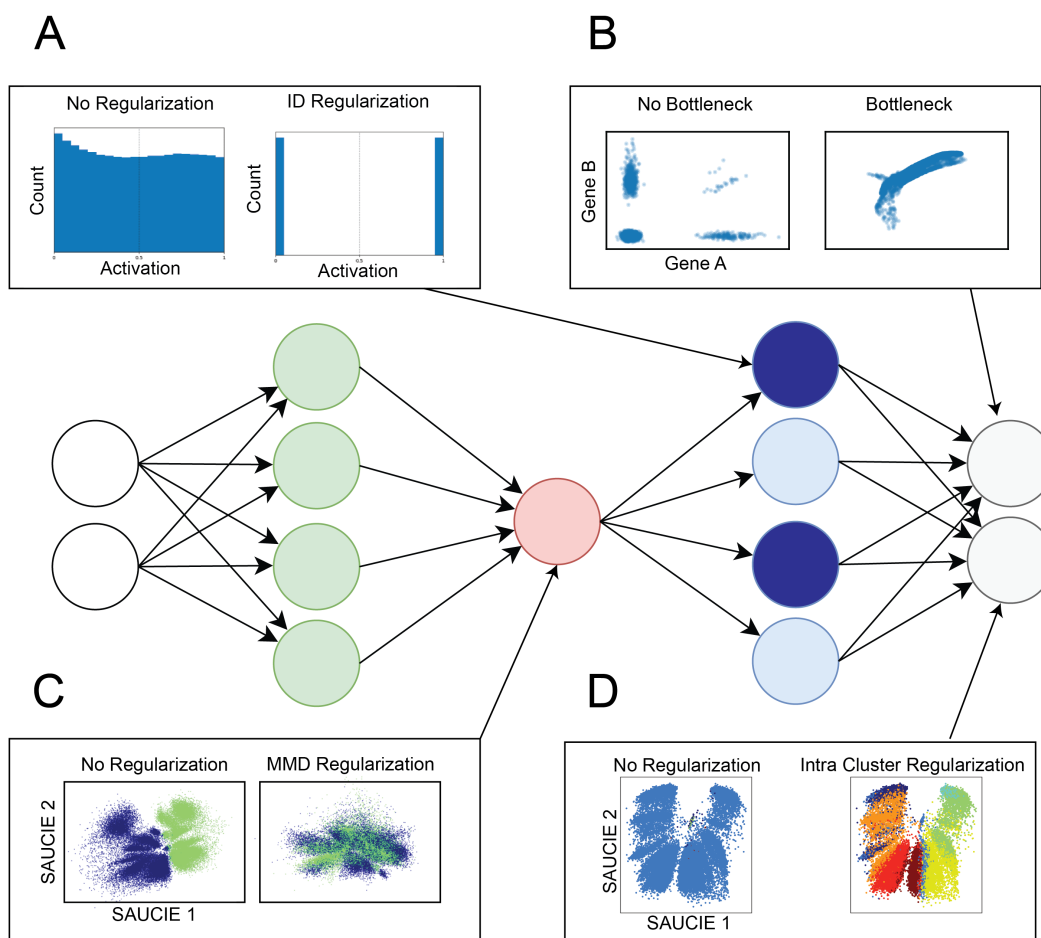


Figure 2: **Regularizations and architecture choices in SAUCIE.** A) the ID regularization applied on the sparse encoding layer produces digital codes for clustering B) the informational bottleneck, i.e. a smaller embedding layer, uses dimensionality reduction to produce denoised data at the output C) the MMD regularization removes batch artifacts D) the within cluster distance regularization applied to the denoised data provides coherent clusters.

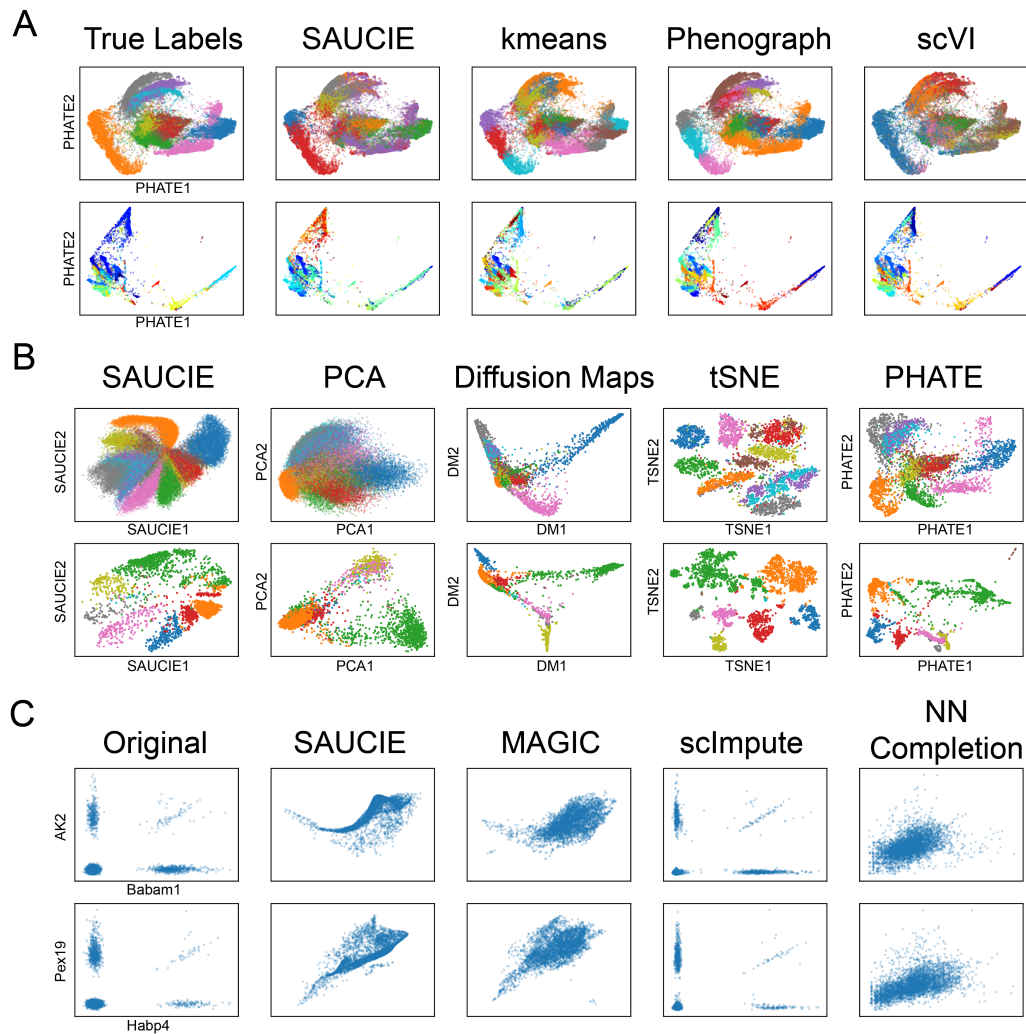


Figure 3: A comparison of the different analysis tasks performed by SAUCIE against other methods. A) A comparison of clustering performance shown on PHATE. SAUCIE compares well to the other methods, producing a coherent clustering. Neither Phenograph nor scVI produces clusters that look coherent. B) A comparison of SAUCIE's visualization. PCA produces a blurry visualization. Diffusion maps shows a much simplified structure. tSNE shatters the space. SAUCIE produces a result similar to PHATE, revealing the structure in the data. C) A comparison of imputation. SAUCIE recovers complex nonlinear shapes of gene-gene relationships.

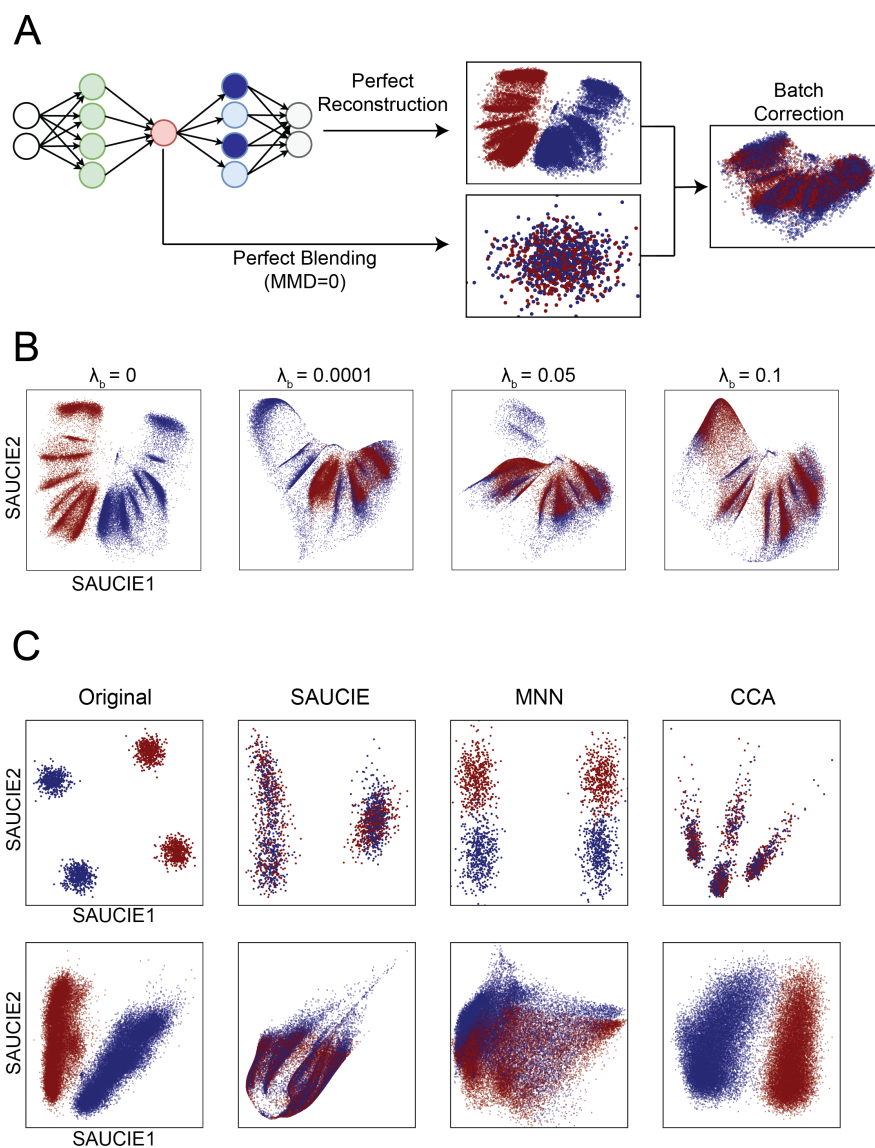


Figure 4: Demonstration of SAUCIE's batch correction abilities. A) SAUCIE batch correction balances perfect reconstruction (which would leave the batches uncorrected) with perfect blending (which would remove all of the original structure in the data) to remove the technical variation while preserving the biological variation. B) The effect of increasing the magnitude of the MMD regularization on the dengue data. Sufficient MMD regularization is capable of fully removing batch effect. C) Results of batch correction on the synthetic GMM data (top) and the dengue data (bottom) shows that SAUCIE better removes batch effects than MNN and better preserves the structure of the data than CCA.

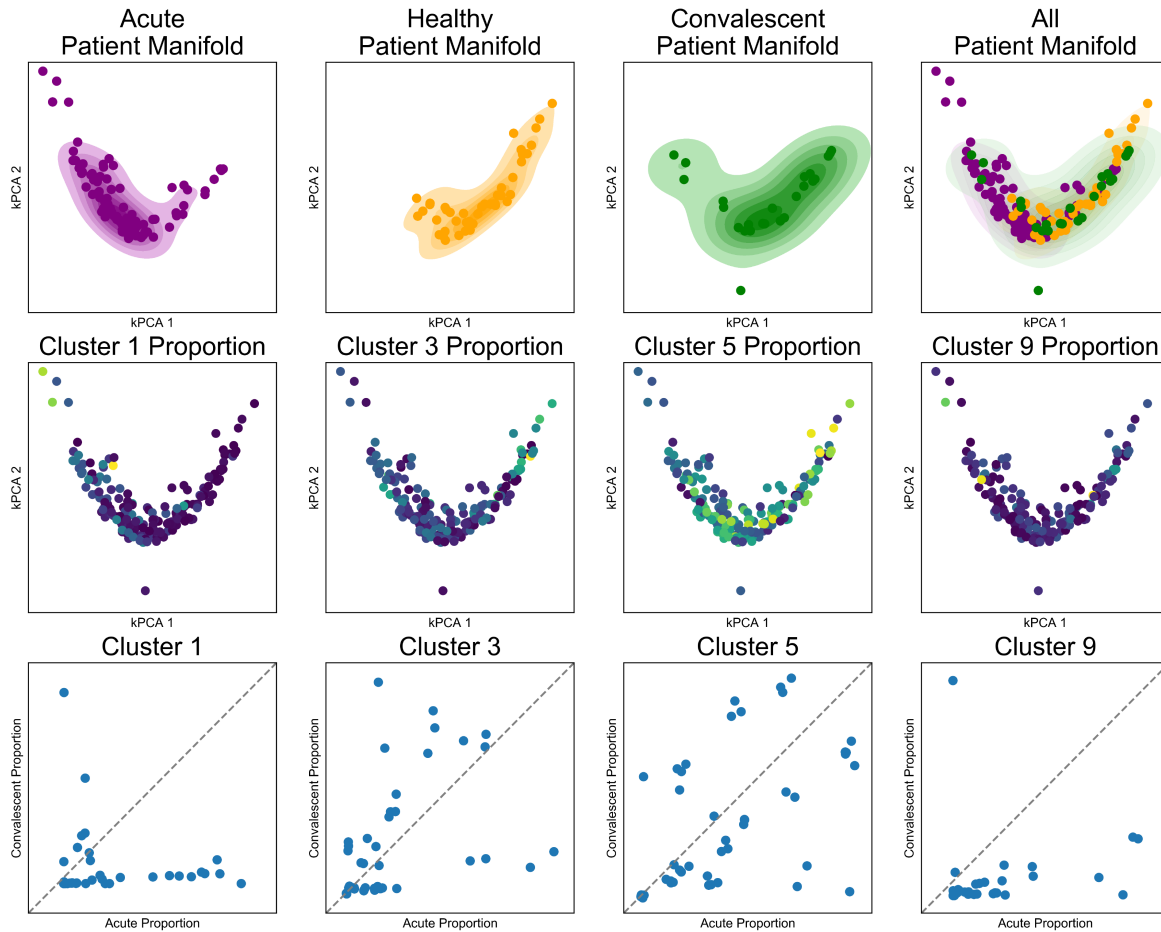


Figure 5: **SAUCIE produces patient manifolds from single-cell cluster signatures.** Top row) The patient manifold identified by SAUCIE cluster proportions, visualized by kernel PCA with acute, healthy, convalescent, and all subjects combined from left to right. The healthy manifold overlaps with the convalescent manifold to a much higher degree than the acute manifold. Middle row) The same patient manifold shown colored by each patient's cluster proportion. Cluster 1 is more prevalent in acute, cluster 3 in healthy, cluster 5 is ubiquitous, and cluster 9 is rare and in acute patients. Bottom row) A comparison of the cluster proportion for acute (X-axis) versus convalescent (Y-axis) for patients that have matched samples.

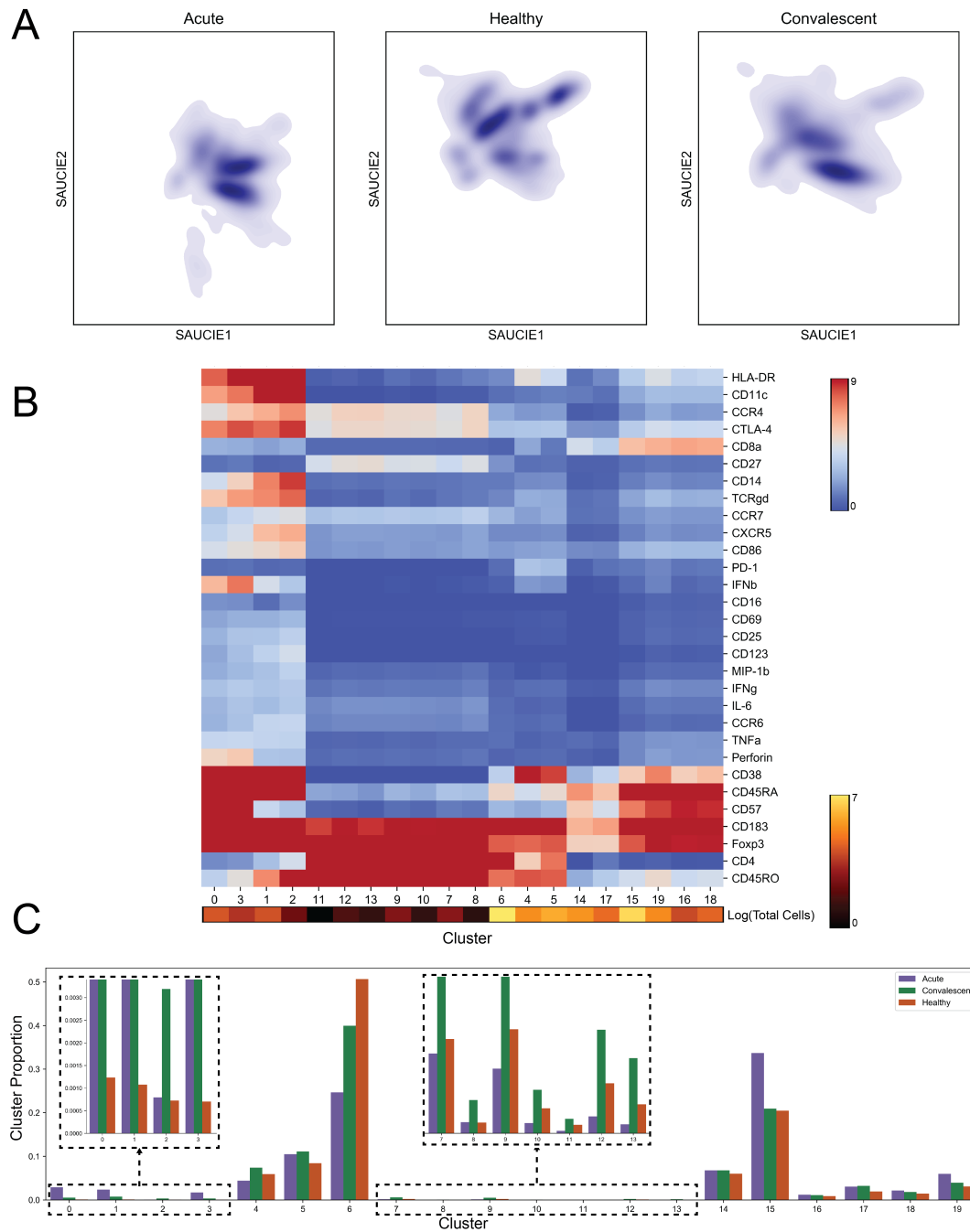


Figure 6: SAUCIE identifies and characterizes cellular clusters, whose proportions can be used to compare patients. A) The cell manifolds identified by the two-dimensional SAUCIE embedding layer for the T lymphocyte subsets from acute, healthy, and convalescent subjects. B) A heatmap showing clusters along the horizontal axis and markers along the vertical axis. Cluster sizes are represented as a color bar beneath the heatmap. C) Cluster proportions for acute, convalescent, and healthy patients.

658 **4 Methods**

659 **4.1 Computational Methods**

660 In this section we explain the SAUCIE framework in greater detail including the philosophy
661 behind using autoencoders for learning the cellular manifold, details of the regularizations used
662 in different layers of SAUCIE to achieve particular data analysis tasks as well as training and
663 implementation details. Finally, we discuss the emergent higher level organization of the patient
664 manifold as a result of the cellular manifold of the subjects learned by SAUCIE.

665 **4.1.1 Multitask manifold learning**

666 A popular and effective approach for processing big high-dimensional data in genomics, as well
667 as other fields, is to intuitively model the intrinsic geometry of the data as being sampled from
668 a low dimensional manifold – this is commonly referred to as the manifold assumption [17].
669 This assumption essentially means that local regions in the data can be linearly mapped to
670 low dimensional coordinates, while the nonlinearity and high dimensionality in the data comes
671 from the curvature of the manifold. Typically, a notion of locality is derived from the data with
672 nearest-neighbor search or adaptive kernels to define local neighborhoods that can approximate
673 tangent spaces of the manifold. Then, these neighborhoods are either used directly for opti-
674 mizing low dimensional embeddings (e.g., in TSNE [48] and LLE [49]), or they are used to
675 infer a global data manifold by considering relations between them (e.g., using diffusion geom-
676 etry [14, 15, 50, 51]). In the latter case, the data manifold enables several applications, including
677 dimensionality reduction [14, 51], clustering [50, 52–54], imputation [15], and extracting latent
678 data features [55–57].

679 The characterization of the intrinsic data geometry as a data manifold is also closely related
680 to the underlying approach in SAUCIE. Indeed, neural networks can be considered as piecewise
681 linear approximations of target functions [58]. In our case, we essentially approximate the data
682 manifold coordinate charts and their inverse with the autoencoder architecture of SAUCIE.
683 The encoder training identifies local patches and maps them to low dimensional coordinates,
684 while sewing these patches together in this embedding to provide a unified visualization. The
685 decoder learns the linear relation between these intrinsic coordinates and the tangent spaces of
686 the manifold, positioned in the high dimension. This also results in a projection of data points on
687 the manifold (via its tangent spaces), which creates a denoising effect similar to the diffusion-
688 based one used recently in MAGIC [15]. Finally, the clustering layer in SAUCIE is trained to
689 recognize and aggregate similar data regions to ensure an appropriate granularity (or resolution)
690 of the identified neighborhoods and prevent excessive fragmentation of the manifold. For more
691 discussion regarding the relations between deep learning and manifold learning we refer the
692 reader to [2, 16, 59].

693 While tools using the scaffold of manifold learning have emerged for various tasks in single
694 cell data analysis, there is currently no unified manifold model that provides all of the necessary
695 tasks in a scalable fashion. For example, MAGIC [15] uses manifold learning to impute the

696 data, but does not address embedding, visualization, or clustering. Diffusion pseudotime [55]
697 provides an organization of the data to infer latent temporal structure and identifies trajectories,
698 but it does not deal with imputation, clustering, or visualization. Furthermore, manifold learning
699 methods do not work well across batches and typically just focus on single batches. Thus, their
700 construction may suffer from batch effects and be dominated by the geometry between batches
701 rather than their biology, as demonstrated by the example of Phenograph in Figure S10.

702 To address these shortcomings, SAUCIE performs all operations on a unified manifold ge-
703 ometry, which is learned implicitly by a deep multitasking neural network. It utilizes the scala-
704 bility of deep learning to process high throughput data and construct a manifold that is jointly
705 optimized for multiple tasks; namely, clustering, visualization, imputation, and batch correc-
706 tion. Therefore, the tasks themselves respect the manifold assumption and have the associated
707 advantages, such as robustness to noise, while also agreeing with each other on a coherent
708 underlying structure of the data.

709 **4.1.2 SAUCIE architecture**

710 SAUCIE consists of three encoding layers, an embedding layer, and then three decoding layers.
711 The default number of neurons per hidden layer in the encoder used were 512, 256, and 128
712 with a symmetric decoder. The GMM dataset, being simpler, was clustered with layers of 50,
713 30, and 10. For batch correction, the best results were achieved with layer sizes of 1024, 512,
714 and 256. The ID regularization was applied to the final decoder layer, which uses a ReLU. The
715 two-dimensional embedding layer uses a linear activation, while all other layers use a leaky
716 rectified linear activation with 0.2 leak. The coefficients λ_d and λ_c were chosen depending on
717 the dataset, with the best values generally being λ_d twice λ_c . Their magnitude was guided by the
718 effect of these two knobs on the granularity (shown in Figure S9). Training was performed with
719 minibatches of 256, mean-squared-error for the reconstruction error function, and the optimizer
720 chosen is ADAM with learning rate 0.001.

721 **4.1.3 Batch correction and MMD Regularization**

A major challenge in the analysis of single-cell data is dealing with so-called batch effects that 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, for example, by assuming measurements are drawn from a Gaussian distribution. Instead of addressing specific explicit models of noise, SAUCIE minimizes a distance metric between distributions. The batch correction term L_b calculates the Maximal Mean Discrepancy (MMD) [60] between batches, as

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

722 where V_{ref} is the visualization layer of one of the replicates, arbitrarily chosen to be considered
723 as a reference batch. MMD compares the average distance from each point to any other point
724 in its own batch, with the distance to points in the other batch. MMD is zero only when two
725 distributions are equal. Thus minimizing this metric encourages SAUCIE to align the batches.
726 MMD has been used effectively to remedy batch effects in residual networks, but here SAUCIE
727 uses it in a feedforward autoencoder and combines it with other tasks of interest in biological
728 exploratory data analysis [21].

729 The choice of reference does not affect the degree to which two distributions can be aligned,
730 but a reference batch is necessary because the encoding layers of a standard network will be
731 encouraged to embed different batches in different places in the visualization layer. It does this
732 because the decoder is required to make its reconstruction \hat{X} match the original data in X , which
733 includes the batch effects. To remedy this, the decoder in SAUCIE is required to reconstruct the
734 reference batch exactly as usual, but other batches must only be reconstructed to preserve the
735 points normalized by mean and variance. Consequently, the MMD regularization term will be
736 minimized when batches are aligned, and the decoder need only be able to reconstruct the exact
737 values of the reference batch and the *relative values* of the non-reference batches. The non-
738 reference batches will be aligned to the reference batch in a way that preserves their internal
739 structure as best as possible.

740 4.1.4 Regularizations and Post-processing for clustering

741 **Information Dimension Regularization** We consider the task of clustering data points by
742 interpreting the sparse layer B in the network as encoding cluster assignments. We note that
743 a common activation function used to introduce nonlinearities in neural networks (including
744 SAUCIE) is the Rectified Linear Unit (ReLU), and it provides a natural threshold for binarizing
745 neuron activation to be either zero or one. These units are either “off” at or below zero or
746 “on” for any positive value, so a small positive value ϵ can be used a threshold to binarize the
747 activations in B . This results in an interpretable clustering layer that creates ‘digital’ cluster
748 codes out of an ‘analog’ hidden layer, thus providing a binary code for each input point of the
749 network. These binary codes are in turn used as cluster identifiers in order to group data points
750 with the same code into a single cluster.

751 In order to automatically learn an appropriate granularity of clusters, we developed a novel
752 regularization that encourages near-binary activations and minimizes the information (i.e., num-
753 ber of clusters) in the clustering layer. Our regularization is inspired by the von Neumann (or
754 spectral) entropy of a linear operator [61], which is computed as the Shannon entropy of their
755 normalized eigenvalues [62, 63]. This entropy serves as a proxy for the numerical rank of the
756 operator [51], and thus provides an estimation of the essential dimensionality of its range. In
757 our case, we extend this notion to the nonlinear transformation of the neural network by treat-
758 ing neurons as our equivalent of eigenvalues, and computing the entropy of their total activation
759 over a batch. We call this entropy ‘information dimension’ (ID) and the corresponding ID reg-
760 ularization aims to minimize this entropy while still encoding sufficient information to allow

761 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

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

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

769 **Intracluster distance regularization** The digital codes learned by SAUCIE create an oppor-
770 tunity to interpret them as clusters, but these clusters would not necessarily be comprised of
771 only similar points. To emphasize that inputs only be represented by the same digital code if
772 they are similar to each other, SAUCIE also penalizes intracluster pairwise distances. Beyond
773 suffering reconstruction loss, using the same code for points that are far away from each other
774 will now incur an even greater loss.

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

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

775 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.

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

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

Cluster merging As the binarized neural network may not converge to the ideal level of
granularity due to the many possible local optima in the loss landscape, we process the SAUCIE
clustering with a cluster merge step to fix the ideal level of granularity everywhere. The cluster
merging is performed by calculating MMD between clusters in the SAUCIE latent space and

merging all clusters $i, j \in C$, where C is the set of all clusters, such that both of the following equations hold

$$\operatorname{argmin}_{\xi \in C} \operatorname{MMD}(i, \xi) = j \quad (2)$$

$$\operatorname{argmin}_{\xi \in C} \operatorname{MMD}(j, \xi) = i \quad (3)$$

784 This merging finds clusters that would be a single cluster in another granularity and fixes them
785 to a single cluster.

786 4.1.5 Patient Manifold Visualization

787 In addition to the cell-level manifold constructed by SAUCIE, we also consider the geometry
788 between samples to provide a coarser patient-level manifold. We construct and embed this
789 manifold in low dimensions by applying kernel-PCA (kPCA) [64] with an RBF kernel to the
790 metric space defined by MMD distances between subjects. This augments the analysis SAUCIE
791 provides of the biological variations identified in the cell space with an analysis of the variation
792 in the patient space. Normally, without batch correction, the two sources of variation would
793 be confounded, and batch effects would prevent clear analysis at either level (patient or cell)
794 across batches. With our approach here we are able to separate them to provide on one hand, a
795 stable (batch-invariant) cell-level geometry by the SAUCIE embedding, and on the other hand,
796 a robust patient geometry provided by kPCA embedding. The patient geometry then allows us
797 to recover patient-level differences and utilize them further for data exploration, in conjunction
798 with the cell-level information. For example, as Figure 5A shows, we have a notable stratifica-
799 tion between the acute and non-acute subjects. There is also a noticeable difference between the
800 convalescent subjects and the acute, albeit a less drastic one than the difference between acute
801 subjects and the others.

802 4.1.6 Training

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

810 For the first run, formally let X be an $n \times d$ input batch, where each row is a single data
811 point, and d is the number of features in the data. It is passed through a cascade of encoding
812 linear and nonlinear transformations. Then, a cascade of decoding transformations reconstruct
813 the denoised batch \hat{X} , which has the same dimensions as the input X and is optimized to
814 reconstruct it.

815 For the next run, the cleaned batch \hat{X} is passed through encoding transformations and a
816 visualization layer denoted by $V \in \mathbb{R}^{n \times 2}$. We also consider a clustering layer in another run
817 where the decoder outputs near-binary activations $B \in \mathbb{R}^{n \times d_B}$, where d_B is the number of
818 hidden nodes in the layer, which will be used to encode cluster assignments, as described below.
819 The activations in B are then passed to the reconstruction \tilde{X} that has the same dimensions as \hat{X}
820 (and X) and is optimized to reconstruct the cleaned batch.

The loss function of all 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 (i.e., $L_r(X, \hat{X}) = \frac{1}{n} \sum_{i=1}^n \|x_i - \hat{x}_i\|^2$, where x_i and \hat{x}_i are the i -th row of X and \hat{X} correspondingly). We note that while MSE is a standard and effective choice in general, other loss functions can also be used here as application-specific substitutes that may be more appropriate for particular types of 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 clustering run then optimizes L_r along with two regularization terms L_c and L_d that together enable SAUCIE to learn clusters:

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

821 The first term L_c guides SAUCIE to learn binary representations via the activations in B using
822 a novel information dimensionality penalty that we introduce in this paper. The second term L_d
823 encourages interpretable clusters that contain similar points by penalizing intra-cluster distances
824 in the cleaned batch \hat{X} , which is fixed for this run.

825 **4.1.7 Runtime Comparison Methodology**

826 For each visualization, clustering, and imputation method, the dataset of size N was given to
827 the method as input and returned the appropriate output. For batch correction, the dataset of
828 size N was divided into two equal-sized batches that were corrected. For the methods that
829 operated on minibatches, minibatches of size 128 were used. For the methods that train by
830 stochastic gradient descent, the number of steps was determined by taking the total number of
831 points and dividing by the size of the minibatch, so that a complete pass through the entire
832 dataset was performed. In order to return clusters, the latent space of scVI must be clustered by
833 another method, and since the number of clusters is not known ahead of time, the fastest method
834 that does not require this to be known (Phenograph) was used. For SAUCIE, batch correction,
835 imputation, clustering, and visualization were all produced in the timed run. All computations
836 were performed on a single machine with 16 CPU cores and a GeForce GTX 1080 GPU.

837 **4.1.8 Number of Clusters**

838 As discussed earlier, the number of clusters resulting from SAUCIE is not specified in advance,
839 but dictated by the structure of the data that the model discovers, and by the choice of regular-
840 ization coefficients λ_d and λ_c . For a given value of λ_d , as λ_c increases, the number of clusters
841 decreases. Increasing λ_d , on the other hand, increases the number of clusters (Figure S9). This
842 is because λ_c penalizes entropy in the activations of the n neurons in the clustering layer of the
843 network. While entropy can be initially decreased by making all n neurons either 0 or 1, it can
844 be further decreased by making all n neurons 0. Thus, as this term is considered more influential
845 in the total loss, in the extreme, all points can be mapped to the same binary code. In contrast,
846 λ_d penalizing intra-cluster distances, so this value can be decreased by making clusters smaller
847 and smaller (and thus getting more of them). In the extreme for this term, every point can be
848 made its own cluster and intra-cluster distances would decrease to 0. By balancing these two,
849 the desired granularity of clustering can be obtained from SAUCIE. In our experiments, we find
850 making λ_d to be between two and three times larger than λ_c , with values around 0.2 generally
851 results in medium coarse-grained clustering. Another consideration that affects the number of
852 clusters is the number of neurons in the clustering layer. We found varying this number does
853 not improve performance and for all experiments here we use a fixed size of 256 neurons.

854 **4.2 Experimental methods**

855 **4.2.1 Study Subjects**

856 Dengue patients and healthy volunteers were enrolled with with written informed consent under
857 the guidelines of the Human Investigations Committees of the NIMHANS and Apollo Hospital,
858 and Yale University [19]. The Human Investigations Committee of each institution approved
859 this study. Patients with dengue virus infection were defined as dengue fever using WHO-
860 defined clinical criteria, and/or laboratory testing of viral load or serotyping at the time of
861 infection. Healthy volunteers included household contacts of dengue patients present in the
862 same endemic area. Participants were of both genders (26.7% female) and were all of Indian
863 heritage. Subjects from the symptomatic and healthy groups were not statistically different for
864 age, gender, or race in this study.

865 **4.2.2 Sample Collection and Cell Isolation**

866 Heparinized blood was collected from patients and healthy volunteers and employed a 42
867 marker panel of metal conjugated antibodies following methods previously described [65, 66].
868 Purification of peripheral blood mononuclear cells (PBMCs) was performed by density-gradient
869 centrifugation using Ficoll-Paque (GE Healthcare) according to the manufacturer's instructions
870 following isolation and cryopreservation guidelines established by the Human Immunology
871 Phenotyping Consortium. PBMCs for CyTOF were frozen in 90% FBS containing 10% DMSO

872 and stored in liquid N₂ for shipping following the guidelines of the DBT. Samples for this study
873 were received in three shipments and viability was average 85% (range 50–98) across the dates.

874 **4.2.3 Mass Cytometry Acquisition**

875 For mass cytometry at Yale University, PBMCs (5 x 10⁶ cells/vial) were thawed incubated in
876 Benzoylase (50U/ml) in RPMI/10% human serum, and seeded in 96-well culture plate (6 x 10³-
877 1.2 x 10⁶ cells/well. Monensin (2μM, eBioscience) and Brefeldin A (3μg/ml, eBioScience)
878 added for the final 4 h of incubation for all groups. Groups of samples (8-13/day) were infected
879 in vitro per day on 5 separate days and included a CD45-labeled spike-in reference sample in
880 every sample. Surface markers were labeled prior to fixation and detailed staining protocols
881 have been described. Briefly, cells were transferred to 96-well deep well plates (Sigma), resus-
882 pended in 25 μM cisplatin (Enzo Life Sciences) for one minute, and quenched with 100% FBS.
883 Cells were surface labeled for 30 min on ice, fixed (BD FACS Lyse), and frozen at -80°C.
884 Intracellular labeling was conducted on batches of cells (12/day). Fixed PBMCs were perme-
885 abilized (BD FACS Perm II) for labeling with intracellular antibodies for 45 min on ice. Cells
886 were suspended overnight in iridium interchelator (125 nM; Fluidigm) in 2% paraformaldehyde
887 in PBS and washed 1X in PBS and 2X in H₂O immediately before acquisition. A single batch
888 of metal-conjugated antibodies was used throughout for labeling panels. Metal-conjugated an-
889 tibodies were purchased from Fluidigm, Longwood CyTOF Resource Core (Cambridge, MA),
890 or carrier-free antibodies were conjugated in house using MaxPar X8 labeling kits according
891 to manufacturer's instructions (Fluidigm). A total of 180 samples were assessed by the Helios
892 (Fluidigm) on 15 independent experiment dates using a flow rate of 0.03 ml/min in the presence
893 of EQ Calibration beads (Fluidigm) for normalization. An average of 112,537 ± 71,444 cells
894 (mean ± s.d.) from each sample were acquired and analyzed by CyTOF. Data was preprocessed
895 with the hyperbolic sine transformation. Additional experimental details will be given in [19].

896 **4.3 Grant Support**

897 This work was supported in part by awards from the NIH (AI089992), the Indo-U.S. Vaccine
898 Action Program. It was also supported by the CZI grant for computational tools.

899 **5 Software**

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

902 References

- 903 [1] J. Tan, G. Doing, K. A. Lewis, C. E. Price, K. M. Chen, K. C. Cady, B. Perchuk, M. T.
904 Laub, D. A. Hogan, and C. S. Greene, “Unsupervised extraction of stable expression sig-
905 natures from public compendia with an ensemble of neural networks,” *Cell systems*, vol. 5,
906 no. 1, pp. 63–71, 2017.
- 907 [2] G. P. Way and C. S. Greene, “Extracting a biologically relevant latent space from cancer
908 transcriptomes with variational autoencoders,” *bioRxiv*, p. 174474, 2017.
- 909 [3] J. Zhou and O. G. Troyanskaya, “Predicting effects of noncoding variants with deep
910 learning–based sequence model,” *Nature methods*, vol. 12, no. 10, p. 931, 2015.
- 911 [4] B. Alipanahi, A. DeLong, M. T. Weirauch, and B. J. Frey, “Predicting the sequence speci-
912 ficities of dna-and rna-binding proteins by deep learning,” *Nature biotechnology*, vol. 33,
913 no. 8, p. 831, 2015.
- 914 [5] J. D. Cohen, L. Li, Y. Wang, C. Thoburn, B. Afsari, L. Danilova, C. Douville, A. A. Javed,
915 F. Wong, A. Mattox, *et al.*, “Detection and localization of surgically resectable cancers
916 with a multi-analyte blood test,” *Science*, p. eaar3247, 2018.
- 917 [6] Y. Bengio, A. C. Courville, and P. Vincent, “Representation learning: A review and new
918 perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35,
919 pp. 1798–1828, 2013.
- 920 [7] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing
921 robust features with denoising autoencoders,” in *ICML*, 2008.
- 922 [8] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural
923 networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- 924 [9] W. Wang, Y. Huang, Y. Wang, and L. Wang, “Generalized autoencoder: A neural network
925 framework for dimensionality reduction,” in *CVPR Workshops*, 2014.
- 926 [10] J. Tan, M. Ung, C. Cheng, and C. S. Greene, “Unsupervised feature construction and
927 knowledge extraction from genome-wide assays of breast cancer with denoising autoen-
928 coders,” in *Pacific Symposium on Biocomputing Co-Chairs*, pp. 132–143, World Scientific,
929 2014.
- 930 [11] J. Tan, J. H. Hammond, D. A. Hogan, and C. S. Greene, “Adage-based integration of pub-
931 licly available pseudomonas aeruginosa gene expression data with denoising autoencoders
932 illuminates microbe-host interactions,” *MSystems*, vol. 1, no. 1, pp. e00025–15, 2016.
- 933 [12] H. Chen, J. Shen, L. Wang, and J. Song, “Leveraging stacked denoising autoencoder in
934 prediction of pathogen-host protein-protein interactions,” in *Big Data (BigData Congress),
935 2017 IEEE International Congress on*, pp. 368–375, IEEE, 2017.

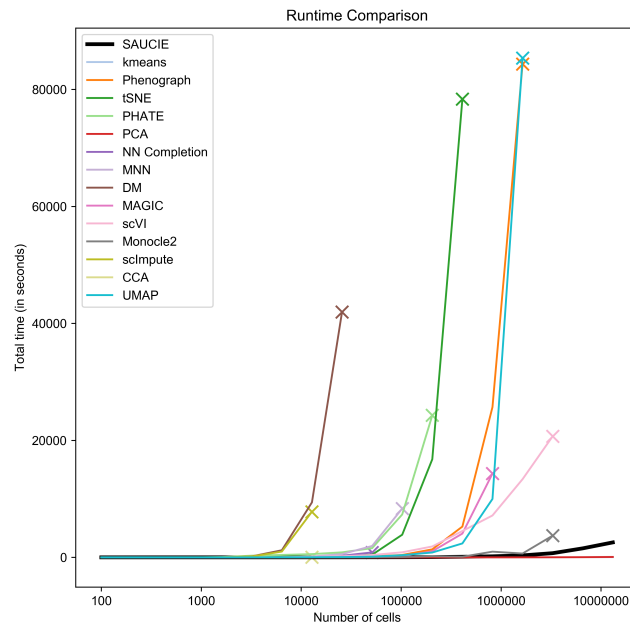
- 936 [13] L. Chen, C. Cai, V. Chen, and X. Lu, “Learning a hierarchical representation of the yeast
937 transcriptomic machinery using an autoencoder model,” *BMC bioinformatics*, vol. 17,
938 no. 1, p. S9, 2016.
- 939 [14] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and computational harmonic
940 analysis*, vol. 21, no. 1, pp. 5–30, 2006.
- 941 [15] D. van Dijk, J. Nainys, R. Sharma, P. Kathail, A. J. Carr, K. R. Moon, L. Mazutis, G. Wolf,
942 S. Krishnaswamy, and D. Pe’er, “Magic: A diffusion-based imputation method reveals
943 gene-gene interactions in single-cell rna-sequencing data,” *BioRxiv*, p. 111591, 2017.
- 944 [16] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing
945 robust features with denoising autoencoders,” in *Proceedings of the 25th international
946 conference on Machine learning*, pp. 1096–1103, ACM, 2008.
- 947 [17] K. R. Moon, J. Stanley, D. Burkhardt, D. van Dijk, G. Wolf, and S. Krishnaswamy, “Mani-
948 fold learning-based methods for analyzing single-cell rna-sequencing data,” *Current Opin-
949 ion in Systems Biology*, 2017.
- 950 [18] R. Lopez, J. Regier, M. Cole, M. Jordan, and N. Yosef, “A deep generative model for
951 single-cell rna sequencing with application to detecting differentially expressed genes,”
952 *arXiv preprint arXiv:1710.05086*, 2017.
- 953 [19] Y. Zhao, M. Amodio, B. V. Wyk, D. van Dijk, K. Moon, X. Wang, A. Malawista, M. E.
954 Cahill1, A. Desai, P. Parthasarathy, M. Ventataswamy, V. Ravi, P. Kumar, S. Krish-
955 naswamy, and R. R. Montgomery, “Dengue virus patients show distinct immune signa-
956 tures and retain immune response to infection with zika virus.” *in preparation*, 2018.
- 957 [20] M. Courbariaux, Y. Bengio, and J.-P. David, “Binaryconnect: Training deep neural net-
958 works with binary weights during propagations,” in *Advances in neural information pro-
959 cessing systems*, pp. 3123–3131, 2015.
- 960 [21] U. Shaham, K. P. Stanton, J. Zhao, H. Li, K. Raddassi, R. Montgomery, and Y. Kluger,
961 “Removal of batch effects using distribution-matching residual networks,” *Bioinformatics*,
962 vol. 33, no. 16, pp. 2539–2546, 2017.
- 963 [22] M. Setty, M. D. Tadmor, S. Reich-Zeliger, O. Angel, T. M. Salame, P. Kathail, K. Choi,
964 S. Bendall, N. Friedman, and D. Pe’er, “Wishbone identifies bifurcating developmental
965 trajectories from single-cell data,” *Nature biotechnology*, vol. 34, no. 6, p. 637, 2016.
- 966 [23] S. Chevrier, J. H. Levine, V. R. T. Zanutelli, K. Silina, D. Schulz, M. Bacac, C. H. Ries,
967 L. Ailles, M. A. S. Jewett, H. Moch, *et al.*, “An immune atlas of clear cell renal cell
968 carcinoma,” *Cell*, vol. 169, no. 4, pp. 736–749, 2017.

- 969 [24] E. Azizi, A. J. Carr, G. Plitas, A. E. Cornish, C. Konopacki, S. Prabhakaran, J. Nainys,
970 K. Wu, V. Kiseliovas, M. Setty, *et al.*, “Single-cell map of diverse immune phenotypes in
971 the breast tumor microenvironment,” *Cell*, vol. 174, no. 5, pp. 1293–1308, 2018.
- 972 [25] E. R. Zunder, E. Lujan, Y. Goltsev, M. Wernig, and G. P. Nolan, “A continuous molecular
973 roadmap to ipsc reprogramming through progression analysis of single-cell mass cytome-
974 try,” *Cell Stem Cell*, vol. 16, no. 3, pp. 323–337, 2015.
- 975 [26] K. Shekhar, S. W. Lapan, I. E. Whitney, N. M. Tran, E. Z. Macosko, M. Kowalczyk,
976 X. Adiconis, J. Z. Levin, J. Nemesh, M. Goldman, *et al.*, “Comprehensive classification
977 of retinal bipolar neurons by single-cell transcriptomics,” *Cell*, vol. 166, no. 5, pp. 1308–
978 1323, 2016.
- 979 [27] F. Paul, Y. Arkin, A. Giladi, D. A. Jaitin, E. Kenigsberg, H. Keren-Shaul, D. Winter,
980 D. Lara-Astiaso, M. Gury, A. Weiner, *et al.*, “Transcriptional heterogeneity and lineage
981 commitment in myeloid progenitors,” *Cell*, vol. 163, no. 7, pp. 1663–1677, 2015.
- 982 [28] A. Zeisel, A. B. Muñoz-Manchado, S. Codeluppi, P. Lönnerberg, G. La Manno, A. Juréus,
983 S. Marques, H. Munguba, L. He, C. Betsholtz, *et al.*, “Cell types in the mouse cortex and
984 hippocampus revealed by single-cell rna-seq,” *Science*, vol. 347, no. 6226, pp. 1138–1142,
985 2015.
- 986 [29] 10x Genomics, “10x Genomics Datasets.” <https://support.10xgenomics.com/single-cell-gene-expression/datasets>.
987
- 988 [30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blon-
989 del, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau,
990 M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,”
991 *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- 992 [31] J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, E. ad D. Amir, M. D. Tadmor,
993 O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder, R. Finck, A. L. Gedman, I. Radtke, J. R.
994 Downing, D. Pe’er, and G. P. Nolan, “Data-driven phenotypic dissection of AML reveals
995 progenitor-like cells that correlate with prognosis,” *Cell*, vol. 162, pp. 184–197, jul 2015.
- 996 [32] L. Haghverdi, A. T. Lun, M. D. Morgan, and J. C. Marioni, “Batch effects in single-
997 cell rna-sequencing data are corrected by matching mutual nearest neighbors,” *Nature*
998 *biotechnology*, vol. 36, no. 5, p. 421, 2018.
- 999 [33] A. Butler and R. Satija, “Integrated analysis of single cell transcriptomic data across con-
1000 ditions, technologies, and species,” *bioRxiv*, p. 164889, 2017.
- 1001 [34] M. Buttner, Z. Miao, A. Wolf, S. A. Teichmann, and F. J. Theis, “Assessment of batch-
1002 correction methods for scrna-seq data with a new test metric,” *bioRxiv*, p. 200345, 2017.

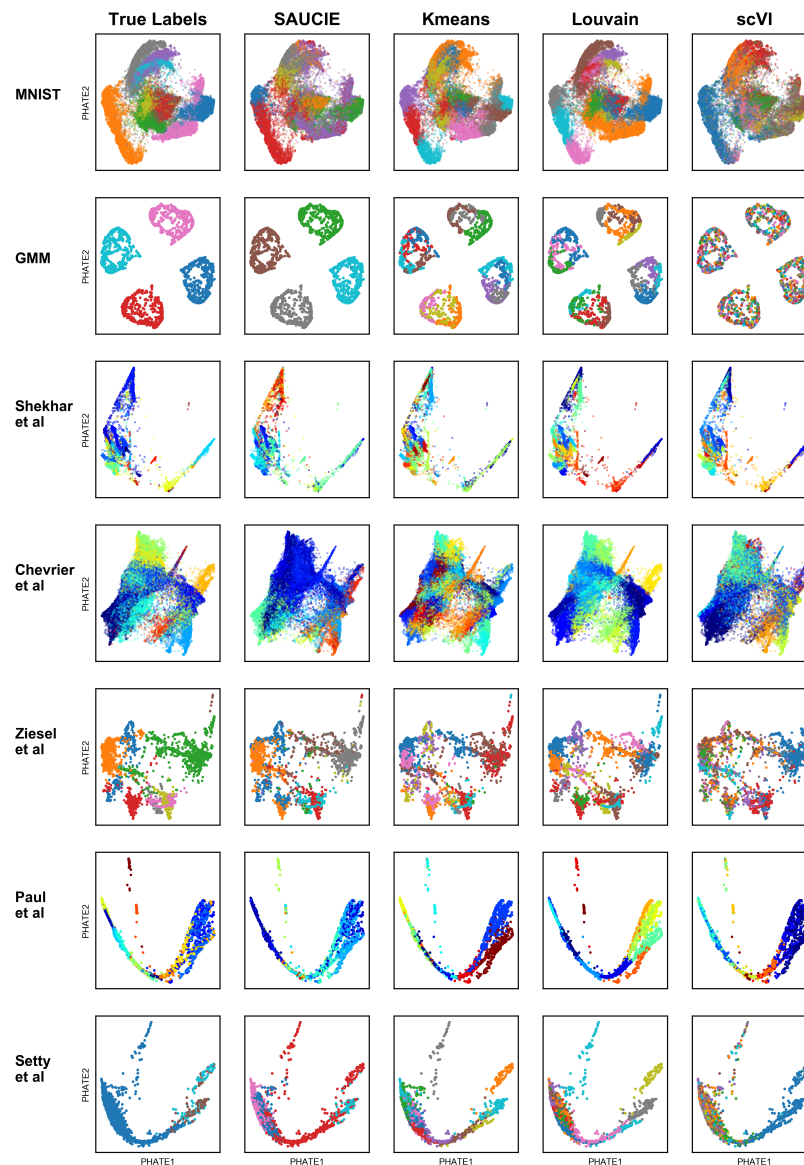
- 1003 [35] B. Luo and E. R. Hancock, “Iterative procrustes alignment with the em algorithm,” *Image*
1004 *and Vision Computing*, vol. 20, no. 5-6, pp. 377–396, 2002.
- 1005 [36] K. Lui, G. W. Ding, R. Huang, and R. McCann, “Dimensionality reduction has quantifi-
1006 able imperfections: Two geometric bounds,” in *Advances in Neural Information Process-*
1007 *ing Systems*, pp. 8461–8471, 2018.
- 1008 [37] A. Tsherniak, F. Vazquez, P. G. Montgomery, B. A. Weir, G. Kryukov, G. S. Cowley,
1009 S. Gill, W. F. Harrington, S. Pantel, J. M. Krill-Burger, *et al.*, “Defining a cancer depen-
1010 dency map,” *Cell*, vol. 170, no. 3, pp. 564–576, 2017.
- 1011 [38] L. C. Katzelnick, L. Gresh, M. E. Halloran, J. C. Mercado, G. Kuan, A. Gordon, A. Bal-
1012 maseda, and E. Harris, “Antibody-dependent enhancement of severe dengue disease in
1013 humans,” *Science*, vol. 358, no. 6365, pp. 929–932, 2017.
- 1014 [39] ScienceMag, *A new dengue vaccine should only be used in people who were previously*
1015 *infected, WHO says*, 2018 (accessed August 9, 2018).
- 1016 [40] A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller,
1017 P. Campbell, P. Carninci, M. Clatworthy, *et al.*, “Science forum: the human cell atlas,”
1018 *Elife*, vol. 6, p. e27041, 2017.
- 1019 [41] A. Panda, F. Qian, S. Mohanty, D. Van Duin, F. K. Newman, L. Zhang, S. Chen, V. Towle,
1020 R. B. Belshe, E. Fikrig, *et al.*, “Age-associated decrease in tlr function in primary hu-
1021 man dendritic cells predicts influenza vaccine response,” *The Journal of Immunology*,
1022 p. ji_0901022, 2010.
- 1023 [42] C.-Y. Tsai, K. H. Liong, M. G. Gunalan, N. Li, D. S. L. Lim, D. A. Fisher, P. A. MacAry,
1024 Y. S. Leo, S.-C. Wong, K. J. Puan, *et al.*, “Type i ifns and il-18 regulate the antiviral
1025 response of primary human $\gamma \delta$ t cells against dendritic cells infected with dengue virus,”
1026 *The Journal of Immunology*, p. 1303343, 2015.
- 1027 [43] B. Garcillán, A. V. Marin, A. Jiménez-Reinoso, A. C. Briones, M. Muñoz-Ruiz, M. J.
1028 García-León, J. Gil, L. M. Allende, E. Martínez-Naves, M. L. Toribio, *et al.*, “gd t lym-
1029 phocytes in the diagnosis of human t cell receptor immunodeficiencies,” *Frontiers in im-*
1030 *munology*, vol. 6, p. 20, 2015.
- 1031 [44] Y.-h. Chien, C. Meyer, and M. Bonneville, “ $\gamma \delta$ t cells: first line of defense and beyond,”
1032 *Annual review of immunology*, vol. 32, pp. 121–155, 2014.
- 1033 [45] E. Cimini, C. Castilletti, A. Sacchi, R. Casetti, V. Bordoni, A. Romanelli, F. Turchi, F. Mar-
1034 tini, N. Tumino, E. Nicastrì, *et al.*, “Human zika infection induces a reduction of ifn- γ
1035 producing cd4 t-cells and a parallel expansion of effector $v\delta 2$ t-cells,” *Scientific reports*,
1036 vol. 7, no. 1, p. 6313, 2017.

- 1037 [46] B. L. Rellahan, J. A. Bluestone, B. A. Houlden, M. M. Cotterman, and L. A. Matis, “Junc-
1038 tional sequences influence the specificity of gamma/delta t cell receptors.,” *Journal of*
1039 *Experimental Medicine*, vol. 173, no. 2, pp. 503–506, 1991.
- 1040 [47] J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, D. A. El-ad, M. D. Tadmor,
1041 O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder, *et al.*, “Data-driven phenotypic dis-
1042 section of aml reveals progenitor-like cells that correlate with prognosis,” *Cell*, vol. 162,
1043 no. 1, pp. 184–197, 2015.
- 1044 [48] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learn-*
1045 *ing Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- 1046 [49] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear em-
1047 bedding,” *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- 1048 [50] B. Nadler, S. Lafon, I. Kevrekidis, and R. R. Coifman, “Diffusion maps, spectral clus-
1049 tering and eigenfunctions of fokker-planck operators,” in *Advances in neural information*
1050 *processing systems*, pp. 955–962, 2006.
- 1051 [51] K. R. Moon, D. van Dijk, Z. Wang, W. Chen, M. J. Hirn, R. R. Coifman, N. B. Ivanova,
1052 G. Wolf, and S. Krishnaswamy, “Phate: A dimensionality reduction method for visualizing
1053 trajectory structures in high-dimensional biological data,” *bioRxiv*, p. 120378, 2017.
- 1054 [52] S. Lafon and A. B. Lee, “Diffusion maps and coarse-graining: A unified framework for
1055 dimensionality reduction, graph partitioning, and data set parameterization,” *IEEE trans-*
1056 *actions on pattern analysis and machine intelligence*, vol. 28, no. 9, pp. 1393–1403, 2006.
- 1057 [53] G. David and A. Averbuch, “Hierarchical data organization, clustering and denoising
1058 via localized diffusion folders,” *Applied and Computational Harmonic Analysis*, vol. 33,
1059 no. 1, pp. 1–23, 2012.
- 1060 [54] G. Wolf, A. Rotbart, G. David, and A. Averbuch, “Coarse-grained localized diffusion,”
1061 *Applied and Computational Harmonic Analysis*, vol. 33, no. 3, pp. 388–400, 2012.
- 1062 [55] L. Haghverdi, M. Buettner, F. A. Wolf, F. Buettner, and F. J. Theis, “Diffusion pseudotime
1063 robustly reconstructs lineage branching,” *Nature methods*, vol. 13, no. 10, p. 845, 2016.
- 1064 [56] R. Talmon and R. R. Coifman, “Empirical intrinsic geometry for nonlinear modeling and
1065 time series filtering,” *Proceedings of the National Academy of Sciences*, vol. 110, no. 31,
1066 pp. 12535–12540, 2013.
- 1067 [57] C. J. Dsilva, R. Talmon, R. R. Coifman, and I. G. Kevrekidis, “Parsimonious representa-
1068 tion of nonlinear dynamical systems through manifold learning: A chemotaxis case study,”
1069 *Applied and Computational Harmonic Analysis*, 2015.

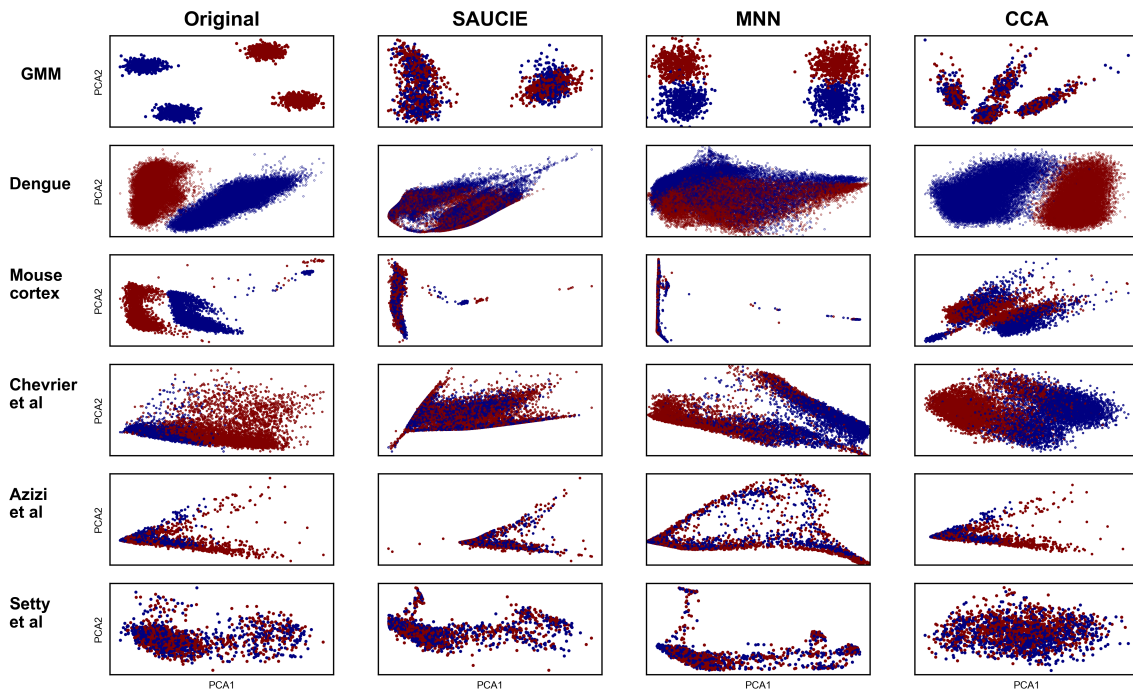
- 1070 [58] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, “On the number of linear regions of
1071 deep neural networks,” in *Advances in neural information processing systems*, pp. 2924–
1072 2932, 2014.
- 1073 [59] G. Alain and Y. Bengio, “What regularized auto-encoders learn from the data-generating
1074 distribution,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3563–3593,
1075 2014.
- 1076 [60] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, “Training generative neural networks via
1077 maximum mean discrepancy optimization,” in *UAI*, 2015.
- 1078 [61] K. Anand, G. Bianconi, and S. Severini, “Shannon and von neumann entropy of ran-
1079 dom networks with heterogeneous expected degree,” *Physical Review E*, vol. 83, no. 3,
1080 p. 036109, 2011.
- 1081 [62] D. Arpit, Y. Zhou, H. Q. Ngo, and V. Govindaraju, “Why regularized auto-encoders learn
1082 sparse representation?,” in *ICML*, 2016.
- 1083 [63] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *AISTATS*,
1084 2011.
- 1085 [64] H. Wang, Z. Hu, and Y. Zhao, “Kernel principal component analysis for large scale data
1086 set,” in *International Conference on Intelligent Computing*, pp. 745–756, Springer, 2006.
- 1087 [65] Y. Yao, D. M. Strauss-Albee, J. Q. Zhou, A. Malawista, M. N. Garcia, K. O. Murray, C. A.
1088 Blish, and R. R. Montgomery, “The natural killer cell response to west nile virus in young
1089 and old individuals with or without a prior history of infection,” *PloS one*, vol. 12, no. 2,
1090 p. e0172625, 2017.
- 1091 [66] Y. Yao, R. Liu, M. S. Shin, M. Trentalange, H. Allore, A. Nassar, I. Kang, J. S. Pober, and
1092 R. R. Montgomery, “Cytof supports efficient detection of immune cell subsets from small
1093 samples,” *Journal of immunological methods*, vol. 415, pp. 1–5, 2014.



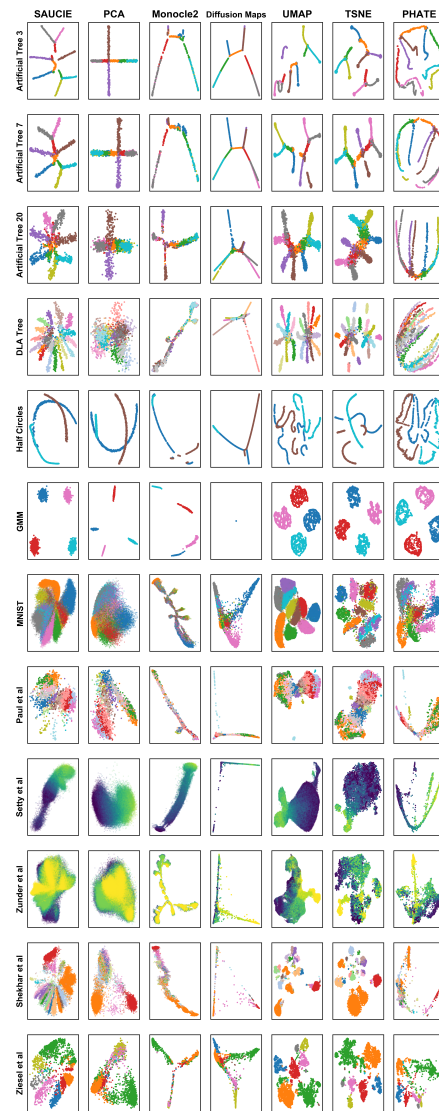
Supplemental Figure S1: Comparison of runtimes on an increasing number of points. The number of points is represented on the horizontal axis and the time in seconds the method took to complete is on the vertical axis. If a method ran out of resources and could not complete a run for a certain number of points, that is demarcated with an 'x' and no further time points were attempted for that method. SAUCIE is the fastest method besides PCA and kmeans.



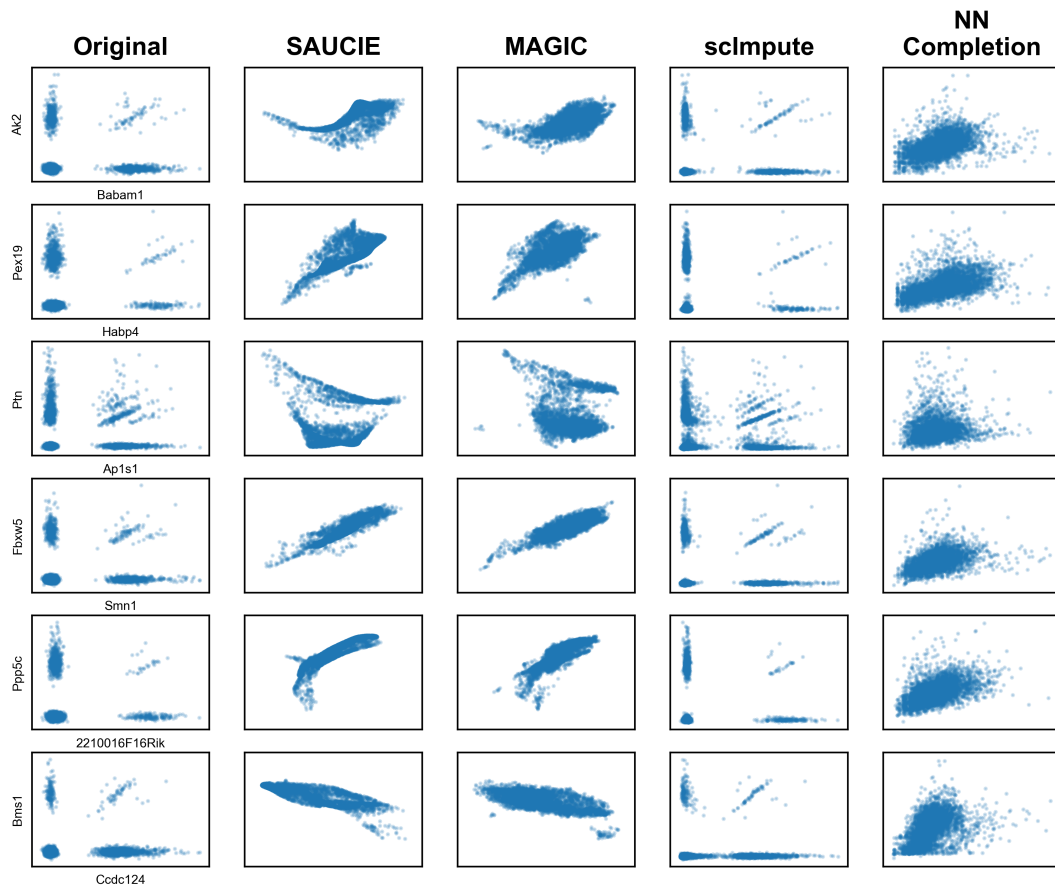
Supplemental Figure S2: A comparison of the SAUCIE clustering to other clustering methods on artificial and real data. Rows show the different datasets. Along with the first two artificial datasets, there are two CyTOF datasets and three scRNA-seq datasets. Columns show the different clustering methods. From left to right: True “ground truth” labels, SAUCIE, kmeans, Phenograph, scVI.



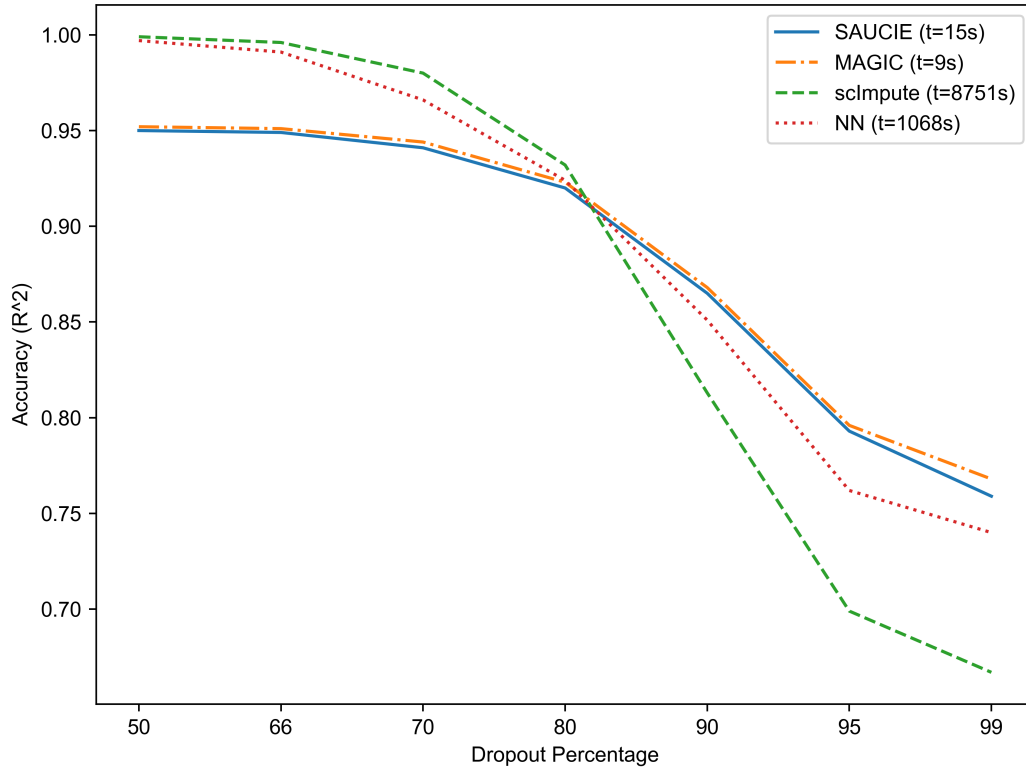
Supplemental Figure S3: A comparison of batch correction with SAUCIE to other methods on an artificial dataset, two technical replicates from the dengue CyTOF data, non-technical replicates on scRNA-seq batches from mouse cortex, and then public data from Chevrier et al, Azizi et al, and Setty et al. Rows show the different datasets. Columns show the different batch correction methods. From left to right: The original data prior to batch correction, SAUCIE, mutual nearest neighbors (MNN), canonical correlation analysis (CCA).



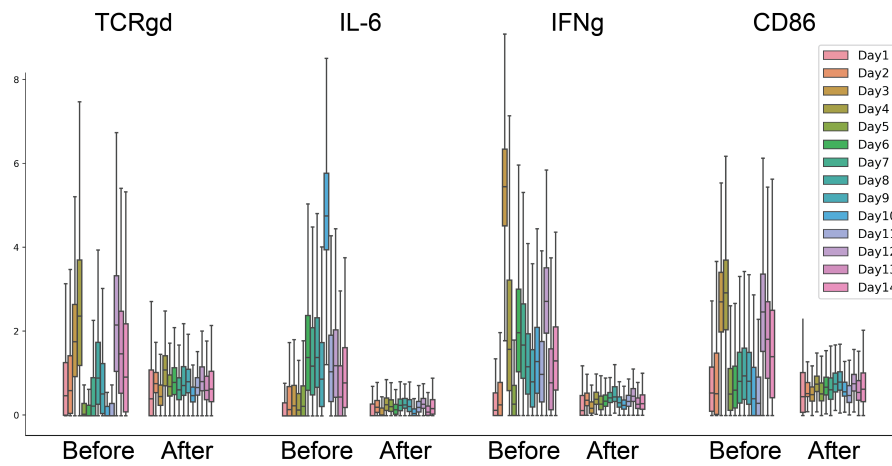
Supplemental Figure S4: A comparison of the SAUCIE visualization to other methods on a number of artificial and real datasets. The columns show the different methods. From left to right: SAUCIE, PCA, Monocle2, Diffusion Maps, UMAP, tSNE, PHATE. The rows show the different datasets. From top to bottom: Artificially generated trees with varying amounts of noise, random tree generated with diffusion limited aggregation (DLA), intersecting half circles, Gaussian mixture model, MNIST, scRNA-seq hematopoiesis from Paul et al. 2015 [27], CyTOF T cell development from Setty et al. 2016 [22], CyTOF ipsc from Zunder at al. 2016 [25], scRNA-seq retinal bipolar cells from Shekhar et al. 2016 [26], scRNA-seq mouse cortex from Zeisel et al. 2015 [28].



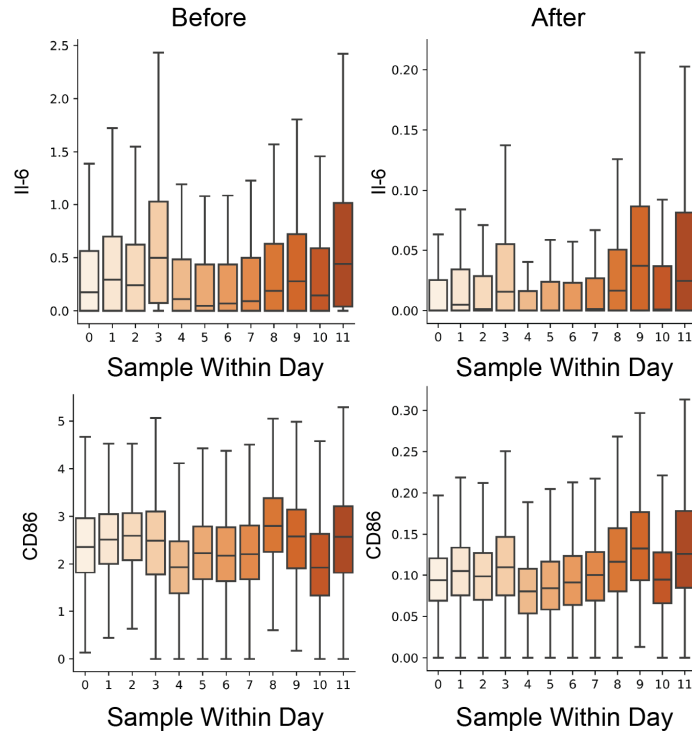
Supplemental Figure S5: A comparison of imputation methods including SAUCIE. Several gene-gene associations are shown from the 10x mouse cortex dataset. From left to right: The original (sparse) data, data after imputation with SAUCIE, MAGIC, scImpute, and nearest neighbor completion.



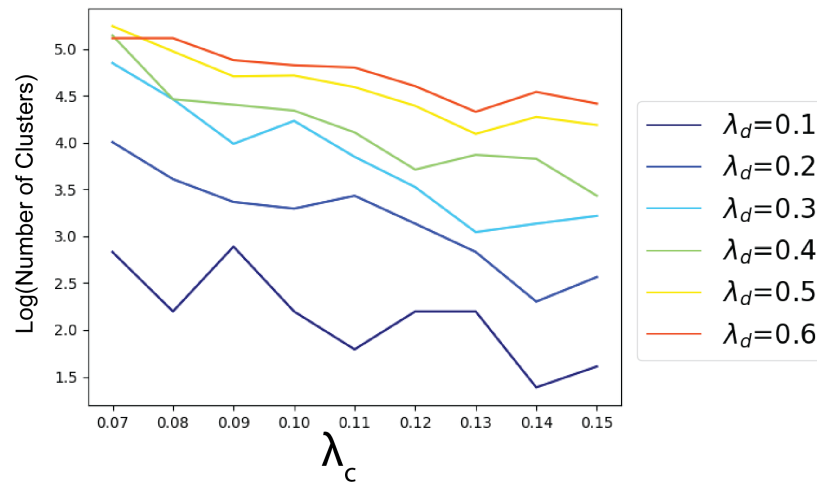
Supplemental Figure S6: A comparison of imputation with SAUCIE to other methods on the simulated dropout experiment. Increasing amounts of dropout are along the horizontal axis from left to right, and the accuracy of each method as measured by R^2 is along the vertical axis. The time each method took to complete is in the legend in seconds.



Supplemental Figure S7: Four select marker abundances with samples grouped by day they were run on the cytometry instrument, with each day having fourteen distinct samples in the group. For each marker, the fourteen samples before batch correction are shown to the left of the same fourteen samples after batch correction.

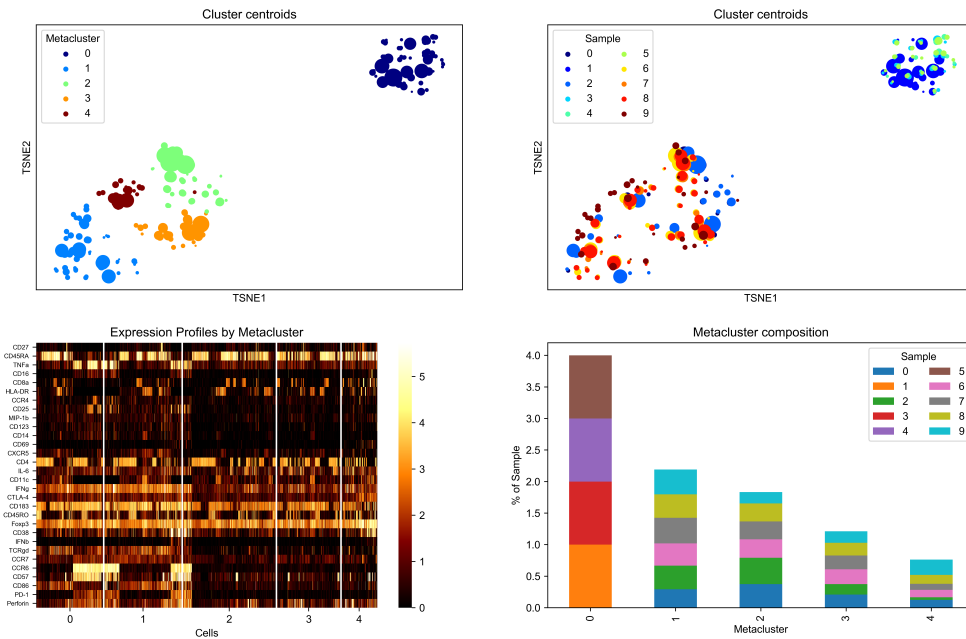


Supplemental Figure S8: Histograms of marker expression (top: IL-6, bottom: CD86) of samples run together on the cytometry instrument on day two, separated by sample. The values for each sample and marker are shown before SAUCIE batch correction (left) and after SAUCIE batch correction (right).



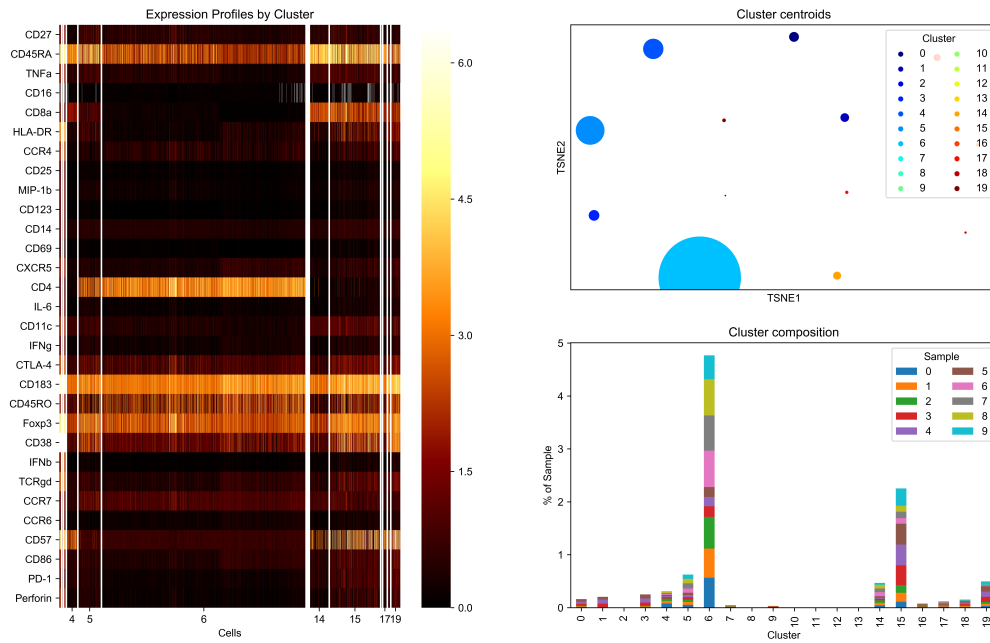
Supplemental Figure S9: The granularity of the clustering, as measured by the total number of clusters found. Each line represents a fixed value of λ_d as λ_c increases from left to right.

10 Samples from Dengue Data



Supplemental Figure S10: An illustration of the metaclustering process on the dengue dataset. Top left: cluster centroids embedded by tSNE and colored by metacluster, sized according to the number of cells in each cluster.. Top right: cluster centroids colored by sample, also sized according to the number of cells in each cluster.. Bottom left: a cell-level heatmap of expression grouped by metacluster. Bottom right: the composition of each metacluster by sample.

10 Samples from Dengue Data



Supplemental Figure S11: An illustration of the SAUCIE pipeline on the dengue dataset. Left: cell-level heatmap of expression grouped by cluster. Top right: cluster centroids embedded by tSNE, sized according to the number of cells in each cluster. Bottom right: the composition of each cluster by sample.

	SAUCIE	kmeans	Phenograph	scVI
MNIST	0.8822/-0.0165	0.8805/0.0535	0.9316/0.0180	0.8592/0.0084
GMM	0.7512/0.8162	0.8917/0.3097	0.9302/0.2662	0.9030/-0.0626
Shekhar et al	0.9662/-0.0602	0.8530/0.0753	0.8981/0.0868	0.93139/0.0593
Chevrier et al	0.9347/-0.3761	0.9517/0.0085	0.9258/-0.0452	0.9330/-0.1967
Zeisel et al	0.8663/-0.1881	0.9138/0.1135	0.9209/0.1529	0.9085/-0.1238
Paul et al	0.8854/-0.3060	0.8930/0.0249	0.8819/0.1802	0.8839/-0.0540
Setty et al	0.6860/0.0425	0.8704/0.0377	0.8912/0.0147	0.8591/-0.0718

Table 1: A comparison of modularity (left) and silhouette (right) scores of each of the clustering algorithms on each dataset.

	Original	SAUCIE	MNN	CCA
GMM	0.999/—	0.630/0.629	0.526/0.620	0.510/0.998
Dengue	0.999/—	0.593/0.532	0.998/0.512	0.992/0.765
Mouse cortex	0.994/—	0.530/0.498	0.898/0.485	0.836/0.923
Chevrier et al	0.880/—	0.540/0.934	0.787/0.232	0.835/0.346
Azizi et al	0.621/—	0.512/0.180	0.560/0.205	0.621/0.000
Setty et al	0.518/—	0.504/0.064	0.514/0.067	0.523/0.698

Table 2: A comparison of mixing (left) and Procrustes (right) scores of each of the batch correction algorithms on each dataset.

	SAUCIE	PCA	Monocle2	DM	UMAP	TSNE	PHATE
Artificial Tree 3	0.993	0.956	0.935	0.963	0.967	0.968	0.947
Artificial Tree 7	0.994	0.951	0.921	0.980	0.986	0.990	0.971
Artificial Tree 20	0.948	0.896	0.854	0.940	0.938	0.940	0.939
DLA Tree	0.865	0.817	0.725	0.819	0.836	0.845	0.847
Half Circles	0.975	0.970	0.940	0.937	0.958	0.946	0.925
GMM	0.999	0.972	0.953	0.500	0.992	0.992	0.969
MNIST	0.777	0.801	0.744	0.498	0.718	0.506	0.507
Paul et al	0.944	0.948	0.896	0.807	0.842	0.865	0.856
Setty et al	0.882	0.870	0.839	0.508	0.501	0.501	0.491
Zunder et al	0.939	0.903	0.884	0.505	0.522	0.513	0.510
Shekhar et al	0.942	0.908	0.918	0.506	0.863	0.508	0.496
Ziesel et al	0.952	0.914	0.903	0.943	0.909	0.881	0.905

Table 3: A comparison of precision-recall area-under-the-curves (AUCs) for each of the visualization algorithms on each dataset.