Discovering Rare Cell Types through Information-based Dimensionality Reduction

Short title: Rare cell type preserving dimensionality reduction

Benjamin DeMeo[2,3] and Bonnie Berger[1,3,*]

[1]Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139, USA

[2]Department of Biomedical Informatics, Harvard University, Cambridge, MA 02138, USA

[3]Department of Mathematics, MIT, Cambridge, MA 02139, USA

Abstract

Dimensionality reduction is the prevailing method for summarizing the complex transcriptomic landscape of single cells, but current approaches favor large cellular populations defined by many genes, and consequently fail to capture the full transcriptional diversity of a sample. Here we present Shannon components analysis (SCA), a dimensionality reduction technique that leverages the information-theoretic notion of "surprisal," and demonstrate its ability to accurately incorporate information about small-scale transcriptional variability into dimensionality reduction of single-cell RNA to enable downstream analyses at unprecedented resolution. On published data, SCA enables recovery of known cell types, both rare and common, with accuracy superior to existing methods. Furthermore, SCA reveals small and subtle subpopulations in complex tissues, including gamma-delta and MAIT cells among cytotoxic T-lymphocytes, which are invisible to existing pipelines. In addition to dimensionality reduction, we leverage SCA's information-theoretic framework on multi-batched data to separate biological signal from systemic batch effects, achieving robust, interpretable, and state-of-the-art batch correction. SCA's

*Correspondence: bab@mit.edu

information-theoretic paradigm offers a principled and effective approach to signal extraction in any high-dimensional data analysis pipeline.

Introduction

Single-cell RNA sequencing (scRNA-seq) produces transcript counts for individual cells, enabling fine-grained analyses of biological tissues. Current technologies allow researchers to record tens of thousands of distinct transcripts in millions of cells. However, due to the relatively low number of transcripts in each cell, and to limited molecule capture efficiency and sequencing depth, the resulting data is noisy and challenging to analyze [1]. Biologically informative patterns, often defined by just a handful of transcriptional programs, are frequently obscured by noisy measurements of transcripts of interest, and by thousands of other irrelevant or misleading transcripts. Consequently, dimensionality reduction, whereby the original transcript counts are represented by a smaller set of variables which encompass the dataset's overall structure, has become a cornerstone of modern scRNA-seq analysis pipelines, as demonstrated by the popularity of Principal Component Analysis (PCA) [2], Independent Components Analysis (ICA) [3], and recent related methods [4, 5, 6, 7]. These reductions are often performed as a first "de-noising" step before informing other downstream analyses (e.g. cell type discovery and trajectory analysis) [1].

Despite its utility, dimensionality reduction carries a substantial and often-overlooked risk of losing essential information about a sample's cellular diversity, especially in rare subpopulations and those defined by a small number of genes. For example, PCA produces features with maximal variance across the entire dataset; yet, as rare cell populations constitute a small fraction of the dataset, they often cannot be recovered from PCA alone. Despite their scarcity, such populations often play important roles in biological and pathological phenomena ranging from immunological memory to cancer pathogenesis. Identifying and characterizing these populations is a key motivator of many single-cell experiments, so the failure of existing techniques to capture them is a substantial limitation in scRNA-seq analysis. In the case of PCA, The assumption that highly-variable features describe cells of interest underpins this shortcoming and illustrates the need for a more nuanced approach.

A variety of approaches have arisen to detect address this issue. GiniClust [7] subsets to genes with high Gini index, a measure of uneven distribution of expression, and clusters the resulting data. The success of this approach relies on the assumption that only high-Gini genes are important, which we shall show is not always the case. RaceID [8] computes an initial k-means clustering, which is then refined by identifying outliers in each cluster and re-assigning cells. Similarly, CellSIUS [9] begins with a coarse clustering, identifies genes with a cluster-specific bimodal distribution, assembles these genes into modules, and identifies subclusters based on module expression. Both of these methods rely on an initial clustering; however, an accurate initial clustering may be difficult to obtain when cell types of interest are rare or subtly defined. Furthermore, purely clustering-based approaches are unsuitable when cells form more continuous transcriptional structures, such as developmental trajectories, which do not neatly partition. FiRE [10] circumvents the limitations of clustering by assigning each cell a continuous rarity score based on its degree of isolation, which is determined by locality-sensitive hashing. However, measuring a cell's isolation depends on a meaningful measure of cell-to-cell distance, which is difficult for cell types that are defined by a very small fraction of the transcripts. Consequently, as we shall show, FiRE fails to highlight such populations.

Here we introduce SCA (Shannon Components Analysis), an information-theoretic dimensionality reduction method that identifies statistically informative signals in transcriptional data to enable deeper insight into complex tissues. SCA newly leverages the notion of *surprisal*, whereby less probable events are more informative when they occur, to assign an *information score* to each transcript in each cell. Rare transcripts or unique transcriptomic patterns defining rare cell types are marked as statistically significant where they are expressed and given high scores, drastically boosting their influence. From these information scores, SCA constructs a set of *Shannon components* with maximal information content, encompassing statistically salient patterns across the dataset. These components act as a reduced set of features which robustly describe transcriptional structures in the data, and which can be input to a wide range of downstream analyses. Each Shannon component is a weighted sum of genes, enabling straightforward biological interpretation, as we shall show.

To demonstrate the utility of our approach, we ran SCA on real and simulated data with subtle differences

between cellular populations. We show that SCA enables detection of small and clinically-relevant populations, such as gamma-delta T-cells and Mucosal-associated Invariant T-cells, which are invisible to existing analytic pipelines. At the same time, SCA reductions better capture larger-scale differences between more common cell types, enabling multi-resolution analysis without the need for re-clustering. In addition, we found that SCA's information scores accurately distinguish biologically informative genes in multi-batched data, enabling simple and interpretable batch correction via feature selection. Despite its simplicity, this approach achieves state-of-the-art performance comparable to that of Harmony [11] and scanorama [12].

SCA is broadly applicable, requiring no information aside from the transcript counts, and does not assume that the data admits a natural clustering structure. The output components have a clear linear relationship with the original transcripts, facilitating straightforward verification and interpretation of their biological meaning. By formalizing a new notion of the "structure of the data" and providing a means to extract it, SCA allows high-throughput RNA-sequencing to take broad steps towards realizing its full potential in the fine-grained study of complex tissues. More broadly, we believe that SCA's information-theoretic approach is a mathematically sound and empirically effective answer to the ever-present problem of extracting signal from noisy and high-dimensional measurements, both in single-cell transcriptomics and elsewhere.

Results

Overview of SCA. SCA takes as an input the transcript count matrix of N cells by M genes and converts it to a matrix that encodes the significance of each gene in a cell. We call this matrix the *Information Score Matrix*, and denote by S. For each cell and gene in this matrix, SCA identifies the k nearest neighbors of the cell and performs a two-tailed binomial test to see whether the gene is expressed in more or fewer of the neighbors than expected, given the gene's global probability of expression across all cells (Figure 1a). The resulting p-value is then corrected for multiple testing using a family-wise error rate (FWER) correction (Methods). The negative logarithm of this corrected p-value can be interpreted as the Shannon information of the a gene's local expression, and provides the magnitude of the information score for that gene in a cell.

The sign of the information score is positive if the gene is over-expressed in the cell's neighborhood, and negative if it is under-expressed.

This strategy gives genes high positive scores where they are markers, scores near zero where they represent noise, and low negative scores where they are conspicuously absent (Figure 1b). Rare genes receive higher scores in regions where they are expressed, representing greater *surprisal* of encountering that gene. Likewise, genes that are near-universally expressed receive strongly negative scores in regions that do not express them. Thus, small populations marked by the presence of a rare gene or the absence of a common one have a greater impact on the information score matrix.

We next sought to identify the combinations of genes with high Shannon information. These are described by the top right-eigenvectors of the information score matrix S, i.e. by its principal component vectors, $v_1, v_2, ..., v_m$ (Methods, Supplementary Information). For a given target dimensionality D, the metagenes represented by $v_1, v_2, ..., v_D$ act as the features for the dimensionality-reduced transcript count matrix X. We call these features *Shannon Components* (SCs) of the input matrix. We linearly project the input dataset to the span of $v_1, .., v_D$, obtaining an $N \times D$ matrix which is then used in downstream analyses (Figure 1a). Figure 1c shows an example of SCA metagene expression over a set of human cells profiled using Smart-seq 3 [13].

Although the process that generates $v_1, ..., v_D$ is nonlinear, requiring nearest-neighbor graphs and binomial score computation, SCA's output is a linear projection of its input onto their span. This places SCA firmly in the linear category, together with PCA and ICA; indeed, for fixed D, the coordinate systems defined by SCA and by these methods are related by rotation in the original high-dimensional space. This linearity restricts the ability of SCA to distort the data, and lends interpretability to the resulting features, as we shall show.

We next sought to address how best to choose a metric for determining the k-nearest neighbors of a given cell which are used to compute the information scores. By default, SCA uses Euclidean distance in PCA space, with user-configurable parameters. However, since we found that SCA's reductions better reflect the structure of the data, we can use an initial SCA reduction to define nearest neighbors in a second application

of SCA, generating another dimensionality reduction on the same dataset. This iterative bootstrapping process can be repeated arbitrarily many times to further boost the signal of rare populations, and improve the discriminative power of the representation. (Figure 2b). Importantly, regardless of the number of iterations, SCA's output remains a linear projection of its input – iteration simply refines the subspace onto which SCA projects. In practice, we observe improved performance with more iterations (Figure 2).

SCA recovers rare populations from noisy synthetic data To test SCA's power to recover rare cell populations in the presence of noise, we generated a synthetic dataset with 500 cells and 10,020 genes. 20 "marker" genes are expressed uniformly in a subpopulation of 25 "rare" cells (5% of the total), and in a random 15% of the remaining cells. Each of the remaining 10,000 genes is expressed in a randomly-selected 20% of the cells, adding further noise. This dataset was designed to mimic a challenging biological scenario with a rare population for which the vast majority of genes are not relevant, and where the marker genes may be sporadically expressed outside of the populations they mark. We reduced this dataset using a standard PCA pipeline, using ICA, and using SCA with 1-4 iterations (Methods). For each reduction, we assessed separation visually using UMAP coordinates [5], and quantitatively by assessing the silhouette score [14]. For further comparison, we also tested four methods for rare cell type recovery: GiniClust [7], CellSIUS [9], and FiRE [10] (See Methods for details and parameters).

SCA drastically outperforms PCA and ICA in recovering the rare population (Figure 2b). PCA with 50 components fails to separate the two populations (silhouette score 0.079), since the noisy genes account for more of the global variance and receive high loadings. ICA with 50 components likewise fails to separate the populations (silhouette score again 0.079). On the other hand, SCA with 50 components produces a very good separation after one iteration, and a near-perfect separation after two, with substantial improvement in the silhouette score. This improvement is due to SCA correctly identifying the marker genes as having high information, and thus giving them higher impact on the final reduction; indeed, the first Shannon component clearly discriminates the two clusters, whereas the first principal component does not (Figure 2c).

GiniClust3 did not identify any of the marker genes as having significantly high Gini index; consequently,

it combined all cells into a single cluster. Indeed, marker genes have similar Gini coefficient to noisy genes in this data (Figure 2d, left). Similarly, CellSIUS did not identify any genes with bimodal distribution, and failed to generate a clustering. FiRE assigned only four distinct scores, which were uncorrelated with the two populations (Figure 2d, right).

SCA distinguishes known cell types in real data We next tested whether SCA's reductions better detect known populations of cells in a real, large-scale single cell dataset. To test this, we obtained a CITE-seq immune dataset from Hao et. al. [15], in which hundreds of thousands of human PBMCs from 8 human donors were subjected in parallel to transcriptomic profiling and to surface receptor profiling with a panel of 228 antibodies. The authors use both modalities, and input from human experts, to produce a confident cell type classification, which we take as a ground truth. Notably, the classification includes some T-cell populations which the authors were not able to detect through analysis of the transcriptomic data alone, such as gamma-delta and MAIT cells, offering a good challenge for SCA. We subset the data to T-cells, yielding 73,000 cells across all donors. To circumvent batch effects, we separate the data by donor and assess performance in each donor individually.

UMAP plots and F1 score analyses for Patient 1 in Figures 3a and 3b. F1 scores for each cell type in all patients are shown in Figure 3c, and UMAP plots for the remaining patients are provided in Supplementary Figure 1. Compared to PCA and ICA, SCA consistently generates more structured UMAP plots downstream, with clearer visual separation between cell types.

We hypothesized that this separation leads to more accurate clusterings downstream. To test this, we performed Leiden clustering [16] on each representation with resolution 1.0, and computed the adjusted mutual information (AMI) between the Leiden clusters and the true cell labels [17]. We also compared against clusters returned by GiniClust3 [7], CellSIUS [9], and RaceID [8]. Across all patients, the SCA-derived Leiden clusterings achieved the highest AMI with the true cell labels (Figure 3). Thus, SCA representations enable more accurate recovery of cellular populations in a sample.

We next assessed the accuracy with which SCA and other methods detect each of the known cellular

populations. To do this, we identify the subset of clusters which best identifies the target population, and report the F1 score (Methods). If the clustering perfectly separates the target population from the other cells, this yields a score of 1, whereas clusters which mix the target population with other cells decrease the score. We use this scoring metric to assess Leiden clusterings computed from PCA, ICA, and SCA embeddings with 50 components each, as well as the clusterings computed by GiniClust3, CellSIUS, and RaceID. For FiRE, which outputs a continuous rarity score, we consider sets containing the top k highest-scoring cells for all values of k ranging from 1 to the full size of the dataset, and report the highest F1 score obtained by any such set. For further details of the implementations and parameters used for each method, see Methods.

Across all donors, we find that clusterings derived from SCA recover most cell types with superior accuracy, with many cell types recoverable only from SCA (Figure 3c). For example, in Patient 1, SCA recovers CD4 cytotoxic T-lymphocytes (CD4 CTL) with F1 score 0.97, where no other method achieves a score higher than 0.61. We observe similar improvements across all patients, with SCA often recovering rare cell types (e.g. MAIT cells, Gamma-delta T-cells, and CD4+ Effector Memory T-cells) where existing methods do not (Figure 3c). Thus, SCA boosts the signal of both rare and common cell types, enabling accurate detection of both in downstream analyses.

SCA captures biological variability in samples We next examined whether SCA can isolate meaningful biological patterns to find novel subpopulations in published data which appear homogeneous when analyzed using existing pipelines. To this end, we applied SCA to a collection of 307 cytotoxic T-cells from an immune dataset profiled using Smart-seq 3 (SS3), which offers excellent read quality at isoform resolution [13]. We first performed a standard *in silico* clustering analysis using SCA, with PCA and ICA for comparison. From the top 20 components of each reduction, we computed a Euclidean 15-nearest neighbor graph using scanpy [18], followed by Leiden clustering [16] and UMAP visualization [5].

The analysis using SCA revealed 11 well-isolated Leiden clusters, clearly visible in the UMAP plot (Figure 4a). By contrast, PCA and ICA-based analysis respectively yielded 8 and 6 clusters, which are less distinct. To validate and characterize SCA's clustering, we assessed concordance with known marker

genes of important T-cell populations, and find that SCA's clusters correspond to well-known cell types (Figure 4b). For example, Gamma-delta ($\gamma\delta$) T-cells by definition express the gamma- and delta- T-cell receptors, represented in this dataset by the genes *TRGV9, TRDV2, TRDC* and *TARP* [19]. Clusters 0, 4, and 7 in the SCA-based analysis robustly capture these populations (Figure 4b). We observe similar clustering concordance for markers of MAIT cells (cluster 1), T-reg cells (clusters 6 and 8), and T-helper cells (cluster 5). These populations are also much more visually obvious in the SCA-derived UMAP plots as compared to PCA and ICA (Figure 4c), facilitating easy discovery in exploratory analyses.

For a quantitative comparison, we assessed the ability of several methods to recover the MAIT and gamma-delta subpopulations. As a ground truth, we identified gamma-delta cells as those expressing at least three of the four marker genes present in the data (*TRGV9, TRDV2, TRDC* and *TARP*); and MAIT cells as those expressing at least two of the three MAIT markers (*SLC4A10*, *TRAV1-2*, and *LTK*). This characterization gives robustness to individual gene dropouts whilst uniquely defining the populations. The output of each rare cell detection method was then converted into an F1 score quantifying the method's ability to detect each subpopulation, as described in the preceding section and in Methods. As shown in Figure 4e, SCA achieves high F1 scores for gamma-delta and MAIT cell recovery (0.921 and 0.899 respectively), whereas no existing method achieves an F1 score higher than 0.79 for gamma-delta cells and 0.7 for MAIT cells.

We suspected that SCA's superior performance stems from its ability to enrich the signal of a small number of biologically informative genes, and capture them in its leading components. To verify this, we examined the gene loadings of each Shannon Component. Figure 4d shows loadings for the genes with highest impact on the two leading Shannon components. The first Shannon component has the highest positive loadings in the three MAIT marker genes, and can itself be interpreted as an MAIT cell marker. Likewise, the second Shannon Component marks Gamma-delta T-cells, with high loadings in three gamma- and delta- receptors.

Thus, SCA enables the detection of small cellular populations of biological, clinical, and methodological importance which are not recovered through standard pipelines, via interpretable and biologically meaningful features.

SCA corrects systemic batch effects. Integrating data from different technical and biological replicates is a central problem in scRNA-seq, and the goal of many existing methods [12, 11, 20, 21, 22]. Batch effects arise from a variety of technical factors, such as capture efficiency, sample quality, and library size. We hypothesized that many batch effects occur uniformly across each sample, producing a noisy signal independent of cell type. Since such signals have little impact on SCA's score matrix, we sought to determine whether SCA could remove systemic batch effects simply by filtering genes based on their scores.

We implemented SCA-based batch correction by keeping only those genes which are highly informative in at least one batch. We applied SCA separately in each batch for three iterations, and kept only genes with high mean Shannon information score across all batches (Methods). As with other SCA functions, this approach is conceptually very simple and easy to interpret (it simply filters genes), and has limited capacity to distort the data. Other methods vary in the degree of batch correction, and may over-correct, leading to variable performance which can be difficult to evaluate (Figure 5).

We ran SCA's integration protocol on two datasets, keeping the top 200 top-scoring genes. On 293T/Jurkat cell mixtures from the Scanorama paper [12], SCA successfully merged the two Jurkat cell populations across batches, performing similarly to Scanorama [12] and Harmony [11] (Figure 5a). Simply taking the top 200 most variable genes failed to achieve this, showing that variable genes are not necessarily informative.

On another dataset consisting of cytotoxic T-lymphocytes from 8 donors [23], SCA again merges cells from different batches, whilst preserving major clusters (Figure 5b). Here, Scanorama and Harmony produce drastically different embeddings: Scanorama merges several small clusters on the periphery, whereas Harmony keeps them separate. SCA's embedding merges some of these clusters while keeping others separate. Notably, taking the top highly variable genes completely loses the finer structural details, reflecting the discriminative power of rare, low-variance genes.

The variability in the outputs of integration methods on this dataset is the result of variable assumptions and stylistic decisions. More aggressive approaches risk merging cells from different biological populations, and losing information. SCA shows that state-of-the-art performance can be achieved via basic feature selection, which is inherently simpler, more conservative, and easier to interpret than existing approaches.

Discussion

SCA provides a mathematically principled approach to extracting signal from noise in high-dimensional datasets that is powerful, interpretable, and broadly applicable. Its power derives from a novel, locality-sensitive, and statistically sound assessment of the significance of each transcript count, amplifying true transcriptional differences whilst reducing transcriptional noise. Rare populations and small-scale structures receive higher scores, enabling their discovery in downstream analyses.

SCA's information scores are similar in principle to Inverse Document Frequency (IDF), a normalization approach widely used in text processing and in some single-cell applications, whereby each feature (gene) is weighted by the logarithm of its inverse frequency. Like GiniClust, this approach is only gene-specific, assigning genes the same importance wherever they are expressed. By incorporating counts from local neighborhoods of each cell, SCA allows genes to have variable scores across the dataset, achieving high-magnitude scores where they are discriminative and near-zero scores where they are noise (Figure 1b). This reflects true biology, where genes may be expressed sporadically across the entire dataset, but mark informative distinctions only within a small subpopulation.

SCA is also conceptually similar to surprisal analysis [24], which compares observed data to a pre-computed equilibrium state to identify meaningful deviations. Originally developed for thermodynamics, these methods have recently found use in the bulk-transcriptomic analysis of biological systems in flux, such as cancer cells undergoing epithelial-to-mesenchymal transition [25] and carcinogenesis [26]. In our case, the "equilibrium state" is represented by the binomial null hypothesis of uniform random gene expression, from which deviations are computed locally. This newly enables detection not just of differences between two pre-defined samples, but between the variety of cell types within a single sample.

SCA is linear, and therefore related to other linear dimensionality reduction methods (e.g. PCA, ICA) by change of basis, which has the advantage of being more interpretable and sample-efficient (able to work with small amounts of data). Intuitively, SCA changes the "perspective" from which the data is viewed. It is remarkable, then, that SCA's reductions look so different in downstream analyses from those of PCA and ICA (see e.g. Figure 4a). This is possible because high-dimensional space offers a far wider variety of

perspectives than the three-dimensional space we often think in, giving linear methods more richness than they are usually credited for.

Single-cell RNA sequencing arose from the need to understand biological tissues at cellular resolution, and as the field evolves, it becomes ever more important to recover clinically relevant cellular populations. SCA offers a robust, scalable, and interpretable framework for doing so.

Software availability. SCA is implemented in Python, and offers integration with scanpy, a popular scRNA processing framework. It is freely available at `github.com/bendemeo/sca`.

Methods

Synthetic Data Experiments The synthetic dataset analyzed in Section 2.2 was generated using numpy [27] and scanpy [18]. Expression levels are binarized, spanning 10,020 genes and 500 cells. 10,000 of the genes are expressed uniformly at random throughout the entire population, with each cell having a 20% chance of expressing each gene. Each of the 20 "marker genes" is expressed constitutively in a group of 25 "rare" cells, and in a randomly-chosen 15% of the remainder of the data. The rare cells are indicated by color in Figure 2b.

For PCA representations, we compute 50 PCs using scanpy's `pca` function [18], which in turn uses the implementation of `scikit-learn` [28], with default parameters. We ran SCA with a neighborhood size of 15 for 1-4 iterations, starting with the PCA embedding and keeping 20 Shannon components after each iteration. In each representation, we computed the 15-nearest-neighbor graph using cosine distance, and used the resulting graphs to generate the UMAP plots shown in Figure 2b. Silhouette scores were computed using the `silhouette_score` method from `scikit-learn` [28, 14].

T-cell data from Hao et. al. We obtained transcriptomic data from the authors' website at `https://atlas.fredhutch.org/nygc/multimodal-pbmc/`. The log-transformed count data was subset to T-cells using the authors' annotations, and split into 8 donor-specific datasets. For each donor, we computed

an SCA reduction using 50 components and 5 iterations, with a neighborhood size of 15. 50 Principal components were computed using scanpy's `pca` function, and 50 Independent components were computed with the `FastICA` Python package [29]. To ensure convergence of `FastICA`, we raised the maximum number of iterations to 500.

We computed 15-nearest-neighbor graphs in each representation using cosine distance. We then ran UMAP [5] and Leiden clustering [16] on the resulting neighborhood graphs, with the default resolution of 1.0.

To compute F1 scores from clusterings, we performed an exhaustive search to identify the subset of clusters achieving the highest F1 score for each target population. More formally, to compute the highest F1 score f_{max} for any subset of clusters for a target population P, we used the following procedure:

1. Initialize $f_{max} = 0$ and $S = \emptyset$ (empty set).
2. Add all clusters that are subsets of P to S. These clusters improve recall without losing specificity, and thus can only increase the F1 score.
3. Denote by O the set of clusters which overlap P, but are not subsets of it. Construct the power-set of O, which we denote power(O).
4. For each collection of sets C in power(O):
   (a) Take the union of all sets in $S \cup C$, and compute the F1 score f.
   (b) If f is the highest F1 score obtained so far (i.e. $f > f_{max}$), set $f_{max} = f$.
5. return f_{max}.

The run-time of this procedure is exponential in the number of clusters overlapping the target population which are not subsets of it; however, in our benchmarks there were usually fewer than 10 such sets, so the computation was tractable.

Cytotoxic T-cell Population Discovery We extracted all cytotoxic T-cells from the Jensen SS3 dataset [13] using the authors' cell type annotations, obtaining 307 cells in total. For PCA, we used scanpy's `pca`

function with 20 components. For ICA, we used the `FastICA` function of `sklearn` [28, 29], again with 20 components. For SCA, we ran five iterations with 20 components each, starting with the PCA representation. 15-nearest neighbors graphs were computed in each representation using Euclidean distance, and the results were used to generate the UMAP plots in Figure 4. Leiden clusters in each representation were computed with the default resolution of 1.0. Matrix plots to show expression of key marker genes were generated using scanpy's `matrixplot` function on the binarized counts.

Integration

293T and Jurkat cell mixtures were downloaded from `http://scanorama.csail.mit.edu/`, provided by the authors of Scanorama [12], who originally obtained them from Zheng et. al. [30]. CD4+ T-lymphocytes were downloaded from the European Bioinformatics Institute [31] under accession E-GEOD-106540, accessed via scanpy's `ebi_expression_atlas` function.

293T/Jurkat data were log-transformed, and T-lymphocyte data binarized, as these transformations produced the clearest visual separation between known clusters regardless of representation. For both methods, we applied Scanorama with default parameters, and Harmony on the PCA representation with 50 PCs. To generate SCA information scores, we used 3 iterations with 20 components each. The same parameters were used to generate a SCA embedding for the representation pictured in the top right of Figure 5b. Highly variable genes were identified for comparison using Scanpy's `highly_variable_genes` function.

The SCA Algorithm

Information Score Matrix Computation Given the input data X with N cells and M genes, we first compute for each cell c a neighborhood $N_k(c)$ containing the k nearest cells. This requires a distance metric. By default, the first SCA iteration uses cosine distance in a 50-dimensional PCA reduction of the data. While this works well for most datasets, The user can override the number of PCs used and the distance metric applied, or directly input a list of neighborhoods for each cell. Successive iterations use the representation output by the previous iteration as input to the chosen metric. For example, if the cosine

distance is chosen, the second iteration of SCA will compute neighborhoods using cosine distance between the SCA representations output by the first iteration. The number of neighbors k is user-specifiable, with a default of 15.

For each cell c and gene g, SCA then assesses the significance of g's expression in $N_k(g)$. Under the null hypothesis, where g is randomly expressed, the number of cells in $N_k(c)$ expressing g obeys a binomial distribution with k trials, each with probability of success equal to the g's global probability of expression p_g (the actual distribution may deviate slightly from binomial since the cells are drawn without replacement, but the effect is very small due to the large number of cells in most datasets). SCA performs a two-tailed binomial test to produces an initial p-value $p_{c,g}$; however, since many genes are tested (often tens of thousands), SCA uses a family-wise error rate to correct for multiple testing. Specifically:

$$\tilde{p}_{c,g} = 1 - (1 - p_{c,g})^{N_t}$$

where N_t is user-specified. If the genes were completely independent from one another, it would make sense to set N_t to the number of genes; however, real scRNA-seq data contains significant inter-gene dependence, so this approach may lead to over-correction and under-performance. In practice, we find that taking $N_t = 50$ is sufficient to avoid over-clustering whilst preserving performance, and we set this as the default.

From the corrected p-values $\tilde{p}_{c,g}$, SCA then computes a signed information score via:

$$I(c, g) = -\text{sgn}(\frac{N_k(c, g)}{k} - p_g \cdot N) \cdot \log(\tilde{p}_{c,g})$$

where $N_k(c, g)$ denotes the number of cells in $N_k(c)$ that express g. Thus, $I(c, g)$ is negative if g is expressed less than expected in c's neighborhood, and positive if expressed more than expected (note that the second term is always negative, since $\tilde{p}_{c,g} < 1$).

Naively, doing this for all cells and all genes requires NM binomial tests. However, note that for each gene, there are only $k + 1$ possible values of $N_k(c, g)$ across the data, namely $0, 1..., k$, and therefore only $k+1$ possible p-values. Thus, we can pre-compute an $M \times k$ table of corrected p-values, and compute $\tilde{p}_{c,g}$ by

accessing the element in the row corresponding to g and the $N_k(c,g)$th column of this pre-computed table. This substantially reduces the number of binomial tests required, to kM from NM, without altering the scores.

As a further speedup and memory reduction, we optionally allow dividing of the unit interval into a fixed number B of bins, and approximating the global gene probabilities p_g by the nearest bin boundary. We then compute corrected p-values for each bin and each $i \in 1, ..., k$, requiring only kB computations. We then approximate $\tilde{p}_{c,g}$ by taking the row corresponding to the bin of g and the $N_k(c,g)$th column of this table. Bin divisions can optionally be chosen so that the resulting approximation to p_g is accurate to within a constant factor, with bins spaced according to a power law. In practice, this means that the binning is denser at lower probabilities, allowing more accurate binomial scores for rare genes. By default, bins are evenly spaced. All of the experiments in this paper were performed without binning.

Computing Shannon Components. From the information scores $I(c,g)$, SCA next seeks to generate an informative linear combination of genes. For a given combination defined by

$$\tilde{g} = \alpha_1 g_1 + \alpha_2 g_2 + ...\alpha_M g_M$$

we say that $\tilde{g}$ has *loadings* $\alpha_1, ..., \alpha_n$. For a fixed cell c, we formulate the information score of $\tilde{g}$ as

$$I(c, \tilde{g}) = \alpha_1 I(c_1, g) + \alpha_2 I(c_2, g) + ... + \alpha_M I(c_M, g)$$

We then define the overall overall information score of g by taking the norm over all cells:

$$I(\tilde{g}) = ||(I(c_1, \tilde{g}), I(c_2, \tilde{g}), ..., I(c_N, \tilde{g}))||$$

or, in matrix notation,

$$I(\tilde{g}) = ||\mathbf{I}\alpha^T||$$

where $\mathbf{I}$ denotes the information score matrix and $\alpha = (\alpha_1, ..., \alpha_M)$.

We now seek the metagene $\tilde{g}$, defined by the loadings $\alpha_1, ..., \alpha_N$, that maximizes $I(\tilde{g})$. Since we can achieve arbitrarily large values of $I(\tilde{g})$ by scaling the coefficients, we constrain the loading coefficients to have norm 1, that is:

$$||(\alpha_1, ..., \alpha_M)|| = 1.$$

It is a standard linear algebra result that this maximum is realized by the leading right-eigenvector of I, i.e. by the first principal component loading vector of $\mathbf{I}$ (proof in the Supplementary Information and [32]). Thus, the first Shannon Component loading vector is simply the first principal component loading vector of $\mathbf{I}$, which we denote $\alpha_{\mathbf{1}}$. To obtain additional Shannon components, we repeat the optimization with the constraint

$$\alpha \perp \alpha_{\mathbf{1}}.$$

It is straightforward to see (Supplementary Information) that this yields the second principal component loading vector $\alpha_{\mathbf{2}}$ of $\mathbf{I}$. Continuing, we see that the loading vectors for Shannon components are identical to those of the principal components on $\mathbf{I}$.

SCA next computes the values of the first D Shannon components over the *input* data (not the information scores). That is,

$$\mathbf{SC}_i(\mathbf{X}) = \mathbf{X}\alpha_i^T$$

(Note that the Shannon components are linear functions of the input data, despite the nonlinear construction of $\mathbf{I}$. Although the loadings are computed on $\mathbf{I}$, the values of the components are computed by applying theses loadings to $\mathbf{X}$). The first D Shannon components then act as features for further downstream analysis.

SCA for Batch Correction

When the input data is combined across B batches $\mathbf{X_1}, ..., \mathbf{X}_B$, we can use batch-specific information scores to assess the informativeness of each gene in each batch. We compute B separate information score matrices $\mathbf{I}_1, ..., \mathbf{I}_B$. We then average the absolute value of the information scores first across cells, and then across

batches, yielding a single score for each gene. That is:

$$\text{score}(g) = \text{mean}(\{\text{mean}(\{|\mathbf{I}_j(g, c_i)|) : i \in 1, ..., N\} : j \in 1, ..., B\}$$

Note that each batch is weighted equally in the outer mean, regardless of its size. Taking the absolute value ensures that genes achieving significant under-expression are preserved.

To batch correct the data, we simply keep the top L genes with highest value of score(g) (in our experiments we take $L = 200$). Intuitively, we keep only genes that give information about within-batch variation, and remove genes that are noisily expressed or highly batch-dependent.

1 Acknowledgements

The authors are grateful to Ashwin Narayan, Brian Hie, Sarah Nyquist, Rohit Singh, Ellen Zhong, and other members of the Berger lab for their continual support and helpful feedback. We also thank Josh Peters for helpful conversations on SCA's applications. This work was supported by the National Institutes of Health [U01CA250554 to B.B.].

References

[1] Hie, B. *et al.* Computational methods for single-cell RNA sequencing. *Annual Review of Biomedical Data Science* (2020).

[2] Ringnér, M. What is principal component analysis? *Nature biotechnology* **26**, 303–304 (2008).

[3] Jutten, C. & Herault, J. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing* **24**, 1–10 (1991).

[4] Narayan, A., Berger, B. & Cho, H. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology* 1–10 (2021).

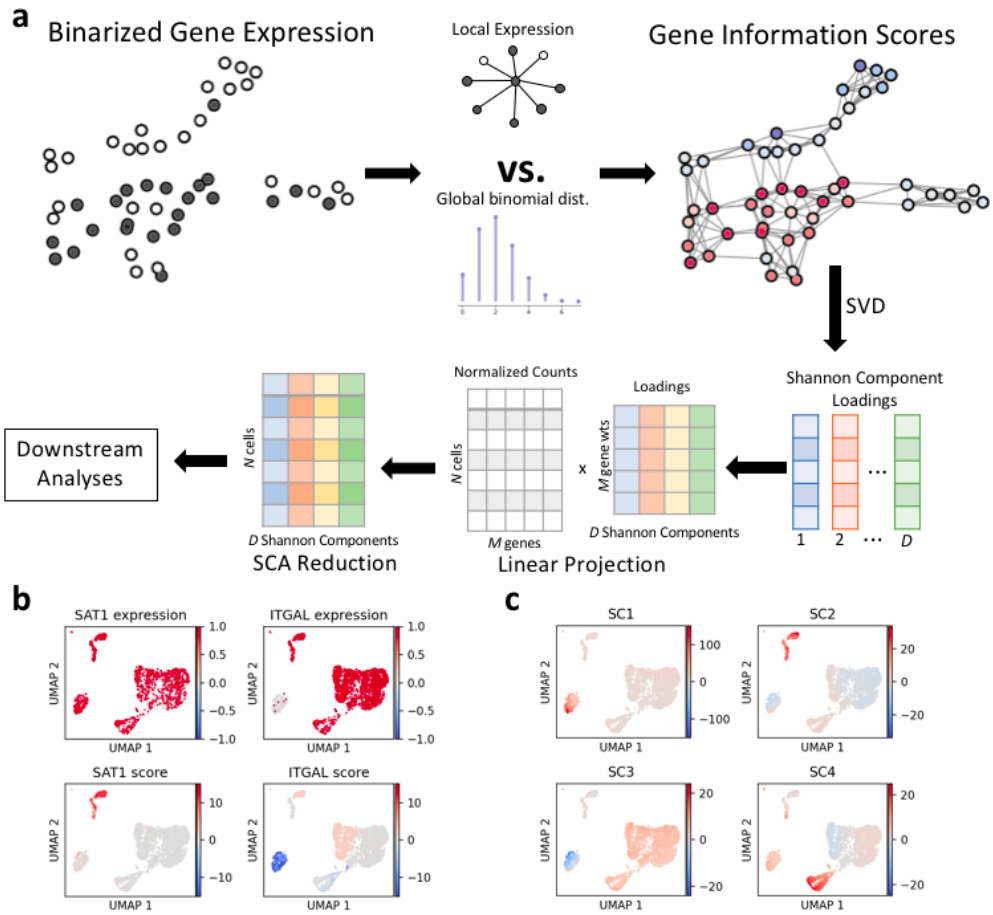

Figure 1: **a**: Overview of SCA. Given the original transcript matrix, we perform a binomial test on the binarized expression to assess over- or under-expression of each gene in each cell's neighborhood (first and second subfigures). The resulting p-values are converted via negative log transform to Shannon information scores (third subfigure), yielding the Information Score Matrix. Singular value decomposition (SVD) on this matrix yields D maximally-informative *Shannon components* (bottom right), which are then used to linearly project the input matrix for downstream clustering and analysis. **b**: Example Information Scores of two genes over a set of human PBMCs profiled via smart-seq 3 [13]. *SAT1* is noisily expressed throughout the data, but achieves a high information score only where it is constitutively expressed. *ITGAL* achieves a strongly negative information score where it is conspicuously absent. **c**: Heat maps of the first four Shannon components over the same data.

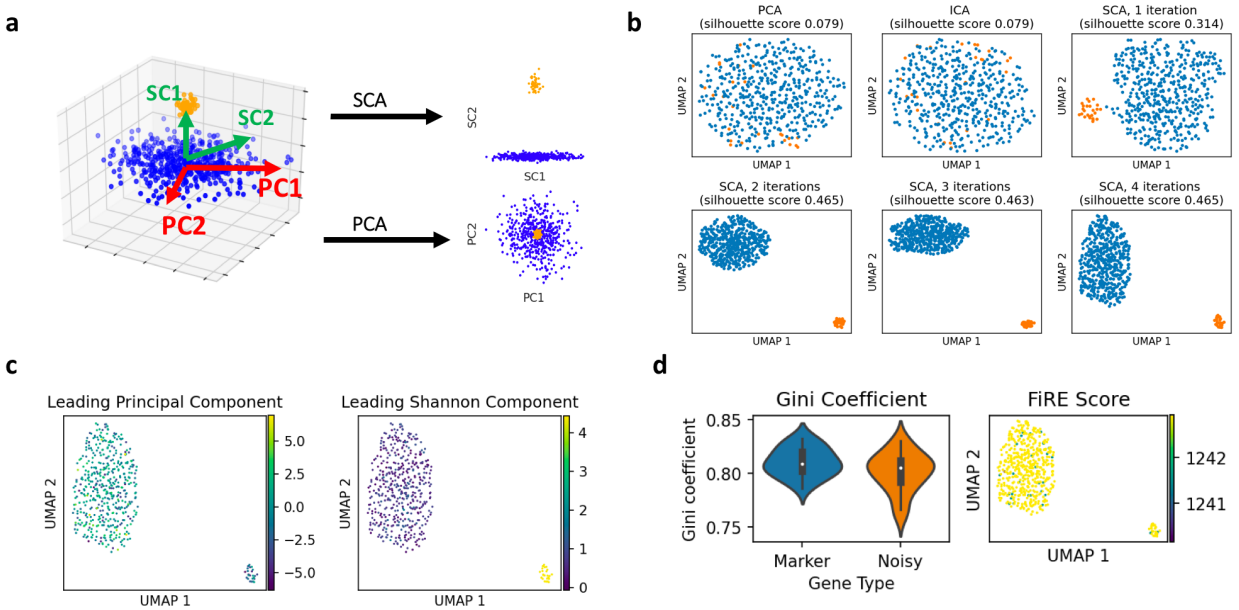

Figure 2: **a**: Schematic cartoon of synthetic test data, explaining the difference in performance. A small population of 25 cells is distinguished from a large and dispersed set of 500 cells with 10,000 randomly-expressed genes by the expression of an additional 20 "marker" genes. Principal components align with the higher-variance cluster, and thus do not distinguish the smaller cluster. SCA detects the smaller cluster due to higher information scores. **b**: UMAP plots generated from the synthetic dataset using PCA and using SCA with 1-4 iterations. Iterating SCA enables recovery of a rare population of 25 cells (5% of the total cells) defined by just 20 marker genes (0.2% of the total genes, with the remainder randomly expressed). **c**: Heatmaps of the leading Principal Component and the leading Shannon Component. The leading Shannon component cleanly separates the two populations, whereas the leading Principal Component does not. **d**: Left: violin plot of Gini coefficients for marker genes and for noisy genes. Marker genes do not have a significantly higher Gini index, despite their discriminative power. Right: Heatmap of rareness scores computed with FiRE [10]. The scores fail to identify the rare population.

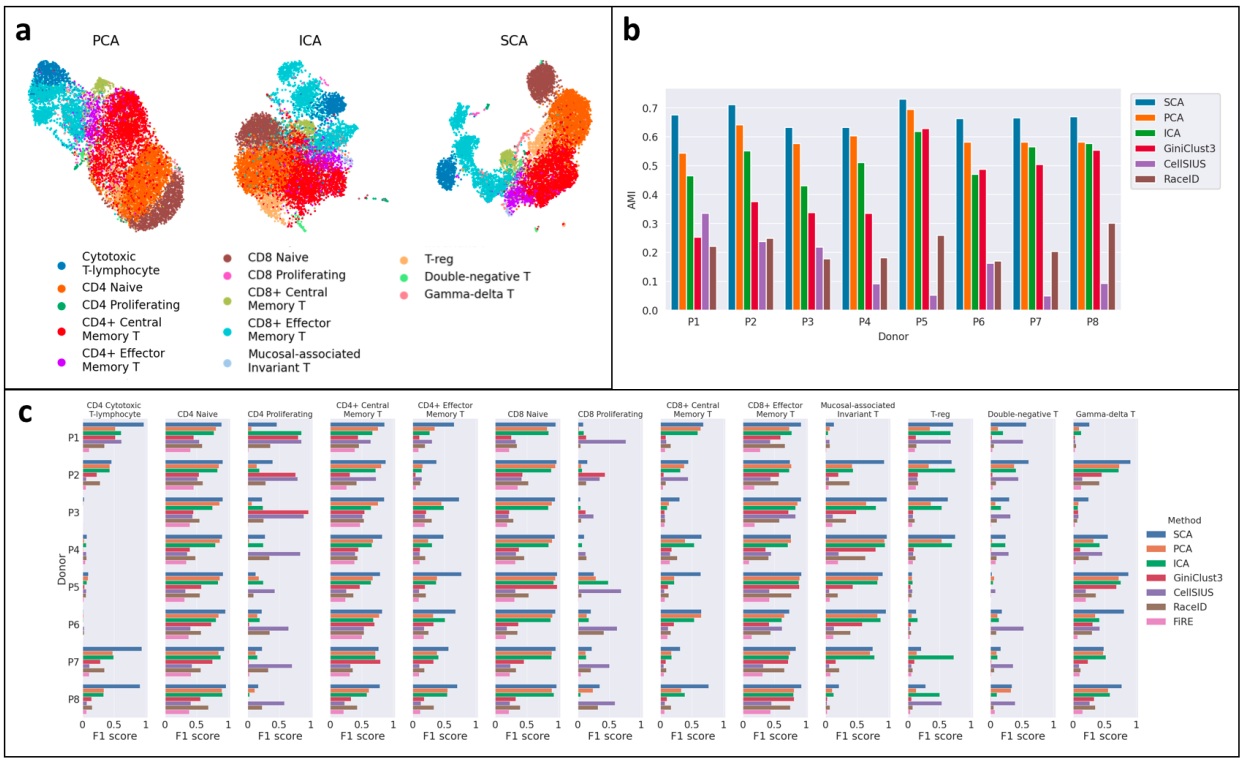

Figure 3: **a**: UMAP plots computed from PCA, ICA, and SCA reductions of T-cell scRNA-seq data for Patient 1 from Hao et. al. [15]. Cell type labels were determined via parallel screening of 228 antibodies using CITE-seq. SCA produces better separation between the annotated cell types, enabling their discovery from transcriptomic data alone. **b**: Adjusted Mutual Information (AMI) of true cell labels with clusters output by each method in each patient. For PCA, ICA, and SCA, we perform Leiden clustering with resolution 1.0 after reduction. SCA-based clusterings consistently have higher AMI with the true labels. **c**: F1 scores for recovery of all T-cell subtypes across all 8 patients of the dataset from Hao et. al., using PCA, ICA, and SCA followed by Leiden clustering with resolution 1.0, and using four existing methods. For each clustering and cell type, the set of clusters best identifying that cell type was identified, and the F1 score reported (Methods). SCA recovers nearly all cell types with superior accuracy, and newly enables discovery of some types (e.g. CD4 cytotoxic T-lymphocytes and CD4+ effector memory T-cells) which are poorly distinguished by PCA and ICA. Thus, SCA facilitates *de novo* discovery of these clinically relevant subpopulations.

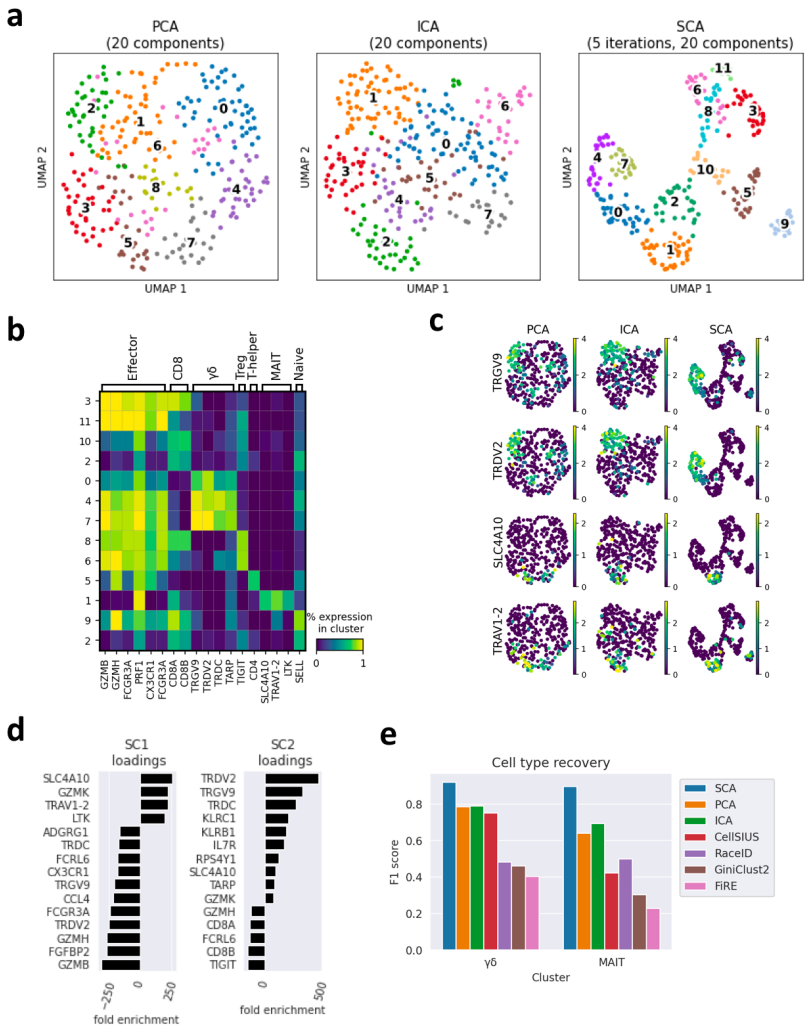

Figure 4: SCA recovers small and subtle subpopulations from a set of 307 cytotoxic T-cells. **a**: Comparison of UMAP embeddings computed from PCA (left), ICA (middle), or SCA (right), colored and labeled by Leiden clustering computed downstream of each dimensionality reduction approach. While the population appears globular under PCA and ICA, SCA's informative components reveal several dsistinct populations upon downstream clustering. **b**: Matrix plot of the expression of key marker genes across clusters computed on the SCA embedding. Brackets on top group indicate the populations the genes mark. Gamma-delta, MAIT, and T-helper cells are clearly discernible from the SCA-derived Leiden clustering. **c**: Expression of key marker genes for Gamma-delta cells (*TRGV9* and *TRDV2*) and MAIT cells (*SLC4A10* and *TRAV1-2*) over UMAP plots derived from PCA, ICA, and SCA embeddings. Whereas these markers are dispersed in the PCA and ICA representations, SCA concentrates them to a single locale, enabling discovery. **d**: 15 Top-loaded genes in the second and fourth Shannon components, including those with high negative weights. Fold-enrichment over the mean loading across all genes is plotted. These components describe $\gamma\delta$ and MAIT cells, respectively, showing that Shannon components reflect true biological variation. **e**: Bar plot of F1 scores for recovery of MAIT and $\gamma\delta$ T-cells by various methods. For SCA, ICA, and PCA, we perform Leiden clustering and convert the clusters to an F1 score (Methods). SCA captures $\gamma\delta$ and MAIT cells with superior accuracy.

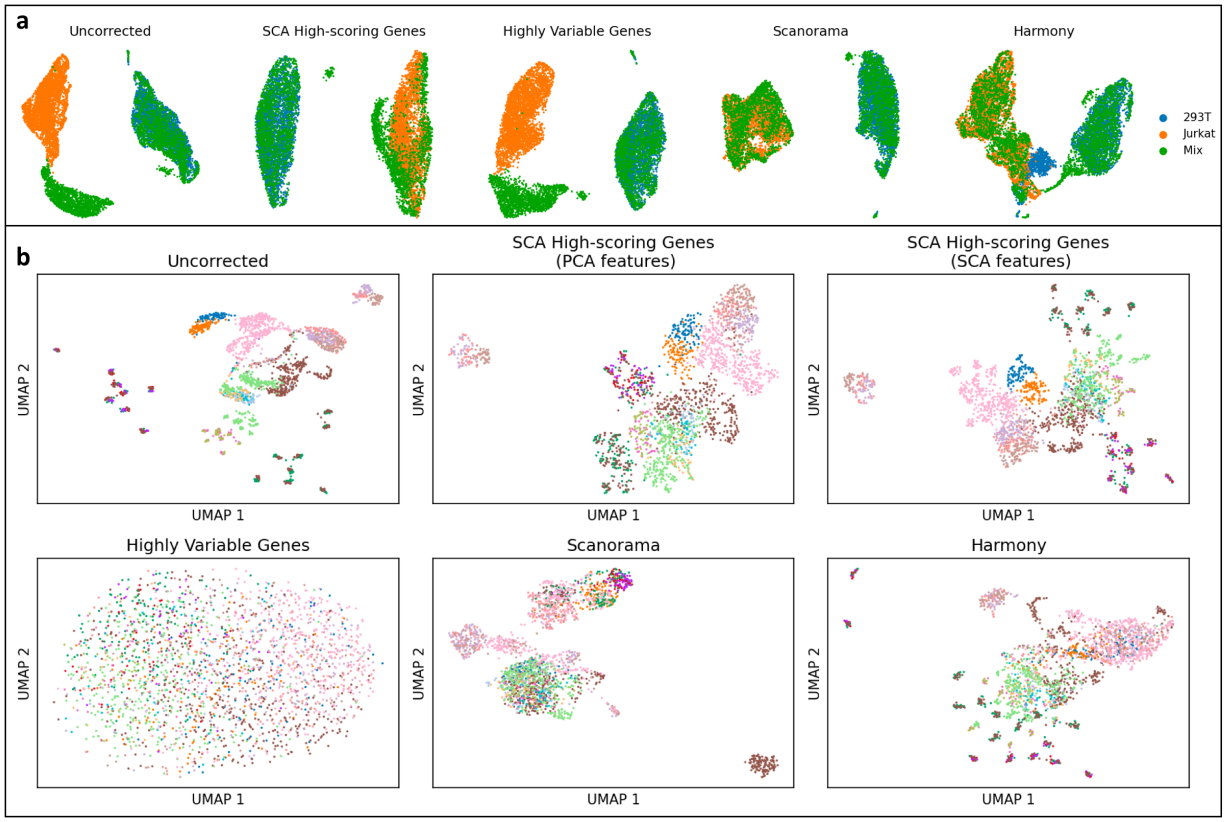

Figure 5: Integration using SCA. **a**: UMAP plots of 293T and Jurkat datasets from the Scanorama paper [12], integrated through various approaches. One batch (blue) contains 293T cells; one (orange) contains Jurkat cells, and one (green) contains an equal mixture of both. Selecting the 200 genes with highest within-sample information score integrates the data, performing similarly to Scanorama and Harmony. On the other hand, selecting the top 200 highest-variance genes fails to integrate the data. **b**: UMAP plots of cytotoxic T-lymphocyte data from Patil et. al. [23] after integration with various approaches. Each UMAP is colored by donor. SCA-based integration produces a representation similar to that of Harmony, whereas taking highly-variable genes loses all structure.

[5] McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[6] Maaten, L. v. d. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9**, 2579–2605 (2008).

[7] Jiang, L., Chen, H., Pinello, L. & Yuan, G.-C. Giniclust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome biology* **17**, 144 (2016).

[8] Grün, D. *et al.* Single-cell messenger rna sequencing reveals rare intestinal cell types. *Nature* **525**, 251–255 (2015).

[9] Wegmann, R. *et al.* Cellsius provides sensitive and specific detection of rare cell populations from complex single-cell rna-seq data. *Genome biology* **20**, 1–21 (2019).

[10] Jindal, A., Gupta, P., Sengupta, D. *et al.* Discovery of rare cells from voluminous single cell expression data. *Nature communications* **9**, 1–9 (2018).

[11] Korsunsky, I. *et al.* Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods* 1–8 (2019).

[12] Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature biotechnology* **37**, 685–691 (2019).

[13] Hagemann-Jensen, M. *et al.* Single-cell RNA counting at allele and isoform resolution using smart-seq3. *Nature Biotechnology* **38**, 708–714 (2020).

[14] Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **20**, 53–65 (1987).

[15] Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *bioRxiv* (2020).

[16] Traag, V. A., Waltman, L. & van Eck, N. J. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports* **9**, 1–12 (2019).

[17] Vinh, N. X., Epps, J. & Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research* **11**, 2837–2854 (2010).

[18] Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology* **19**, 15 (2018).

[19] Holtmeier, W. & Kabelitz, D. $\gamma\delta$ t cells link innate and adaptive immune responses. In *Mechanisms of epithelial defense*, vol. 86, 151–183 (Karger Publishers, 2005).

[20] Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods* **15**, 1053–1058 (2018).

[21] Polański, K. *et al.* Bbknn: fast batch alignment of single cell transcriptomes. *Bioinformatics* **36**, 964–965 (2020).

[22] van Dijk, D. *et al.* Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data. *BioRxiv* 111591 (2017).

[23] Patil, V. S. *et al.* Precursors of human cd4+ cytotoxic t lymphocytes identified by single-cell transcriptome analysis. *Science immunology* **3** (2018).

[24] Procaccia, I. & Levine, R. Potential work: A statistical-mechanical approach for systems in disequilibrium. *The Journal of Chemical Physics* **65**, 3357–3364 (1976).

[25] Zadran, S., Arumugam, R., Herschman, H., Phelps, M. E. & Levine, R. Surprisal analysis characterizes the free energy time course of cancer cells undergoing epithelial-to-mesenchymal transition. *Proceedings of the National Academy of Sciences* **111**, 13235–13240 (2014).

[26] Gross, A. & Levine, R. D. Surprisal analysis of transcripts expression levels in the presence of noise: A reliable determination of the onset of a tumor phenotype. *PLoS One* **8**, e61554 (2013).

[27] Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).

[28] Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *Journal of machine learning research* **12**, 2825–2830 (2011).

[29] Oja, E. & Hyvarinen, A. Independent component analysis: algorithms and applications. *Neural networks* **13**, 411–430 (2000).

[30] Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature communications* **8**, 1–12 (2017).

[31] Madeira, F. *et al.* The EMBL-EBI search and sequence analysis tools apis in 2019. *Nucleic acids research* **47**, W636–W641 (2019).

[32] (https://math.stackexchange.com/users/22857/martin argerami), M. A. What vector x will maximize the norm of $\|ax\|_2/\|x\|_2$ (norm 2). Mathematics Stack Exchange. URL `https://math.stackexchange.com/q/3112310`. URL:https://math.stackexchange.com/q/3112310 (version: 2021-01-26), `https://math.stackexchange.com/q/3112310`.