Unsupervised cell functional annotation for single-cell RNA-Seq

Dongshunyi Li[1], Jun Ding[2], and Ziv Bar-Joseph*[1,3]

[1]Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2]Meakins-Christie Laboratories, Department of Medicine, McGill University Health Centre, Montreal, Quebec, H4A 3J1, Canada
[3]Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

November 21, 2021

Abstract

One of the first steps in the analysis of single cell RNA-Sequencing data (scRNA-Seq) is the assignment of cell types. While a number of supervised methods have been developed for this, in most cases such assignment is performed by first clustering cells in low-dimensional space and then assigning cell types to different clusters. To overcome noise and to improve cell type assignments we developed UNIFAN, a neural network method that simultaneously clusters and annotates cells using known gene sets. UNIFAN combines both, low dimension representation for all genes and cell specific gene set activity scores to determine the clustering. We applied UNIFAN to human and mouse scRNA-Seq datasets from several different organs. As we show, by using knowledge on gene sets, UNIFAN greatly outperforms prior methods developed for clustering scRNA-Seq data. The gene sets assigned by UNIFAN to different clusters provide strong evidence for the cell type that is represented by this cluster making annotations easier.

Software: `https://github.com/doraadong/UNIFAN`

Keywords

Cell annotation, Cell type identification, Clustering, Gene expression

*Correspondence: Ziv Bar-Joseph, zivbj@cs.cmu.edu

1 Introduction

The large increase in studies profiling RNA-Sequencing data in single cells [1] raises several computational challenges. One of the first, and most important, steps in the analysis of such studies is cell type assignment [2]. Several methods have been developed for such assignment, including supervised and unsupervised methods. Supervised methods mainly use previously annotated datasets to annotate new datasets [3]. This is done either by directly classifying each cell [4] or by learning an alignment between the datasets to classify groups of cells in a new study [5].

While supervised methods are useful in some cases, they cannot be applied to all cases since reference datasets are not available for most organs, tissues and conditions. Another challenge with supervised methods is their inability to identify new cell types that are often one of the major goals of the study [3]. Thus, the most popular way to annotate single cell data is by using unsupervised methods. These are often based on clustering cells in a low-dimensional space and manually annotating each cluster using known marker genes or cluster specific differentially expressed genes. Several methods for clustering single cell data have been developed and used. These include SIMLR [6] which clusters cells by using multiple kernel functions to construct a similarity matrix between cells, Leiden clustering [7] and Seuratv3 [8] which use k-nearest neighbors (k-nn) based graph partitions to group cells, and methods based on deep neural networks, such as DESC [9], which uses autoencoders to reduce the dimensions of the data and then clusters cells in the reduced dimension space.

While several clustering methods have been developed and used for scRNA-Seq data, to date these methods have only relied on the observed expression data. However, there are several additional complementary datasets that can be used to improve clustering and reduce noise related grouping. Specifically, gene sets [10] have been compiled to characterize many processes, pathways and conditions. While the exact processes or functions that are activated in specific cells or clusters are unknown, we can use these sets to guide the grouping of cells by placing more emphasis on co-expression of genes in known sets when clustering single cell data. Since cells of the same type likely share many of the biological processes, such design can both, improve the identification of good clusters and help in annotating them based on the function of the sets associated with each cluster.

Here we introduce UNIFAN (**Un**supervised S**i**ngle-cell **F**unctional **An**notation) to simultaneously cluster and annotate cells with known biological processes (including pathways). For each cell, we first infer its gene set activity scores based on the co-expression of genes in known gene sets. We also use an autoencoder that outputs a low-dimensional representation learned from the expression of all genes. We combine both, the low dimension representation and the gene set activity scores to determine the cluster for each cell. The process is iterative and we define a target function and show how to learn model parameters to optimize it. In addition to the cell clusters, the method also outputs the gene sets associated with each cluster and these can be used to annotate and assign cell types to different clusters.

We applied UNIFAN to several mouse and human datasets spanning multiple organs, cell types and labs. Our results indicate that by using gene sets as input, UNIFAN can improve on current single cell clustering methods. In addition, in most cases, the gene sets selected for each cluster serve as a very good source for their annotations.

2 Materials and Methods

2.1 Datasets and Data Preprocessing

We used both human and mouse datasets from several tissues to test our method. The human samples include three scRNA-Seq datasets from The Human BioMolecular Atlas Program (HuBMAP) consortium [11]. These include "HuBMAP spleen", "HuBMAP thymus" and "HuBMAP lymph_node". We use Scanpy [12] for the data preprocessing leading to 34,515 cells and 26,092 genes for "HuBMAP spleen", 22,367 cells and 24,396 genes for "HuBMAP thymus", and 24,311 cells and 20,946 genes for "HuBMAP lymph_node". The "Atlas lung" uses the healthy control samples from [13]. After filtering, this dataset is composed of 96,282 cells and 17,315 genes. The "pbmc28k" data is from [14] and has 25,185 cells and 19,404 genes. The "pbmc68k" data is from [15], having 68,551 cells and 17,788 genes. Mouse datasets are from the Tabula Muris paper [16]. Following [17], we end up with 21 datasets each for a single tissue. They all have 22,904 genes and the number of cells ranges from 366 (Aorta) to 4,433 (Heart). See Supplementary Methods for the preprocessing details.

In addition to expression data, UNIFAN uses gene sets to guide clustering. For this we use 7481 gene sets derived from the GO Biological Process ontology (termed c5.go.bp in MSigDB [10]), 2922 gene sets from pathway databases (c2.cp in MSigDB [10]) and 335 sets of targets of transcription factors from [18]. Names for biological process sets start with "GOBP". Pathway sets use a prefix representing the pathway database they are extracted from (e.g., "KEGG", "WP", "REACTOME"). We purposely did not use cell type marker gene sets (c8.all in MSigDB) since we wanted to keep the method unsupervised and marker lists are often based on DE analysis of labeled cell type data.

2.2 Clustering and Annotating Single Cells Using Gene Sets

To enable the use of prior knowledge on gene function and regulation for clustering single cells, we developed a deep learning model, UNIFAN (**Un**supervised **S**ingle-cell **F**unctional **An**notation). For each single cell, UNIFAN first infers gene set activity scores associated with this cell using the input gene sets. Next, UNIFAN clusters cells by using the learned gene set activity scores and a reduced dimension representation of the expression of genes in the cell. The gene set activity scores are used by an "annotator" to guide the clustering such that cells sharing similar biological processes are more likely to be grouped together. Such design allows the method to focus on the key processes when clustering cells and so can overcome issues related to noise and dropout while simultaneously selecting marker gene sets which can be used to annotate clusters.

2.2.1 Learning Gene Set Activity Scores for Cells

For each cell, we first infer its gene set activity scores $\mathbf{r} \in \mathbb{R}^{L}_{\geq 0}$ (L: number of gene sets), which represent the activity of known biological processes or pathways in the cell. For this, we design a special autoencoder whose decoder, instead of being fully-connected, is composed of a binary matrix $D \in \mathbb{R}^{G \times L}$, where G is the total number of genes profiled. Each column in D corresponds to a known gene set for a biological process or pathway where the values are indicators for whether

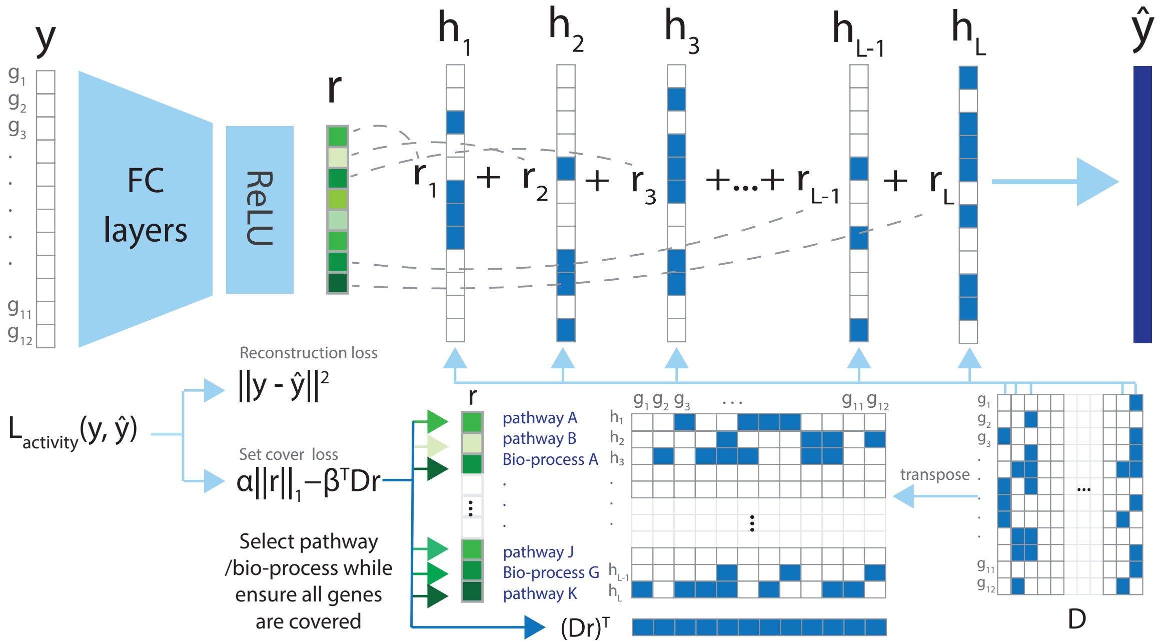

Figure 1: Assigning single-cell gene set activity scores using an autoencoder. The autoencoder is designed such that the decoder is composed of binary vectors with values indicating if a profiled gene belongs to a known gene set or not. The output of the encoder, **r**, serve as coefficients for the gene set vectors, showing how related a cell is to a known pathway/biological processes. **r** thus can be seen as the gene set activity scores for this cell. The set cover loss is designed to select uncorrelated pathways/processes to better annotate cells. FC layers: fully-connected layers.

a gene belongs to this set or not. For a cell with expression **y**, the encoder, which composes of fully-connected layers, outputs a low-dimensional representation **r**. **r** is then multiplied by the binary matrix D which leads to a reconstructed expression vector $\hat{\mathbf{y}}$, as shown in Figure 1. Values in **r** serve as weights / coefficients for known gene sets for this cell. Parameters for the fully-connected encoder are optimized such that the combination of the gene sets, weighted by **r**, can be used to reconstruct the observed expression **y** for all genes in the cell. Thus, **r** can be seen as the activity levels of pathways and processes in the cell.

To construct the gene set matrix D which serves as an input to UNIFAN, we collected gene sets representing biological processes (including canonical pathways and targets of specific regulators) from MSigDB [10] and [18], which resulted in a total of roughly 10K sets. We expect that only a small subset of these biological processes are active for each single cell and so we employ regularization to select active gene sets for each cell. First, we constrain **r** to be non-negative by using ReLU for the output layer, which results in most values in **r** being assigned to 0. Next, we use a regularizer inspired by the classical set cover algorithm, which aims to find the least number of sets that covers all elements (in our case, profiled genes of the cell). By employing this regularizer, we aim to find a small subset of non-overlapping gene sets that can cover as many of genes as possible [19]. For this, our regularizer optimizes the following function $\alpha\|\mathbf{r}\|_1 - \beta^T D\mathbf{r}$, where α and β are hyperparameters (see section 2.3 on selecting values for hyperparameters in our model). Using mean-squared error for the reconstruction loss, our overall loss function for a single cell is

$$\mathcal{L}_{activity}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 + \alpha\|\mathbf{r}\|_1 - \beta^T D\mathbf{r}.$$

2.2.2 Clustering Cells Using Gene Set Activity Scores

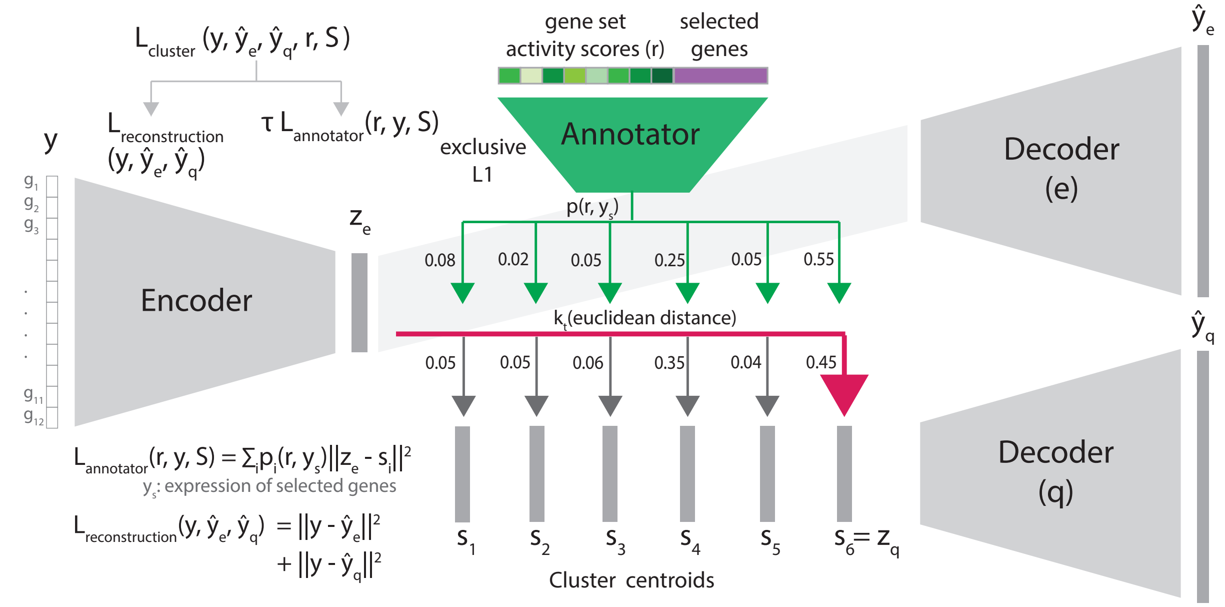
L_cluster (y, ŷ_e, ŷ_q, r, S)
L_reconstruction (y, ŷ_e, ŷ_q)
τ L_annotator (r, y, S)
gene set activity scores (r)
selected genes
Annotator
exclusive L1
p(r, y_s)
y
g1
g2
g3
g11
g12
Encoder
z_e
0.08
0.02
0.05
0.25
0.05
0.55
k_t(euclidean distance)
0.05
0.05
0.06
0.35
0.04
0.45
s1
s2
s3
s4
s5
s6 = z_q
Cluster centroids
Decoder (e)
ŷ_e
Decoder (q)
ŷ_q
L_annotator(r, y, S) = Σ_i p_i(r, y_s)||z_e - s_i||²
y_s: expression of selected genes
L_reconstruction(y, ŷ_e, ŷ_q) = ||y - ŷ_e||² + ||y - ŷ_q||²

Figure 2: Jointly clustering and annotating cells. The autoencoder contains two parts: the cluster assignment part (grey in Figure 2) uses a low-dimensional representation $\mathbf{z_e}$ to assign a cell to clusters; the "annotator" (green in Figure 2) uses the learned gene set activity scores and selected genes' expression to refine clustering and annotate clusters. Gene sets and genes selected as predictive by the annotator, in turn, provide useful annotations for each cell cluster. We set the number of clusters M as 6 in this figure for illustration purposes.

To cluster cells using the inferred gene set activity scores, we use an autoencoder-based method. It is composed of two parts, an expression based cluster assignment part ("grey" parts in Figure 2) and the "annotator" part ("green" parts in Figure 2) which uses the gene set activity scores discussed above as input.

The cluster assignment part only uses the expression profile for each cell. It consists of an encoder and two decoders (Decoder(e) and (q) in Figure 2), modified based on [20, 21]. For a single cell, we first use an encoder on the expression of genes in the cell $\mathbf{y}$, resulting in a low-dimensional representation $\mathbf{z_e}$, as shown in Figure 2. After initialization, we start with a guess of M clusters and cluster centroids. Among M cluster centroids $S = \{\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_M}\}$, we identify the centroid closest to $\mathbf{z_e}$ by first calculating the euclidean distances between $\mathbf{z_e}$ and all centroids and then transforming the distances using a t-distribution kernel $k_t(d) = (1 + \frac{d^2}{v})^{-\frac{v+1}{2}}$, following [9, 22, 23]. d stands for the distance and v stands for the degrees of freedom, which is fixed at 10 for all experiments. We then take the closest centroid as the discrete representation $\mathbf{z_q}$ of the cell and assign the cell to the corresponding cluster. We obtain the reconstructed expression $\hat{\mathbf{y}}_\mathbf{q}$ using decoder (q) which only takes $\mathbf{z_q}$ as input and so all cells in the same cluster have the same reconstructed expression. We optimize the reconstruction error $||\mathbf{y} - \hat{\mathbf{y}}_\mathbf{q}||^2$ to find the best $\mathbf{z_q}$, cluster centroids S, and decoder (q), in a manner similar to finding the best cluster centroids in k-means clustering.

Since we assign cell clusters using k-means (i.e. discrete assignment), the encoder cannot be learned using backpropogation. To enable the iterative refinement of model parameters using gradients, we follow [20] by adding another decoder, decoder (e). Decoder (e) takes $\mathbf{z_e}$ as input and outputs another reconstructed expression $\hat{\mathbf{y}}_\mathbf{e}$. By optimizing $||\mathbf{y} - \hat{\mathbf{y}}_\mathbf{e}||^2$, we can now update

$\mathbf{z_e}$ and the encoder. The overall loss function for a single cell in the cluster assignment part is thus $\mathcal{L}_{\text{reconstruction}}(\mathbf{y}, \hat{\mathbf{y}}_\mathbf{e}, \hat{\mathbf{y}}_\mathbf{q}) = \|\mathbf{y} - \hat{\mathbf{y}}_\mathbf{e}\|^2 + \|\mathbf{y} - \hat{\mathbf{y}}_\mathbf{q}\|^2$. All neural networks mentioned above are composed of fully-connected layers.

So far we only discussed clustering using expression data. We next use the learned gene set activity scores for each cell to refine cluster assignments as well as to annotate cell clusters. For this, we add an "annotator", a logistic classifier, to the network model. For each cell, the annotator uses the gene set activity scores $\mathbf{r}$ for that cell as input and outputs $\mathbf{p}(\mathbf{r})$, the probability of the cell being assigned to each cluster. We use the annotator's output to refine the cluster assignment by adding $\mathcal{L}_{annotator}(\mathbf{r}, \mathbf{y}, S) = \sum_i^M p_i(\mathbf{r}) d_i^2 = \sum_i^M p_i(\mathbf{r}) \|\mathbf{z_e} - \mathbf{s_i}\|^2$ to the existing loss function. Since it uses both, the low-dimensional representation of the cell $\mathbf{z_e}$ and the cluster centroids S, such loss encourages cells being assigned to clusters based on the probability specified by $\mathbf{p}(\mathbf{r})$. In other words, by employing $\mathcal{L}_{annotator}$ and the annotator, we are using prior knowledge about gene membership in key biological processes to guide the dimension reduction and cluster assignment. Gene sets selected as predictive by the annotator, in turn, provide useful annotations for each cell cluster.

To allow the selection of marker genes for each cluster, we also tested the use of the UNIFAN with a subset of the most variable genes selected using Seuratv3 [8]. Using such set the annotator loss becomes: $\mathcal{L}_{annotator}(\mathbf{r}, \mathbf{y}, S) = \sum_i^M p_i(\mathbf{r}, \mathbf{y_s}) \|\mathbf{z_e} - \mathbf{s_i}\|^2$, where $\mathbf{y_s}$ are the expression of the selected genes. The overall loss function for the cluster assignment part is thus

$$\mathcal{L}_{\text{cluster}}(\mathbf{y}, \hat{\mathbf{y}}_\mathbf{e}, \hat{\mathbf{y}}_\mathbf{q}, \mathbf{r}, S) = \mathcal{L}_{\text{reconstruction}}(\mathbf{y}, \hat{\mathbf{y}}_\mathbf{e}, \hat{\mathbf{y}}_\mathbf{q}) + \tau \mathcal{L}_{annotator}(\mathbf{r}, \mathbf{y}, S)$$

, where τ is a weighting hyperparameter.

The annotator is trained to optimize its own loss function. We use cross-entropy loss to train the annotator: $\mathcal{L}_{accuracy}(\mathbf{r}, \mathbf{y_s}, \mathbf{c}) = -\sum_i^M c_i \log(p_i(\mathbf{r}, \mathbf{y_s}))$, where $c_i = \mathbb{1}(\text{cell clustered to } i)$. To select marker gene sets and genes specific to each cluster, we use the exclusive LASSO regularizer [24] for the annotator. The regularizer takes the form of $\mathcal{L}_{\text{exclusive}}(B) = \sum_{j=1}^L (\sum_{k=1}^M \mid B_{jk} \mid)^2$, where B are the parameters of the logistic classifier. Thus the overall loss function for the annotator is

$$\mathcal{L}_{classification}(\mathbf{r}, \mathbf{y_s}, \mathbf{c}, B) = \mathcal{L}_{accuracy}(\mathbf{r}, \mathbf{y_s}, \mathbf{c}) + \gamma \mathcal{L}_{\text{exclusive}}(B)$$

, where γ is a weighting hyperparameter.

2.3 Training UNIFAN and Hyperparameter Selection

We first train the autoencoder for the gene set activity scores and obtain the gene set activity scores for all cells. We then pretrain the encoder and the decoder (e) of the autoencoder for clustering on the expression data to obtain an initial low-dimensional representations of the cells. We then run Leiden clustering [7] on these representations to obtain a guess of the number of clusters M, the initial cluster assignment and the cluster centroids S. Both the number of clusters M and cluster centroids S are refined as part of the training. Specifically, clusters with no cell assigned to them are removed. The annotator is then pretrained using the inferred gene set activity scores and the selected genes, if available. We use the cluster assignment initialized as described above as the true labels.

Finally, we train the annotator together with the cluster assignment part (the encoder and decoder (e) & decoder(q)). In each epoch, the annotator is trained by using the clustering results as the true label for each cell. The output from the annotator $\mathbf{p}(\mathbf{r})$ is in turn used to evaluate the annotator loss $\mathcal{L}_{annotator}$ for the cluster assignment part. As described previously, the annotator is optimized using its own loss function, separately from the cluster assignment part. See Supplementary Methods for details in training.

We use 32 dimensions for the low-dimensional representation $\mathbf{z_e}$ of a cell. To select the values for hyperparameters, including the neural network configuration and the weighting hyperparamters for the loss functions, we conducted a grid search using the Tabula Muris dataset and selected those hyperparameters values leading to the best performance over tissues. Unless specifically mentioned, the same set of values were applied to all datasets in all experiments. See Supplementary Methods for details on how we select the values. As discussed in Supplementary Results, our method is robust to different choices of hyperparameter values.

2.4 Performance Evaluation and Comparison to Other Methods

To evaluate the performance of UNIFAN and to compare it to prior methods including Leiden clustering [7], Seuratv3 [8], SIMLR [6], and DESC [9], we run each method on each dataset ten times with different initializations. For the Tabula Muris data, we run methods on each tissue separately. We use adjusted Rand index (ARI) and Normalized Mutual Information (NMI) implemented in scikit-learn [25] to compare clusters with ground truth annotations. Since calculating ARI for large datasets is time consuming, we use stratified random sampling when computing ARI for large datasets (>5e4 cells).

See Supplementary Methods for details on how we used prior methods, including hyper-parameter settings for these methods and for information on the evaluation strategy including how we compute enrichment p-values of the cell type marker sets in the highly weighted genes learned by the annotator.

3 Results

We developed UNIFAN to simultaneously cluster and annotate cells (and cell clusters) with known biological processes or pathways. We show that by integrating prior information about gene sets with observed expression data, we can improve clustering results while simultaneously making the clusters more interpretable.

3.1 UNIFAN Correctly Clusters Cells and Identifies Relevant Biological Processes

We first evaluated if UNIFAN can accurately cluster cells and reveal key pathways and cellular functions activated in cells assigned to different clusters. For this, we used the "pbmc28k" scRNA-seq dataset (Methods). UNIFAN clusters successfully captured different cell types when compared to manual annotations (ARI: 0.81, NMI: 0.77). Figure 3 presents UMAP [26] visualizations of $\mathbf{z_e}$ output from UNIFAN for each cell. As can be seen, clusters are mostly composed of cells from the same type which is a large improvement over other methods including Leiden clustering (shown in Figure 3 C), as we discuss below. By relying on known gene sets, UNIFAN is robust to noise and only focuses on relevant co-expressed sets of genes leading to much more coherent clusters. We observed similar results for the other datasets we tested as can be seen in Figures S4 - S6.

To annotate cell clusters, we examined the coefficients assigned by the "annotator" to different gene sets for each cluster. Figure 3 D presents some of top ranked sets for the different clusters. We observe that for cluster 0, the set "GOBP POSITIVE REGULATION OF T CELL RECEPTOR SIGNALING PATHWAY" is assigned a large weight and this cluster is annotated as CD4+ T cells in the original paper. For cluster 5 (which mainly includes CD8+ T cells), one of the top scoring sets is "REACTOME NEF MEDIATED CD8 DOWN REGULATION". Cluster 1 cells labeled as CD56 (dim) natural killer (NK) cells in the original paper. UNIFAN correctly assigns "GOBP REGULATION OF NK CELL MEDIATED IMMUNITY" and "KEGG NK CELL MEDIATED CYTOTOXICITY" as two of the top gene sets for this cluster. Cluster 3 and 6 correspond to classical monocyte (cMonocyte) and non-classical monocyte (ncMonocyte) respectively. While UNIFAN assigns biological processes related to "antigen presentation" and "inflammation" to both clusters, the biological process related to wound healing "GOBP REGULATION OF INFLAMMATORY RESPONSE TO WOUNDING" only appears in cluster 6. One of the main differences between ncMonocyte and cMonocyte is their role in wound healing [27] and so such assignment can make it much easier to correctly annotate this cluster of cells. In addition to the gene sets, we also evaluated genes highly weighted by the annotator by comparing them to known cell type marker sets. As shown in Figure 3 E, the most enriched cell type marker sets for each cluster correspond very well to the true cell labels, indicating that UNIFAN can indeed identify the marker genes for each cell type (cluster).

We observed similar performance in terms of cluster annotations for other datasets we tested. For example, for the "Atlas lung" dataset, UNIFAN successfully separated macrophage (cluster 2) and alveolar macrophage (cluster 5) as shown in Figure S4 A. The annotator selected"GOBP MACROPHAGE FUSION", "WP MACROPHAGE MARKERS" and "GOBP NEGATIVE REGULATION OF RESPONSE TO INTERFERON GAMMA" for both clusters. It also selected "GOBP REGULATION OF COLLAGEN FIBRIL ORGANIZATION" for cluster 8, which agrees well with the labels of cells in that cluster (fibroblasts). It selected "GOBP CILIUM MOVEMENT" for cluster 10, again in agreement with the types of cells in this cluster (ciliated). Similarly, the most enriched cell type marker sets for each cluster, learned from the highly weighted genes, corresponded very well to the true cell type labels (Figure S4 E).

3.2 UNIFAN Improves upon Prior Methods

We compared UNIFAN's clustering performance on all seven datasets with several prior methods proposed for clustering scRNA-Seq data. The number of cells in the datasets we used to compare the methods ranges from 366 (Aorta in Tabula Muris) to 96,282 ("Atlas lung" dataset) and so they can provide a good representation of the scRNA-Seq datasets being analyzed by researchers. The methods we compared to included two graph-based methods Leiden clustering [7] and Seuratv3 [8], a kernel-based method SIMLR [6] and a deep-learning based method DESC [9]. For each dataset, we ran each method ten times using different initializations. Results are presented in Figure 4 and

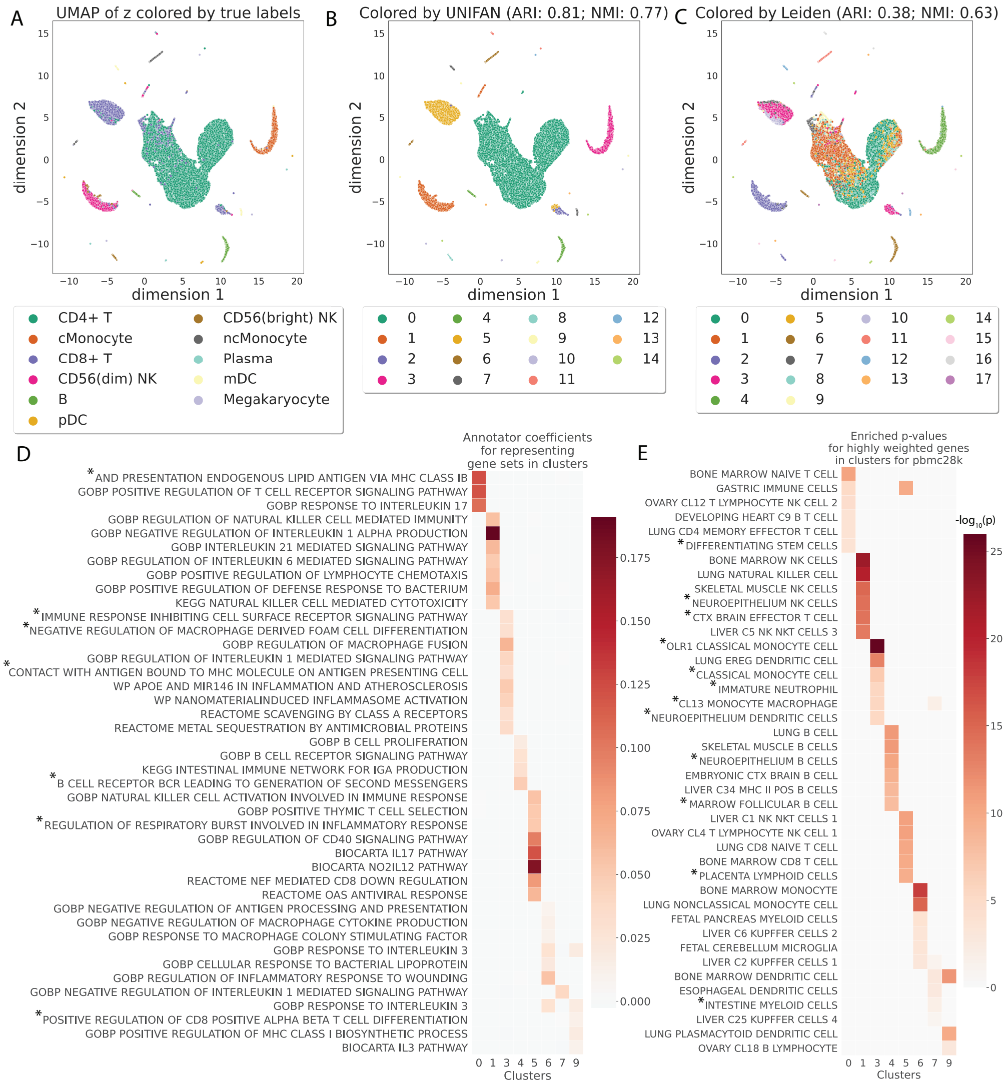

Figure 3: UNIFAN accurately clusters cells and correctly identifies biological processes / pathways. Results presented for the "pbmc28k" dataset. **A, B and C**: UMAP visualization of the low-dimensional representation $\mathbf{z_e}$ of cells output from UNFIAN. **A**: Colored by true cell type labels; **B**: colored by the clusters found by UNIFAN; **C**: colored by Leiden clustering. **D**: Coefficients learned by the annotator for highly ranked gene sets for some of the clusters. **E**: Enrichment p-values of cell type marker sets in the highly weighted genes learned by the annotator. Here we show the result from the best run for both UNIFAN and Leiden. Due to space limit, some gene set names in D and E are truncated (marked with *). See Table S1 and S2 for the full names.

S7. As can be seen, for all datasets, UNIFAN outperforms all other methods regardless of the evaluation metric being used (e.g., average ARI of UNIFAN and the best performing prior method on "pbmc28k": UNIFAN-0.72, Leiden-0.37; on "HuBMAP Spleen": UNIFAN-0.75, DESC-0.31; on "Tabula Muris": UNIFAN-0.70, SIMLR-0.53). The large improvement may result from the ability of UNIFAN's to focus on the more relevant sets of co-expressed genes rather than on co-expression that may results from noise or the large number of genes being profiled.

To further evaluate the different parts of UNIFAN in order to determine which input and processing is contributing the most to its success, we compared different versions of UNIFAN. These included "UNIFAN no annotator" which is composed of only the clustering part without the annotator, "UNIFAN random" which uses randomly generated features for the annotator and several other variations differing in the biological features used by the annotator including "UNIFAN gene sets" which uses only gene set activity scores and "UNIFAN genes & gene sets" (the default version) which uses both gene set activity scores and the selected genes.

As shown in Figure 4 and S8, the two "UNIFAN" variations using gene sets constantly outperformed the other versions which either did not use an annotator or used randomly generated values as features for the annotator. This results indicate that the use of the annotator to focus on the relevant co-expressed sets of genes is crucial to the performance of UNIFAN.

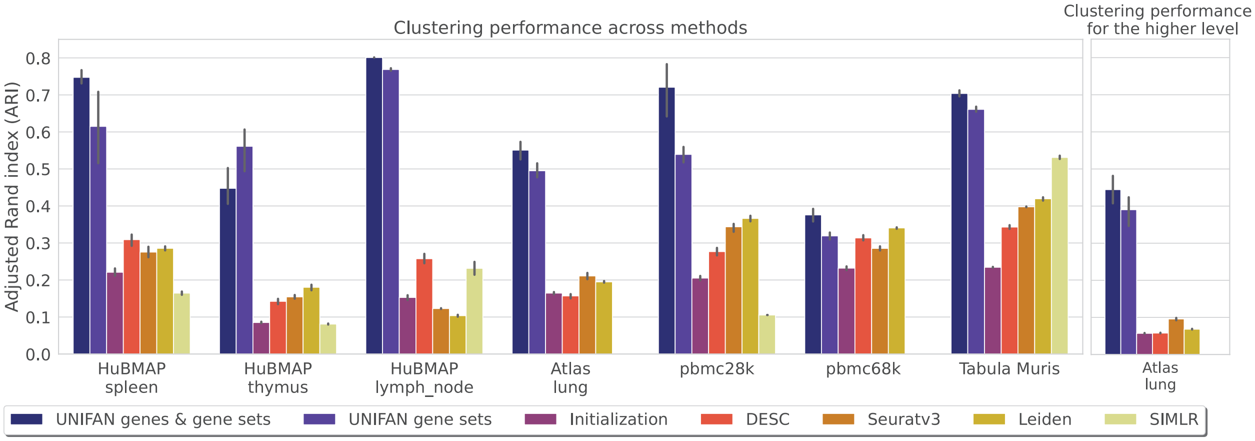

Figure 4: UNIFAN significantly outperforms other methods across all datasets. "UNIFAN genes & gene sets" is the default UNIFAN version using both gene set activity scores and a subset of genes as features for the annotator. "UNIFAN gene sets" uses only the gene set activity scores. "Initialization" is the initialization clustering results. The others are the prior methods we used for comparison. For the Tabula Muris data, we take the average over all tissues. See Figure S9 and S10 for tissue specific results. The "Atlas lung" data provides two levels of cell type annotations and so we show results for both (less detailed annotation comparison shown on the right). SIMLR was unable to cluster the "pbmc68k" and "Atlas lung" data since it ran out of memory. See Supplementary Methods for details.

3.3 Models are Transferable across Tissues and Species

Since different tissues from the same species or the same tissue across species may share cell types, we next explored if an autoencoder for gene set activity scores which is pre-trained on one tissue / species can also be useful for another tissue/species. The importance of such pre-training is that training of the autoencoder for gene set activity scores of UNIFAN is time consuming and so if this can be done offline (i.e. using prior data), then the application of the method to a new dataset can be much faster.

For this, we pre-train a gene set activity scores model using all available human datasets and apply the learned model to infer the gene set activity scores for Tabula Muris mouse datasets. We then run the clustering and annotation using these inferred scores and compare the results with those inferred from a model that was directly trained on the Tabula Muris data as discussed above. Given we focus on the usefulness of gene set activity scores, we use only these scores as features for the annotator ("UNIFAN gene sets") for this comparison.

Figure S13 presents the results. As expected, we see an overall decline in the average performance over tissues when comparing the results of pretrained and de novo models. However, for those mouse tissues that are also profiled in the human datasets we used, we observe similar performance when using the pre-trained human model. This is most apparent for spleen, lung and a few adipose tissues including SCAT (subcutaneous adipose tissue) and GAT (gonadal adipose tissue), as shown in Figure S13 and Figure S14-S15. These adipose tissues contain many immune cell types which are also present in many of the human tissues we used for pre-training (spleen, thymus, lymph nodes and PBMC). We further tested pre-trained models for individual tissues (i.e., training using spleen in human and testing only on spleen in mouse). As shown in Figure S13, for such analysis the performance is even better for the most part when compared to using the generally trained model. The only exception is thymus, where the mouse and human annotations differ significantly in the datasets we used. The major cell type in the Tabula Muris thymus data (thymocyte) does not appear in the HuBMAP human thymus data.

4 Discussion

Cell type assignment is one of the most important steps in scRNA-seq analysis. In most cases, such assignment is performed by first clustering cells and then assigning each cluster with a cell type based on differentially expressed genes or the expression of known cell markers.

Here we presented UNIFAN which improves both clustering and cluster annotations by using a large collection of gene sets [10]. UNIFAN infers gene set activity scores and uses them to regularize the clustering of cells. Such design improves the ability to identify biologically meaningful co-expressed genes and to use these to group cells. In addition to leading to improved clustering, UNIFAN also assigns a subset of the gene sets to clusters which can help characterize their cell type.

We compared UNIFAN to several popular methods for clustering scRNA-Seq data using datasets spanning a large number of organs from both human and mouse. As we show, UNIFAN consistently outperforms other methods across these datasets. We also analyzed the gene sets selected by UNIFAN for various clusters and demonstrated that they match well with the known cell types.

Analysis of the various parts of UNIFAN identified the annotator and the gene sets and genes it uses as the main sources for the improvement. The fact that adding variable genes as input improves performance is likely the result of the fact that current gene sets, while very useful, are incomplete. It is likely that we are still missing from current collections sets of genes characterizing some less known biological processes. In such cases, the selected genes capture groupings that are missed by the known gene sets.

UNIFAN can be slow on large datasets (Table S3 in Supplementary Results). The main time consuming part is training the gene set activity scores model for the data being clustered. To speed up the analysis, we have applied UNIFAN to a new dataset using a gene set activity scores model pretrained on another dataset. This greatly reduced run time (Supplementary Results) but did lead to drop in performance for tissues whose cell types were not well-represented in the pre-training dataset. As we obtain more data from tissues and conditions, we expect that we can further improve the ability to use pretraining to improve runtime.

UNIFAN is written in Python using PyTorch [28] and is freely avialable from `https://github.com/doraadong/UNIFAN`.

References

[1] Tanay, A. & Regev, A. Scaling single-cell genomics from phenomenology to mechanism. *Nature* **541**, 331–338 (2017). URL https://doi.org/10.1038/nature21350.

[2] Clarke, Z. A. *et al.* Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods. *Nat. Protoc.* **16**, 2749–2764 (2021). URL https://doi.org/10.1038/s41596-021-00534-0.

[3] Abdelaal, T. *et al.* A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome Biol.* **20**, 1–19 (2019). URL https://doi.org/10.1186/s13059-019-1795-z.

[4] Alavi, A., Ruffalo, M., Parvangada, A., Huang, Z. & Bar-Joseph, Z. A web server for comparative analysis of single-cell rna-seq data. *Nat. Commun.* **9**, 1–11 (2018). URL https://doi.org/10.1038/s41467-018-07165-2.

[5] Kiselev, V. Y., Yiu, A. & Hemberg, M. scmap: projection of single-cell rna-seq data across data sets. *Nat. Methods* **15**, 359–362 (2018). URL https://doi.org/10.1038/nmeth.4644.

[6] Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. & Batzoglou, S. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nat. Methods* **14**, 414–416 (2017). URL https://doi.org/10.1038/nmeth.4207.

[7] Traag, V. A., Waltman, L. & Van Eck, N. J. From louvain to leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 1–12 (2019). URL https://doi.org/10.1038/s41598-019-41695-z.

[8] Stuart, T. *et al.* Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902 (2019). URL https://doi.org/10.1016/j.cell.2019.05.031.

[9] Li, X. *et al.* Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nat. Commun.* **11**, 1–14 (2020). URL https://doi.org/10.1038/s41467-020-15851-3.

[10] Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.* **102**, 15545–15550 (2005). URL https://www.pnas.org/content/102/43/15545. https://www.pnas.org/content/102/43/15545.full.pdf.

[11] Consortium, H. *et al.* The human body at cellular resolution: the nih human biomolecular atlas program. *Nature* **574**, 187 (2019). URL https://doi.org/10.1038/s41586-019-1629-x.

[12] Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 1–5 (2018). URL https://doi.org/10.1186/s13059-017-1382-0.

[13] Adams, T. S. *et al.* Single-cell rna-seq reveals ectopic and aberrant lung-resident cell populations in idiopathic pulmonary fibrosis. *Sci. Adv.* **6**, eaba1983 (2020). URL https://www.science.org/doi/abs/10.1126/sciadv.aba1983. https://www.science.org/doi/pdf/10.1126/sciadv.aba1983.

[14] Van Der Wijst, M. G. *et al.* Single-cell rna sequencing identifies celltype-specific cis-eqtls and co-expression qtls. *Nat. Genet* **50**, 493–497 (2018). URL https://doi.org/10.1038/s41588-018-0089-9.

[15] Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 1–12 (2017). URL https://doi.org/10.1038/ncomms14049.

[16] Consortium, T. M. *et al.* Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature* **562**, 367–372 (2018). URL https://doi.org/10.1038/s41586-018-0590-4.

[17] Brbić, M. *et al.* Mars: discovering novel cell types across heterogeneous single-cell experiments. *Nat. Methods* **17**, 1200–1206 (2020). URL https://doi.org/10.1038/s41592-020-00979-3.

[18] Ernst, J., Vainas, O., Harbison, C. T., Simon, I. & Bar-Joseph, Z. Reconstructing dynamic regulatory maps. *Mol. Syst. Biol.* **3**, 74 (2007). URL https://doi.org/10.1038/msb4100115.

[19] Lu, Y., Rosenfeld, R., Simon, I., Nau, G. J. & Bar-Joseph, Z. A probabilistic generative model for go enrichment analysis. *Nucleic Acids Res.* **36**, e109–e109 (2008). URL https://doi.org/10.1093/nar/gkn434.

[20] Fortuin, V., Hüser, M., Locatello, F., Strathmann, H. & Rätsch, G. Som-vae: Interpretable discrete representation learning on time series. *arXiv preprint arXiv:1806.02199* (2018).

[21] Oord, A. v. d., Vinyals, O. & Kavukcuoglu, K. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937* (2017).

[22] Xie, J., Girshick, R. & Farhadi, A. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, 478–487 (PMLR, 2016).

[23] van der Maaten, L. & Hinton, G. Visualizing high-dimensional data using t-sne. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

[24] Zhou, Y., Jin, R. & Hoi, S. C.-H. Exclusive lasso for multi-task feature selection. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 988–995 (JMLR Workshop and Conference Proceedings, 2010).

[25] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

[26] Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using umap. *Nat. Biotechnol.* **37**, 38–44 (2019). URL https://doi.org/10.1038/nbt.4314.

[27] Schmidl, C. *et al.* Transcription and enhancer profiling in human monocyte subsets. *Blood* **123**, e90–e99 (2014). URL https://doi.org/10.1182/blood-2013-02-484188.

[28] Paszke, A. *et al.* Automatic differentiation in pytorch (2017).

[29] Xin, J. *et al.* High-performance web services for querying gene and variant annotation. *Genome Biol.* **17**, 1–7 (2016). URL https://doi.org/10.1186/s13059-016-0953-9.

[30] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization (2014). arXiv:1412.6980.