EasyChair Preprint

№ 4571

Vector Representation of Gene Co-expression in Single Cell RNA-Seq

Nicholas Ceglia, Florian Uhlitz and Andrew McPherson

November 15, 2020

Vector Representation of Gene Co-expression in Single Cell RNA-Seq

Nicholas Ceglia[1], Florian Uhlitz[1] and Andrew McPherson[1]

Memorial Sloan Kettering Cancer Center, New York NY 10065, USA

Abstract. The sparsity of gene expression is a well known problem in single cell RNA-seq data. Known as dropout, the gene expression observed for each cell is only a fraction of the total transcriptome. Several techniques have been adopted to address this challenge including variable gene selection and expression imputation. We present an approach for finding dense vector representations of genes from co-expression that can be used in place of the sparse expression profile over cells. By leveraging co-expression across all cells, each gene vector is a meaningful representation that is independent of missing data from individual cells. Similar genes, measured by cosine similarity between vectors, are found to correspond to known cell type markers. Using latent space arithmetic, these gene vectors have the additive capacity to accurately describe each cell and can be used to generate a low dimensional cell embedding. It is also possible to decompose and subtract sources of variation including batch effects. Any feature that can be described as a set of genes can be represented as a composite of vectors. We demonstrate the application of these vectors in identification of cell type markers, dimensionality reduction, and batch correction.

Keywords: Single Cell RNA-Seq · Gene Expression · Vector Representations · Dimensionality Reduction · Batch Effect Correction.

1 Introduction

Most single cell RNA-seq analysis involves normalization and dimensionality reduction of the initial gene-by-cell count matrix using principal component analysis (PCA). PCA allows for a reduction in sparsity while preserving explanatory variation in each cell. This is an ideal input for building the nearest neighbor graph for unsupervised clustering algorithms [11] and visualization techniques including t-SNE and UMAP [5, 6]. To understand the gene expression that is driving this variation, a secondary step is performed to map the normalized expression to the identified structure in lower dimensional space. This step is most often the identification of differentially expressed genes between unsupervised clusters. The initial normalization may be negatively effected by dropout [9, 3]. Failure to normalize correctly may have consequences on the differential expression and clustering results. Methods such as *sctransform* [3] have been developed to more faithfully perform normalization in the presence of missing data. An alternative to correcting missing values in the sparse expression profile is to define

a dense representation of each gene that utilizes a more robust feature such as gene co-expression.

The approach presented in this paper generates vector representations of genes from co-expressed pairs across a single cell RNA-seq dataset based on the *Word2Vec* technique [7]. Within natural language processing, vector representations of words are used to model the syntactic and semantic similarities in a corpus of text. These methods have proven remarkably effective at generating meaningful lower dimensional representations that capture subtle word associations. *Word2Vec* learns a distributed embedding of word specific vectors within a neural network constructed of a single hidden layer with linear activation. The network can be trained to predict either a target word given a set of context words or the set of context words given the target word. The later method is described as the *Skip-Gram* architecture and is the chosen model for the approach described in this paper. For each word in the original text, the associated context words are found surrounding the target word within a predefined window size. A sliding window generates pairs of target and context words that are used as training data. After training, the hidden layer contains a vector of equivalent size for each word in the original text. In the presented method, the analog of each word is a gene and a gene's context is defined as those genes co-expressed in a single cell.

By sampling input and output gene pairs from each cell's co-expressed genes, we apply the *word2vec* model to single cell RNA-seq and obtain a learned vector representation for each expressed gene. The similarity of co-expression between genes can be measured by cosine similarity between vectors. If similar genes are represented by vectors with roughly equivalent magnitude and direction, the average vector of any set of genes can be used as a single representation of their co-expression. It follows that each cell can be represented by an average vector of the expressed genes each weighted by the normalized expression. An average weighted vector for each cell can be combined into a single cell embedding.

A similar technique has been previously used to generate vector representations of genes in bulk RNA-seq [2]. *Gene2Vec* uses the co-expression of genes found in 984 human GEO datasets across varied tissues and conditions with greater than 30 individual samples. In this context, co-expression was defined as gene pairs with a pearson correlation coefficient greater than 0.9 between samples within a single dataset. A t-SNE embedding is used to highlight that functionally related genes in known *MSigDB* pathways form distinct clusters in the low dimensional space. Utilizing a similar framework, this approach is extended to single cell RNA-seq by our approach.

Unsupervised clustering of gene vectors can be used to define groups of robustly co-expressed genes. We demonstrate that these clusters accurately capture distinct cell types in benchmark datasets and that the most similar genes define known cell types with equal specificity of differentially expressed genes. We demonstrate that when used as input to t-SNE, the visualizations are comparable to those generated with traditional PCA. Finally, we demonstrate that it is possible to derive an average vector for any feature that can be decomposed into

a set of genes. By defining an average vector for a known batch, it is possible to subtract batch effects with results comparable to state of the art methods. Our approach enables many of the standard analysis performed on single cell RNA-seq within a single embedding framework.

2 Methods

2.1 Model

To construct the gene pair input to the model, a set of non-zero expression gene symbols is generated for each cell. A complete list of gene pairs is concatenated from the length two permutations of each cell's co-expressed set of genes. By using permutations, the input enforces the property that the co-expression, and ultimately the similarity, of any two genes is symmetric. The frequency of a gene pair in the global list is determined by the number of cells in which that pair is found to be co-expressed. The final list of pairs is subsequently used as input to a neural network.

The neural network consists of a single hidden layer with linear activation. The size of the input and output layers is determined by the total number of expressed genes across all cells. The number of hidden layer units defines the length of the learned vector representations. Sub-sampling frequent words has been previously proposed as a means of training optimization [7]. While these methods aim to more aggressively sample lower frequency input pairs, proportionally increasing the sampling of these genes decreases the similarity between higher frequency and more representative gene pairs. Instead, a uniform probability of discarding any co-expressed gene pair was used to preserve the overall frequency distribution and decrease the number of redundant training examples. The neural network is trained in a feed-forward manner and optimized using stochastic gradient descent. Figure 1 summarizes the model architecture.

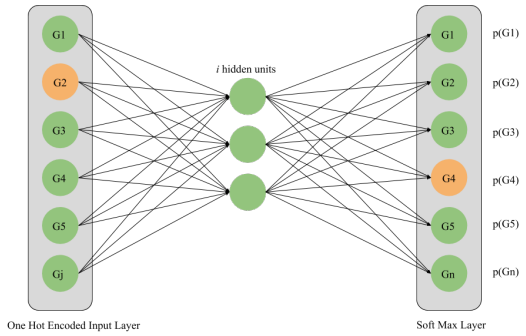


Fig. 1: Neural network constructed with a single hidden layer of linear units connecting a one hot encoded input to a soft-max output layer. Each training example maps a pair of co-expressed genes within the dataset.

2.2 Lower Dimensional Embeddings

The weight matrix connecting the hidden layer to the output layer contains the lower dimensional gene embedding defined as G. The gene embedding dimensions are given as j genes and i latent dimensions corresponding to the number of hidden units in the neural network.

$$G_{j,i} = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,i} \\ g_{2,1} & a_{2,2} & \cdots & g_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ g_{j,1} & g_{j,2} & \cdots & g_{j,i} \end{pmatrix}$$

A representative cell vector can be computed as the average vector of the gene embedding weighted by the normalized expression of each gene in the given cell. Equation 1 describes this vector for cell C where w_i is the log-normalized expression of gene i for cell j.

$$\vec{C} = \left(c_1\ c_2\ c_3\ ...\ c_k\right) where \tag{1}$$

$$c_k = \frac{\sum_{i=1}^{j} w_i g_{\mathrm{i,k}}}{\sum_{i=1}^{j} w_i}$$

Similarly, a representative vector can be defined for any subset of cells. An average vector can be computed across n cells using Equation 1 where w_i is substituted for the mean expression of the j^{th} gene over the set of n cells.

$$\overline{w_i} = \frac{w_1 + w_2 + \cdots + w_n}{n} \tag{2}$$

3 Results

3.1 Human PBMCs

Peripheral blood mononuclear cells (PBMCs) originally made available by 10x Genomics are used to demonstrate results from our approach. Gene vectors are computed and similarities between genes is shown to capture expected marker gene co-expression. We map unsupervised clustering results to known cell types using cosine similarity and demonstrate that these clusters can be used to identify marker genes. Finally, we compute a cell embedding for t-SNE visualization based on the learned gene vectors. This dataset has been featured in a number of tutorials demonstrating single cell RNA-seq analysis [10, 13] and provides a well understood platform for comparing the quality of presented results with standard techniques. The log-normalized count matrix can be downloaded from the Scanpy API [13]. The dataset consists of 2700 cells and 32,738 genes. Cell type labels have been generated from *leiden* clustering using a set of 14 known gene markers corresponding to the expected cell types. Overall, there are eight cell types labeled as CD4 and CD8 T cells, NK cells, B cells, FCERG1+ and CD14+ monocytes, dendritic cells, and megakaryocytes.

After training the neural network with a hidden layer of 100 units, we obtain a gene vector for each of the 32,738 genes. The cosine similarity between marker genes is computed and Figure 3 highlights the similarity between each marker. Markers of the same cell types are found to be most similar. Markers of similar cell types, such as T and NK cells, are more similar than distinctly different cell types. Given the distinct expression of marker genes in the each cell type, this result demonstrates cosine similarity as a valid measure of gene co-expression. Cosine similarity can also be computed between any two expressed genes. Figure 2 displays the top 10 most similar genes to each marker in a graph format. Marker gene nodes are colored by cell type to highlight similarity between like cell types. T Cell associated genes including CD3D, CD3E, and CD3G can be found connected to CD4 and CD8 T cell markers.

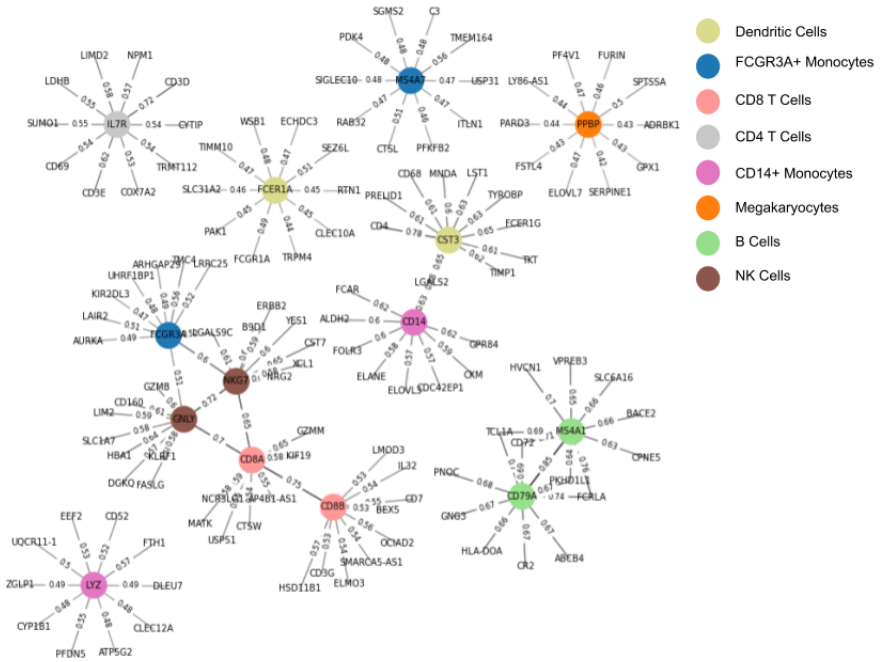


Fig. 2: Top 10 most cosine similar gene for each marker. Markers are colored by cell type. Similar cell types such as NK and CD8 T cells are directly connected by similar marker genes. Known T Cell associated genes CD3E, CD3D, and CD3G are found similar to both CD4 and CD8 T cells.

Gene co-expression patterns can be identified through unsupervised clustering of the gene embedding. Using hierarchical agglomerative clustering with a cosine distance metric [8], 20 clusters were generated with unique gene expression

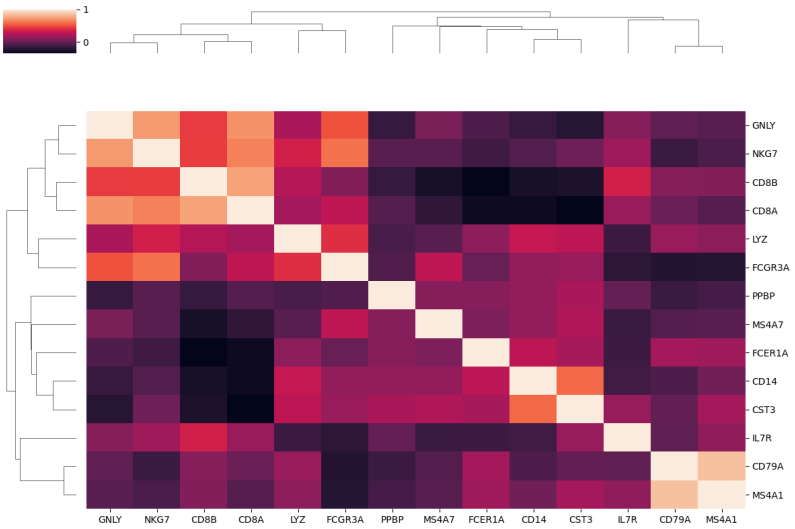


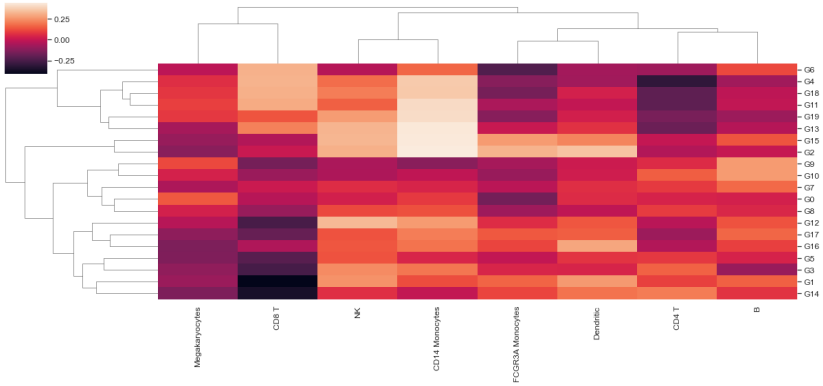
Fig. 3: Cosine Similarity of marker genes for 2700 PBMCs. Marker genes are found to be clustered in the heatmap by associated cell type. Markers for NK cells (NKG7 and GNLY) are found to be more similar to CD8 T cell markers (CD8A and CD8B). Markers for more transcriptionally distinct cell types, such as B Cells, are found to be clearly unique.

patterns. For each cluster, the average vector of associated genes is computed. The cosine similarity of each gene with the average vector of each cell type is show in Figure 4a. The cosine similarity of each cluster can be mapped to individual cells. Figure 4 shows how the similarity of each cluster maps to cells in the t-SNE embedding.

A cell embedding based on Equation 1 can be computed for each of the 2700 PBMC cells. Figure 5 displays the resulting t-SNE visualization of the cell embedding compared to the t-SNE computed with PCA in Scanpy [13]. Qualitatively, the results are similar to the previously generated low dimensional embedding. The cell embedding generated from our approach demonstrates that our method can be used to generate lower dimensional visualizations with results comparable to accepted methods. Furthermore, clustering of the gene embedding can be used as a replacement for differential analysis to identify cell type markers.

3.2 Mouse Cell Atlas

A diverse selection of cells from two mouse cell atlas datasets [1, 4] curated for comparison of batch effect correction methods [12] are used to further demonstrate batch correction with our method. The dataset has previously undergone quality control [12] and the resulting normalized count matrices are publicly available for download. The two datasets were generated on different sequencing



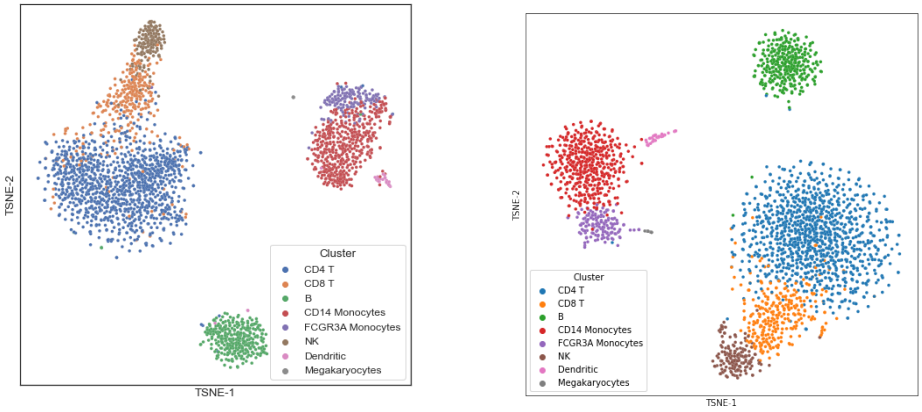
(a)

(b)

Fig. 4: Gene cluster similarity with cell types and individual cells. (a) Gene cluster similarity with given cell types. (b) t-SNE colored by cosine similarity between each gene cluster and each cell. Distinct clusters can be visualized for several cell types.

platforms. The first batch was generated using Microwell-Seq and the second batch was generated using Smart-Seq2. Batch 1 and Batch 2 will be used respectively to label the sequencing platform. The final datasets includes 6067 cells and 1962 genes.

The cell embedding computed on both batches was clustered by agglomerative hierarchical clustering and the resulting clusters are used as reference for applying local batch correction using vectors. For each cluster, average vectors were computed for both batches using Equation 2. Using Batch 1 as a reference,



(a) t-SNE generated from gene embedding.

(b) t-SNE generated from PCA embedding.

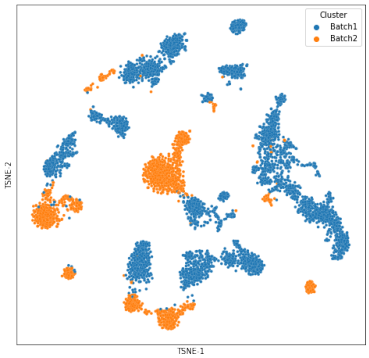
Fig. 5: Comparison of t-SNE.

a corrective vector was computed by subtracting the Batch 2 vector from the Batch 1 vector. The corrective vector is then added to each of the cells from Batch 2 within the cluster. Figure 6 displays the cell embedding colored by batch before correction and after correction. Cell types found to be distinct in the same embedding for each batch have been corrected into a single cell type in the t-SNE visualization.

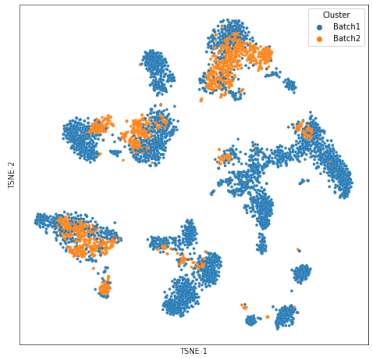
4 Discussion

The method presented in this paper provides a platform for investigating gene co-expression in single cell RNA-seq data. The results obtained from this approach are comparable to accepted methods. The ability to create composite vectors from any subset of genes provides a novel method for generating low dimensional embeddings. Through vector arithmetic, these vectors provide a means to describe and decompose variation in the dataset. Furthermore, the ability to cluster gene expression patterns and relate these clusters directly to cells allows the identification of gene expression patterns without traditional differential analysis. Importantly, gene vectors are a distributed representation of co-expression derived from the entire set of cells. If critical genes are co-expressed in some subset of like cells, this method will be able to capture that relationship regardless of dropout in the cell population of interest.

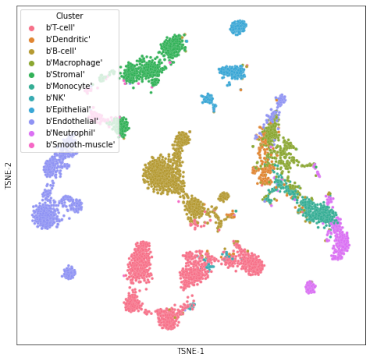
Training gene vectors without optimization is computationally intensive. Several techniques for decreasing training time and increasing quality of results have been proposed for *word2vec* models [7]. Negative sampling can be implemented to decrease the number of weight updates for each training example given the sparsity of the input and output vectors. Using previously described parameters,



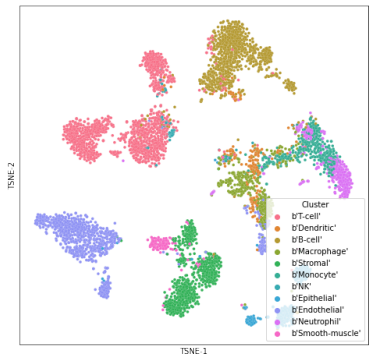
(a) Uncorrected cell embedding with Microwell-Seq (Batch 1) and Smart-Seq2 (Batch 2).



(b) Corrected cell embedding after applying local corrective vectors for size unsupervised clusters.



(c) Corrected cell embedding with cell type labels. T cells are visually separated by batch label.



(d) Corrected cell embedding where T cells populations can be visualized in a single cluster.

Fig. 6: Batch correction using local corrective vectors derived from unsupervised clustering.

training time can be significantly reduced without general loss of results in our method. The current software library supports this optimization with significant performance improvements.

This work can be extended to develop vector representations for known pathways of interest. By selecting genes to construct an average vector, it is possible to test the similarity of a given pathway to each cell. The current library supports the creation of vectors of interest for any subset of genes. Additionally, vectors can be trained on different features other than co-expression. Some examples might include the use of known transcription factor targets and gene regulatory networks to select gene pairs for learning vector representations. The method presented and the software library developed can be used as an initial platform for further exploration in developing gene representations that do not rely on sparse normalized expression results.

5 Software Availability

A python software library using PyTorch is available on Github (*github.com/nceglia/compass*). Jupyter notebooks are available for the PBMCs dataset and the two technology mouse cell atlas dataset.

References

1. Consortium, T.M., et al.: Single-cell transcriptomics of 20 mouse organs creates a tabula muris. Nature **562**(7727), 367 (2018)
2. Du, J., Jia, P., Dai, Y., Tao, C., Zhao, Z., Zhi, D.: Gene2vec: distributed representation of genes based on co-expression. BMC genomics **20**(1), 82 (2019)
3. Hafemeister, C., Satija, R.: Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression. Genome biology **20**(1), 1–15 (2019)
4. Han, X., Wang, R., Zhou, Y., Fei, L., Sun, H., Lai, S., Saadatpour, A., Zhou, Z., Chen, H., Ye, F., et al.: Mapping the mouse cell atlas by microwell-seq. Cell **172**(5), 1091–1107 (2018)
5. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
6. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
9. Qiu, P.: Embracing the dropouts in single-cell rna-seq analysis. Nature communications **11**(1), 1–9 (2020)
10. Satija, R., Farrell, J.A., Gennert, D., Schier, A.F., Regev, A.: Spatial reconstruction of single-cell gene expression data. Nature biotechnology **33**(5), 495–502 (2015)

11. Traag, V.A., Waltman, L., van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Scientific reports **9**(1), 1–12 (2019)
12. Tran, H.T.N., Ang, K.S., Chevrier, M., Zhang, X., Lee, N.Y.S., Goh, M., Chen, J.: A benchmark of batch-effect correction methods for single-cell rna sequencing data. Genome biology **21**(1), 1–32 (2020)
13. Wolf, F.A., Angerer, P., Theis, F.J.: Scanpy: large-scale single-cell gene expression data analysis. Genome biology **19**(1), 15 (2018)