GeneVector: Identification of transcriptional programs using dense vector representations defined by mutual information

Nicholas Ceglia[a,1], Zachary Sethna[a], Florian Uhlitz[a], Viktoria Bojilova[a], Nicole Rusk[a], Bharat Burman[a], Andrew Chow[a], Sohrab Salehi[a], Farhia Kabeer[b,c], Samuel Aparicio[b,c], Benjamin Greenbaum[a], Sohrab P. Shah[a], Andrew McPherson[a]

[a] Computational Oncology, Department of Epidemiology and Biostatistics, Memorial Sloan Kettering Cancer Center, New York, NY, USA
[b] Department of Molecular Oncology, British Columbia Cancer Research Centre, Vancouver, British Columbia, Canada'
[c]Department of Pathology and Laboratory Medicine, University of British Columbia, Vancouver, British Columbia, Canada.
[1] Correspondence: ceglian@mskcc.org, 1275 York Ave, New York, NY 10065

Abstract

Deciphering individual cell phenotypes from cell-specific transcriptional processes requires high dimensional single cell RNA sequencing. However, current dimensionality reduction methods aggregate sparse gene information across similar cells, without directly measuring the relationships that exist between genes. By performing dimensionality reduction with respect to gene co-expression, low-dimensional features can model these gene-specific relationships and leverage shared signal to overcome sparsity. We describe GeneVector, a scalable framework for dimensionality reduction implemented as a vector space model using mutual information between gene expression. Unlike other methods, including principal component analysis and variational autoencoders, GeneVector uses latent space arithmetic in a lower dimensional gene embedding to identify transcriptional programs and classify cell types. Using four single cell RNA-seq datasets, we show that GeneVector was able to capture phenotype-specific pathways, perform batch effect correction, interactively annotate cell types, and identify pathway variation with treatment over time. GeneVector is available as an open source python package at https://github.com/nceglia/genevector.

Introduction

Maintenance of cell state and execution of cellular function are based on the coordinated activity within networks of related genes. To approximate these connections, transcriptomic studies have conceptually organized the transcriptome into sets of co-regulated genes, termed *gene programs* or *metagenes* (J. M. Stuart 2003; Svensson et al. 2020). The first intuitive step to identify such co-regulated genes is the reduction of dimensionality for sparse expression measurements: high dimensional gene expression is compressed into a minimal set of explanatory features that highlight similarities in cellular function. However, to map existing biological knowledge to each cell, the derived features must be interpretable at the gene level.

To find similarities in lower dimensions, biology can borrow from the field of natural language processing. NLP commonly uses dimensionality reduction to identify word associations within a body of text (Pennington, Socher, and Manning 2014). To find contextually similar words, NLP methods make use of vector space models to represent similarities in a lower dimensional space. Similar methodology has been applied to bulk RNA-seq expression for finding co-expression patterns (Pennington, Socher, and Manning 2014; Du et al. 2019). Inspired by this work, we developed a tool that generates gene vectors based on single cell RNA (scRNA)-seq expression data. While current methods reduce dimensionality with respect to sparse expression across each cell, our tool produces a lower dimensional embedding with respect to each gene. The lower dimensional vectors derived from GeneVector provide a framework for identifying metagenes within a gene similarity graph and relating these metagenes back to each cell using latent space arithmetic.

The most pervasive method for identifying the main sources of variation in scRNA-seq studies is principal component analysis (PCA). PCA allows for a reduction in sparsity while preserving explanatory variation in each cell. Importantly, the relationship of principal components to gene expression is linear, allowing lower dimensional structure to be directly related to expressional variation. This is an ideal input for building the nearest neighbor graph for unsupervised clustering algorithms (Traag, Waltman, and van Eck 2019) and visualization methods including t-SNE (Pezzotti et al. 2017) and UMAP (McInnes et al. 2018). However, the assumption of a continuous multivariate gaussian distribution creates distortion in modeling read counts arising from a negative binomial distribution.

More sophisticated methods address this issue. The single cell variational inference (scVI) framework generates an embedding using non-linear autoencoders that can be used in a range of analyses including normalization, batch correction, gene-dropout correction, and visualization (Svensson et al. 2020; Lopez et al. 2018). Results from the scVI embeddings show improved performance over traditional PCA-based analysis in these tasks. However, scVI embeddings have a non-linear relationship to the original count matrix that may distort the link between structure in the generated embedding and potentially identifiable gene programs (Svensson et al. 2020).

To maintain this linear relationship, scVI was extended to include a non-Gaussian linearly decoded variational autoencoder (LDVAE). LDVAE generates a low dimensional embedding that is linearly related to variation in gene expression with the assumption of a negative binomial distribution of read counts and produces interpretable results that relate to individual cells (Svensson et al. 2020) . However, the relationship between gene expression and cell representation is tied to correlated variation across cells. In cases where gene expression is highly entropic without a linear dependence between genes, this relationship can be obscured. In these cases, the mutual information shared between the empirical probability distribution of counts can elucidate the relationship, summarized by how much information is shared between both genes. We suggest that combining vector space models with information theoretic measurements enables model construction of a correlation-independent linear representation between genes.

We present GeneVector (**Figure 1**) as a framework for generating linear embeddings constructed from the mutual information between genes. GeneVector summarizes the co-expression of genes as mutual information between the probability distribution of read counts across cells. Mutual information scores are used to train a vector space model encoding the link between genes, as opposed to latent patterns found in the expression matrix. After model fitting, latent space arithmetic can be applied to produce a cell-based embedding and a similarity graph constructed from the cosine distance between each gene vector.

Using peripheral blood mononuclear cells (PBMCs) subjected to interferon beta stimulation, we first demonstrate that GeneVector is capable of identifying metagenes that correspond to cell-specific transcriptional processes that include canonical phenotype markers and interferon activated gene expression. Compared to results obtained from Leiden clustering or LDVAE, metagenes identified are more coherent in terms of known interferon signaling pathway signatures. We further show that latent space arithmetic can be used to label cell types with a pseudo-probability by generating representative vectors from known gene markers. These results are consistent with previous annotations in a collection of diverse cells from multiple cancer types. We use vector space arithmetic to directly map metagenes to site specific changes in primary and metastatic sites in high grade serous ovarian cancer. Finally, we show that GeneVector can identify subtle transcriptional programs in human cancer that change over time with respect to cisplatin treatment in patient-derived xenografts. Using vector space arithmetic, we deconstruct treatment labels into vectors and demonstrate that these vectors identify genes that reflect drug resistance pathways with exposure to cisplatin.

Results

We developed GeneVector as a tool to generate low dimensional gene embeddings and identify metagenes from a co-expression similarity graph. To do this, a single layer neural network is trained over all possible gene pairs. The input weights (w_i) and output weights (w_j) are updated with adaptive gradient descent (AdaGrad) (Duchi et al. 2011). Gene co-expression relationships are modeled from the product of the mutual information, computed from a joint probability distribution, and the probability of co-expression over all cells. The model identifies the strongest relationships between gene pairs that both share information in count expression and are co-expressed in a large number of cells. Training loss is evaluated as the mean squared error of this value with the model output.

The final latent space is a matrix *W* defined as a series of vectors w_g for each gene *g*. Gene vectors weighted by expression in each cell are combined to generate the cell embedding. The cell embedding can be batch corrected by selecting a reference set of cells and using vector subtraction to shift each set of non-reference cells by the corresponding difference in the latent space. The batch label corresponding to the largest set of cells is selected as the reference. After batch correction, a fully connected graph is constructed in which each node is defined as a gene and each edge is weighted by cosine similarity. Independent downstream analysis of the cell embedding and co-expression similarity graph includes the generation of metagenes and phenotype assignment based on sets of marker genes (**Figure 1**).

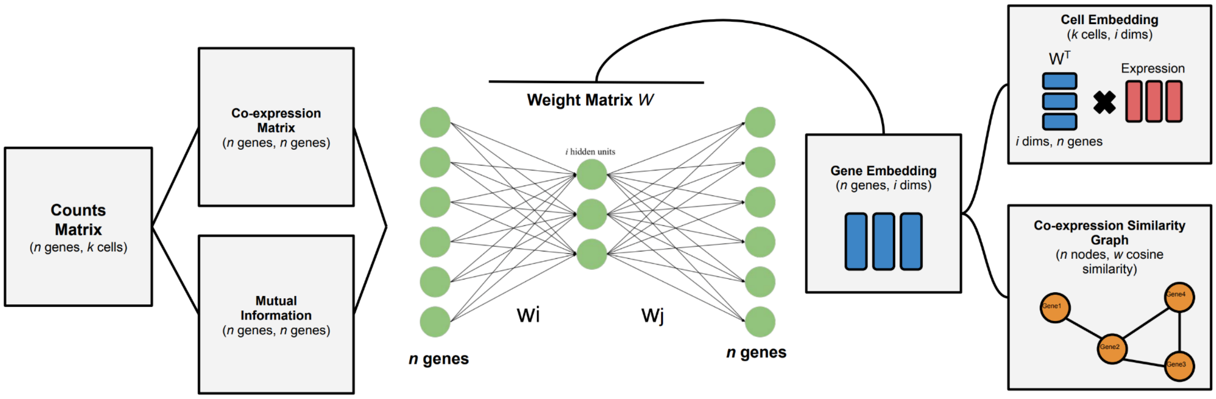

Figure 1: GeneVector Framework

Overview of GeneVector framework starting from single cell read counts. The products of the co-expression matrix and mutual information are used to train a single layer neural network. From the resulting weight matrix a gene embedding, cell embedding, and co-expression similarity graph is constructed. Downstream analyses include cell type classification and transcriptional program generation.

After generating the co-expression similarity graph, we applied Leiden clustering (Traag, Waltman, and van Eck 2019)to identify metagene clusters . The clustering algorithm is applied to the gene embedding, rather than the cell embedding as is often done to identify transcriptional clusters. Our approach requires selection of a resolution parameter, where larger values result in a smaller number of genes included in each metagene. We find that the same metagenes exist at multiple resolutions and that larger metagenes are functionally related aggregates of metagenes found at smaller resolutions.

To perform cell type assignment, a set of known marker genes is used to generate a representative feature vector for each cell type. For a given cell, the cosine similarity of each possible phenotype is computed between the cell vector and the marker gene vector. Softmax is applied to cosine distances to obtain a pseudo-probability over each phenotype. Discrete labels can be assigned cells by selecting the phenotype corresponding to the maximum pseudo-probability. Inversely, cosine distance between each gene vector and a representative feature vector for a set of cells in the dataset can be sorted to generate a ranked list of genes with predictive power for those cells.

Comparison of interferon stimulated transcriptional programs in 10k Human PBMCs

To identify cell-specific metagenes related to interferon beta stimulation and compare to transcriptional programs identified with differential expression analysis of Leiden clustering and LDVAE loadings, we computed a co-expression similarity graph over a peripheral blood mononuclear cells (PBMCs) dataset consisting of scRNA-seq data from 6,855 quality control filtered cells, representing an interferon beta stimulated sample and a control sample (Kang et al. 2018). The original count matrix was subset to 1,000 highly variable genes using the Seurat V3 method (Butler et al. 2018) as implemented in Scanpy (Wolf, Angerer, and Theis 2018). We used previously annotated cell types generated from unsupervised clustering as ground truth labels (Butler et al. 2018; T. Stuart et al. 2019). The comparison of the uncorrected UMAP embedding, computed on the GeneVector cell embedding, and GeneVector-based batch correction (**Figure 2A**) demonstrates correction of the large variation resulting from interferon stimulation. Additionally, cell types from both conditions are correctly aligned on the batch corrected UMAP (**Figure 2B**).

As a method of both validation and exploration, GeneVector provides the ability to query similarity in genes. For a given target gene, a list of the closest genes sorted by cosine distance can be returned in a single function call. This is useful in both validating known markers and identifying the function of unfamiliar genes by context. For example, the genes most similar to CD8A, shown as a network with edges weighted by cosine similarity (**Figure 2C**), include all well known markers of CD8 T cells (van der Leun et al. 2020). This graph also highlights the similarity of CD8 T cell markers to the long non-coding RNA transcript MIAT. While not a canonical marker of CD8 T cells, MIAT expression has been found to be correlated with immune cell types in breast cancer and shown to correlate with interferon gamma pathway and cytotoxicity scores in 14 additional cancer types (Ye et al. 2021; Li et al. 2020).

The gene embedding can be visualized as a UMAP, similar to the familiar cell-based visualizations pervasive in scRNA-seq studies (**Figure 2D**). In total, GeneVector identified 123 metagenes and those corresponding to several cell types (CD8 T cells, B cells, FCGR3A+/CD14+ monocytes, and megakaryocytes) and pathways (Interferon Stimulated Genes and MHC Class II) are highlighted (**Figure 2D**). We note that the metagene related to interferon pathway activation is composed of exclusively interferon stimulated genes (ISGs) (IFIT1, IFIT2, IFIT3, ISG20, LY6E, MX1, RSAD2, IRF7, OAS1, SAMD9L, TNFSF10, MT2A, and GBP1) and the expression of each of these genes is higher in interferon-stimulated cells, as opposed to control, in each cell type (**Figure 2E**).

To compare the GeneVector with LDVAE, we trained an LDVAE model using 10 latent dimensions for 250 epochs with control and stimuli as batch labels in the SCVI framework. In contrast to the specificity of the GeneVector ISG metagene to only interferon stimulated genes, the nearest LDVAE loading (Z_7) includes only a few interferon-related genes (RSAD2, IFIT2, TNFSF10, and CXCL10) in addition to CCL13, SCIN, CSRP2, NEXN, and others (**Figure 2F: LDVAE ISG Signature**). Overall, GeneVector identified a more comprehensive and specific ISG specific signature compared to LDVAE.

We also compare results obtained with GeneVector to those computed from differential expression analysis of Leiden clustering as implemented in Scanpy (Wolf et al. 2018). Leiden clustering identified 17 transcriptional clusters and cluster 3, found in CD4 T cells, was noted to primarily differentially express ISG (**Figure 2F: Leiden ISG Signature**). Here, we are given the erroneous impression that CD4 T cells are characterized almost entirely by ISGs genes, whereas GeneVector reveals a cell type agnostic ISG signature that is found in each cell type (**Figure 2G**). We scored the GeneVector ISG signature using the gene module score method (Satija et al. 2015) over all cells and displayed these values alongside the Leiden derived ISG signature (blue) (**Figure 2G**). While we do find increased module scores in CD4 T cells, the signature is not exclusive to CD4 T cells.

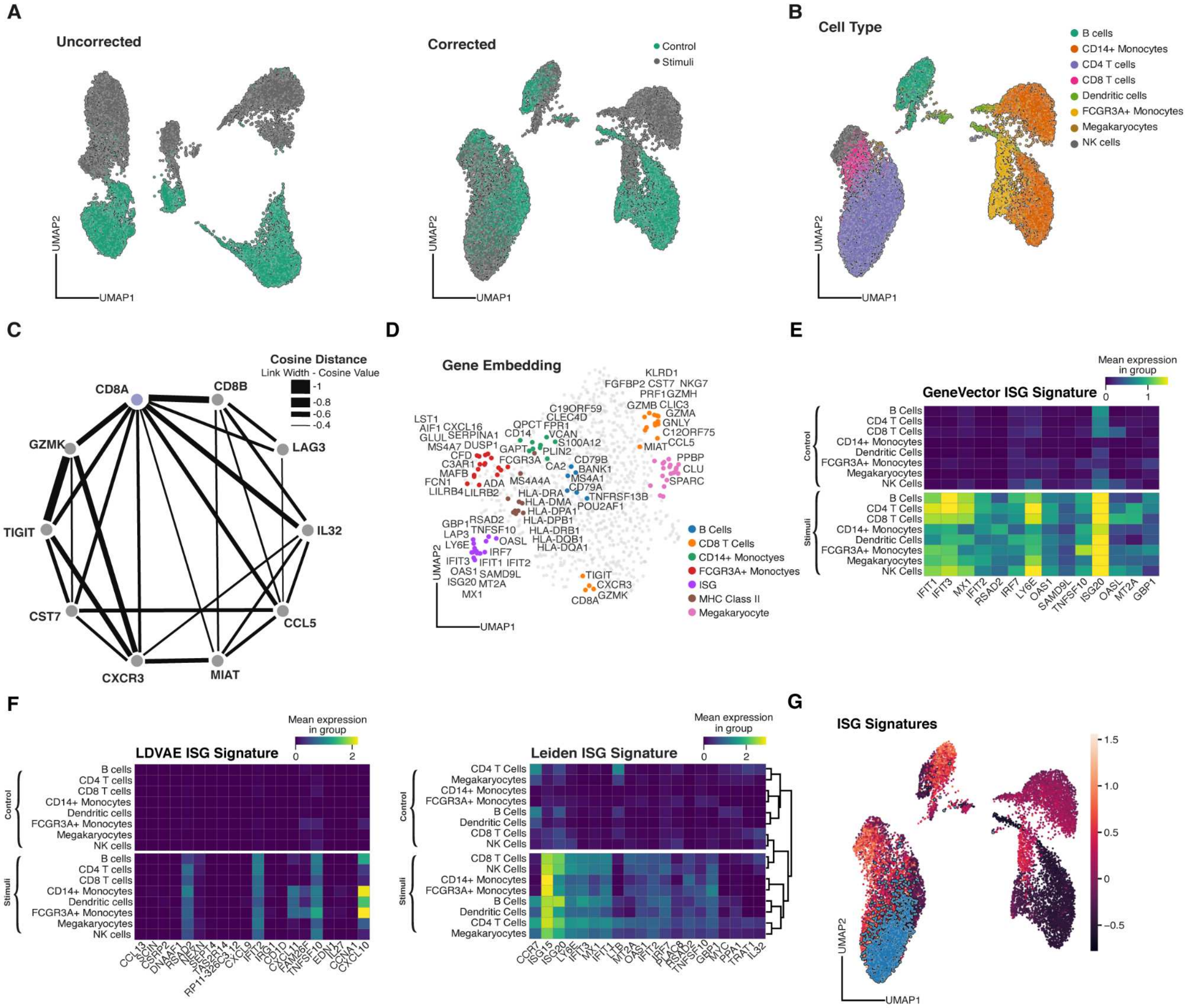

Figure 2: Comparing Methods with Interferon Beta Stimulated 10K PBMCs

A) Uncorrected (left) and batch corrected GeneVector UMAPs (right) on 10k PBMCs with control and interferon beta stimulated cells (stimuli). B) GeneVector UMAP showing original cell type annotations. C) Similarity graph of genes related to CD8A weighted by cosine distance. D) UMAP of gene embedding with gene annotations and labels for metagenes defining CD8 T cells, B Cells, CD14+/FCGR3A+ monocytes, Megakaryocytes, ISGs, and MHC Class II antigen presentation E) Gene expression over cell types and conditions for GeneVector ISG metagene. F) Gene *expression over cell types and batches for Leiden derived ISG signature and LDVAE ISG-associated signature. G) Leiden derived ISG signature cells (blue) as compared with scored expression from GeneVector ISG signature.*

Fast and accurate cell type classification using GeneVector

Comparative analysis of gene expression programs across large cohorts of patients can potentially identify transcriptional patterns in common cell types shared between many cancer types. However, classification of cell types using methods such as CellAssign (Zhang et al. 2019) are computationally expensive. Furthermore, the large number of covariates in these datasets makes disentangling patient-specific signals from disease and therapy difficult. GeneVector provides a fast and accurate method of cell type classification. Using GeneVectors, we perform cell type classification on a subset of 23,764 cells from the Tumor Immune Cell Atlas (TICA) composed of 181 patients and 18 cancer types (Nieto et al. 2021). The dataset was subset to 2,000 highly variable genes. Using the original cell type labels, cells were grouped into three main immune cell types: T cells, B cells, and Myeloid cells (**Figure 4A: TICA Coarse Cell Type**). We select three canonical markers for each cell type (T cells: CD3D, CD3G ,CD3E, B cells: CD79A, CD79B, MS4A1, and Myeloid cells: LYZ, CST3, AIF1) for cell type classification.

To generate a set of unbiased cell type labels as a target for classification, we computed a gene embedding based on expression from only the nine selected markers. For each cell, we then computed the pseudo-probabilities over cell type and assigned a ground truth label to each cell (**Figure 3A: Generated Ground Truth**). Using these labels, we assessed the performance of GeneVector in classifying cells based on predefined markers. We recomputed the gene embedding using the total set of genes and performed the cell type classification with the same nine markers. We found that 94.7% of both T cells and myeloid cells and 79.1% of B cells were correctly classified with respect to the generated ground truth labels. When comparing classifications to the original cell type labels, we observed slightly increased performance on T cells (99.0%) and myeloid (97.3%) with a decreased accuracy of B cells (72.1%). We generated confusion matrices to compare classification performance with both ground truth and coarse cell type and GeneVector classifications (**Figure 3B)**, They revealed that pseudo-probabilities within B cells were found to be more uniformly distributed, consistent with the misclassification, when compared to the high probabilities of T or myeloid cells (**Figure 3C**).

While several tools provide the ability to perform probabilistic cell type classification, these tools are computationally expensive and thus unable to rapidly update annotations with new marker genes. GeneVector, on the other hand, allows rapid testing of different marker genes and phenotypes in exploratory analysis settings. Increased performance in classification is important given the large variation of markers used to define the same phenotypes across different studies. After generating a gene embedding, cell type prediction can be computed interactively and in real-time. The assignment is computationally bounded by $O(n+m)$, where n is the number of cells and m is the number of gene markers. An additional advantage of having a probability is the ability to map markers from known pathways to a continuous value in each cell. In both phenotype and pathway, demonstration of continuous gradients across cells provides a measure of change and activation that cannot be seen from unsupervised clustering.

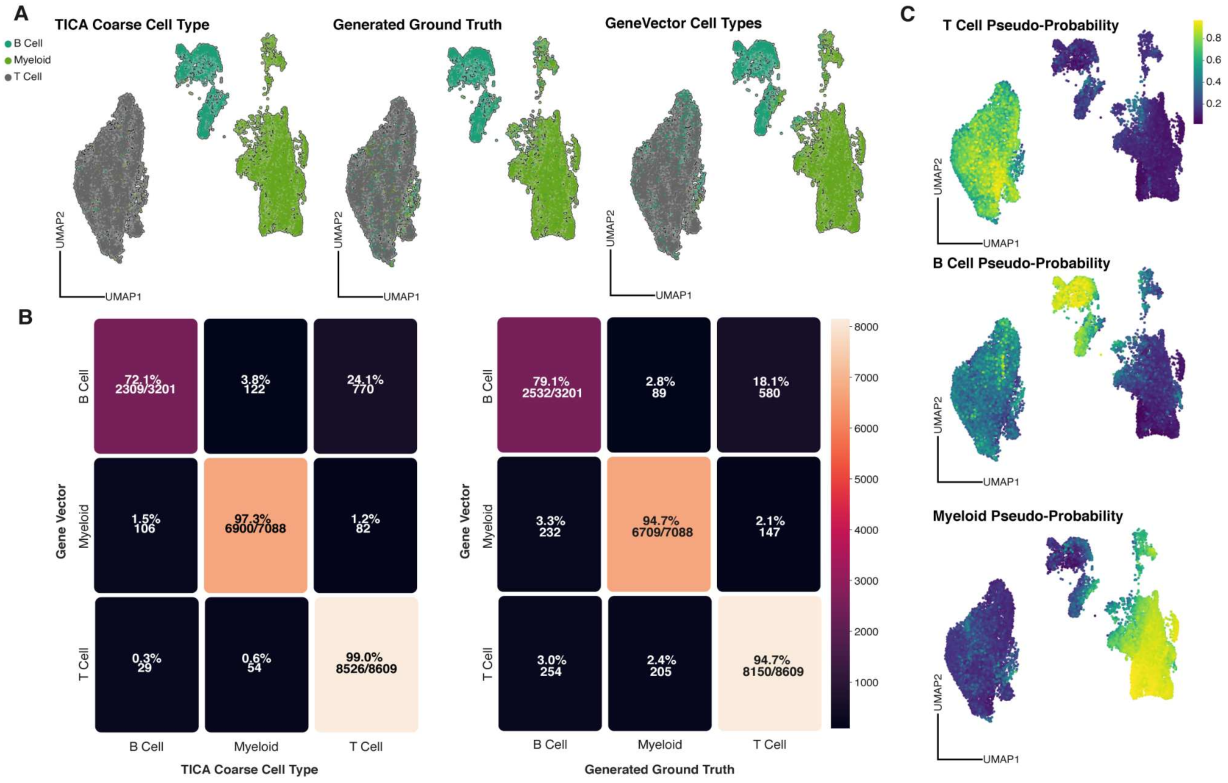

Figure 3: GeneVector Accurately Classifies Cells in TICA Cell Atlas

A) Cells annotated by coarse cell type summarized from the original TICA provided annotations, ground truth labels generated from a subset of nine markers by GeneVector, and cell type labels computed by nine markers using GeneVector. B) Confusion matrices comparing GeneVector classification accuracy with TICA coarse cell type labels and generated ground truth labels. C) Pseudo-probability values for each coarse cell type mapped to each cell in the UMAP.

Transcriptional Changes between Primary and Metastatic Site in HGSOC

Studies with scRNA-seq data sampled from multiple tumor sites in the same patient provide a wide picture of cancer progression and spread in diseases such as high grade serous ovarian cancer (HGSOC). As these datasets grow larger and more complete, understanding the transcriptional changes that occur from primary to metastatic sites can help identify mechanisms that aid in the process of the invasion-metastasis cascade. GeneVector provides a framework for asking such questions in the form of latent space arithmetic. By defining the difference between two sites as a vector, where the direction defines transcriptional change, we identify metagenes associated with expression loss and gain between primary and metastasis sites from an HGSOC patient (022) in the Memorial Sloan Kettering Cancer Center SPECTRUM cohort (Vázquez-García et al. 2021).

A set of 17,329 previously annotated and quality control filtered cancer cells from left or right adnexa and bowel were processed with GeneVector (**Figure 4A**). The dataset was subset to 2,000 highly variable genes. Feature vectors were defined for adnexa (v_{adnexa}) as the primary site and bowel (v_{bowel}) as the site of metastasis. The vector representing expression gain in metastasis was defined as $v_{gain} = v_{adnexa} - v_{bowel}$ and the inverse vector, $v_{loss} = v_{bowel} - v_{adnexa}$, was representative of a loss in expression. A total of 368 identified metagenes were sorted by cosine similarity to either vector. Gene Set Enrichment Analysis (GSEA) using GSEAPY with Hallmark gene set annotations from Enrichr (Kuleshov et al. 2016) was performed on the top five most similar metagenes to the vector v_{gain}, .The most representative metagene for transcriptional change from primary to metastatic sites was Epithelial-to-Mesenchymal Transition (EMT) (**Figure 4B**). Conversely, the set of metagenes representative of loss from adnexa to bowel included MHC Class I genes (HLA-A, HLA-B, HLA-C, HLA-E, and HLA-F) and the transcriptional regulator B2M, suggesting a means of immune escape via loss of MHC Class I expression. For both the EMT and MHCI metagenes, pseudo-probabilities computed using GeneVector highlight pathway activity localized to either site in the UMAP (**Figure 4A**). The ability to phrase questions about transcriptional change as vector arithmetic provides a powerful platform for more complex queries than can be performed with differential expression analysis.

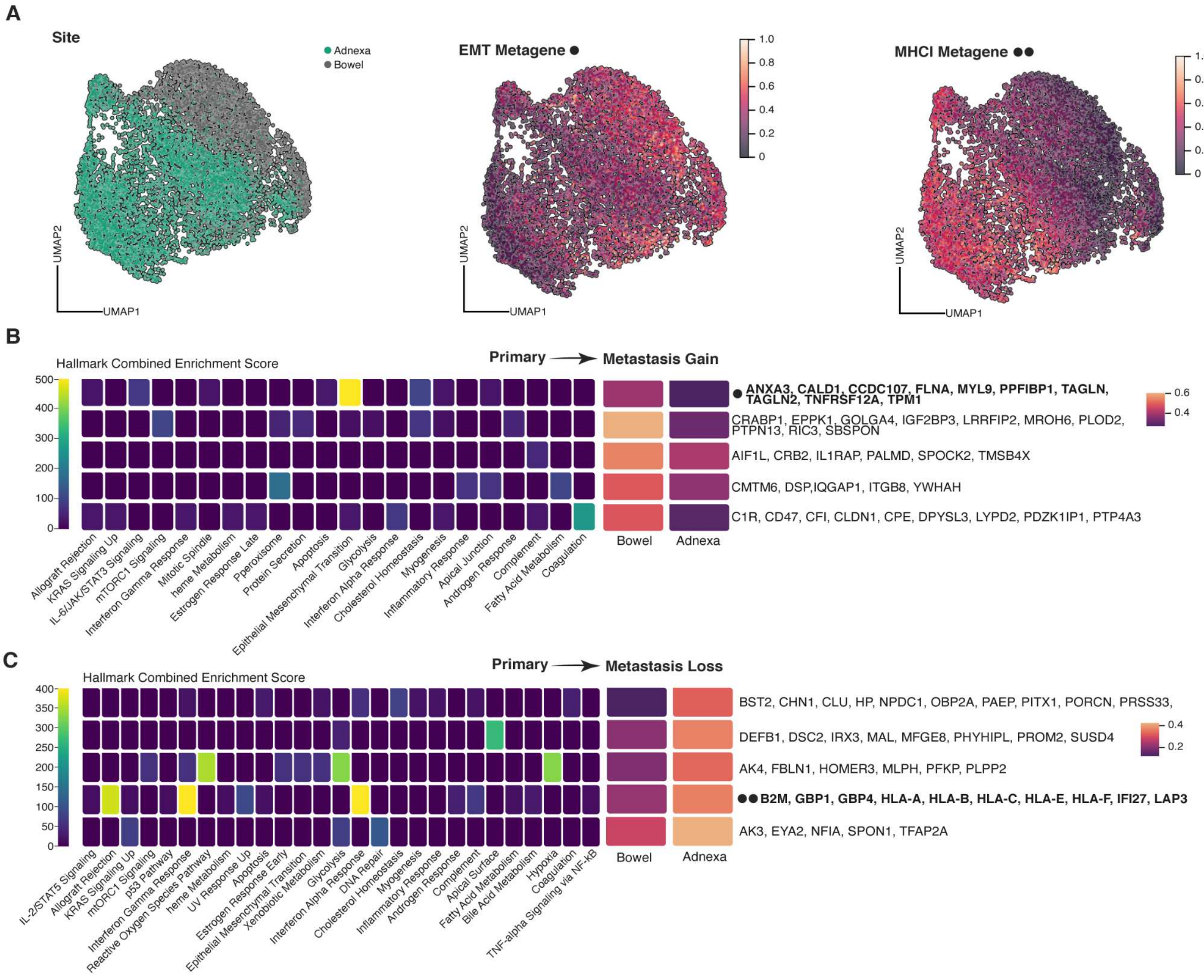

Figure 4: Metagenes associated with directional difference in HGSOC cancer cells from adnexa to bowel. *A) GeneVector UMAP of HGSOC cells with site labels and pseudo-probabilities for metagenes associated with up-regulation in bowel to adnexa (Epithelial-to-Mesenchymal Transition) and down-regulation (MHC Class I). B) Top five metagenes most similar by cosine distance to vector describing expression gain from adnexa to ball. Hallmark combined enrichment scores for each metagene highlights the most descriptive metagene defines Epithelial-to-Mesenchymal Transition. C) Top five metagenes by cosine distance to vector describing expression loss from adnexa to bowel includes loss of MHC Class I gene expression.*

PDX Samples in Presence and Absence of Cisplatin Treatment

Understanding the transcriptional processes that generate resistance to chemotherapies in cancer cells has immense clinical value. However, the transcriptional organization of resistance is complex with many parallel mechanisms contributing to cancer cell survival (Shen et al. 2012). GeneVector provides a framework for enumerating functional units, as metagenes, that change with treatment in a given set of cells. We analyzed single cell RNA-seq collected from triple negative breast cancer patient-derived xenograft model (SA609 PDX) along a treated and untreated time series (Salehi et al. 2021). Over a total of 19,799 cancer cells with treatment labels (**Figure 5A**), we generated metagenes to identify programs potentially related to cisplatin resistance.

We identified a total of 85 metagenes and generated gene module scores for each metagene together with separate linear regression coefficients over the four time points. Positive coefficients identify gene programs that increase over the four timepoints, and conversely, negative coefficients highlight decreasing expression. We classified gene programs by coefficient into sets that increase in both treated and untreated, decrease in both, increase in treated only, and increase in untreated only (**Figure 5B**). We identified the five metagenes that have the highest coefficient in treated time points, while simultaneously decreasing in untreated samples.

Our pathway enrichment on each of the five metagenes using GSEAPY with Hallmark gene set annotations from Enrichr (Kuleshov et al. 2016) showed the metagene 141 to be enriched for IL-6/JAK/STAT3 signaling (**Figure 5C**), a pathway frequently up-regulated in chemoresistant cancers known to benefit from STAT3 inhibition (Sun et al. 2019). This metagene includes genes ID1, ID2, ID3, ID4, and STAT3, all of which have been associated in chemo-resistant samples of several cancers (Zhang et al. 2006; Yamano et al. 2010; Roberts et al. 2005). As further validation of increased expression changes, the individual normalized expression profiles for each gene are shown to increase only over treated timepoints (**Figure 5D**).

GeneVector also identified global expression differences between treated and untreated cells from the set of most similar genes to vectors defined from treated and untreated cells. From these gene sets (**Figure 5E**), we concluded that genes associated with treated cells were also potentially related to cisplatin resistance. One example is FOXP1, the most predictive gene for treated cells. Several studies have implicated multiple resistance mechanisms involving FOXP1 including transcriptional regulation, immune response, and MAPK signaling (Zhu et al. 2015; Choi et al. 2016; Hu et al. 2015). Another example is EPCAM, whose high expression has been associated with increased viability of cancer cells in diverse cancer types (Sun et al. 2018; Imrich, Hachmeister, and Gires 2012). Additionally, EPCAM has been shown to have a role in resistance to chemotherapy in both breast and ovarian cancers through WNT signaling and Epithelial-Mesenchymal Transition (EMT) (Tayama et al. 2017; Latifi et al. 2011). Among these genes, we also identified several of the genes comprising metagene 141 including STAT3 and ID family genes.

In summary, GeneVector metagenes can be used to understand transcriptional dynamics at the single cell level in time series data. By identifying metagenes that change expression in time, GeneVector delivers candidate transcriptional programs in an unsupervised fashion that provide a foundation for further investigation into drug resistance mechanisms and many potential time-dependent studies.

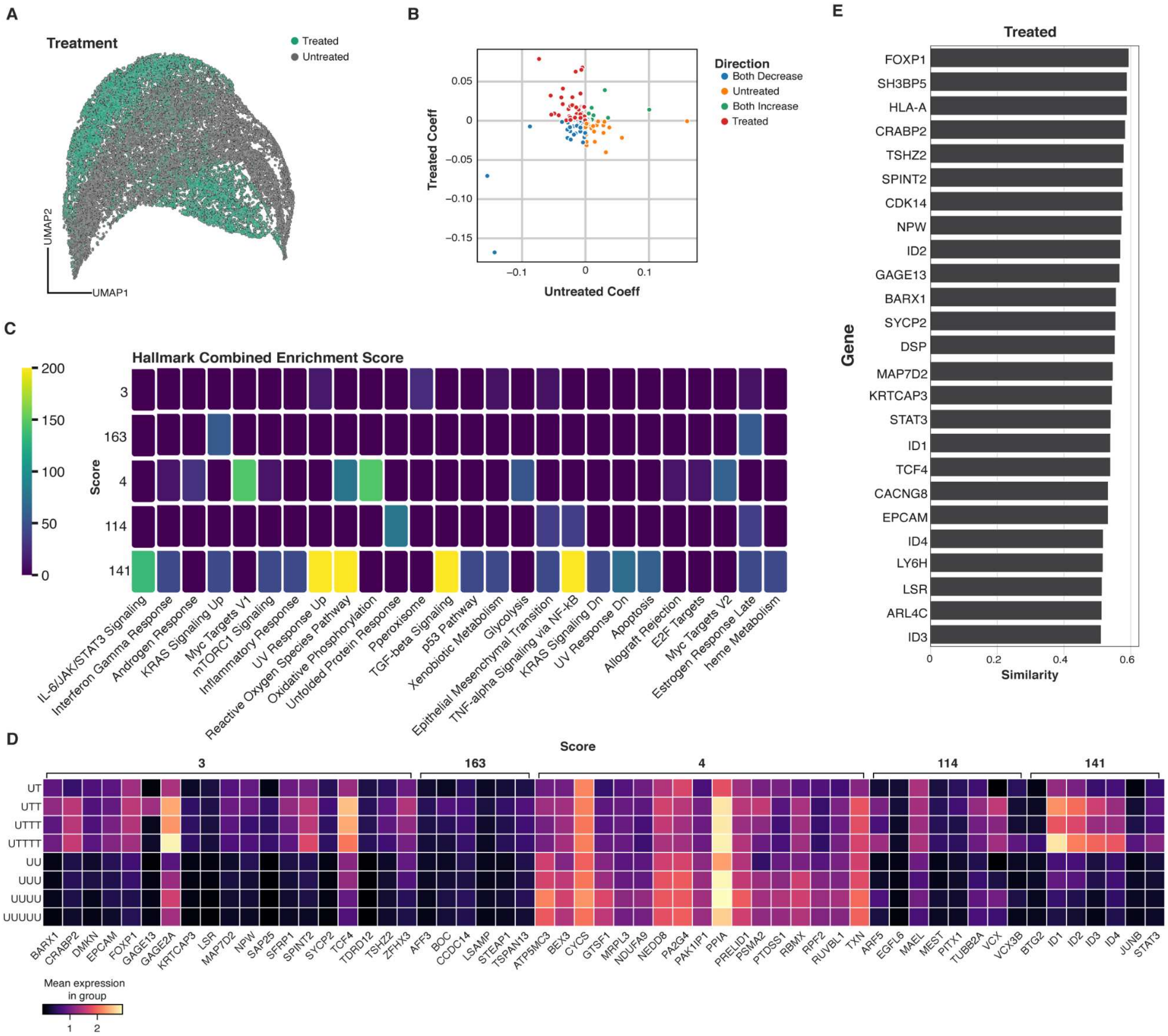

Figure 5: Analysis of Metagenes in Cisplatin-treated PDX Time Series

A) UMAP of cisplatin-treated PDX cells annotated by time point and treatment. B) Metagenes plotted with respect to regression coefficients over four timepoints in either treated or untreated cells with p-values < 0.001. C) Hallmark combined enrichment scores for metagenes increasing only in treated cells. D) Gene expression profiles for each timepoint for the five metagenes associated with increase in treatment. Metagene 141 is associated with IL-6/JAK/STAT3 includes STAT3 and ID family genes with known cisplatin resistance function. E) Most predictive genes for treated cells without respect to time points.

Discussion

GeneVector is a vector space model built on mutual information that allows a multitude of analyses from a single embedding. By borrowing expression signal across genes, GeneVector overcomes sparsity issues to generate a dense representation of each gene. Using vector arithmetic, we generate gene co-expression graphs that can be used to identify transcriptional programs, or metagenes, in an unsupervised fashion. These metagenes can be related to a cell embedding to identify transcriptional changes related to conditional labels or time points. We show that gene vectors can additionally be used to annotate cells with a pseudo-probability, and that these labels are accurate with respect to previously defined cell types and generated ground truth labels based on a set of known marker genes.

In interferon-stimulated PBMCs, we identify a cell type independent ISG metagene that summarizes interferon-stimulation across cell types that cannot be recovered with LDVAE. Additionally, we show that differential expression analysis of Leiden clusters associates a similar ISG signature with a single cell type (CD4 T cells). Failing to identify all cell types that respond to a stimulus with changes in gene programs could obscure a true biological process and conceal potential targets for intervention. We demonstrate accurate cell type assignment across 18 different cancers in over 100 patients described in the TICA cell atlas. Since GeneVector is computationally nimble it can perform this analysis in real time and with different markers to assess the change in assignment as a function of cell type label alterations. This is of particular interest in large cohorts and for rare cell types, where the cell type labels may still need to be optimized. Current methods are too computationally expensive for such a real-time task. In high grade serous ovarian cancer, we identify metagenes that describe transcriptional changes from primary to metastatic sites. Our results implicate the loss of MHC class I gene expression as a potential immune escape mechanism in cancer spread. In cisplatin-treated TNBC PDXs, GeneVector uncovers transcriptional signatures that are active in drug resistance, most notably metagenes enriched in JAK/STAT3 signaling. This signaling pathway is a cornerstone in cancer progression since it modulates immune surveillance; it is the target of various therapies, but success has been mixed (Owen et al. 2019) making it all the more important to employ tools that identify the multitude of players contributing to therapy response.

Beyond what has been demonstrated, there are other potential uses for the GeneVector framework. The integration of single cell TCR sequencing data may allow the generation of transcriptional programs related to individual clonotypes. This provides an alternative method for quantifying TCR functional similarity, beyond motif or sequence distance-based metrics, that leverages transcriptional signature. As CITE-seq, the simultaneous measurement of protein and transcriptome, is becoming more commonplace, GeneVector will be able to identify relationships between protein and gene expression and thereby extend the use of a single protein as a proxy measurement for several genes. Finally, with the increased interest in spatial transcriptomics, new methods will emerge to incorporate distance into the study of gene interactions. We propose that the GeneVector model can be easily extended to include spatial distance, instead of the number of co-expressed cells, as measure of interactivity.

While GeneVector is the first of its kind, we can also envision possible improvements to GeneVector. The construction of a 2D histogram of co-occurrence provides an empirical measure of the joint probability distribution between any two genes. However, mutual information can be sensitive to sparse data. It is likely that our method would benefit from smoothing over read count data. By fitting a negative binomial distribution to each gene, it may be possible to increase the reliability of mutual information and speed computation through an analytical calculation of the mutual information between genes. Currently, the construction of a 2D histogram of count co-occurrences between genes is the most expensive computational step.

While identification of metagenes from the co-expression similarity graph using Leiden clustering yields consistent transcriptional programs across datasets, improvements can be made to this methodology. The selection of the best resolution for clustering remains untested and is left to the user definition. The desired size of gene programs may differ with application. The choice of Leiden clustering may also be replaced by other types of network clustering methods. One such possible solution may be the use of cliques. In graph theory, the maximal clique is a complete, or fully connected, subgraph that cannot be found in any other clique (Chang, Kloks, and Lee 2001). In bioinformatics applications, cliques have been previously used to cluster gene expression microarray data (Chang, Kloks, and Lee 2001; Tanay, Sharan, and Shamir 2002). However, this method allows association of single genes to many metagenes, resulting in redundancy, which may or may not be desirable.

Finally, our model requires a distance metric between genes that is symmetric and positive. While we present results using mutual information scaled by co-expression, other symmetric distance metrics may be substituted for mutual information, such as the Pearson correlation coefficient or spatially derived physical distance. Additionally, many transcription factors are repressive in function, and these associations are missed by only examining positive distances. However, gene programs that include both activation and repression associations would be difficult to interpret without annotation of directionality in the metagene definitions.

Overall, we see GeneVector as a first step in a direction that makes use of information theoretic measures and vector space models that leverage gene-specific interactions to overcome sparsity in scRNA-seq data. GeneVector allows high quality end to end analysis of single cell RNA-seq data from a single embedding. The framework is flexible and intuitive, providing a platform to ask powerful questions in the context of diverse datasets. GeneVector is extendable and future applications include multimodal analysis using TCR-seq, CITE-seq, and spatial transcriptomics. The software library is available on Github https://github.com/nceglia/genevector.

Online Methods

Gene Expression Mutual Information

In NLP applications, vector space models are trained by defining an association between words that appear in the same context. In single cell RNA sequencing data, we can redefine this textual context as co-expression within a given cell and mutual information across cells. The simplest metric to define association is the overall number of co-expression events between genes. However, the expression profiles over cells may differ due to both technical and biological factors. To summarize the variability in this relationship, we generate a joint probability distribution on the co-occurrence of binned read counts. The ranges of each bin are defined separately for each gene based on a user defined number of quantiles. By defining the bin ranges separately, the lowest counts in one gene can be compared directly to the lowest counts in another gene without need for further normalization. Using the joint probability distribution, we compute the mutual information between genes defined in Equation 1. The product of this measure with the number of co-expression events is subsequently used as the target value in training the model, allowing us to highlight the relationship between genes independent of normalization methods as a simple, single-valued quantity. Returning to the analogy of words, the product of the mutual information and the probability of co-expression between any two genes provides a measurement of the contextual similarity within single cells.

$$I\left(G_{i},\ G_{j}\right) = \sum_{i}^{n}\sum_{j}^{n} p\left(G_{i},\ G_{j}\right) log\left(\frac{p\left(G_{i}, G_{j}\right)}{p\left(G_{i}\right), p\left(G_{j}\right)}\right)$$

Equation 1: Mutual information between G_i and G_j computed on the empirical joint probability distribution over all possible integer count values *n* for G_i and G_j.

Model Training

A neural network is constructed from a single hidden layer corresponding to the size of the latent space vectors. A set of independently updated weights connects the one-hot encoded input and output layers that are defined from each pair of genes. These weights, *w1* and *w2*, are matrices with dimensions equal to *N* expressed genes by *I* hidden units. Initial values for *w1* and *w2* are generated uniformly on the interval -1 to 1. The objective function, as a minimization of least squares, is defined in Equation 2. The final latent space, defined as the *gene embedding*, can be taken as the vector mean of weights w_i and *wj*. For NLP applications, this is a preferred approach over selecting either weight matrix (Pennington, Socher, and Manning 2014). Each co-expressed gene pair is used as a single training example. The maximum number of examples for a full training epoch is given by the total number of co-expressed gene pairs. Weights are batch updated with adaptive gradient descent (AdaGrad) (Duchi et al. 2011). Training is halted at either a maximum number of epochs, or when the change in loss falls below a specified threshold. The model is implemented in PyTorch using standard library functions

$$J = \left(w_i^T \widehat{w}_j - I(G_i, G_j)\, X_{ij} \right)^2$$

Equation 2: Objective function for weights corresponding to gene G_i and G_j, where $I(G_i, G_j)$ is the mutual information and X_{ij} is the number of co-expression events between both genes over the total number of cells.

Gene Vectors

The cosine distance function, defined in Equation 3, is used to measure similarity between vectors. Values closer to 1 indicate strong association within the dataset. A *gene vector* is defined as the learned weights in the gene embedding for a particular gene. *Feature vectors* describing higher order variables are generated by computing a weighted average vector, described in Equation 4, from a set of individual gene vectors.

$$cosine\ similarity = 1 - \frac{A*B}{||A||*||B||}$$

Equation 3: Cosine similarity as the dot-product between feature vectors A and B.

To assign a vector to each cell, the average vector is computed across the gene embedding weighted by counts observed in each cell. The matrix of all cell vectors in the dataset is defined as the *cell embedding*. This cell embedding can be used in place of PCA or embeddings obtained with variational auto-encoders. Each cell vector maintains a linear relationship with the gene embedding. The feature vector is described in Equation 4.

$$C_k = \frac{\sum_{i=1}^{n} w_{i*} \bar{x}_i}{\sum_{i=1}^{n} w_i}$$

Equation 4: Feature vector C_k defined as the kth component of the cell embedding computed from the average mean of vectors $\bar{x}_{i \to n}$ where n is equal to the number of hidden units.

Co-expression Similarity Graph

A co-expression similarity graph is constructed from cosine similarity between each pair of genes. A node in the graph represents a single gene and edges are weighted by cosine similarity. Metagenes are identified through hierarchical agglomerative clustering in Scikit-learn (Garreta and Moncecchi 2013) using complete linkage with a cosine distance. To identify the optimal threshold distance, the silhouette coefficient is computed over cosine similarity values ranging from 0.0 to 1.0 at a 0.01 resolution (Rousseeuw 1987). The cosine similarity corresponding to the maximum silhouette coefficient is selected as input for generating the set of metagene clusters.

Phenotype Assignment

A set of phenotypes with associated marker genes are used to perform a pseudo-probabilistic phenotype assignment for each cell. A representative feature vector is computed for each marker gene set. Cosine similarity values for the given phenotypes are passed through a softmax function, given in Equation 6, to provide a pseudo-probability distribution for each phenotype. The argument maximum of this distribution is used to classify the most likely phenotype for a given cell.

$$\sigma(z)_i = \frac{e^{Z_i}}{\sum_{j=1}^{n} e^{Z_i}}$$

Equation 6: Softmax function where z is the set of cosine similarities for n phenotypes.

Generation of Gene Programs

The method of phenotype assignment can be reversed to produce a set of genes that are most similar to any given grouping of cells. Without the need for the softmax function, a feature vector can be constructed from a set of cells corresponding to a given label. Gene vectors can be sorted by cosine similarity to produce a ranked list of candidate genes that are most similar to the set of cells.

Time-series Analysis of Gene Programs

After generating gene module scores over a series of time points, linear regression yields a p-value and slope coefficient from each series. Gene programs with a p-value < 0.05 are sorted by the direction of change indicated from the linear coefficient.

References

Butler, Andrew, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. 2018. “Integrating Single-Cell Transcriptomic Data across Different Conditions, Technologies, and Species.” *Nature Biotechnology* 36 (5): 411–20.

Chang, Maw-Shang, Ton Kloks, and Chuan-Min Lee. 2001. “Maximum Clique Transversals.” *Graph-Theoretic Concepts in Computer Science*. https://doi.org/10.1007/3-540-45477-2_5.

Du, Jingcheng, Peilin Jia, Yulin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. 2019. “Gene2vec: Distributed Representation of Genes Based on Co-Expression.” *BMC Genomics* 20 (Suppl 1): 82.

Imrich, Sannia, Matthias Hachmeister, and Olivier Gires. 2012. “EpCAM and Its Potential Role in Tumor-Initiating Cells.” *Cell Adhesion & Migration*. https://doi.org/10.4161/cam.18953.

Kang, Hyun Min, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, et al. 2018. “Multiplexed Droplet Single-Cell RNA-Sequencing Using Natural Genetic Variation.” *Nature Biotechnology* 36 (1): 89–94.

Lopez, Romain, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. 2018. “Deep Generative Modeling for Single-Cell Transcriptomics.” *Nature Methods*. https://doi.org/10.1038/s41592-018-0229-2.

McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. “UMAP: Uniform Manifold Approximation and Projection.” *Journal of Open Source Software*. https://doi.org/10.21105/joss.00861.

Nieto, Paula, Marc Elosua-Bayes, Juan L. Trincado, Domenica Marchese, Ramon Massoni-Badosa, Maria Salvany, Ana Henriques, et al. 2021. “A Single-Cell Tumor Immune Atlas for Precision Oncology.” *Genome Research* 31 (10): 1913–26.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. “Glove: Global Vectors for Word Representation.” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://doi.org/10.3115/v1/d14-1162.

Pezzotti, Nicola, Boudewijn P. F. Lelieveldt, Laurens van der Maaten, Thomas Hollt, Elmar Eisemann, and Anna Vilanova. 2017. “Approximated and User Steerable tSNE for Progressive Visual Analytics.” *IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/tvcg.2016.2570755.

Salehi, Sohrab, Farhia Kabeer, Nicholas Ceglia, Mirela Andronescu, Marc J. Williams, Kieran R. Campbell, Tehmina Masud, et al. 2021. “Clonal Fitness Inferred from Time-Series Modelling of Single-Cell Cancer Genomes.” *Nature* 595 (7868): 585–90.

Stuart, J. M. 2003. “A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules.”

Science. https://doi.org/10.1126/science.1087447.

Stuart, Tim, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck 3rd, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. 2019. "Comprehensive Integration of Single-Cell Data." *Cell* 177 (7): 1888–1902.e21.

Sun, Xuan, Robert C. G. Martin, Qianqian Zheng, Russell Farmer, Harshul Pandit, Xuanyi Li, Kevin Jacob, Jian Suo, and Yan Li. 2018. "Drug-Induced Expression of EpCAM Contributes to Therapy Resistance in Esophageal Adenocarcinoma." *Cellular Oncology* 41 (6): 651–62.

Svensson, Valentine, Adam Gayoso, Nir Yosef, and Lior Pachter. 2020. "Interpretable Factor Models of Single-Cell RNA-Seq via Variational Autoencoders." *Bioinformatics* 36 (11): 3418–21.

Tanay, Amos, Roded Sharan, and Ron Shamir. 2002. "Discovering Statistically Significant Biclusters in Gene Expression Data." *Bioinformatics* 18 Suppl 1: S136–44.

Traag, V. A., L. Waltman, and N. J. van Eck. 2019. "From Louvain to Leiden: Guaranteeing Well-Connected Communities." *Scientific Reports* 9 (1): 5233.

Vázquez-García, Ignacio, Florian Uhlitz, Nicholas Ceglia, Jamie L. P. Lim, Michelle Wu, Neeman Mohibullah, Arvin Eric B. Ruiz, et al. 2021. "Immune and Malignant Cell Phenotypes of Ovarian Cancer Are Determined by Distinct Mutational Processes." *bioRxiv*. https://doi.org/10.1101/2021.08.24.454519.

Wolf, F. Alexander, Philipp Angerer, and Fabian J. Theis. 2018. "SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis." *Genome Biology* 19 (1): 15.

Zhang, Allen W., Ciara O'Flanagan, Elizabeth A. Chavez, Jamie L. P. Lim, Nicholas Ceglia, Andrew McPherson, Matt Wiens, et al. 2019. "Probabilistic Cell-Type Assignment of Single-Cell RNA-Seq for Tumor Microenvironment Profiling." *Nature Methods* 16 (10): 1007–15.