A unified analysis of atlas single cell data

Hao Chen [1], Nam D. Nguyen [1], Matthew Ruffalo [1], and Ziv Bar-Joseph [*1,2]

Abstract

Recent efforts to generate atlas-scale single cell data provide opportunities for joint analysis across tissues and across modalities. Most of the existing methods for single cell atlas analysis use cells as the reference unit to combine datasets. However, such methods suffer from the limitations of inability to effectively integrate cross-modality data, hindering downstream gene-based analysis, and loss of genuine biological variations. Here we present a new data integration method, GIANT, which is for the first time designed for the atlas-scale analysis from the gene perspective. GIANT first converts datasets from different modalities into gene graphs, and then recursively embeds genes in the graphs into a latent space without additional alignment. Applying GIANT to the HuBMAP datasets creates a unified gene embedding space across multiple human tissues and data modalities, where gene representations reflect the functions of genes in their cells. Further evaluations demonstrate the usefulness of GIANT in discovering diverse gene functions, and underlying gene regulations in cells of different tissues.

[1]Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

[2]Machine Learning Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

[*]To whom correspondence should be addressed. Email: `zivbj@cs.cmu.edu`

Introduction

Several recent efforts have attempted to generate atlas-scale single cell data. Examples include The Human BioMolecular Atlas Program (HuBMAP) [1], the Human Cell Atlas (HCA) [2], the Cellular Senescence Network (SenNet) [3] and many more. In all of these efforts, researchers aim to collect and analyze data from multiple tissues, modalities, conditions and more. While the initial phases of the programs often focused on the coverage of different tissues, and on the separate, modality-specific analysis of each dataset, we are now at a point where, for the first time, we can perform joint cross-tissue and cross-modality analysis, on single cell data at the atlas scale.

The joint analysis is challenging for many reasons. First, one needs to decide on a common reference unit for the analysis. While most of the prior methods for atlas analysis focused on cells [4–7], such analysis makes it very hard to combine data across modalities (for example, in many cases chromatin accessibility is not a good predictor of gene expression levels [8]). In addition, cell based analysis makes it more difficult to identify shared pathways between different cell types [9]. Another issue is the type of analysis to be performed. Cell clustering based on gene expression similarity, which has been commonly used [10–12], may not be appropriate for spatial technologies which provide information on cell relationships in space [13–15]. Finally, different signal-to-noise levels across datasets [9], and the large volume of data in atlas studies create considerable challenges for joint analysis.

To date, there have been only a few attempts at a large-scale joint analysis of data from different modalities (not just in biology). The main approach used for such integration utilizes alignment-based multi-view learning [6, 16–18]. The goal is to transform multi-view (multi-dataset) inputs, which form multimodal distribution, into a homogeneous unimodal distribution for downstream learning tasks. The key idea of these methods is to project the manifolds of multiple datasets into a common space such that both, the locality of each dataset is preserved and the distance between the same entities in different datasets (for example, the same individuals in different social networks) are minimized. While this may work for some biological datasets, we often observe over-alignment problems with this approach [9].

To improve on these issues and to enable large-scale joint analyses of atlas-level data, we

developed a new method GIANT (Gene-based data Integration and ANalysis Technique), that focuses on genes rather than on cells. Genes provide a level of granularity which on the one hand can serve as anchor points common to all modalities, while on the other hand makes it easy to perform higher-level analysis. We first convert cell clusters from each dataset and each modality into gene graphs based on the specific characteristics of that modality. Genes are more basic building blocks for joint analysis than cells or pathways and, as we show, such graphs can be extracted from different high-throughput methods that are being profiled in large atlas studies. To integrate these graphs across modalities and tissues, we project genes from all the graphs into a latent space (embedding) using recursive projections. In recursive projection [19, 20], a dendrogram is used to enforce similarity constraints across graphs while still allowing genes with multiple functions to be projected to different locations in the embedding space. In other words, the recursive method can be seen as a hierarchical alignment along embedding, in which graphs nearby in the tree structure are embedded closely in the latent space.

We applied the method to the HuBMAP data from 10 different tissues, 3 different major modalities, and 9 different single cell platforms. As we show, despite the considerable heterogeneity, our joint processing and analysis method was able to successfully group genes based on their function and pathways, associate specific pathways with tissues, discover new functions for a large number of genes, and identify tissue-specific regulations of genes based on their cross-modality clustering.

Results

We developed GIANT to construct a unified embedding for genes from 10 different tissues and 3 different data modalities. Figure 1 presents an overview of GIANT. We first construct gene graphs for cell clusters from each tissue and data modality based on expression or epigenetic correlations (Methods). Next, all gene graphs are connected in a cell cluster dendrogram. A recursive graph embedding method which takes into account both the dendrogram and gene identity connections is applied. The final result is an embedding (a location in a latent space) for each gene in each of the graphs. See "Methods" section for details.

GIANT integrates multi-modality and multi-tissue data

The gene embedding space produced by GIANT enables joint functional analyses across data modalities, tissues, and cell clusters. Therefore, the location of a gene in the space is expected to reflect the function the gene performs (which could be the same across several different cell types and tissues), instead of the specific data source which generates the value for that gene. For example, genes from different tissues can be closely embedded in space as long as they perform similar functions in these tissues. To investigate whether different data sources are well integrated, we visualized the entire embedding space and highlighted the locations of genes based on the sources of data that they are from. Figure 2 presents the distribution of genes in the map based on their modalities (Fig. 2A), tissues (Fig. 2B), and cell clusters (Fig. 2C) respectively. As can be seen, the locations of genes are not determined by their data modalities or tissues, allowing us to use the embeddings to investigate shared functionalities across tissues and mechanisms being studied by each of the data modalities we used.

To further investigate which data sources tend to co-cluster, we clustered the entire embedding space (Fig. 2D), and then looked at the data sources assigned to each of the resulting clusters (embedding components). Specifically, we applied the Leiden algorithm [21] to cluster all genes in the embedded space into 172 components (Methods). We then determined the data modality or tissue for each gene in each component. Figure 2E shows the mapping from data modalities to components. Genes from Slide-seq expression and Slide-seq spatial graphs are always assigned to the same components, as these represent different measurements for the same gene. For the other modalities, we found that 14.5% (25 out of 172) of the components contain genes from at least two of the three modalities, scRNA-seq, scATAC-seq, and Slide-seq. Figure 2F shows the mapping from tissues to components. Interestingly, genes from Heart (blue) and Lung (orange), both parts of the circulatory system [22], frequently appear in the same components. Tissues from the immune system, including Lymph Node (purple), Thymus (pink), and Spleen (brown) [23], are also frequently co-clustering. These results combined show that GIANT successfully integrated data from different modalities, tissues, and cell clusters, and is able to group genes based on common functions rather than data sources.

Genes closely embedded in the space show tissue-specific GO enrichment

We next tested the functional enrichment of nearby genes in the embedding space (Methods). Due to the large number of tissues we analyzed, such analysis allowed us to identify a large number of different functional groups spanning most of the Gene Ontology categories. Specifically, this analysis identified 272,848 enriched Biological Process (BP) GO terms, 44,952 enriched Cellular Component (CC) GO terms, and 56,199 enriched Molecular Function (MF) GO terms (adjusted $P \leq 0.01$) in the 1,199 components (Supplementary Table 1).

We next analyzed GO terms discovered in embedding components dominated by different tissues. Specifically, for each component, we looked at the top three tissues for genes assigned to that component. Next, we selected the top 5% enriched BP GO terms with the largest fold changes and adjusted $P \leq 1\text{e} - 5$, and associated these GO terms with each of the top three tissues. To summarize the enriched GO categories for each tissue, we propagated these GO terms to their parent terms along the GO hierarchies. GO categories that are prominently enriched in each tissue are shown in Figure 3A, where the size of each dot indicates the ratio of enriched GO terms in the tissue (row), who are children terms of the noted GO category (column), while the color of each dot indicates the fold change of the ratio over the average value of ratios in the column.

We found that several relevant prominent GO categories for each tissue (dark red dots in Fig. 3A) are strongly consistent with the known cellular functions of that tissue. For example, muscle related GO categories (*e.g.*, GO:0003012 muscle system process, GO:0006936 muscle contraction) are prominent for Heart, Lung, and Large Intestine, which contain abundant cells of cardiac muscles, respiratory muscles, and smooth muscles [24–26]. For immune tissues (Lymph Node, Thymus, and Spleen), immunity related GO categories dominated the assignments, including GO: 0002376 immune system process, and GO:0006955 immune response. Lipid metabolism related GO categories (*e.g.*, GO:0010876 lipid localization) are prominently discovered for Liver, and metabolic process related GO categories are significantly found for Pancreas, all consistent with the specific functions of these tissues [23, 27, 28].

Examples of some tissue-specific GO terms are shown in Figure 3B. Each of them is enriched in different components in the embedding space (reddish regions), while they are more likely

found in components where the related tissues reside in (dark grey regions).

Inferring gene functions from multi-modality multi-tissue embeddings

The precise function of a specific protein (and the genes that code for it) may differ between tissues and cell types [29]. Given the observations above that GIANT can group functionally similar genes together across data sources, we used the embeddings to infer novel gene functions for all the genes we embedded. We first clustered each gene's embeddings and then inferred the gene's functions in each cluster based on other co-clustered genes with known functions in the graphs of this cluster (Methods).

We tested the coverage and precision of the functions that we can assign using GO. We found that 47.4% of all known GO functions for the genes are identified in at least one of their clusters, which means that our predictions cover roughly half of all known functions for these genes (recall). This number is significant, and much higher than the expected number given the number of GO assignments (compared to 6.9% expected for random assignments). On the other hand, the ratio of known functions in our GO assignments for genes (precision) is 25.1% which is also significant compared to the ratio of known functions in the whole prediction space (1.5%) (Fig. 4A). Precision and recall of our predictions considering only children terms of other GO categories are shown in Figure 4A.

Given this large coverage and precision, we next investigated genes that are assigned very different functions in at least two different clusters, which are termed as "bimodal genes" here. In particular, genes are selected as bimodal that met the following criteria: among all the gene's clusters with at least 5 assigned GO terms, which are BP terms associated with no more than 200 genes to focus on specific GO terms that may show tissue specificity, at least two clusters do not have any GO overlap. If a gene has more than two such clusters, the two with the most GO terms assigned are considered in this analysis. As a result, we identified 1063 bimodal genes, and 487 (45.8%) among them have known functions assigned in at least one cluster. For 121 (11.4%) of them, both clusters have functions that are already known even though they can be very different (Fig. 4B).

Figure 4C presents examples of bimodal genes with different known functions in different clusters. MEF2C is essential in B-lymphopoiesis, as well as for B-cell survival and proliferation

in response to stimulation [30, 31]. On the other hand, it is known to be involved in vascular development and controls cardiac morphogenesis [32, 33]. Our method correctly assigns MEF2C the functions of B cell activation, lymphocyte activation, and regulation of immune response in the group of Lymph Node and Thymus cell clusters, while it assigns this gene the functions of positive regulation of developmental process, and tube morphogenesis in a group of Heart and Lung cell clusters. Similarly, RYR2 has been found to play an important role in triggering cardiac muscle contraction and to be related to cardiac arrhythmia [34], while it also mediates the flow of calcium ions across cell membranes in large intestine [35]. Again, RYR2 is assigned the functions of regulation of heart rate, muscle contraction, and regulation of blood circulation in the group containing a Heart cell cluster, while in the Large Intestine cell cluster, it is assigned ion transmembrane transport function. A complete list of predicted bimodal genes and their predicted functions can be found in Supplementary Table 2.

Gene embeddings enable the discovery of tissue-specific gene regulations

Identifying the regulatory programs of cells helps understand their diverse gene expression patterns. We further demonstrate the ability of our gene embeddings to discover gene regulations in cell clusters of different tissues. Similar to the previous subsection, we calculated the enrichment of genes regulated by the same transcription factor (*i.e.*, TF regulon) in each embedding component (Methods). Specifically, this analysis identified 5,815 enriched TF regulons (adjusted $P \leq 0.001$) in the 1,199 components. To compare the TFs identified in cells of different tissues, we associated discovered TFs with tissues using the top three tissues for each component as discussed above. TF overlaps among different subsets of tissues are shown in Figure 5A. Although a large number of TFs (104 out of 435, 23.9%) were identified as significant for all the tissues, most TFs show specificity for different tissues. These observations are consistent with previous findings in the literature [36, 37]. The full list of TFs identified in different tissues can be found in Supplementary Table 3.

To further validate our discoveries, we examine whether TFs found in each tissue are enriched with TFs known to be specific to this tissue. For this, we used a prior list of tissue marker TFs [38]. We find enrichment for tissue markers in all tissues and significant enrichment for three of the eight tissues (Figure 5B).

Discussion

Recent efforts to generate atlas-scale single cell data provide opportunities for joint analysis across tissues and modalities, which can enable new insights into cell functional plasticity, gene function, cell regulation, and disease. A key challenge for such analysis is how to effectively integrate data from different sources, such as tissues, modalities, and conditions, while enabling the discovery of system-level organizations and regulations.

We developed a new method which is the first to fully integrate atlas-level data from multiple tissues and modalities in a single unified embedding. For this, we use genes as the anchor points common to all datasets, modalities and tissues. Our method first converts cell clusters in different tissues from different data modalities to gene graphs and then recursively embeds these graphs such that connected genes within a single graph are encouraged to have similar embeddings, while genes from graphs generated from similar cells are projected closely in the embedding space. Finally, a unified embedding space is produced where the locations of genes reflect the functions they perform in the cells.

We applied our method to HuBMAP database with 10 different tissues, 3 different major modalities and 9 different single cell platforms. As we have shown, despite the considerable heterogeneity of the input data, our method was able to successfully group genes based on their function and pathways. We further used the method to assign new functions to previously characterized genes, and explore tissue similarities and regulations.

An interesting finding is the number of genes that have at least two sets of functions (termed "bimodal genes"). Even with only 10 tissues, we identified over 1000 genes with two or more diverse functions depending on the tissues. While several studies have pointed out genes with multiple functions [39] mostly in different cell types or tissues [30, 31], these prior findings were anecdotal and focused on very few genes. By analyzing much larger atlas data we were able to determine that such bimodal genes are actually quite common. It is likely that the number of genes with multiple functions is actually much larger given the large number of tissues that were not included in our study. These findings indicate that even though the majority of genes have at least one associated function in large annotation databases (for example, GO), there is still much work needed to fully characterize the functions of human genes.

Our work mainly focused on tissue-level analysis as the first step for atlas-level integration. While we used cell clustering (which likely represents cell types) as input, these clusters were not assigned to specific cell types and so our method can currently only provide information on tissue-level functions and regulations. Our method can be directly extended to cell type analysis as well and we expect that when such information is available, the method would provide an even better ability to infer cell type specific gene functions and regulations.

Our embedding method is implemented in Python and is available at https://github.com/chenhcs/GIANT. We hope that as the analysis of atlas-level data becomes more common, our method would be able to be applied to additional atlases and datasets to integrate large-scale single cell data.

Methods

Datasets

We integrated single cell data from more than 60 individuals profiling 10 different tissues and 3 major data modalities from the Human BioMolecular Atlas Program (HuBMAP) consortium [1].

Single cell RNA-seq (scRNA-seq) datasets: scRNA-seq datasets were obtained for 8 tissues: Heart, Kidney, Large Intestine, Liver, Lung, Lymph Node, Spleen, and Thymus. For Heart we obtained 14 datasets and a total of 30,241 cells, 15 datasets and 109,055 cells for Kidney, 6 datasets and 13,128 cells for Large Intestine, 5 datasets and 7,916 cells for Liver, 9 datasets and 68,390 cells for Lung, 4 datasets and 40,449 cells for Lymph Node, 10 datasets and 93,737 cells for Spleen, 3 datasets and 10,838 cells for Thymus. This data was obtained from 5 platforms (Supplementary Table 4) and uniformly processed. The uniform processing used the Salmon's Alevin method [40] for sequencing reads mapping and abundance quantification, leading to a cell-by-gene count matrix for each dataset.

Single cell ATAC-seq (scATAC-seq) datasets: scATAC-seq datasets were obtained for 8 tissues, including Heart, Kidney, Large Intestine, Liver, Lung, Pancreas, Small Intestine, and Spleen. There are 14 datasets and in total 312,579 cells for Heart, 19 datasets and 272,526 cells for Kidney, 6 datasets and 8,368 cells for Large Intestine, 2 datasets and 50,748 cells for Liver, 9 datasets and 205,048 cells for Lung, 3 datasets and 68,294 cells for Pancreas, 3 datasets and 7,774 cells for Small Intestine, 4 datasets and 16,260 cells for Spleen. Again, data from 3 platforms (Supplementary

Table 4) was combined using a uniform processing pipeline. For this, we used SnapTools [41] for reads mapping and abundance quantification. SnapTools segments the genome into uniform-sized bins of 5,000 base pairs (bp) and a cell-by-bin matrix is created for each dataset with each element indicating the count of sequencing fragments overlapping with a given bin in a certain cell.

Slide-seq datasets: Slide-seq [42], a spatial transcriptomics methods, was used to profile Kidney samples. There are in total 551,342 cells (DNA-barcoded beads) included in 19 Slide-seq datasets. Slide-seq generates short read sequencing data that is equivalent to scRNA-seq data, and was processed in the same way we processed all scRNA-Seq data. In addition, the spatial location associated with each bead is provided.

Clustering cells with the HuBMAP modality-specific pipelines

For each tissue in each data modality, cells are merged by joining the data matrices of different datasets along the cell axis. The HuBMAP modality-specific data analysis pipelines, described below, are then used to cluster cells in each tissue of each data modality.

The HuBMAP scRNA-seq pipeline uses the Scanpy [43] package to perform cell clustering. Scanpy first filters cells that express less than 200 genes and genes that are expressed in less than three cells. Each cell is then normalized by its total counts over all genes. The produced count matrix is log transformed and scaled so that each gene has unit variance and zero mean, after which it is used as input to dimensionality reduction via PCA, computation of a k-nearest-neighbor (k-NN) graph in PCA space, and finally clustering with the Leiden algorithm [21]. We further identify differentially expressed genes for each cell cluster against all the other clusters in the tissue using the Wilcoxon rank-sum test, allowing for estimating similarities between cell clusters.

The scATAC-seq processing pipeline uses the SnapATAC package [41] for data processing, which starts from each cell-by-bin count matrix. High-quality cells are identified for subsequent analysis based on three criteria: (1) total number of unique fragment count [>2000]; (2) UMI count [>1000]; and (3) mitochondrial read ratio [<50%]. Bins are then filtered out if they overlap with the ENCODE blacklist (downloaded from http://mitra.stanford.edu/kundaje/akundaje/release/blacklists/), are in mitochondrial chromosomes, or belong to the top 5% invariant bins ranked by the cover-

age. From the filtered cell-by-bin matrix, we create a gene activity matrix. Gene coordinates are extracted and extended to include the 2,000 bp upstream region [5]. Gene activities are then computed based on the number of fragments that map to each of these regions, using the "calGmatFromMat" function in the SnapATAC package, and saved as a cell-by-gene matrix. Cell clusters are identified from the cell-by-gene matrix by using it as the input of the same scRNA-seq cell clustering workflow with the steps of normalization and rescaling, dimensionality reduction, k-NN graph computation, and clustering with the Leiden algorithm. After cell clustering, MACS2 [44] is used to call accessibility peaks over all cells for each cluster with at least 100 cells. Genes that are differentially active for each cell cluster against all the other clusters in the tissue are again calculated using the Wilcoxon rank-sum test from the gene activity matrix.

Since Slide-seq yields gene count matrices that are equivalent to those of scRNA-seq, the same cell clustering pipeline that was used for scRNA-seq is applied to Slide-seq data. The spatial information of cells is used to construct the gene spatial graphs as discussed below.

Constructing gene graphs for cell clusters

Gene graphs are constructed for cell clusters in all tissues and modalities. To focus on cell type specific genes, we filter genes based on their variance in the datasets. Specifically, the union of the 500 most variable genes from each tissue are selected, using the Scanpy package with the criteria defined in Seurat [45]. The variable genes are identified from gene expression matrices for scRNA-seq and Slide-seq, and gene activities matrices for scATAC-seq. The union of these sets results in 7,000 genes that are selected and used for building gene graphs and further analyses.

An unweighted undirected k-NN gene graph is computed for each scRNA-seq cell cluster from its normalized and log transformed expression matrix. For each pair of genes, the Pearson correlation of their expression values across cells in the datasets is computed. Each gene is then connected to the top 10 other genes with the strongest correlation.

A gene-transcription factor (TF) hypergraph is constructed for each scATAC-seq cell cluster by associating the peaks identified in each cell cluster with TFs and target genes. We identify peaks that contain TF motifs using the motifmatchr package which provides information on 633 Human TF binding motifs collected from the JASPAR 2020 [46] database. A binary peaks-by-TF matrix is generated, where an element "1" indicates that the TF's motifs are found in the region covered by

the corresponding peak. A complimentary gene-by-peak matrix is generated, where elements are set to "1" if the corresponding peaks are within the promoter regions of the corresponding genes. Here the promoter region is defined as the 500 bp upstream region of the transcription start site as was previously done by Aibar, *et al.* [47], which considers core and proximal promoters. The dot product of the two aforementioned matrices results in a gene-by-TF matrix. We then construct a hypergraph by treating the gene-by-TF matrix as the incidence matrix, where each column that corresponds to a TF is treated as a hyperedge, and genes with elements of "1" in this column are nodes connected by this hyperedge.

The expression matrix of Slide-seq is equivalent to that of scRNA-seq data. Therefore for each Slide-seq cell cluster, we construct an unweighted undirected co-expression k-NN graph using the same method used for scRNA-seq data. In addition, we construct another spatial co-expression graph for each bead cluster which links co-expressed genes in neighboring beads. For this, we identify the neighbors of each bead using the Delaunay triangulation of the bead's spatial coordinates. Then for each cluster, we compute the correlation of each pair of genes, where one side is the expression of a gene in this cluster and another side is the mean expression of a gene in the neighboring clusters. Again, each gene is connected to the top 10 most co-expressed genes and the graph is converted to a hypergraph where each neighboring-cell gene is treated as a hyperedge.

Finally, we constructed 701 graphs with 1,318,103 nodes in total for 10 tissues and 3 data modalities.

Connecting gene graphs through a cell cluster dendrogram

We build a cell cluster dendrogram using hierarchical clustering [48], where cell clusters that are close together in the dendrogram are likely functionally similar. For this, we first construct a distance matrix between cell clusters using the obtained top 50 differentially expressed or active genes of each cell cluster. Each Slide-seq cell cluster has two copies, one for the expression graph and one for the spatial graph. Given two sets of differentially expressed or active genes S_A and S_B of cell clusters A and cell cluster B, their distance is defined as:

$$1 - \frac{|S_A \cap S_B|}{|S_A \cup S_B|}, \tag{1}$$

where $\frac{|S_A \cap S_B|}{|S_A \cup S_B|}$ is the Jaccard index of the two gene sets. That is, if two cell clusters have totally different differentially expressed or active gene sets, their distance will be 1, while if they have the same gene sets, their distance will be 0. We then build the dendrogram from the distance matrix using the agglomerative clustering algorithm with the criterion of minimizing the average of distances when merging pairs of branches. The clustering process results in a dendrogram where each leaf is associated with a gene graph of a cell cluster.

Learning gene embeddings from gene graphs and the cell cluster dendrogram

To identify genes with similar function or genes participating in the same pathway we aim to embed each gene in each gene graph into a d-dimensional feature space in an unsupervised fashion. For this, we use the multi-layer graph learning objective proposed by Zitnik, *et al.* [19], which relies on connections between genes within a single graph and combines this with inter-graph relationships between genes in different graphs. Formally, let V be a given set of N genes $\{v_1, v_2, ..., v_N\}$, and $G_i = (V_i, E_i)$ be a gene graph ($i = 1, 2, ..., K$), where E_i denotes the edges between nodes $V_i \subseteq V$ within the graph. The dendrogram is a full binary tree T defined over a set of $2K - 1$ elements, where each of the K leaf elements is a gene graph. We define $\pi(i)$ as the parent of the element i in the tree, and reversely, $c(i)$ as the children of the element i in the tree. The problem is thus to learn mapping functions $f_1, f_2, ..., f_K$, each attached with one graph, such that function f_i maps nodes in V_i to feature representations in $\mathbb{R}^d$. f_i is equivalent to a matrix of $|V_i| \times d$ parameters.

We started from the assumption that genes with similar neighborhoods in each single graph perform similar functions and therefore should be embedded closely together. The following objective function is used to achieve this goal:

$$p(N_i(v)|v) = \prod_{u \in N_i(v)} \frac{\exp(f(u)^\mathsf{T} f(v))}{\sum_{z \in V_i} \exp(f(z)^\mathsf{T} f(v))}, \tag{2}$$

where $N_i(v)$ are the neighbors of node v in graph G_i. The conditional likelihood $p(N_i(v)|v)$ is equivalent the product of a series of softmax functions, maximization of $p(N_i(v)|v)$ therefore tries

to maximize the classification probabilities of v's neighbors based on learned node embeddings. With that, the objective for each graph i is set to:

$$O_i = \sum_{v \in V_i} \log p(N_i(v)|v), \text{ for } i = 1, 2, ..., K. \tag{3}$$

We assume that nearby cell clusters in the dendrogram are functionally similar, which implies that the same gene should perform similar functions in these cell clusters and therefore its embeddings in the corresponding graphs should be similar. For this, we introduce a mapping function f_i (where $i = K+1, ..., 2K-1$) for each internal, *i.e.*, non-leaf, element, similar to those of leaf elements, which maps nodes to feature representations in $\mathbb{R}^d$. Recursively, the nodes in an internal element is the union of the nodes in its children elements, $V_i = \cup_{j \in c(i)} V_j$ (where $i = K+1, ..., 2K-1$). The following regularization term is then introduced for nodes in each element i of T:

$$L_i(v) = \frac{1}{2} ||f_i(v) - f_{\pi(i)}(v)||_2^2, \tag{4}$$

where node v in the child elements is encouraged to share similar embeddings with the node v in the parent element under the Euclidean norm. We obtain the regularization for an element by summing over the terms of all its nodes: $L_i = \sum_{v \in V_i} L_i(v)$. For an internal element i, its mapping function f_i can be solved by the following closed-form solution given the mapping functions of all the other elements in T, which minimizes the regularization terms involving f_i:

$$f_i(v) = \frac{1}{|c(i)| + |\pi(i)|} (f_{\pi(i)}(v) + \sum_{j \in c(i)} f_j(v)). \tag{5}$$

With the objective function and regularization term defined, the model tries to solve the following optimization problem to find the best mapping functions $f_1, f_2, ..., f_K$ of leaf elements:

$$\max_{f_1, f_2, ..., f_K} \sum_{i \in [1,K]} O_i - \lambda L_i, \tag{6}$$

where λ is the hyperparameter of the regularization strength. To optimize this function, an iterative learning process is performed, where leaf mapping functions and internal mapping functions are updated alternately. The leaf mapping functions $f_1, f_2, ..., f_K$ are updated by performing one

epoch of stochastic gradient descent that optimizes Equation (5), while the internal mapping functions $f_{K+1}, f_{K+2}, ..., f_{2K-1}$ are updated using Equation (4).

In practice, we set the dimension hyperparameter d as 128 and the regularization strength λ as 1. We tested the robustness of GIANT to a wide set of hyperparameters values for d and λ (Extended Data Fig. 1) and observed that the results stay the same across a large range of parameter choices. The neighborhood $N(v)$ for each node in each graph is constructed using Node2vec's random walk algorithm [49]. We implemented distributed training which applies gradients on each node parallelly based on the Gensim package [50] written in Cython. With this, the algorithm is scalable to hundreds of graphs and millions of nodes.

Splitting embedding space into components and identifying enriched GOs and TF regulons in them

The learned mapping functions map each gene in each cell cluster to a point in the embedding space. To identify overlapping data sources and discover functions and pathways enriched in different regions of the space, we split the embedding space into embedding components by clustering gene points in it. To do the clustering, we generate another graph in which each point in the space (gene) is connected to its 30 nearest neighbors. The Leiden algorithm [21] with the resolutions of 1 (for identifying overlapping data sources) or 50 (for discovering enriched functions) is then applied to cluster the genes. GO enrichment is performed for genes in each component against all the genes clustered using the GOATOOLS package [51], in which the same gene from different cell clusters in the component are treated as unique copies. GO annotations with the evidence code "IEA", *i.e.*, electronic annotation without manual review, are ignored in the analysis to avoid circular reasoning. GO annotations are propagated through the GO hierarchies with "is_a" and "part_of" relationships. False discovery rate (FDR) correction with Benjamini/Hochberg is applied to adjust the enrichment P-values.

We also download manually curated TF-target regulatory relationships from the TRUST v2 [52] database. Enrichment of TF regulons (genes regulated by the same transcription factor) in each component is computed using the same method described above.

Inferring multicellular functions of genes

To identify new functions for genes we relied on genes with known functions that are co-embedded with them. Specifically, for one gene, we calculate the Euclidean distance between each pair of the gene's embeddings of different graphs, and then group all its graphs into clusters using agglomerative clustering, where two clusters will stop being merged if the distance between them is above 1.5. Then for each graph in a cluster, the gene's 50 nearest neighbors in the graph are identified based on the Euclidean distance between gene embeddings. A "common neighbor" set for the gene in this cluster is finally selected based on two criteria: (1) the neighbor gene must appear in the nearest neighbors of at least half of the graphs in the cluster; (2) the neighbor gene is in the 50 most frequent neighbors based on the number of its appearance in nearest neighbors. Next, GO enrichment analysis is performed using the common neighbors.

Data availability

Accession numbers for the HuBMAP datasets used in this study are listed in Supplementary Table 4.

Code availability

The source code of GIANT is available at https://github.com/chenhcs/GIANT.

Acknowledgements

The work in this article was partially supported by National Institutes of Health (NIH) grants OT2OD026682, 1U54AG075931 and 1U24CA268108 to Z.B.-J.

Author contributions

H.C., N.D.N. and Z.B.-J. conceptualized the study. H.C., N.D.N. and Z.B.-J. designed the algorithm and methodology. M.R. and H.C. performed data preprocessing. H.C. and N.D.N. devel-

oped the software. H.C. and Z.B.-J. performed the analysis. H.C., N.D.N. and Z.B.-J. wrote the manuscript. All authors read and approved the final manuscript.

References

[1] HuBMAP Consortium et al. "The human body at cellular resolution: the NIH Human Biomolecular Atlas Program". In: *Nature* 574.7777 (2019), p. 187.

[2] Orit Rozenblatt-Rosen et al. "The Human Cell Atlas: from vision to reality". In: *Nature* 550.7677 (2017), pp. 451–453.

[3] Patty Lee et al. "NIH SenNet Consortium: Mapping Senescent Cells in the Human Body to Understand Health and Disease". In: *Preprints* 2022.2022070160 (2022).

[4] Tim Stuart et al. "Comprehensive integration of single-cell data". In: *Cell* 177.7 (2019), pp. 1888–1902.

[5] Yuhan Hao et al. "Integrated analysis of multimodal single-cell data". In: *Cell* 184.13 (2021), pp. 3573–3587.

[6] Andrew Butler et al. "Integrating single-cell transcriptomic data across different conditions, technologies, and species". In: *Nature biotechnology* 36.5 (2018), pp. 411–420.

[7] C Dominguez Conde et al. "Cross-tissue immune cell analysis reveals tissue-specific features in humans". In: *Science* 376.6594 (2022), eabl5197.

[8] BJ Strober et al. "Dynamic genetic regulation of gene expression during cellular differentiation". In: *Science* 364.6447 (2019), pp. 1287–1290.

[9] Ricard Argelaguet et al. "Computational principles and challenges in single-cell data integration". In: *Nature biotechnology* 39.10 (2021), pp. 1202–1215.

[10] Vladimir Yu Kiselev et al. "SC3: consensus clustering of single-cell RNA-seq data". In: *Nature methods* 14.5 (2017), pp. 483–486.

[11] Euxhen Hasanaj et al. "Interactive single-cell data analysis using Cellar". In: *Nature communications* 13.1 (2022), pp. 1–6.

[12] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. "Challenges in unsupervised clustering of single-cell RNA-seq data". In: *Nature Reviews Genetics* 20.5 (2019), pp. 273–282.

[13] Haotian Teng, Ye Yuan, and Ziv Bar-Joseph. "Clustering spatial transcriptomics data". In: *Bioinformatics* 38.4 (2022), pp. 997–1004.

[14] Ye Yuan and Ziv Bar-Joseph. "GCNG: graph convolutional networks for inferring gene interaction from spatial transcriptomics data". In: *Genome biology* 21.1 (2020), pp. 1–16.

[15] Dongshunyi Li, Jun Ding, and Ziv Bar-Joseph. "Identifying signaling genes in spatial single-cell expression data". In: *Bioinformatics* 37.7 (2021), pp. 968–975.

[16] Joshua D Welch, Alexander J Hartemink, and Jan F Prins. "MATCHER: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics". In: *Genome biology* 18.1 (2017), pp. 1–19.

[17] Brian Hie, Bryan Bryson, and Bonnie Berger. "Efficient integration of heterogeneous single-cell transcriptomes using Scanorama". In: *Nature biotechnology* 37.6 (2019), pp. 685–691.

[18] Nam D Nguyen and Daifeng Wang. "Multiview learning for understanding functional multiomics". In: *PLoS computational biology* 16.4 (2020), e1007677.

[19] Marinka Zitnik and Jure Leskovec. "Predicting multicellular function through multi-layer tissue networks". In: *Bioinformatics* 33.14 (2017), pp. i190–i198.

[20] Siddharth Gopal and Yiming Yang. "Recursive regularization for large-scale classification with hierarchical and graphical dependencies". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 257–265.

[21] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. "From Louvain to Leiden: guaranteeing well-connected communities". In: *Scientific reports* 9.1 (2019), pp. 1–12.

[22] HE Fessler. "Heart-lung interactions: applications in the critically ill". In: *European Respiratory Journal* 10.1 (1997), pp. 226–237.

[23] Suzanne Y Felten and David L Felten. "Neural-immune interactions". In: *Progress in brain research* 100 (1994), pp. 157–162.

[24] Nicholas J Severs. "The cardiac muscle cell". In: *Bioessays* 22.2 (2000), pp. 188–199.

[25] Anat Ratnovsky, David Elad, and Pinchas Halpern. "Mechanics of respiratory muscles". In: *Respiratory physiology & neurobiology* 163.1-3 (2008), pp. 82–89.

[26] Kenton M Sanders et al. "Regulation of gastrointestinal motility—insights from smooth muscle biology". In: *Nature reviews Gastroenterology & hepatology* 9.11 (2012), pp. 633–645.

[27] PHUONGTRANG Nguyen et al. "Liver lipid metabolism". In: *Journal of animal physiology and animal nutrition* 92.3 (2008), pp. 272–283.

[28] Sayali A Pendharkar et al. "Relationship between pancreatic hormones and glucose metabolism: a cross-sectional study in patients after acute pancreatitis". In: *American journal of physiology-gastrointestinal and liver physiology* 311.1 (2016), G50–G58.

[29] Esti Yeger-Lotem and Roded Sharan. "Human protein interaction networks across tissues and diseases". In: *Frontiers in genetics* 6 (2015), p. 257.

[30] Dustin Khiem et al. "A p38 MAPK-MEF2C pathway regulates B-cell proliferation". In: *Proceedings of the National Academy of Sciences* 105.44 (2008), pp. 17067–17072.

[31] Peter R Wilker et al. "Transcription factor Mef2c is required for B cell proliferation and survival after antigen receptor stimulation". In: *Nature immunology* 9.6 (2008), pp. 603–612.

[32] Qing Lin et al. "Requirement of the MADS-box transcription factor MEF2C for vascular development". In: *Development* 125.22 (1998), pp. 4565–4574.

[33] Qing Lin et al. "Control of mouse cardiac morphogenesis and myogenesis by transcription factor MEF2C". In: *Science* 276.5317 (1997), pp. 1404–1407.

[34] Michael J Bround et al. "Cardiac ryanodine receptors control heart rate and rhythmicity in adult mice". In: *Cardiovascular research* 96.3 (2012), pp. 372–380.

[35] Anne Marie O'Donnell, Hiroki Nakamura, and Prem Puri. "Altered ryanodine receptor gene expression in Hirschsprung's disease". In: *Pediatric Surgery International* 35.9 (2019), pp. 923–927.

[36] Abhijeet Rajendra Sonawane et al. "Understanding tissue-specific gene regulation". In: *Cell reports* 21.4 (2017), pp. 1077–1088.

[37] Lukas Steuernagel et al. "Computational identification of tissue-specific transcription factor cooperation in ten cattle tissues". In: *PLoS One* 14.5 (2019), e0216475.

[38] Mingcong Xu et al. "TF-Marker: a comprehensive manually curated database for transcription factors and related markers in specific cell and tissue types in human". In: *Nucleic acids research* 50.D1 (2022), pp. D402–D412.

[39] Yuri Pritykin, Dario Ghersi, and Mona Singh. "Genome-wide detection and analysis of multifunctional genes". In: *PLoS computational biology* 11.10 (2015), e1004467.

[40] Rob Patro et al. "Salmon provides fast and bias-aware quantification of transcript expression". In: *Nature methods* 14.4 (2017), pp. 417–419.

[41] Rongxin Fang et al. "Comprehensive analysis of single cell ATAC-seq data with SnapATAC". In: *Nature communications* 12.1 (2021), pp. 1–15.

[42] Samuel G Rodriques et al. "Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution". In: *Science* 363.6434 (2019), pp. 1463–1467.

[43] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome biology* 19.1 (2018), pp. 1–5.

[44] Jianxing Feng et al. "Identifying ChIP-seq enrichment using MACS". In: *Nature protocols* 7.9 (2012), pp. 1728–1740.

[45] Rahul Satija et al. "Spatial reconstruction of single-cell gene expression data". In: *Nature biotechnology* 33.5 (2015), pp. 495–502.

[46] Oriol Fornes et al. "JASPAR 2020: update of the open-access database of transcription factor binding profiles". In: *Nucleic acids research* 48.D1 (2020), pp. D87–D92.

[47] Sara Aibar et al. "SCENIC: single-cell regulatory network inference and clustering". In: *Nature methods* 14.11 (2017), pp. 1083–1086.

[48] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[49] Aditya Grover and Jure Leskovec. "node2vec: Scalable feature learning for networks". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 2016, pp. 855–864.

[50] Radim Rehurek and Petr Sojka. "Software framework for topic modelling with large corpora". In: *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*. Citeseer. 2010.

[51] DV Klopfenstein et al. "GOATOOLS: A Python library for Gene Ontology analyses". In: *Scientific reports* 8.1 (2018), pp. 1–17.

[52] Heonjong Han et al. "TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions". In: *Nucleic acids research* 46.D1 (2018), pp. D380–D386.

[53] Jake R Conway, Alexander Lex, and Nils Gehlenborg. "UpSetR: an R package for the visualization of intersecting sets and their properties". In: *Bioinformatics* (2017).

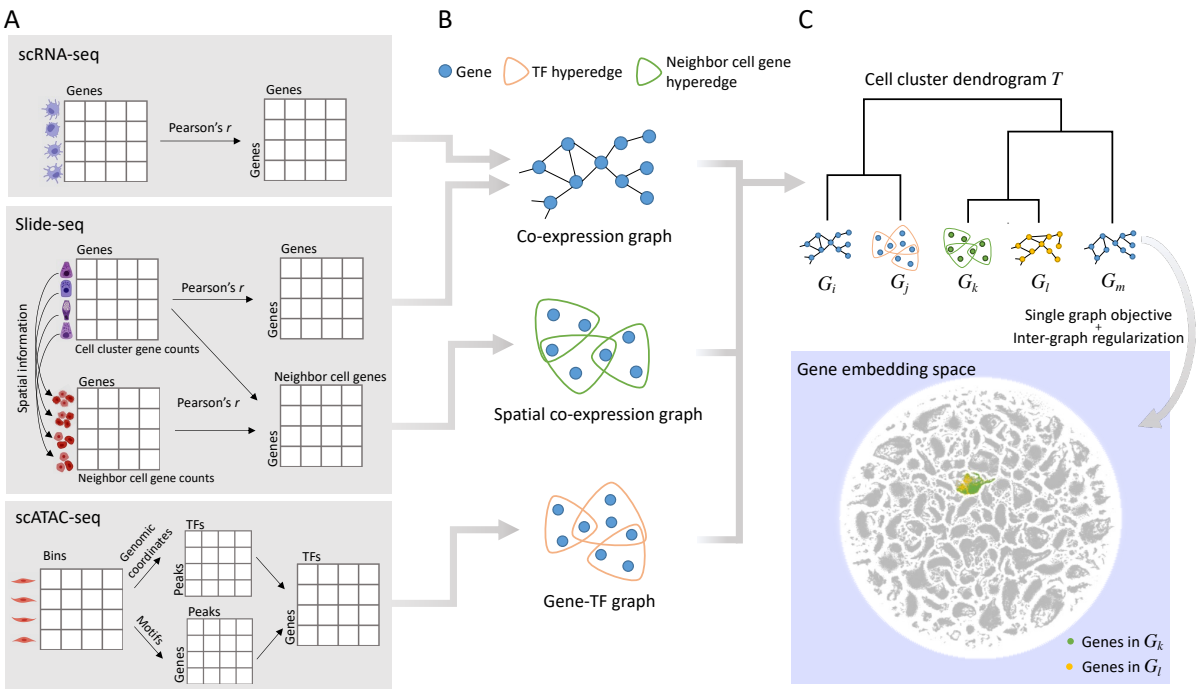


Figure 1: Schematic illustration of the method. (A) Cell clusters are identified from each tissue of each data modality. For each scRNA-seq cell cluster, a gene co-expression matrix is constructed for all the gene pairs in the cell cluster. For each Slide-seq cell cluster, the same gene co-expression matrix is also constructed, and another spatial co-expression matrix is built for gene pairs between the cells of this cell cluster and their neighboring cells. For each scATAC-seq cell cluster, identified peaks from the bin matrix are associated with TFs and genes, which forms a gene-TF matrix. (B) We obtain gene co-expression or co-activity graphs for each of these modalities based on the specific matrices computed in (A). (C) A dendrogram for graphs in (B) is constructed by hierarchical clustering of cell clusters, and a multi-layer graph learning algorithm is applied using the tree structure. Each gene in each graph is then encoded into an embedding space, where genes with similar functions are embedded together.

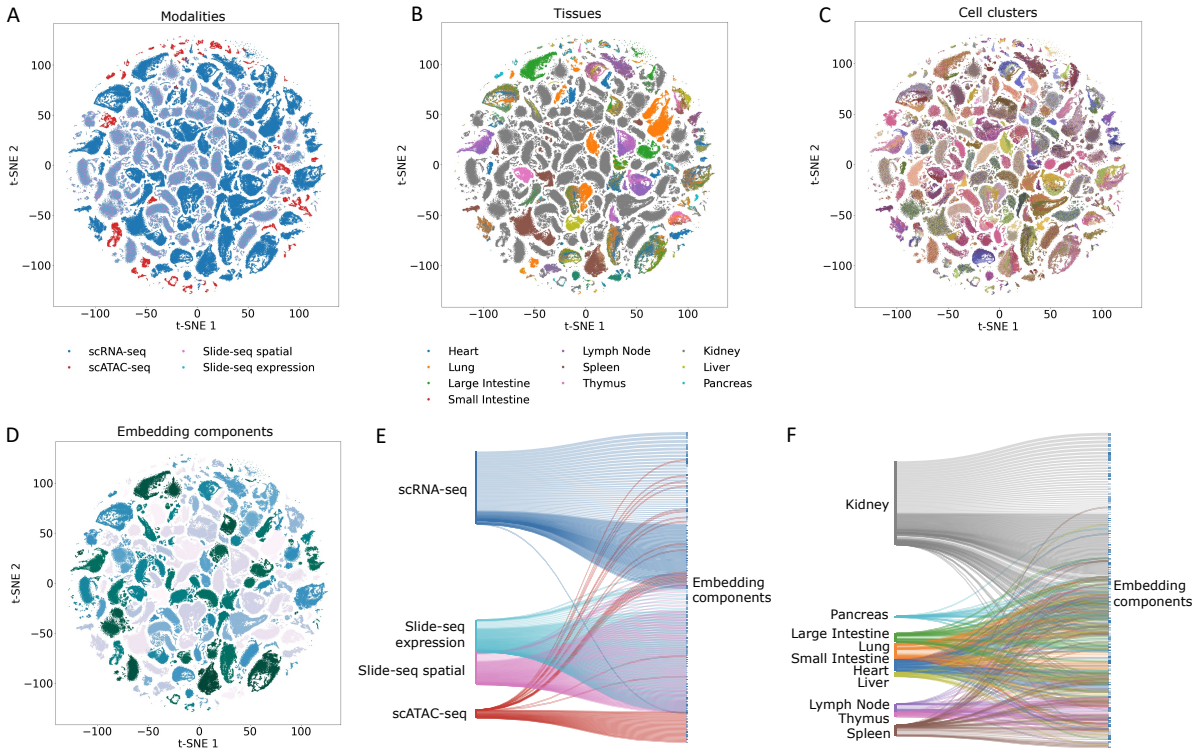


Figure 2: Distributions of data modalities, tissues, and cell clusters in the entire embedding space. (A) Distributions of data modalities in the embedding space, where genes are colored by data modalities. (B) Distributions of tissues in the embedding space, where genes are colored by tissues. (C) Distributions of cell clusters in the embedding space, where genes are colored by cell clusters. (D) The embedding space is split into components, where genes in different components are shown in different colors. (E) Cross clustering between data modalities (F) Cross clustering between tissues. As can be seen, genes from several different tissues, and from multiple modalities, are clustered together, allowing for a joint analysis of this large dataset.

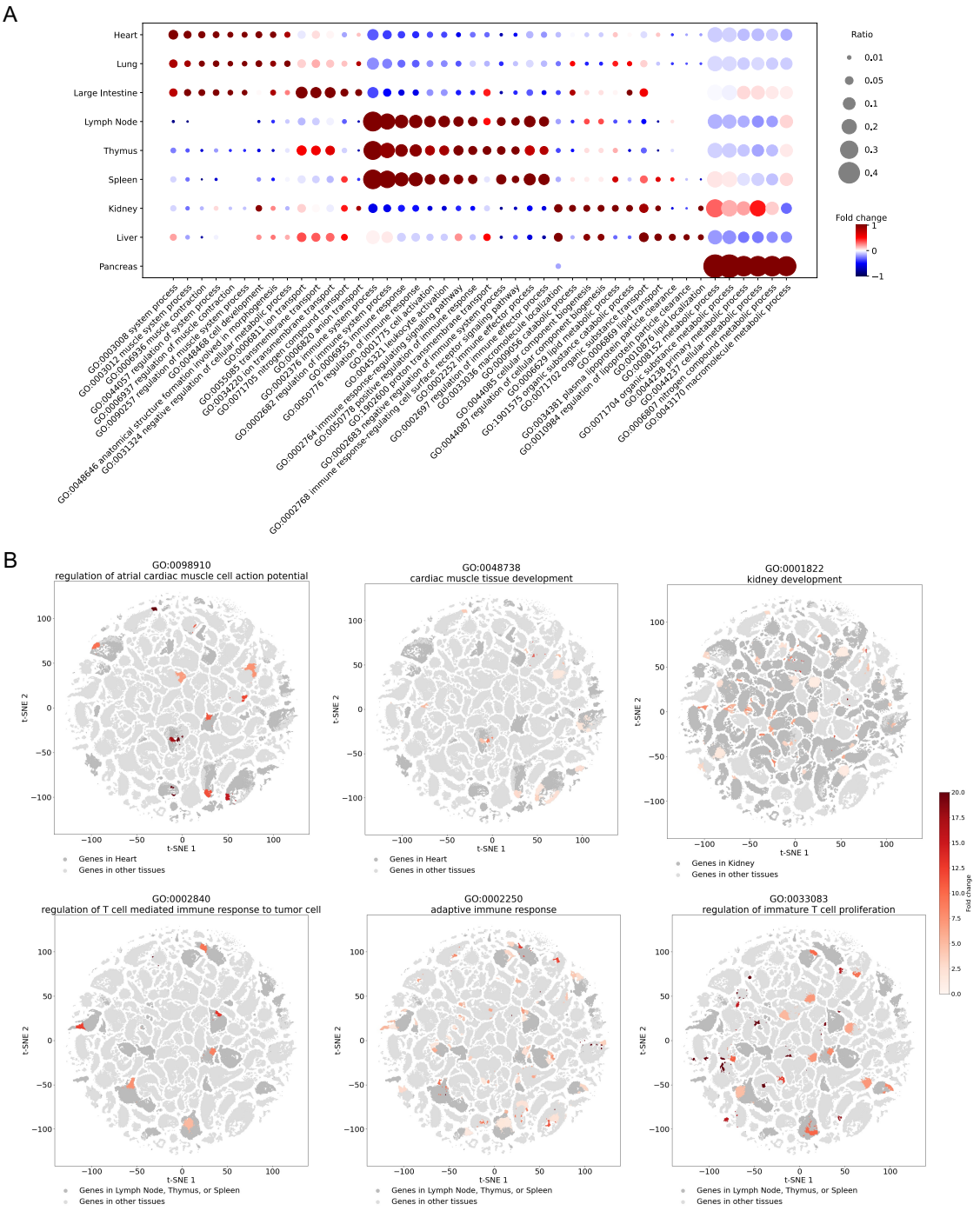

A
Heart
Lung
Large Intestine
Lymph Node
Thymus
Spleen
Kidney
Liver
Pancreas
Ratio
0.01
0.05
0.1
0.2
0.3
0.4
Fold change
1
0
−1
GO:0003008 system process
GO:0003012 muscle system process
GO:0006936 muscle contraction
GO:0044057 regulation of system process
GO:0006937 regulation of muscle contraction
GO:0090257 regulation of muscle system process
GO:0048646 anatomical structure formation involved in morphogenesis
GO:0031324 negative regulation of cellular metabolic process
GO:0044468 cell development
GO:0055085 transmembrane transport
GO:0006811 ion transport
GO:0034220 ion transmembrane transport
GO:0071705 nitrogen compound transport
GO:0006820 anion transport
GO:0002682 regulation of immune system process
GO:0002376 immune system process
GO:0050776 regulation of immune response
GO:0006955 immune response
GO:0002764 immune response-regulating signaling pathway
GO:0001775 cell activation
GO:0030778 positive regulation of leukocyte activation
GO:0045321 leukocyte activation
GO:0002683 negative regulation of immune system process
GO:1903706 regulation of hemopoiesis
GO:0002768 immune response-regulating cell surface receptor signaling pathway
GO:0002697 regulation of immune effector process
GO:0002253 activation of immune response
GO:0033036 macromolecule localization
GO:0044087 regulation of cellular component biogenesis
GO:0009056 catabolic process
GO:0044085 cellular component biogenesis
GO:1901575 organic substance catabolic process
GO:0006629 lipid metabolic process
GO:0071702 organic substance transport
GO:0034381 plasma lipoprotein particle clearance
GO:0008869 lipid transport
GO:0010984 regulation of lipoprotein particle clearance
GO:0010876 lipid localization
GO:0008152 metabolic process
GO:0071704 organic substance metabolic process
GO:0044238 primary metabolic process
GO:0008807 nitrogen compound metabolic process
GO:0044237 cellular metabolic process
GO:0043170 macromolecule metabolic process
B
GO:0098910
regulation of atrial cardiac muscle cell action potential
GO:0048738
cardiac muscle tissue development
GO:0001822
kidney development
GO:0002840
regulation of T cell mediated immune response to tumor cell
GO:0002250
adaptive immune response
GO:0033083
regulation of immature T cell proliferation
t-SNE 1
t-SNE 2
Genes in Heart
Genes in Kidney
Genes in Lymph Node, Thymus, or Spleen
Genes in other tissues
Fold change
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

Figure 3: GO terms enriched in different embedding components. (A) GO terms enriched in each embedding component are associated with the dominant tissues of the component. The size of each dot indicates the ratio of GO terms who are children terms of the GO category (column), among all the terms enriched for the tissue (row). The color of each dot indicates the fold change of the ratio over the average value of ratios in the column. (B) GO enrichment in different embedding components. Components with the GO term enriched (adjusted $P < 0.01$) are colored by the enrichment levels of the GO term, while the other components are shown in grey. Genes from tissues related to the GO term are highlighted by dark grey, while genes in the other tissues are colored by light grey.

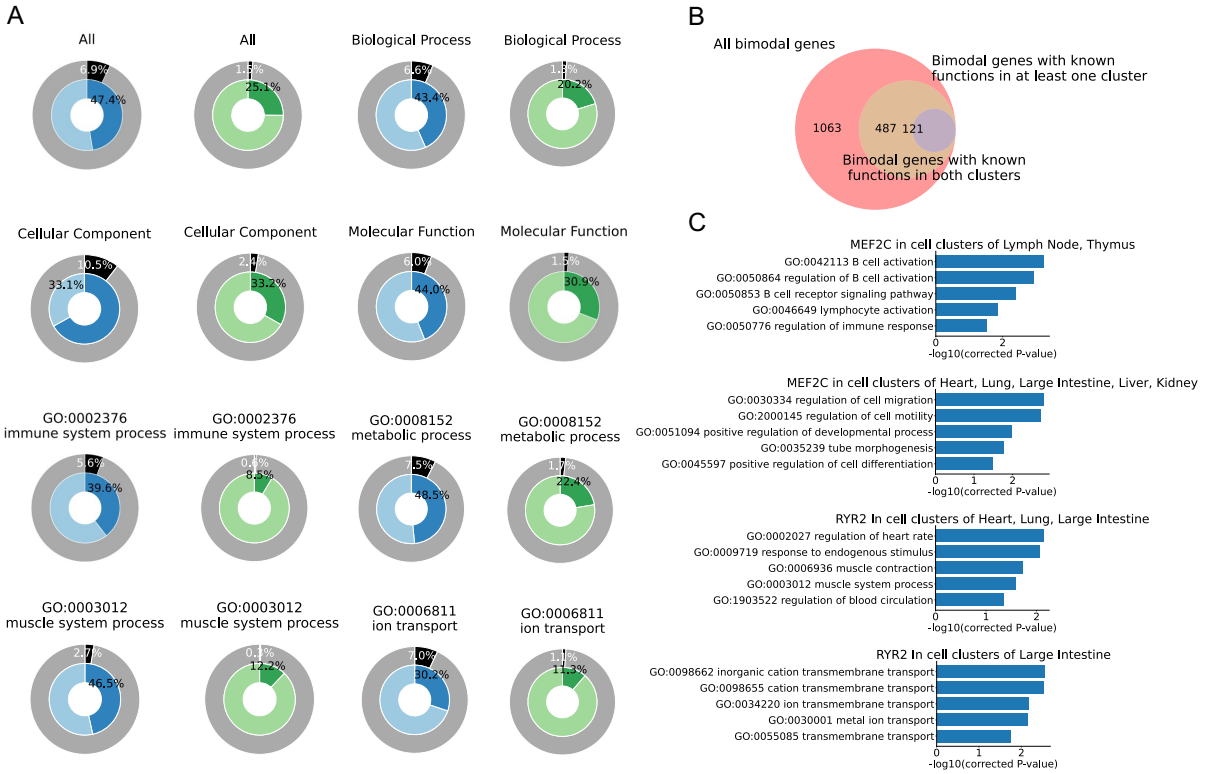


Figure 4: Inference of gene functions in cell clusters. (A) Blue circles show the ratios of known gene functions that are covered by our predictions (recall), while their outside circles indicate the ratios of our predictions in the whole prediction space (all the possible GO assignments). Green circles show the ratios of known gene functions in our predictions (precision), while their outside circles indicate the ratios of known functions in the whole prediction space. Titles indicate children GO terms of which categories are considered. Besides the whole GO space and three main GO branches, some representative GO categories in Figure 3A are also shown here. (C) The number of bimodal genes, and the number of bimodal genes for which one or both functions are confirmed. (D) Inferred functions of two genes in different cell clusters.

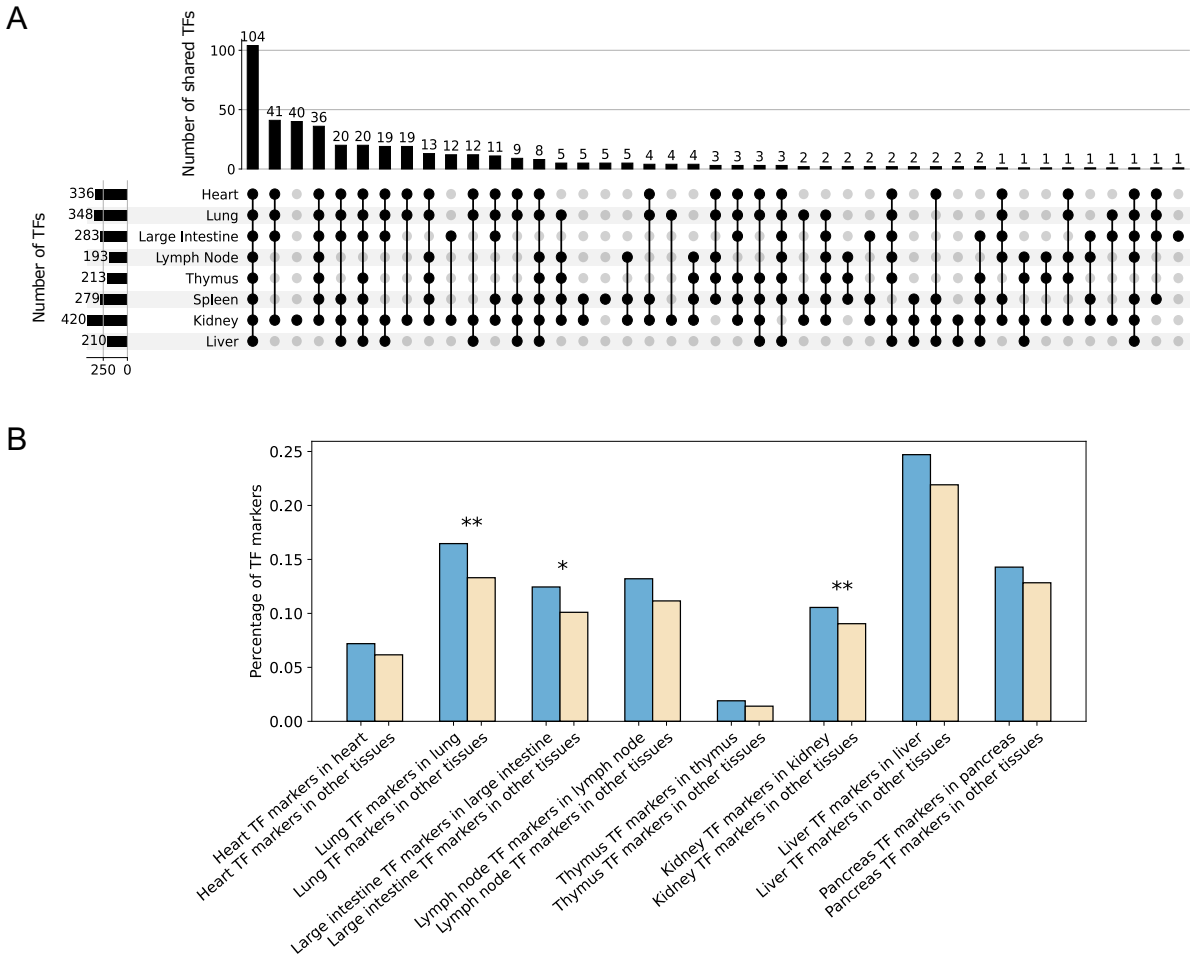
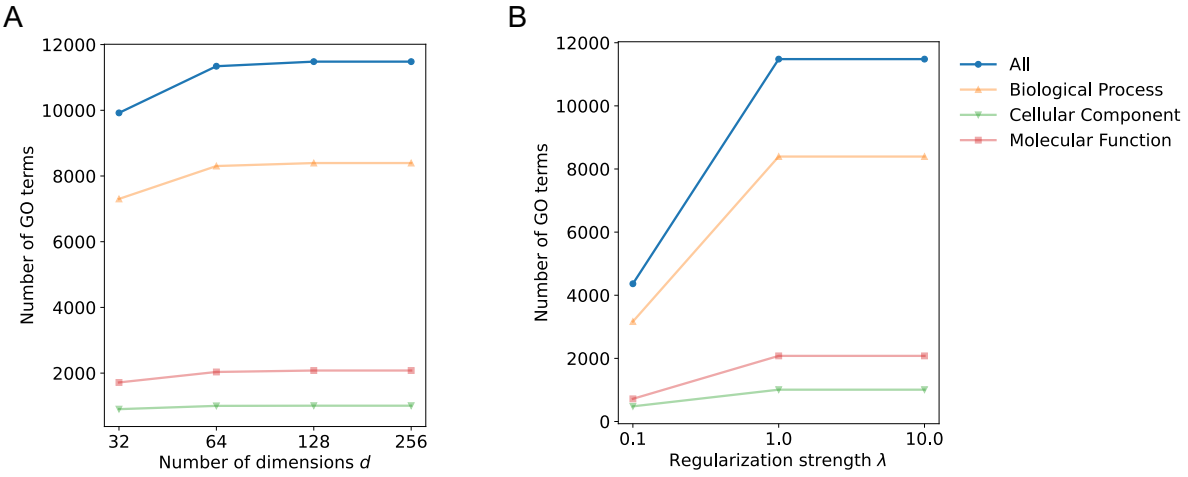


Figure 5: TFs discovered in different tissues. (A) TF intersections among different tissues. Bars on the left show the number of TFs identified in each tissue, while each bar on the top shows the number of TFs shared among tissues marked by the dots below it. Tissues with fewer than 20 TFs identified are omitted in this figure. The figure was drawn using UpSetR [53]. (B) The percentage of tissue-specific TF markers in the corresponding tissue (blue) compared to their percentage in the other tissues (light yellow). Significant enrichment in blue groups are marked by stars (*: $P \leq 0.05$, **: $P \leq 0.01$, P-values are evaluated using one-sided Fisher's exact tests).



Extended Data Fig 1: Robustness of GIANT to the choices of hyperparameters of embedding dimensions (d) and regulation strength for balancing two learning objectives (λ). We measure the performance using the number of different types of GO terms that can be discovered from different embedding components (Methods). (A) While the performance reaches the highest when the number of dimensions is up to 128, the performance is overall stable across a wide-range of values. (B) The performance drops when the model basically considers only the single graph learning objective ($\lambda = 0.1$), but when λ is in the wide range of 1.0 to 10.0, the model shows stable performance.