nature computational science

Article

https://doi.org/10.1038/s43588-024-00734-0

Mapping the gene space at single-cell resolution with gene signal pattern analysis

Received: 17 November 2023

Accepted: 30 October 2024

Published online: 20 December 2024

Check for updates

Aarthi Venkat[1], Sam Leone[2], Scott E. Youlten[3], Eric Fagerberg[4], John Attanasio[4], Nikhil S. Joshi[4], Michael Perlmutter[5,6] & Smita Krishnaswamy[1,2,3,7,8]

In single-cell sequencing analysis, several computational methods have been developed to map the cellular state space, but little has been done to map or create embeddings of the gene space. Here we formulate the gene embedding problem, design tasks with simulated single-cell data to evaluate representations, and establish ten relevant baselines. We then present a graph signal processing approach, called gene signal pattern analysis (GSPA), that learns rich gene representations from single-cell data using a dictionary of diffusion wavelets on the cell–cell graph. GSPA enables characterization of genes based on their patterning and localization on the cellular manifold. We motivate and demonstrate the efficacy of GSPA as a framework for diverse biological tasks, such as capturing gene co-expression modules, condition-specific enrichment and perturbation-specific gene–gene interactions. Then we showcase the broad utility of gene representations derived from GSPA, including for cell–cell communication (GSPA-LR), spatial transcriptomics (GSPA-multimodal) and patient response (GSPA-Pt) analysis.

A variety of techniques are used to map the cellular state space in single-cell RNA-sequencing (scRNA-seq) analysis, including dimensionality reduction approaches PCA, UMAP and PHATE[1–3]. These methods build low-dimensional embeddings based on the transcriptional similarity between cells across thousands of genes, revealing the organization of the cellular landscape, including clusters of similar cells and trajectories along phenotypic continuums. Gene expression is also highly organized, coordinated into complexes, biological processes and pathways. However, despite numerous techniques for cell embeddings, it has not been possible to apply them analogously to understand the gene landscape. High and variable degrees of biological and technical noise[4], including 'dropout', or the lack of detection of an expressed gene due to sampling inefficiency[5], affect our ability to quantify gene–gene similarity.

Rather than building gene embeddings from single-cell data directly, we frame genes as signals on a cell–cell graph viewed as a discretization of an underlying cellular manifold. While graph feature or signal embeddings have been largely understudied in machine learning, graph signal processing has made important advances by modeling features on graph nodes as signals or functions, extending classical signal processing concepts such as Fourier and wavelet analysis to graph data[6].

We therefore employ diffusion-based manifold learning to represent signals on graphs. We first model random walks across the cell–cell graph and construct a diffusion operator $\mathbf{P}$ describing transition probabilities between graph nodes. Powering $\mathbf{P}$ to t gives the transition probabilities of a t-step random walk, where smaller t capture local representations and larger t capture global representations[7]. We leverage different t to produce multiscale representations of the cell–cell graph. Inspired by classical wavelet constructions[8], diffusion wavelets[9] power $\mathbf{P}$ to increasing scales and construct matrices defined by the differences between scales (for example, for two values of t, t_1 and

[1]Computational Biology and Bioinformatics Program, Yale University, New Haven, CT, USA. [2]Applied Math Program, Yale University, New Haven, CT, USA. [3]Department of Genetics, Yale University, New Haven, CT, USA. [4]Department of Immunobiology, Yale University, New Haven, CT, USA. [5]Department of Mathematics, Boise State University, Boise, ID, USA. [6]Program in Computing, Boise State University, Boise, ID, USA. [7]Department of Computer Science, Yale University, New Haven, CT, USA. [8]Wu Tsai Institute, Yale University, New Haven, CT, USA. e-mail: smita.krishnaswamy@yale.edu

https://doi.org/10.1038/s43588-024-00734-0

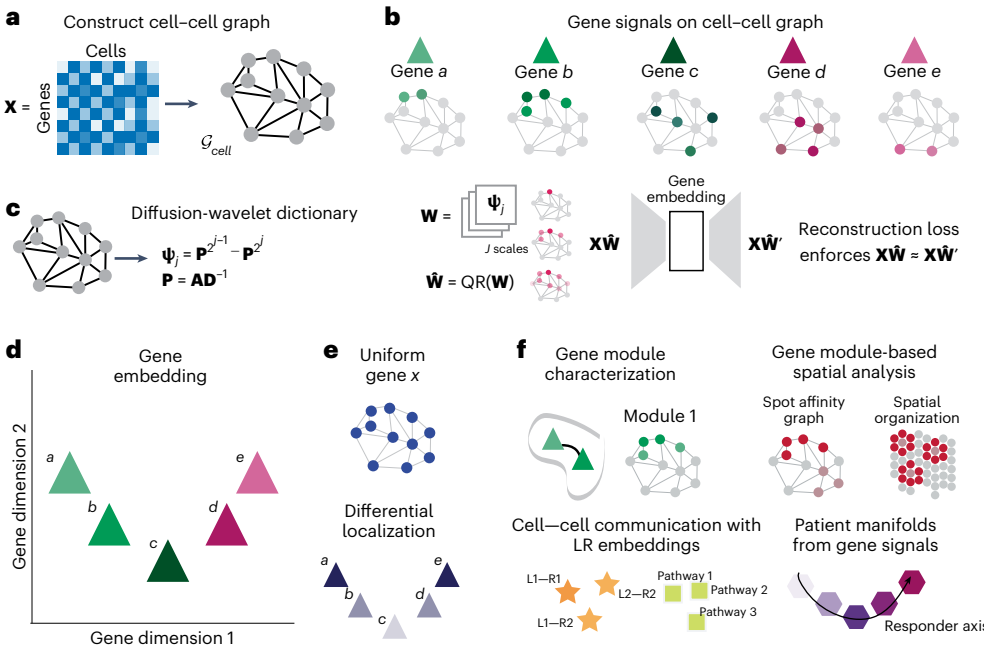


Fig. 1 | Overview of GSPA. a, Construction of a cell–cell graph, where nodes are cells and edges are affinities between cells based on similarity of transcriptomic measurements. **b**, Five demonstrative gene signals (triangles), where signals are continuous functions defined on nodes of cell–cell graph. **c**, Construction of diffusion wavelet Ψ_j at scale **j** and diffusion wavelet dictionary **W**, or QR-factorized dictionary $\hat{\mathbf{W}}$ consisting of diffusion wavelets for scales 1, …, *J*. Gene signals are projected onto the wavelet dictionary and gene embeddings are learned via an autoencoder architecture. **d**, Demonstrative gene embedding, where similar gene patterns are embedded closer together in the low-dimensional space, and far gene patterns are embedded far apart in the low-dimensional space. **e**, Differential localization determines how diffusely expressed gene signals are on a graph, where very diffusely expressed signals do not explain cell–cell variation. Genes *a* and *e* are most localized and gene *c* is least localized. **f**, Example downstream applications of GSPA, where gene embeddings enable cell-type-independent characterization of gene modules, cell–cell communication, spatial transcriptomics and patient manifolds.

t_2, $\mathbf{P}^{t_1} - \mathbf{P}^{t_2}$), allowing them to play a powerful role in graph signal processing[6].

Here we present gene signal pattern analysis (GSPA), a method for embedding genes in single-cell datasets using diffusion wavelets and deep learning. First, we build a cell–cell graph (Fig. 1a) and define genes as signals on the graph (Fig. 1b). Then we decompose gene signals using a dictionary of diffusion wavelets of varying scales centered on graph vertices (Fig. 1c). The result is a representation of each gene as a set of graph diffusion wavelet coefficients. We pair this representation with an autoencoder framework to reduce dimensionality, making it suitable for downstream tasks (Fig. 1d), including a new type of analysis that we term 'differential localization' (Fig. 1e), which identifies genes localized to distinct regions of the manifold without prior assumptions about its shape. We further create benchmarking tasks using three synthetic datasets with ground truth to test (1) the preservation of gene–gene relationships and (2) the ability to capture gene localization, which in turn enables (3) the visualization of the gene space and (4) the characterization of genes associated with gene modules, trajectories and archetypes.

We demonstrate the utility of GSPA for biological interpretation in multiple settings (Fig. 1f). In an analysis of a newly generated scRNA-seq dataset of $CD8^+$ T cells during acute and chronic infection at three timepoints[10,11], we build gene networks corresponding to T cell differentiation programs, demonstrating that GSPA uniquely identifies gene sets enriched for type 1 interferon signaling. We introduce GSPA-based ligand–receptor (LR) analysis (GSPA-LR), which identifies related LR patterns and pathways within and across cell types for cell–cell communication analysis. GSPA-LR recovers known communication pathways in a peripheral tolerance model[12]. We also introduce GSPA-multimodal, which uses a diffusion operator created from multiple modalities to learn gene embeddings. In spatial transcriptomic data of a human lymph node[13], GSPA-multimodal identifies gene modules, captures spatially variable genes and characterizes microenvironmental signaling events in key substructures. Finally, we present GSPA-Pt, which constructs patient vectors based on GSPA embeddings for improved prediction and interpretability, demonstrating effectiveness on single-cell datasets from patients with melanoma pre- and post-immunotherapy[14].

These results demonstrate the utility of considering gene-expression measurements as signals on the cell–cell graph and learning multiscale representations of graph signals. The code for GSPA and generating the results is available at https://github.com/KrishnaswamyLab/Gene-Signal-Pattern-Analysis (ref. 15).

Results

Gene embedding problem set-up

Despite the high-dimensionality of single-cell data (measuring *m* genes in *n* cells organized into an $m \times n$ matrix **X**), cells are often modeled on an underlying manifold[16]. Manifold learning methods build a cell–cell similarity graph $\mathcal{G}_{\text{cell}} = (V_{\text{cell}}, E_{\text{cell}})$, representing a discretization of the cellular manifold, where vertices are cells and edges describe transcriptional similarity (Methods). As gene measurements may also be compressed into a lower-dimensional space, we aim to learn a gene representation with respect to the cellular manifold. We seek a reduced dimensional map $\Theta : \mathbb{R}^n \to \mathbb{R}^d$, for some low dimension $d \ll n$, which (1) preserves local and global distances between signals, (2) is robust to noise and (3) is flexible to downstream tasks (Methods). We present GSPA to achieve these desired properties.

Model overview

To construct map Θ, we make the critical observation that $\mathbf{X}_i$ for gene *i* can be described as a signal (function) defined on the nodes of cell–cell graph $\mathcal{G}_{\text{cell}}$. This framing allows us to compare the similarity of gene patterns based on distances on the cellular manifold. We detail the steps of GSPA in Supplementary Algorithm 1 and Methods.

First, we build $\mathcal{G}_{\text{cell}}$ based on similarity of transcriptomic profiles between columns of $\mathbf{X}$ (Methods). Cell–cell similarity can be flexibly defined, such as from multiple modalities and multiple sequencing runs (Methods). Where batch effect affects downstream analysis, GSPA accepts batch-corrected cell measurements or cell embeddings as input or corrects for batch in the cell–cell graph through mutual nearest-neighbors (MNN) graph construction[17] (Methods and Extended Data Fig. 1). In addition, GSPA utilizes diffusion condensation[18,19] to summarize large graphs into coarse-grained graphs for improved scalability (Methods and Supplementary Fig. 1).

After constructing $\mathcal{G}_{\text{cell}}$, we use data diffusion to model random walks. With affinity matrix $\mathbf{A}$ and degree matrix $\mathbf{D}$, we define diffusion operator $\mathbf{P} = \mathbf{AD}^{-1}$ containing transition probabilities between cells on the basis of their similarity (Methods). We then use this operator to construct a dictionary of diffusion wavelets, powering $\mathbf{P}$ to different t to capture local (small t) and global (large t) representations. Encoding multiple scales shows improved representation for signal-based tasks over a single choice of t (Supplementary Figs. 2 and 3). Each diffusion wavelet is defined by the difference of these powered diffusion operators[9] (Methods). Wavelets are then organized into dictionary $\mathbf{W}$ of shape $n \times Jn$, where the number of scales J is defined as the log of the number of cells n based on Lemma 1 introduced and proven in ref. 20 (Methods). In general, a small set of large wavelets can be used to describe the graph at a coarse resolution, and we can leverage rank-revealing QR factorization as in ref. 9, resulting in a compressed wavelet dictionary (Methods). We evaluate gene embeddings with (GSPA+QR) and without (GSPA) compression.

To represent gene signals, we decompose each signal using the wavelet dictionary, encoding the local and global topology of the nodes (cells) it is defined on in addition to the signal (gene) itself. Given a gene signal $\mathbf{X}_i$ and wavelet dictionary $\mathbf{W}$ (or compressed wavelet dictionary $\widehat{\mathbf{W}}$) we project $\mathbf{X}_i$ onto the dictionary. Theorem 1 shows this wavelet projection $\mathbf{X} \to \mathbf{XW}$ is continuous with respect to an unbalanced diffusion earth mover's distance (UDEMD)[21] (Methods), which describes how similar two gene signals $\mathbf{X}_{i_1}$ and $\mathbf{X}_{i_2}$ are in a manner informed by the geometry of the cellular graph.

Finally, we learn a meaningful low-dimensional embedding of the wavelet-based representations $\mathbf{XW}$ and $\mathbf{X}\widehat{\mathbf{W}}$ using an autoencoder $D \circ E$, consisting of encoder E and decoder D:

$$\text{GSPA}(\mathbf{X}) = E(\mathbf{XW}) \qquad \text{GSPA+QR}(\mathbf{X}) = E(\mathbf{X}\widehat{\mathbf{W}})$$

where $\mathbf{W}$ and $\widehat{\mathbf{W}}$ are uncompressed and compressed wavelet dictionaries (respectively), E is the encoder, and GSPA and GSPA+QR are taken to be the map Θ.

Achieving desired representation properties with GSPA

Theorem 1 guarantees preservation of gene distances with respect to the cellular manifold; we will have $\text{GSPA}(\mathbf{X}_i) \approx \text{GSPA}(\mathbf{X}_j)$ whenever $\mathbf{X}_i$ is close to $\mathbf{X}_j$ with respect to the UDEMD (see Methods for proof of Theorem 1 and detailed discussion of UDEMD). GSPA also achieves noise robustness, where acting on a signal $\mathbf{X}_i$ by $\mathbf{P}^t$ preserves the portion of the signal aligned with the first eigenvector and depresses the portion of the signal corresponding to the other eigenvectors by a factor of eigenvalue λ^t. As t increases, this denoises by suppressing the high-frequency portion of the signal. Therefore, we can restrict the dictionary to wavelets that decompose only lower frequencies by initially multiplying each wavelet by $\mathbf{P}^t$. We note the distance preservation result from Theorem 1 shows the wavelet projection is continuous with respect to the UDEMD, which may be viewed as a form of noise robustness (Methods).

Finally, the low-dimensional representation is flexible to downstream tasks. In the following sections, we demonstrate this flexibility. First, we describe two gene rankings enabled by GSPA representations. Then we benchmark GSPA against baselines for preservation of gene–gene distances and gene localization. We showcase GSPA for characterizing $CD8^+$ T cell states, then present three adaptations for downstream tasks beyond cellular heterogeneity: GSPA-LR for cluster-independent cell–cell communication analysis, GSPA-multimodal to integrate multiple modalities for gene embeddings, and GSPA-Pt to build patient-level representations for downstream exploration.

Ranking based on embedded distances from synthetic signals

Beyond preserving relationships within the gene space, GSPA also preserves distances to any signal defined on the cell–cell graph, allowing us to rank genes by distance to synthetic signals. Here we describe two such approaches to calculate cell-type association and gene localization.

For datasets with assigned cell types, GSPA naturally enables identification of cell-type-specific genes. Where each cell is assigned a cell type, for each cell type C, we can define a set indicator signal $\mathbf{1}_C$ on all vertices of the cell–cell graph V_{cell}, where $\mathbf{1}_C(v) = 1$ if $v \in C$ and $\mathbf{1}_C(v) = 0$, otherwise for vertex $v \in V_{\text{cell}}$. Then we can rank genes in $\mathbf{X}$ by closeness to $\mathbf{1}_C$ in the dictionary representation (Methods).

However, it can be non-trivial to assign cluster labels and define cell-type-specific genes[22], including for datasets with cellular trajectories, fine-grained subtypes within coarse-grained cell types, rare cell types and highly plastic cell types. Thus, there is a need to identify genes specific to particular populations to characterize cell–cell variation without prior clustering or annotation.

We present an alternative strategy to calculate the specificity of gene signal $\mathbf{X}_i$, termed 'differential localization'. We observe that uniformly expressed genes are least likely to be involved in cell-state-defining biological processes, in line with previous work[23]. Leveraging this, we calculate the distance between each gene signal and a multiscale representation of a uniform (constant) signal $\mathbf{u} = \frac{1}{\sqrt{n}}\mathbf{1}$, normalized as other signals, where the gene localization score corresponds to the distance from the uniform signal representation (Methods). Localized genes are considered more relevant for characterizing cell–cell variation and can be used for feature selection without the underlying assumption of clusters.

Comparison with alternative gene-mapping strategies. We evaluate embeddings on three single-cell datasets simulated with Splatter (one trajectory, two branches and three branches)[24]. Splatter allows us to generate corresponding noisy counts $\mathbf{X}$ and unseen true (noiseless) counts $\mathbf{X}'$ and increase dropout probability to reflect true single-cell data (84.3–85.3% sparse; Fig. 2a). We adapt ten baselines for comparisons and assess the embeddings for preservation of co-expression, fidelity of visualization, coherence of downstream analyses and ability to capture signal localization. We compare against approaches from graph signal processing and representation learning for benchmarking tasks[25–31] (Extended Data Fig. 2 and Methods).

For our first task, we aimed to ensure that co-expressed genes are close in low-dimensional space, while genes expressed in different cells remain distant. We defined the co-expression between genes i_1 and i_2 as the correlation of the true counts $\mathbf{X}'_{i_1}$ and $\mathbf{X}'_{i_2}$. We then learned gene embeddings e_{i_1} and e_{i_2} from the noisy counts $\mathbf{X}$ and computed their distance. The correlation between this distance and their co-expression served as our evaluation metric (Methods, Fig. 2b and Extended Data Fig. 3a). GSPA+QR performed best on all benchmarking datasets, followed by GSPA (Fig. 2b and Extended Data Fig. 3b,c). GSPA and GSPA+QR also surpassed other approaches across data normalizations and graph constructions, with approaches using the cell–cell graph showing better performance overall (Methods and Extended Data Fig. 4). This supports our assertion that analyzing genes with respect to the cell–cell graph can improve gene–gene analysis.

We next evaluated the coherence of downstream analyses. For visualization, GSPA and GSPA+QR are largely unaffected by differences in the number of cells expressing each gene. For most comparisons,

https://doi.org/10.1038/s43588-024-00734-0

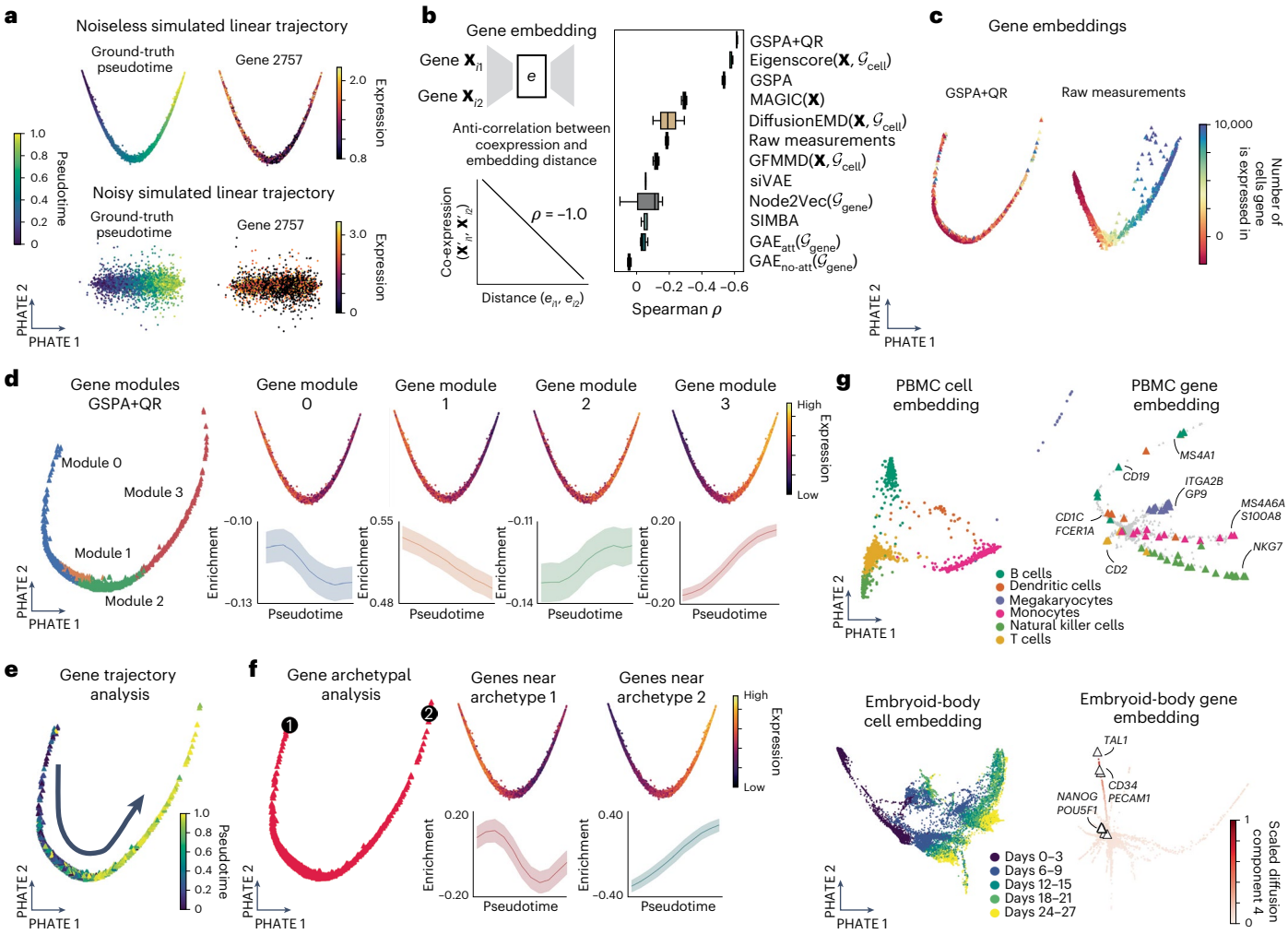


Fig. 2 | Capturing coherent visualization and gene modules, trajectories and archetypes. a, Noiseless (top) and noisy (bottom) cell embeddings of simulated linear trajectory, colored by ground-truth pseudotime provided by the simulation engine (left) and example gene expression (right). **b**, Experimental set-up (left) and Spearman correlation (ρ) evaluating performance on task for all comparisons across three runs (right). **c**, Gene embeddings of GSPA+QR and raw measurements colored by number of cells gene is expressed in. **d**, Gene modules detected by Leiden clustering (left) for GSPA+QR, and gene module enrichment and expression over time (right). Expression over time presented as mean expression of genes within module ±1 s.d. **e**, GSPA+QR gene embedding colored by time at which gene peaks. **f**, GSPA+QR gene embedding with archetypes identified via AAnet (left), with gene enrichment and expression over time visualized for 'archetypal' genes (genes closest to each archetype). Expression over time presented as mean expression of genes within module ±1 s.d. **g**, PBMC cell embedding and gene embedding with key PBMC markers annotated from PanglaoDB (top) and embryoid-body cell embedding and gene embedding (bottom), colored by diffusion eigenvector and key hemangioblast lineage markers annotated from ref. 3.

the major axis of variation correlates with this measure, suggesting a confounding factor for gene embedding analysis (Fig. 2c and Supplementary Fig. 4). For comparison of clusters from the gene embedding space (termed gene modules), genes within GSPA and GSPA+QR modules show enriched activation in distinct regions of the embedding and a shared expression trend over time (Methods, Fig. 2d and Extended Data Fig. 3d,e). Other approaches failed to produce interpretable module enrichment scores, including those methods designed for gene module detection[32] (Supplementary Fig. 5). Beyond modules, gene embeddings facilitate trajectory analysis and archetypal analysis, also used for cell-based analysis. Gene trajectory analysis revealed GSPA+QR and GSPA embeddings order genes by the pseudotime at which they peak (Fig. 2e and Supplementary Fig. 5). Gene archetypal analysis revealed genes near each archetype are enriched at the beginning and end of the trajectory (Fig. 2f and Supplementary Fig. 5). Almost all other comparisons did not capture these trends (Supplementary Fig. 5).

GSPA+QR also identified known gene relationships from real single-cell data (Methods and Fig. 2g). For peripheral blood mononuclear cells (PBMCs)[33], GSPA+QR grouped cell-type-specific genes from PanglaoDB[34]. In embryoid-body cell lineages[3], one gene trajectory reflected a hemangioblast lineage identified in ref. 3, with early trajectory genes linked to embryonic stem cells (*NANOG*, *POU5F1*) and late trajectory genes linked to hemangioblast-specific signatures (*CD34*, *PECAM1*, *TAL1*).

We next benchmarked embeddings for their ability to capture gene localization (Fig. 3a). While simulated datasets consist of genes with varying localization, we do not have ground-truth scores. Thus, we used the same simulated datasets and generated artificial graph signals of varying localization. We derived 'ground truth' localization scores based on the insight that we could constrain the size of the region where signals were defined, termed 'window'. The size of this window is inversely related to the true localization score (Methods and Extended Data Fig. 5).

We then predicted signal localization by embedding the uniform signal and computing the multiscale distance of each signal to the uniform signal (as described in 'Computation of differential localization' in Methods; Fig. 3b). GSPA+QR outcompeted baselines for all three simulated datasets, followed by GSPA (Fig. 3c and Extended Data Fig. 6),

https://doi.org/10.1038/s43588-024-00734-0

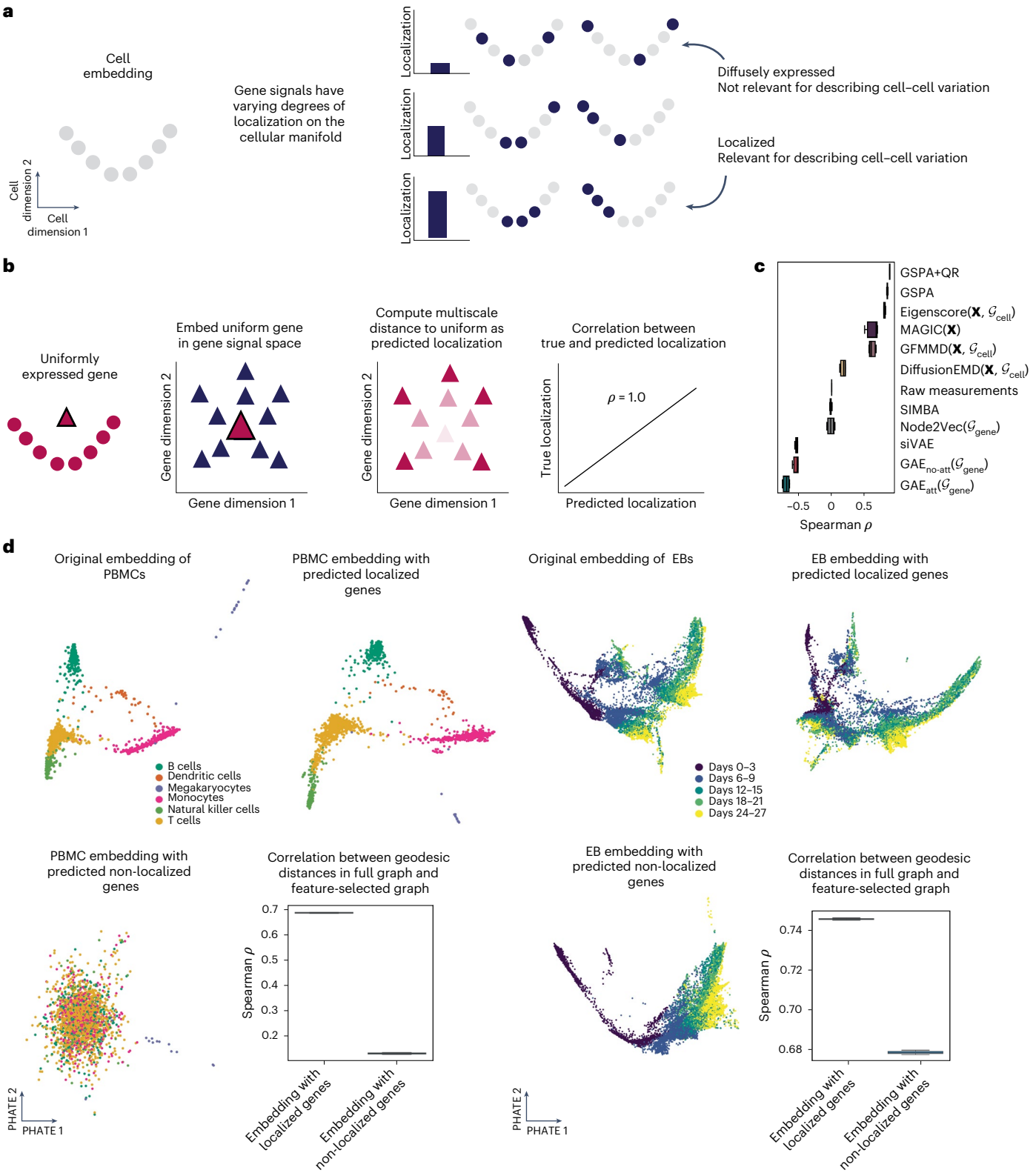


Fig. 3 | Differential localization analysis enabled by GSPA. a, Differential localization diagram. **b**, Diagram of how localization reveals genes that are most distant from uniform. **c**, Spearman correlation evaluating performance for all comparisons across three runs. **d**, Original cell embedding versus cell embedding generated with predicted localized genes or predicted non-localized genes only; correlation between geodesic distances in original cell–cell graph versus feature-selected cell–cell graphs (for 100,000 pairwise distances subsampled twice). Shown for PBMCs (left) and embryoid-body data (right). EB, embryoid body.

maintaining performance across data normalizations and graph constructions (Methods and Extended Data Fig. 4), with those approaches using the cell–cell graph showing better overall performance.

Localized genes inform cell–cell variation, suggesting their use for topologically informed feature selection (Methods). For PBMCs, visualizing cells using all genes versus only top predicted localized

genes showed a similar cell embedding, suggesting that localized genes capture cell-type variation and preserve information about the entire transcriptional space. By contrast, visualizing cells using the least localized genes lost all cell-type variation. The correlation between pairwise geodesic distances between cells in the cell–cell graph derived from all versus selected genes indicates that localized genes better preserve information about cell–cell relationships (Fig. 3d). For the embryoid-body dataset, predicted localized genes captured major and minor trajectories, while non-localized genes identified only the major timecourse axis. The correlation between pairwise geodesic distances demonstrated stronger preservation of cell–cell distances by localized genes (Fig. 3d).

Co-expression in CD8$^+$ T cells with gene embeddings

For our first case study, we investigated CD8$^+$ T cell differentiation in response to infection, characterized by highly heterogeneous and plastic substates[35]. However, the gene signaling pathways and relationships defining these transitions are not fully known and can provide insights for therapeutic intervention.

We analyzed a newly developed dataset comprising 39,704 sorted CD8$^+$ Tetramer$^+$ T cells, sequenced at three timepoints (day 4, day 8 and day 40) from acute and chronic lymphocytic choriomeningitis virus (LCMV) infections[10] (Methods). Visualizing the cellular manifold reveals that cells do not separate into clusters or trajectories. Clustering cells and identifying differentially expressed genes captures genes highly expressed in more than one cluster (for example, *Rps19*; Extended Data Fig. 7a,b). T cell markers also show lack of cluster specificity, and some gene signatures, for example, proliferation, overlap others, for example, activation (Fig. 4a and Extended Data Fig. 7b), motivating mapping the gene space to capture signatures at single-cell resolution.

We computed representations and localization scores for highly variable genes with GSPA+QR, clustering them into six modules (Methods, Fig. 4b and Supplementary Table 1). Localized genes were on the periphery of the embedding due to commitment to a particular region of the cell–cell graph (Fig. 4c), and localization scores are not associated with number of cells expressing each gene ($\rho = -0.159$).

To compare differential localization and differential gene expression, we designed a cell-clustering rank based on maximum z-score across cell clusters, finding a slight positive correlation with localization scores (Methods; $\rho = 0.327$) (Fig. 4d). Genes highly ranked by both categories were enriched in a particular cluster (for example, *S1pr5*), and genes lowly ranked by both showed low or non-specific enrichment overall (for example, *Trav13d-1*). Other genes (for example, *Rps20*) were ranked highly by cell clustering but lowly by localization due to high expression in one cluster, but high overall expression. Genes (for example, *Tox*) ranked highly by localization but lowly by cell clustering showed varied but specific enrichment. Localization can thus identify cluster-specific signatures without clustering, prioritize subtle, lowly expressed signatures, and deprioritize ubiquitously expressed genes. Furthermore, re-embedding cells using the top localized genes better preserves the overall manifold structure than re-embedding cells using the bottom localized genes, highlighting localization for topologically informed feature selection (Supplementary Fig. 6).

Next we characterized each gene module (Fig. 4e). Module 0 was memory-specific, enriched in cells from acute infection at day 40 (acute day 40) and containing hallmark genes *Il7r* and *Bcl2*[36]. Module 1 included naive and memory genes, enriched in acute day 4 and day 40. Module 2 was proliferation-specific, enriched in acute day 4 and chronic day 4. Module 3 included effector genes, enriched in acute day 8 and chronic day 8, and module 4, with late effector and exhaustion genes, was enriched in acute day 8, chronic day 8 and chronic day 40 (Fig. 4e). Module 5 captured an interferon response signature, interestingly present in both acute and chronic day 4 but persistent in chronic settings, in line with findings that type 1 interferon can enforce T cell exhaustion in chronic infection[37,38] (Fig. 4e). We built gene module-specific networks using highly localized genes and showed with STRINGdb[39] protein–protein interaction analysis that all modules had significantly higher interaction than expected ($P < 1.0 \times 10^{-16}$ for all modules), suggesting representation of each gene module in previous literature (Methods and Extended Data Fig. 7c).

To understand whether GSPA facilitates biological discovery beyond existing approaches, we constructed an experiment based on gene set enrichment of type 1 interferon signaling (Methods). The top differentially expressed genes from each cell cluster show no type 1 interferon enrichment, despite its known relevance to T cells in chronic conditions (Fig. 4f). Similarly, the top localized genes in the gene module of interest from consensus non-negative matrix factorization (cNMF) and gene embedding approaches show low enrichment scores. By contrast, the top localized genes in the gene module of interest from GSPA and GSPA+QR reveal strong enrichment, suggesting gene module and localization analysis with GSPA can uniquely identify this signature.

Finally, we analyzed perturbation effects by comparing gene–gene networks for acute day 8 negative control cells versus cells from a *Tbx21* knockout and a *Klf2* knockout, visualizing only gene–gene interactions knocked out by the knockout but present in the control[11] (Fig. 4g, Extended Data Fig. 7d and Methods). This uncovered networks lost in the *Klf2* knockout, including surrounding *Bach2* and *Id2*, implicated in memory and effector differentiation respectively[40,41]. Interestingly, in the *Tbx21* knockout network, *Cd69* and *Batf* appear, possibly reflective of the role of *Tbx21* in enforcing effector differentiation[42]. Such analysis highlights the utility of GSPA for capturing perturbation-specific gene coordination.

Cluster-independent cell–cell communication with GSPA-LR

Characterizing gene signals beyond cell-state boundaries is especially useful for cell–cell communication analysis, as interactions can involve multiple cell states or a small stimulated subset of cells. Traditional communication analysis often overlooks such interactions[43].

Using known LR pairs, GSPA-LR obtains ligand (L) and receptor (R) embeddings individually, then concatenates them into a pair representation. Mapping LR pairs reveals shared patterning on the cellular manifold across and within cell types. We can additionally map the pathway space based on LR–LR pair similarity and pathway attributes defined for each pair (Methods and Fig. 5a).

We applied GSPA-LR to examine the role of immune inhibitory receptor PD-1 in immune-related adverse events[12] (Methods). The original work showed the principal ligand of PD-1, PD-L1, is upregulated by myeloid cells to bind to PD-1 on CD8$^+$ T cells and maintain healthy tissue, with PD-1-blocking immunotherapies resulting in adverse events. We sought to recover the antigen-specific PD-1–PD-L1 interaction using our pipeline.

We analyzed 21,178 skin cells from three conditions: antigen off (NO AG), antigen-expressing (AG) and antigen-expressing treated with checkpoint inhibitors (AG CPI) (Fig. 5b). This dataset was cell-type-annotated (Fig. 5c), where the gene embedding showed separate regions enriched for each cell type (Extended Data Fig. 8a), and genes predicted to be associated with each cell type were strongly and specifically enriched (Extended Data Fig. 8b).

Despite cell-type separation, many communication patterns were enriched across multiple cell types. *Ccl5* was enriched in epithelial, myeloid and T cells, and *Ccr5* in myeloid cells and T cells, both activated by antigen (AG and AG CPI). *Cd274* (encoding PD-L1) was enriched in myeloid cells, and *Pdcd1* (encoding PD-1) in T cells, also activated by antigen (Fig. 5d). CellPhoneDB[44] identified interactions between *Ccl5* in myeloid cells and T cells to *Ccr5* in all cell types, but misses *Ccl5* enrichment in epithelial cells and the condition-specific nature of the interaction. CellPhoneDB also captures the low expression of *Cd274* across all cell types, but fails to highlight its strong antigen-specific enrichment in myeloid cells, as validated in ref. 12 (Fig. 5e).

https://doi.org/10.1038/s43588-024-00734-0

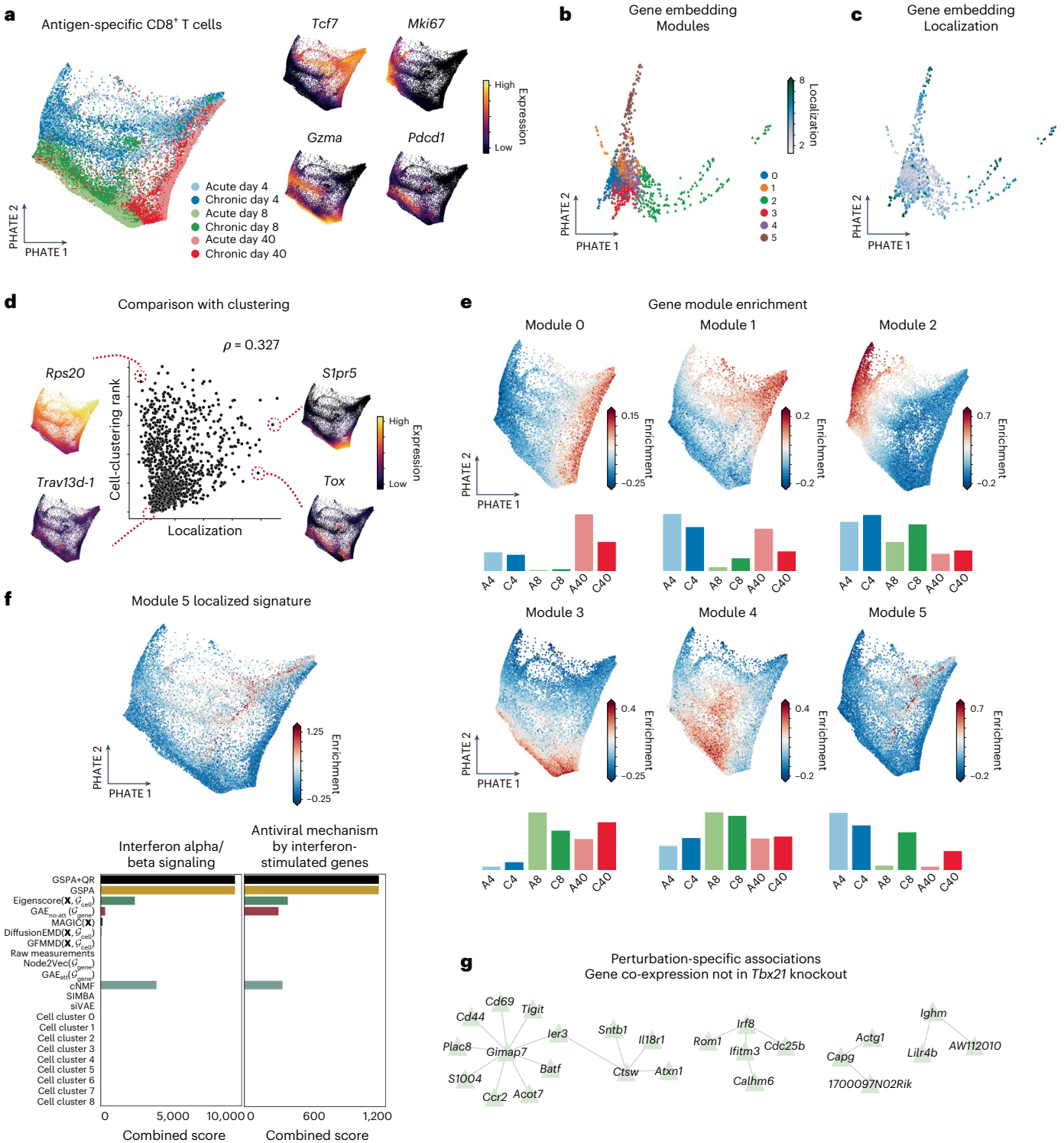


Fig. 4 | Gene–gene co-expression in CD8⁺ T cells during acute and chronic infection. a, PHATE embedding of antigen-specific CD8⁺ T cells from six experimental conditions (left) and marker genes visualized (right). **b**, Gene embedding visualized with PHATE, colored by gene module assignment. **c**, Gene embedding visualized with PHATE, colored by computed localization score. **d**, Cell clustering rank versus localization score, with representative genes visualized to demonstrate similarities and differences. **e**, Gene module enrichment across all cells and per condition (A, acute; C, chronic) and timepoint (days 4, 8 and 40). **f**, Enrichment of top localized genes enriched in gene module 5 for GSPA+QR (top), and gene set enrichment scores for type 1 interferon gene sets for top genes from all comparisons (bottom). **g**, *k*NN graph of gene–gene co-expression relationships that were knocked out in *Tbx21* knockout.

GSPA-LR reveals LR pair modules capturing a diverse range of ligand and receptor profiles without leveraging cell-type annotation (Methods, Fig. 5f, Supplementary Table 2 and Supplementary Fig. 7). Module 5 includes *Ccl5*–*Ccr5* and shows ligand enrichment in AG and AG CPI subsets of epithelial, myeloid and T cells and receptor enrichment in AG and AG CPI subsets of myeloid and T cells. Gene set enrichment analysis indicates condition-specific processes related to migration and effector activity. Module 19 includes *Cd274*–*Pdcd1*

https://doi.org/10.1038/s43588-024-00734-0

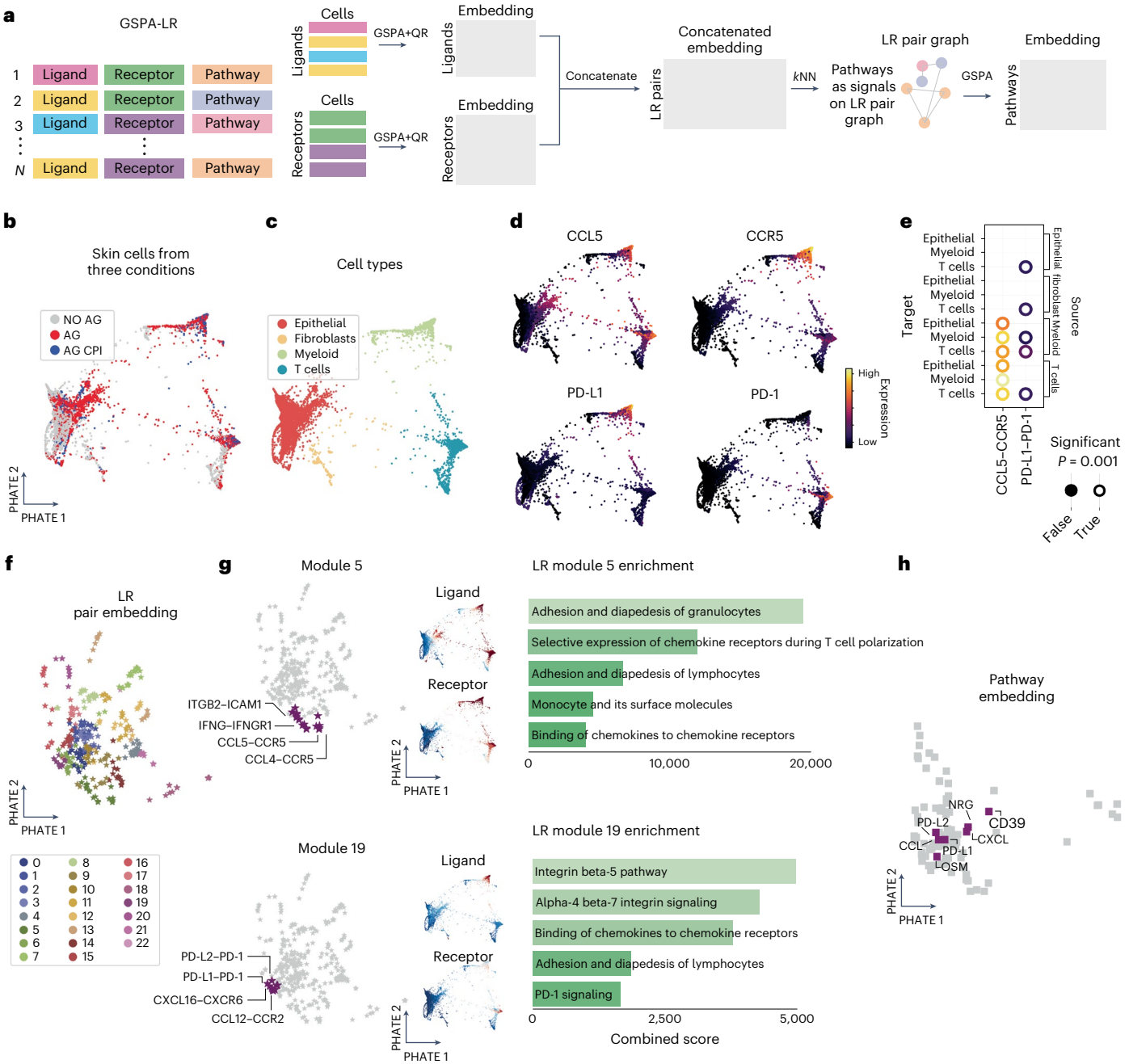


Fig. 5 | Cluster-independent LR signal patterns in peripheral tolerance skin model. a, Schematic of the GSPA-LR pipeline. **b**, Skin cells from no antigen (NO AG), antigen (AG) and antigen with checkpoint inhibitor (AG CPI) conditions visualized with PHATE. **c**, Skin cells colored by previously annotated cell types. **d**, Skin cells colored by CCL5, CCR5, PD-L1 and PD-1. **e**, Permutation test result from CellPhoneDB. **f**, LR pair embedding visualized with PHATE. **g**, Visualization of pairs, ligand and receptor enrichment, and gene set enrichment scores for module 5 (top) and module 19 (bottom). **h**, Pathway embedding visualized with PHATE.

and shows ligand enrichment in AG and AG CPI-specific myeloid cells and receptor enrichment in AG and AG CPI-specific T cells, as previously described[12]. Module 19 is enriched for integrin activity and PD-1 signaling, associated with T cell activation and exhaustion (Fig. 5g). GSPA-LR also maps pathway relationships, revealing co-localization of PD-L1/2 and T cell cytokine/chemokine secretion (Fig. 5h). Overall, GSPA-LR captures pair–pair and pathway–pathway relationships, demonstrating its capability for multiscale gene signaling analysis.

Spatially organized gene modules with GSPA-multimodal

GSPA can derive feature embeddings for modalities beyond scRNA-seq (GSPA-multimodal). Through integrated diffusion[45,46], GSPA creates a joint diffusion operator $\mathbf{P}_{\text{integrated}}$ from multimodal data, then constructs an integrated wavelet dictionary informed by all modalities (Methods and Fig. 6a).

We applied GSPA-multimodal to 10x Visium spatial transcriptomic data from human lymph node[13] (Methods and Fig. 6b), constructing gene embeddings informed by gene expression and spatial affinity and identifying spatially variable genes using localization. Gene modules and localized genes highlight the spatial localization of each signature (Methods, Fig. 6c,d and Supplementary Table 3).

We show that spatially variable genes from SpatialDE[47] correspond to our localized genes (Methods). GSPA-multimodal's integration of expression and spatial graphs allows us to further identify biologically relevant genes, such as *CD34* and *MMRN2*, enriched in the adventitia of

https://doi.org/10.1038/s43588-024-00734-0

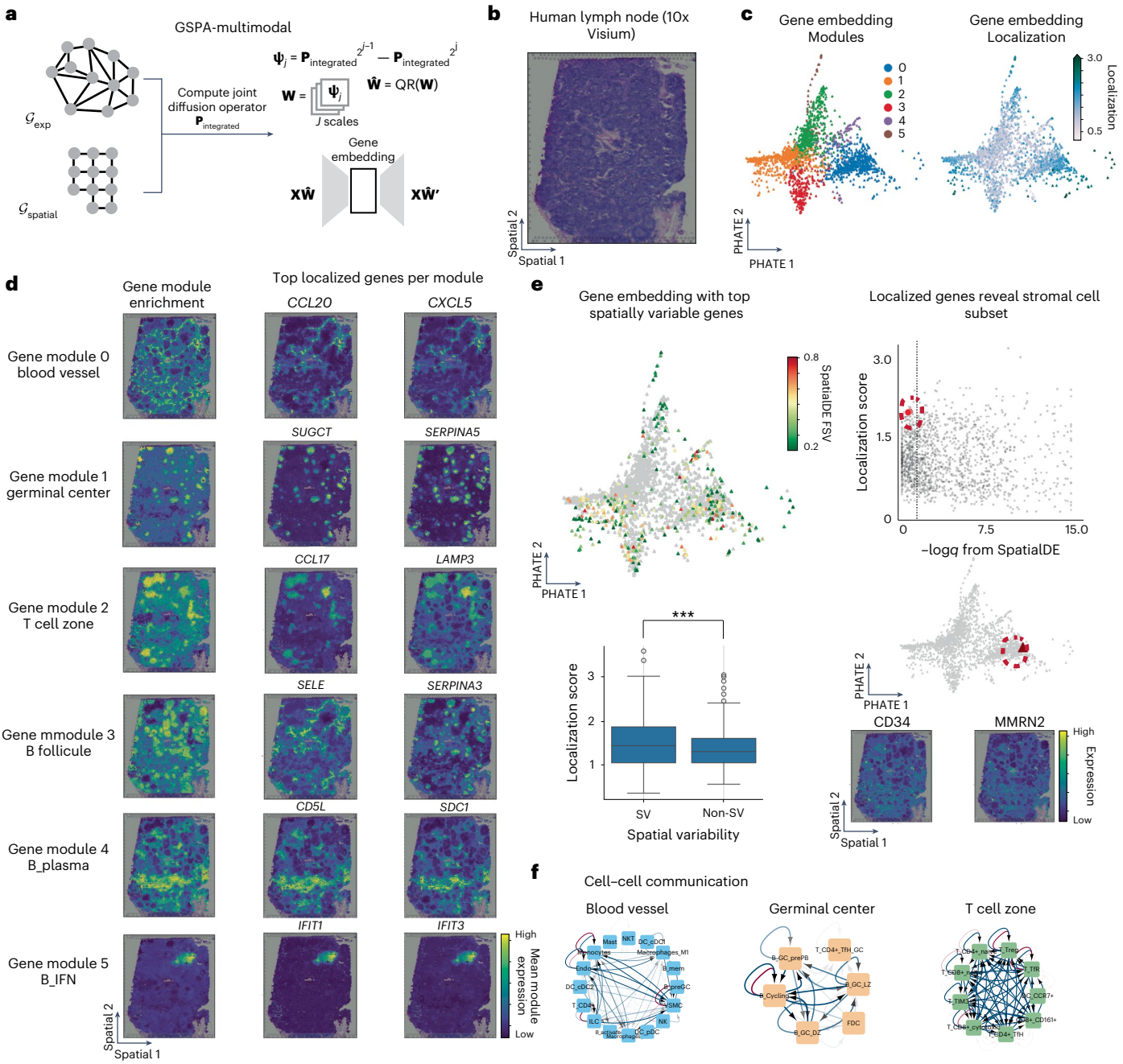


Fig. 6 | Spatially localized gene signaling and immune hubs in 10x Visium human lymph node. a, Schematic of GSPA-multimodal using integrated diffusion on spatial transcriptomic data[45,46]. **b**, Hematoxylin and eosin stain of human lymph node tissue. **c**, PHATE visualization of gene embedding, colored by gene module assignment (left) and localization score (right). **d**, Enrichment of gene modules spatially and visualization of top localized genes. **e**, Gene embedding with top spatially variable genes, where localization score corresponds with spatial variability (left) for $n = 1{,}969$ highly variable genes (one-sided Wilcoxon rank sums test, $P 9.47 \times 10^{-7}$). Localized genes that are not significant by SpatialDE reveal stromal subset (right). **f**, Cell–cell communication networks derived from gene–gene interactions with OmniPathDB.

the vasculature and previously implicated as progenitors to fibroblast subsets[48] (Fig. 6e).

To impute single-cell resolution, we leveraged a human secondary lymphoid organ atlas from ref. 49 and the cell2location[49] mapping to determine cell types enriched for each gene module, revealing spatially organized enrichment of monocytes, macrophages, mast cells, endothelial cells, follicular dendritic cells and vascular smooth muscle cells for module 0 (blood vessel); germinal center-committed B cells, T follicular helper cells and follicular dendritic cells for gene module 1 (germinal centers); non-germinal center-specific B cells for gene module 3 (B follicule); B plasma cells for gene module 4; and interferon signaling B cells for gene module 5 (Supplementary Fig. 8). Given high cell–cell interaction within each spot, we leveraged OmniPathDB[50] to capture active intercellular (blue) and intracellular (red) gene–gene interactions. We then identified cell types involved in active signaling within each module, capturing complex multicellular networks (Methods and Fig. 6f).

Together, GSPA-multimodal enables spatial analysis and demonstrates the ability of GSPA to extend beyond single-cell frameworks to represent multimodal features effectively.

Patient manifolds and outcome prediction with GSPA-Pt

Finally, we show that GSPA-Pt can be used to map patient sample manifolds. We hypothesized that constructing patient vectors from GSPA gene embeddings could enhance response prediction by capturing the cell–cell graph topology and gene co-expression. In addition, as features of the patient vector correspond to genes, we could explore genes predictive of response (Methods and Extended Data Fig. 9a).

We tested GSPA-Pt in 48 melanoma samples pre- and post-checkpoint immunotherapy[14] to interpret immunological programs of patient response. We mapped each patient sample, corresponding to a scRNA-seq dataset of $CD45^+$ cells, and classified response from the patient embedding using logistic regression.

The GSPA-Pt visualization shows separation between responders and non-responders, as do two comparisons (GSPA embeddings of patient set indicator signals on the cell–cell graph and mean expression across all cells). Patient embeddings by cell cluster proportion and $CD8^+$ subcluster proportion do not clearly distinguish responders and non-responders (Extended Data Fig. 9b). The GSPA-Pt gene embeddings achieved the highest classification performance, indicating learning multiscale features enhances response prediction (Extended Data Fig. 9c).

As patient embeddings comprise gene features, the coefficients of the logistic regressor reflect the importance of different genes for prediction (Extended Data Fig. 9d and Supplementary Table 4). Many important genes were related to T cell function, reflecting their role in tumor recognition and control[14,51]. Genes most associated with non-response include *NKG7* (rank 1), *GZMA* (rank 5) and *CD38* (rank 28), resembling known terminal differentiation programs[14,52,53]. Genes associated with response include *IL7R* (rank 3), *CCR7* (rank 4) and *TCF7* (rank 16), linked to T cell progenitor states such as stemness, memory, activation and survival[14,54], and reflecting the known role of progenitor T cell states as immunotherapy targets[55,56]. While mean expression-based embeddings showed comparable gene rankings for some markers (*NKG7* (rank 3), *IL7R* (rank 17) and *CCR7* (rank 3)), other markers are ranked lower, including *GZMA* (rank 115), *CD38* (rank 176) and *TCF7* (rank 499).

The patient manifold revealed information beyond T cell signatures. Five samples were visually distinct due to a high (>40%) proportion of B cells (Extended Data Fig. 9e). In addition, the shape of the patient manifold enables understanding of patient trajectories (Extended Data Fig. 9f). For patient 1, characterized by resistance[14], three samples were obtained (baseline, day 0, lesion classified as responder, post-therapy I biopsy, day 48, (responder), and post-therapy II biopsy, day 437, non-responder). Despite baseline and post-therapy I biopsy classification as responder, the patient manifold shows that these samples embed near non-response samples and show a trajectory toward non-response, suggesting that the tumor microenvironment resembles non-responding cells. For patient 3, determined to have *B2M* deficiency associated with resistance and less cytotoxic T cell infiltration[57], although all samples were determined to be non-responders, they embed near responders on the patient manifold and reflect changes in proportions of memory and activated $CD8^+$ T cells. Finally, two samples from patient 20, who showed mixed response, were determined as non-responsive. The patient manifold shows a trajectory, while still in the non-response region, in the direction of response, suggesting that mixed response may be through intertumoral differences in immune infiltration[58] or $CD8^+$ T cell heterogeneity. These findings collectively underscore the utility of GSPA-Pt and interpretable patient representations for analysis of clinical outcomes.

Discussion

Although there is much interest in understanding gene–gene relationships, mapping the gene space has not been sufficiently investigated and motivated. In this work, we defined baselines and designed experiments to evaluate gene embeddings, establishing the groundwork for future research in this space. We demonstrated the utility of framing genes as graph signals, and we described our method of decomposing gene signals using diffusion wavelets and learning a low-dimensional representation. We also highlighted the superior ability of this technique to capture gene–gene relationships, meaningful visualization and salient gene modules.

We introduced GSPA for ranking genes based on cell-type association, where cell types are defined, and by differential localization, where localized genes are specific to regions of the manifold without any assumptions about its shape. These rankings are broadly applicable to single-cell analysis, which traditionally relies on statistical testing of mean differences between clusters. By bridging localization and gene embeddings, GSPA allows for identification of localized genes per module, ensuring a diversity of selected features not guaranteed by other feature selection approaches[23,59].

We demonstrated the utility of gene embeddings in several vignettes. In a scRNA-seq dataset of $CD8^+$ T cells during acute and chronic infection at three timepoints, we identified key gene modules corresponding to enrichment at different points of the T cell differentiation process. We also analyzed gene–gene relationships from a single-cell CRISPR screen, highlighting co-expression events affected by *Tbx21* and *Klf2* knockouts.

Beyond cellular-state characterization, GSPA-LR performed cell–cell communication analysis without cell-type annotation in a peripheral tolerance model[12], recovering condition-specific communication between subsets of multiple cell types. GSPA-multimodal on spatial transcriptomic data identified spatially enriched gene modules and microenvironmental signaling events. GSPA-Pt mapped melanoma samples using the gene space to create interpretable representations for response prediction.

This work opens avenues for further exploration. While we identified gene and LR pair modules in our case studies, the continuous structure may be further explored with gene trajectory analysis and gene archetypal analysis. Furthermore, characterizing additional types of gene patterns against graph signals, such as cellular pseudotime or condition/perturbation labels, could yield new insights. GSPA is also flexibly defined and can be combined with supervised losses for jointly trained gene embeddings to predict treatment response or cell state.

Overall, GSPA represents a key contribution to graph feature or signal representation learning, with opportunities beyond single-cell graphs. For example, localization scores can indicate specialization or commitment to a particular group of nodes in diverse settings. We thus expect future work to benefit from GSPA to extract feature representations from large-scale high-dimensional data.

Methods

Manifold learning and diffusion geometry

The manifold hypothesis is the belief that many high-dimensional datasets have a low-dimensional intrinsic structure. More specifically, given a dataset $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^m$ of high-dimensional observations, the manifold hypothesis assumes that the datapoints lie on a d-dimensional manifold (that is, a d-dimensional subset of $\mathbb{R}^m$ that is locally equivalent to $\mathbb{R}^d$) for some $d \ll m$. For instance, in high-dimensional, single-cell experiments, each cell is described by a vector of counts per feature, where the number of features range from tens of key markers to thousands of genes. The measured cells make up the 'cellular-state space', representing different possible cell states defined by the experimental set-up. Redundancy in these measurements and high levels of coordination between the genes puts constraints on cellular behavior and reduces the effective degree. Thus, arbitrary points in $\mathbb{R}^m$ do not correspond to plausible cellular states. Instead, the set of plausible states can be modeled as lying on a lower-dimensional manifold (see ref. 16 for further discussion of modeling single-cell data on a manifold).

Many popular manifold learning methods are based upon constructing a graph $\mathcal{G} = (V, E)$, whose vertices are the datapoints x_i, which serves as a discrete approximation of the (unknown) underlying manifold. Weighted edges are constructed using a kernel K such as the Gaussian kernel

$$K_\epsilon(x_i, x_j) = \exp(-\| x_i - x_j \|^2/\epsilon),$$

and one then defines a weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ by $\mathbf{A}_{i,j} = K_\epsilon(x_i, x_j)$. The parameter ϵ, sometimes referred to as the bandwidth, controls the scale of the kernel, based on the idea that Euclidean distances $\|x_i - x_j\|$ are equivalent to manifold distances (lengths of shortest path along the manifold) at small scales, but not at larger scales. Thus, ϵ should be chosen to be sufficiently small so that the kernel only gives high weight to intrinsically meaningful connections, but not so small that the graph becomes disconnected.

Given $\mathbf{A}$, one can then define the diffusion operator $\mathbf{P} = \mathbf{A}\mathbf{D}^{-1}$, where $\mathbf{D}$ is the diagonal degree matrix ($\mathbf{D}_{i,i} = \sum_j \mathbf{A}_{i,j}$, $\mathbf{D}_{i,j} = 0$ if $i \neq j$). This matrix $\mathbf{P}$ describes the transition probabilities of a lazy random walker exploring the vertices of the graph one step at a time. It can also be interpreted as an operator to diffuse the values of signal $\mathbf{x}$ based on the values at neighboring points. Using higher powers of $\mathbf{P}$, that is, $\mathbf{P}^t\mathbf{x}$, can be seen as averaging $\mathbf{x}$ over t-step random walks.

On the basis of these observations, Coifman and Lafon introduced diffusion maps[7], a method for embedding the datapoints into a low-dimensional Euclidean space $\mathbb{R}^d$, $d \ll m$, parameterized by a diffusion-time parameter, which controls the scale (level of locality) of the representation. Formally, we take the eigenvalues $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ and (corresponding) eigenvectors $\{\phi_j\}_{j=1}^N$ of $\mathbf{P}$ and map each point $x_i \in X$ to a d-dimensional vector

$$\Phi_t(x_i) = [\lambda_1^t\phi_1(x_i), \ldots, \lambda_N^t\phi_d(x_i)]^\top, \tag{1}$$

where t represents a diffusion time (note that the λ_i^t are the eigenvalues of the powered diffusion operator $\mathbf{P}^t$). Small choices of t lead to more localized, small-scale representations, whereas larger values of t lead to larger-scale, more global representations. However, in the latter case, we emphasize that these representations are larger scale in the sense that they consider long-range dependencies between points on the manifold as measured by lengths of shortest paths, which are not the same as Euclidean distances.

Given the representations $\{\Phi_t(x_i)\}_{i=1}^N \subseteq \mathbb{R}^d$, one may then analyze the data in reduced-dimension space, rather than the original high-dimensional ambient space. Indeed, using diffusion maps and related methods to model single-cell data on a manifold provides insight into key geometric and topological features of the cell–cell affinity graph, enabling characterization of the cellular-state space despite sparsity, artifacts and complex nonlinear relationships. As such, it has become a key tool in single-cell analysis for cell-type identification, imputation[25], visualization[2,3] and multiscale analysis[19]. Below, we highlight two methods related to diffusion maps, MAGIC and diffusion wavelets.

MAGIC[25] is a denoising method based on the above-mentioned diffusion operator. It is based on the intuition the many signals measured on, for example, cell–cell graphs have extremely high noise levels, but that the noise typically shows high-frequency behavior. It is known that the $\{\phi_i\}_{i=1}^N$ form a basis for $\mathbb{R}^N$ and therefore any arbitrary graph signal $\mathbf{x}$ can be decomposed by

$$\mathbf{x} = \sum_i c_i\phi_i,$$

where the coefficient c_i represents how much of $\mathbf{x}$ lies at frequency λ_i. One may verify that applying the diffusion operator yields

$$\mathbf{P}^t\mathbf{x} = \sum_i \lambda_i^t c_i\phi_i.$$

In addition, we note that the eigenvectors are ordered in terms of smoothness/frequency with ϕ_1 being the most smooth and each ϕ_k being smoother than ϕ_{k+1}. Here, smoothness is quantified by

$$\sum_{i,j} \frac{\mathbf{A}_{i,j}|(\phi_k)_i - (\phi_k)_j|^2}{\mathbf{D}_{i,i}}. \tag{2}$$

so that ϕ_k is smooth if $(\phi_k)_i \approx (\phi_k)_j$ whenever there is a large edge-weight $\mathbf{A}_{i,j}$. In analogy to traditional Fourier analysis, we refer to the smooth eigenvectors, which vary slowly over the graph, as 'low frequency' and refer to the eigenvectors that vary rapidly over the graph as high frequency.

Recalling that $1 = \lambda_1 > \lambda_2 > \lambda_3 \ldots$, we see that $\mathbf{P}^t$ can be interpreted as a 'low-pass filter' which preserves the low-frequency portion of $\mathbf{x}$ (corresponding to the larger eigenvalues, that is, smaller values of k) and suppressing the higher-frequency portion of the signal, which is typically noisy. This is the basis for the the MAGIC algorithm, which replaces the initial data matrix X (whose columns are the datapoints x_i) with a denoised data matrix defined by

$$X_{\text{denoised}} = \mathbf{P}^t X. \tag{3}$$

Diffusion wavelets[9] aims to build on the diffusion-maps approach to produce a multiscale data representation. The diffusion map, defined as in equation (1), is based on the eigendecomposition of the powered diffusion operator $\mathbf{P}^t$ (as $\mathbf{P}^t\phi_k = \lambda_k^t\phi_k$). The diffusion-time parameter, t, can be thought of as representing the scale of the representation since $\mathbf{P}^t$ performs a t-step random walk over the graph. To achieve a multiscale representation of the data, one can leverage multiple different values of t. This, along with classical wavelet constructions for data such as images (see, for example, ref. 8), inspired Coifman and Maggioni to introduce diffusion wavelets of the form

$$\mathbf{\Psi}_j = \mathbf{P}^{2^{j-1}} - \mathbf{P}^{2^j}.$$

As the diffusion-time parameter t represents the data scale, these $\mathbf{\Psi}_j$ model changes in the data across multiple scales and thereby produce a multiscale data representation. Since their introduction, these wavelets have played a powerful role in graph signal processing[6] and have also been used to construct neural networks for graphs, manifolds and other forms of geometric data[60,61]. In addition, we note that the choice of dyadic scales 2^j was initially inherited from traditional Euclidean wavelets.

GSPA detailed overview

Gene embedding problem set-up. We assume that we are given a single-cell sequencing dataset consisting of m genes and their measurements in n cells, organized into an $m \times n$ matrix $\mathbf{X}$. Here the measured cells (columns of $\mathbf{X}$) can be viewed as a sampling of possible cell states represented in the experiment and together make up the 'cellular-state space'. Despite the apparent high-dimensionality of this state space, cells are often modeled on an underlying manifold[16]. Manifold learning approaches have enabled myriad downstream tasks in single-cell analysis by modeling the constrained cellular state as having comparatively low intrinsic dimension. The majority of these approaches first build a cell–cell similarity graph $\mathcal{G}_{\text{cell}}$, where the vertices correspond to cells and edges correspond to cells with similar gene profiles. This graph represents a discretization of the cellular manifold.

Thus, with the insight that gene measurements may also be compressed into a lower-dimensional space for analyzing gene–gene relationships, our goal is to obtain a low-dimensional representation of each gene that preserves the inherent structure of the gene space with respect to the cellular manifold. In particular, we seek a reduced dimensional map $\Theta : \mathbb{R}^n \to \mathbb{R}^d$, $d \ll n$, which satisfies the desired properties enumerated below.

Desired properties.

(1) Preserving local and global distances between signals. A good gene embedding should produce similar representations of genes $\mathbf{X}_{i_1}$ and $\mathbf{X}_{i_2}$ (viewed as rows of $\mathbf{X}$) if they have similar measurement profiles. To ensure that we capture meaningful information, we aim to preserve distances based on the geometry of the underlying cell–cell graph $\mathcal{G}_{\text{cell}}$, rather than the naive pointwise distance, between gene signals.
(2) Noise robustness. Addressing biological noise, such as cell-to-cell variation, and technical noise, such as dropout, have been longstanding concerns in single-cell analysis and best practices[4,5,25]. Due to variability in noise between genes with different expression levels[4], noise robustness is especially relevant for constructing gene embeddings. We thus seek a representation Θ such that $\| \Theta(\mathbf{X}_{i_1}) - \Theta(\mathbf{X}_{i_2})\|_2 \approx \| \Theta(\mathbf{X}_{i_1} + \epsilon_{i_1}) - \Theta(\mathbf{X}_{i_2} + \epsilon_{i_2})\|_2$ where ϵ_{i_1} and ϵ_{i_2} are measurement noise associated with genes $\mathbf{X}_{i_1}$ and $\mathbf{X}_{i_2}$.
(3) Flexibility to downstream tasks. Finally, we want to ensure our embedding Θ is flexibly defined for various additional tasks, whether concurrently trained with the learned embedding or downstream of the embedding.

In the following section, we present our approach, GSPA, to achieve these desired properties.

Model overview. To construct the map Θ, we make the critical observation that the expression pattern $\mathbf{X}_i$ for gene i can be described as a signal (function) defined on the nodes of a cell–cell similarity graph $\mathcal{G}_{\text{cell}}$. Through this framing, we can compare how gene-expression patterns are similar to and different from each other based on distances along the cellular manifold.

We first model random walks on the cell–cell similarity graph with a diffusion operator $\mathbf{P}$ that contains transition probabilities between cells on the basis of their similarity. We use this operator to construct a dictionary of diffusion wavelets for decomposing gene signals on the cell similarity matrix. To construct these wavelets, we power $\mathbf{P}$ to different values of t to capture local (small t) and global (large t) representations. Each diffusion wavelet is then defined by the difference of these powered diffusion operators[9].

We then represent gene signals with respect to these local and global geometric structures. To this end, we decompose each signal using the wavelet dictionary, encoding the topology of the nodes (cells) it is defined on in addition to the signal (gene) itself. Finally, we learn a meaningful low-dimensional embedding using an autoencoder for reducing the wavelet-based representations (Algorithm 1). This reduced embedding can be used for many downstream analyses including visualization, gene module identification or differential localization.

Constructing a cell–cell similarity graph from single-cell data. In scRNA-seq, each cell is measured as a vector of gene-expression counts. That is, the output of a scRNA-seq experiment is a matrix $\mathbf{X}$ of shape $m \times n$, where m corresponds to the number of genes, and n the number of cells. The first step of the GSPA algorithm is to build a graph $\mathcal{G}_{\text{cell}} = (V_{\text{cell}}, E_{\text{cell}})$, where each node in V_{cell} corresponds to a cell, and each edge $E_{\upsilon_1\upsilon_2}$ in E_{cell} describes the similarity between cell υ_1 and cell υ_2.

To build the graph $\mathcal{G}_{\text{cell}} = (V_{\text{cell}}, E_{\text{cell}})$, we compute the Euclidean distances between all pairs of cells and apply an α-decaying kernel to calculate affinities[3,62]. The α-decaying kernel is defined as

$$K_{k,\alpha}(\upsilon_1, \upsilon_2) = \frac{1}{2}\exp\left(-\left(\frac{\| \upsilon_1 - \upsilon_2\|_2}{\varepsilon_k(\upsilon_1)}\right)^{\alpha}\right) + \frac{1}{2}\exp\left(-\left(\frac{\| \upsilon_1 - \upsilon_2\|_2}{\varepsilon_k(\upsilon_2)}\right)^{\alpha}\right), \quad (4)$$

where υ_1 and υ_2 are cells $\in V_{\text{cell}}$, viewed as points in $\mathbb{R}^m$ corresponding to columns of $\mathbf{X}$, $\varepsilon_k(\upsilon_1)$ and $\varepsilon_k(\upsilon_2)$ are the distance from υ_1 and υ_2 to their kth nearest neighbors (kNNs), respectively, and α is a parameter that controls the decay rate (that is, heaviness of the tails) of the kernel. This construction generalizes the popular Gaussian kernel typically used in manifold learning while addressing some of its key limitations, as explained in ref. 16.

This defines a fully connected and weighted graph between cells such that the connection between cells υ_1 and υ_2 is given by $K(\upsilon_1, \upsilon_2)$. To increase the computational efficiency, we sparsify the graph by setting very small edge weights (that is, $\leq 1 \times 10^{-4}$) to 0. In addition, $\mathbf{A}$ is defined as the adjacency/affinity matrix of $\mathcal{G}_{\text{cell}}$, or binarized K, and $\mathbf{D}$ is the diagonal degree matrix of $\mathcal{G}_{\text{cell}}$.

Owing to the flexibility of graph construction, GSPA can handle a variety of use cases. Graphs can be constructed with affinities derived from multiple modalities, then used with GSPA for integrated gene analysis. Moreover, graphs can consist of cells from multiple sequencing runs. In such cases, downstream analysis is often affected by batch effects, in which expression of genes systematically differs between batches, resulting in cellular affinities defined by batch rather than cell type or other underlying biology. Batch effect affects all downstream analysis, including gene embeddings derived from our approach, where genes separate by enrichment within a particular batch (Extended Data Fig. 1a). GSPA handles batch effect through either accepting batch-corrected cell measurements or cell embeddings as input, or correcting for batch in the cell–cell graph through MNN graph construction[17]. This results in gene embeddings corrected for differences in batch (Extended Data Fig. 1b). For large graphs, GSPA utilizes diffusion condensation, a coarse-graining process that iteratively condenses datapoints toward local centers of gravity[18,19]. This technique allows GSPA to summarize the underlying topology of the data manifold in a smaller coarse-grained cell–cell graph for improved scalability (Supplementary Fig. 1a). For more discussion on such cases for graph construction, see 'GSPA for multiple modalities, datasets and large graphs'.

Building dictionary of graph diffusion wavelets for gene representation. Like many other tools from signal processing, wavelet analysis can naturally be extended to graphs and manifolds. In classical signal processing, for example, the analysis of temporal data, a wavelet dictionary is defined by taking a function ψ and a set of transformations of this function by time and scale. To adapt these methods to graphs, where there is no concept of time or linear space, we center wavelets at vertices on the graph and change scales via diffusion. Given the diffusion operator $\mathbf{P}$ (renormalized to have largest eigenvalue 1), we follow the construction of diffusion wavelets in ref. 9. Using ideas related to ref. 63, we use $\mathbf{P}$ to induce a multiresolution analysis, interpreting powers of $\mathbf{P}$ as dilation operators acting on functions, and constructing downsampling operators to efficiently represent the graph at fine-grained and coarse-grained resolutions.

Given the cellular graph $\mathcal{G}_{\text{cell}}$, we define $\mathbf{P} = \mathbf{A}\mathbf{D}^{-1}$ as the diffusion operator. Each wavelet of scale j centered at vertex υ can thus be calculated by computing the matrix $\mathbf{\Psi}_j = \mathbf{P}^{2^{j-1}} - \mathbf{P}^{2^j}$ for $1 \leq j \leq J$ (and $\mathbf{\Psi}_0 = \mathbf{I} - \mathbf{P}$, where $\mathbf{I}$ is the identity matrix), and then extracting the υth row via $\delta_\upsilon^\top \mathbf{\Psi}_j$, where δ_υ is the Kronecker delta centered at the υth vertex. Then $\{\mathbf{\Psi}_j^\top \delta_\upsilon\}_{\upsilon \in V_{\text{cell}}, j \in 0,1\ldots J}$ defines our wavelet dictionary $\mathbf{W}$ (where we use $(\delta_\upsilon^\top \mathbf{\Psi}_j)^\top = \mathbf{\Psi}_j^\top \delta_\upsilon$ because our projection $\mathbf{X} \to \mathbf{X}\mathbf{W}$ is performed via multiplication on the right). $\mathbf{W}$ is an $n \times Jn$ matrix (every wavelet takes values on the whole vertex set, and we have a wavelet for every vertex at each scale).

The number of scales for the wavelet dictionary J is defined as the log of the number of cells n based on the following lemma introduced in ref. 20 and proven in the original work.

Lemma 1. There exists a $K = O(\log|V|)$ *such that* $\mu_i^{(2^K)} \simeq \phi_0$ *for density estimate* μ *at every* $i = 1, \ldots, n$, *where* ϕ_0 *is the trivial eigenvector of* $\mathbf{P}$ *associated with the eigenvalue* $\lambda_0 = 1$.

This is based on the reasoning that if the Markov process converges in polynomial time with respect the number of nodes $|V|$, then one can ensure that beyond $O(\log|V|)$, all density estimates would be indistinguishable from each other.

Because the diffusion operator $\mathbf{P}$ is smoothing, we make the assumption that the numerical rank decreases as we take powers of the operator[9]. The faster the spectrum of $\mathbf{P}$ decays, the smaller the numerical rank of the powers of $\mathbf{P}$, the more these can be compressed, and the faster the construction of $\mathbf{\Psi}_j$ for large j. Therefore, to decrease the size of the wavelet dictionary, remove redundant wavelets and increase their interpretability, we can perform QR factorization. This results in a compressed wavelet dictionary $\hat{\mathbf{W}} = \{\tilde{\mathbf{\Psi}}_j^{\top}\delta_v\}_{v\in V_{\text{cell}}, j\in 0,1\ldots J}$, where for each j, $\tilde{\mathbf{\Psi}}_j$ is a set of linear combinations of wavelets at j that account for the most variance. For large j, QR factorization naturally computes the numerical rank of $\mathbf{\Psi}_j$ by taking a linear combination to form $\tilde{\mathbf{\Psi}}_j$ such that the total error in projecting $\mathbf{\Psi}_j$ onto $\tilde{\mathbf{\Psi}}_j$ is less than some ϵ fraction of the norm of the whole $\mathbf{\Psi}_j$. That is

$$\frac{1}{||\mathbf{\Psi}_j||^2}||\mathbf{\Psi}_j - \tilde{\mathbf{\Psi}}_j(\tilde{\mathbf{\Psi}}_j^{\top}\tilde{\mathbf{\Psi}}_j)^{-1}\tilde{\mathbf{\Psi}}_j^{\top}\mathbf{\Psi}_j||^2 < \epsilon \tag{5}$$

As raising $\mathbf{P}$ to the power of t diminishes the higher-frequency eigenvalues, diffusion wavelets support noise robustness and continuity properties by ensuring that $\mathbf{\Psi}_j\delta_{v_1} \approx \mathbf{\Psi}_j\delta_{v_2}$ if vertex v_1 is near vertex v_2 on the graph. We test gene signal embeddings both with (GSPA+QR) and without (GSPA) QR factorization.

Projecting gene signals onto the wavelet dictionary. Each gene signal $\mathbf{X}_i$ of shape $1\times n$ corresponds to the expression of the gene in the cellular-state space. Importantly, we consider each measured gene feature as a signal on the cellular graph. Given a gene signal $\mathbf{X}_i$ and wavelet dictionary $\hat{\mathbf{W}}$ (or $\mathbf{W}$), we project $\mathbf{X}_i$ onto the dictionary, which, for all gene signals, corresponds to $\mathbf{X}\hat{\mathbf{W}}$ or $\mathbf{XW}$. Alternatively, the solution can be framed as a set of m inner products provided by $\hat{\mathbf{W}}\mathbf{X}_i^{\top}$ or $\mathbf{W}\mathbf{X}_i^{\top}$ as a transformation of $\mathbf{X}_i$, or $\hat{\mathbf{W}}^{\top}\mathbf{X}^{\top}$ or $\mathbf{W}^{\top}\mathbf{X}^{\top}$ for all gene signals. This transformation reveals each gene signal's spatial and frequency information over the corresponding cell–cell graph $\mathcal{G}_{\text{cell}}$. Theorem 1 stated below shows that the wavelet projection $\mathbf{X}\to\mathbf{XW}$ is continuous with respect to a UDEMD[21]. Intuitively, this UDEMD describes how similar two gene signals $\mathbf{X}_{i_1}$ and $\mathbf{X}_{i_2}$ are in a manner informed by the geometry of the cellular graph.

Learning a low-dimensional representation of gene signal projections with an autoencoder. $\mathbf{X}\hat{\mathbf{W}}$ is unnecessarily high-dimensional, where each feature corresponds to the gene signal projection of a particular cell at a particular diffusion scale. To reduce redundancy and improve computational tractability for downstream analysis, we reduce the dimensionality with autoencoder $D\circ E$ where the objective is to minimize the mean squared error. That is

$$\mathbf{X}\hat{\mathbf{W}}' \approx D(E(\mathbf{X}\hat{\mathbf{W}})), \tag{6}$$

so that $\|\mathbf{X}\hat{\mathbf{W}} - \mathbf{X}\hat{\mathbf{W}}'\|_2^2 = \sum_{i\in\mathbf{X}}\|\mathbf{X}_i\hat{\mathbf{W}} - \mathbf{X}_i\hat{\mathbf{W}}'\|_2^2$ is as small as possible. The latent representation $E(\mathbf{X}\hat{\mathbf{W}})$ is the embedding we evaluate and characterize in downstream analysis, that is

$$\text{GSPA}(\mathbf{X}) = E(\mathbf{XW}) \qquad \text{GSPA+QR}(\mathbf{X}) = E(\mathbf{X}\hat{\mathbf{W}})$$

where $\mathbf{W}$ and $\hat{\mathbf{W}}$ are uncompressed and compressed wavelet dictionaries (respectively), E is the encoder discussed above, and GSPA and GSPA+QR are taken to be the map Θ discussed in the problem set-up.

Achieving desired representation properties with GSPA. *Distance preservation*. Our first desired property is an embedding that preserves distances (quantified in a manner informed by the geometry of the cellular-state space). Theorem 1 shows that GSPA is able to achieve this goal as it guarantees we will have $\text{GSPA}(\mathbf{X}_i)\approx\text{GSPA}(\mathbf{X}_j)$ whenever $\mathbf{X}_i$ is close to $\mathbf{X}_j$ with respect to the UDEMD. This distance, a variant of traditional earth mover's distance (EMD), views the signals $\mathbf{X}_i$ and $\mathbf{X}_j$ (when properly normalized) as probability distributions on the graph.

EMDs, alternatively referred to as Monge–Kantorovich or Wasserstein distances, are a useful way of computing the distances between two signals. In the case where the signals correspond to probability distributions μ and ν, these distances can be thought of as the 'cost' of moving a collection of points distributed according to μ to a collection of points distributed according to ν, where the cost of moving each point depends on the distance it must travel (defined with respect to some ground distance). In ref. 21 (see also ref. 20), it was shown that the Wasserstein distance (with a truncated geodesic distance as ground distance) could be approximated by the UDEMD defined below. Here we show that the metric induced by our wavelets is continuous with respect to this UDEMD, that is, $\|\mathbf{X}_{i_1}\mathbf{W} - \mathbf{X}_{i_2}\mathbf{W}\|_2 \lesssim \text{UDEMD}(\mathbf{X}_{i_1}, \mathbf{X}_{i_2})$.

In ref. 21, the UDEMD[21] between two signals (genes) $\mathbf{X}_{i_1}, \mathbf{X}_{i_2}$ is defined as

$$\text{UDEMD}_{\beta,J}(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}) := \sum_{j=0}^{J}\left\|T_{\beta,k}(\mathbf{X}_{i_1}) - T_{\beta,k}(\mathbf{X}_{i_2})\right\|_1, \tag{7}$$

where $0<\beta<1/2$ is a meta-parameter used to balance long- and short-range distances and J is the maximum scale considered here, and $T_{\beta,j}$ is defined by

$$T_{\beta,j}(\mathbf{X}_i) := 2^{-(J-j)\beta}\left(\boldsymbol{\mu}_i^{(2^j)} - \boldsymbol{\mu}_i^{(2^{j-1})}\right), \quad \boldsymbol{\mu}_i^{(t)} := \mathbf{P}^t\mathbf{X}_i, \tag{8}$$

for $1\le j\le J$ and $T_{\beta,0}(\mathbf{X}_i) = 2^{-J\beta}(\mathbf{P}-\mathbf{I})\mathbf{X}_i$.

Theorem 1. For $0<\beta<1/2$, the diffusion wavelet transform $\mathbf{W}$ (with maximal scale J) is Lipschitz continuous with respect to $\text{UDEMD}_{\beta,J}$, that is, there exists a constant $C>0$ (depending on β and J and the ratio between the largest and smallest vertex degrees) such that

$$\|\mathbf{X}_{i_1}\mathbf{W} - \mathbf{X}_{i_2}\mathbf{W}\|_2 \le C\times\text{UDEMD}_{\beta,J}(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}). \tag{9}$$

Proof. Let $\mathbf{X}_{i_1}\neq\mathbf{X}_{i_2}$. (The inequality holds trivially in the case where $\mathbf{X}_{i_1} = \mathbf{X}_{i_2}$.) We may compute

$$\begin{aligned}
\|\mathbf{X}_{i_1}\mathbf{W} - \mathbf{X}_{i_2}\mathbf{W}\|_2^2 &= \|\mathbf{W}^{\top}\mathbf{X}_{i_1}^{\top} - \mathbf{W}^{\top}\mathbf{X}_{i_2}^{\top}\|_2^2\\
&= \sum_{v=1}^{n}\sum_{j=0}^{J}|\delta_v^{\top}\mathbf{\Psi}_j\mathbf{X}_{i_1}^{\top} - \delta_v^{\top}\mathbf{\Psi}_j\mathbf{X}_{i_2}^{\top}|^2\\
&= \sum_{j=0}^{J}\|\mathbf{\Psi}_j\mathbf{X}_{i_1}^{\top} - \mathbf{\Psi}_j\mathbf{X}_{i_2}^{\top}\|_2^2\\
&\le \sum_{j=0}^{J}\|\mathbf{\Psi}_j\mathbf{X}_{i_1}^{\top} - \mathbf{\Psi}_j\mathbf{X}_{i_2}^{\top}\|_2\,\|\mathbf{\Psi}_j\mathbf{X}_{i_1}^{\top} - \mathbf{\Psi}_j\mathbf{X}_{i_2}^{\top}\|_1\\
&\le C\max_{0\le j\le J}\|\mathbf{\Psi}_j\mathbf{X}_{i_1}^{\top} - \mathbf{\Psi}_j\mathbf{X}_{i_2}^{\top}\|_2\sum_{j=0}^{J}2^{-(J-j)\beta}\|\mathbf{\Psi}_j\mathbf{X}_{i_1}^{\top} - \mathbf{\Psi}_j\mathbf{X}_{i_2}^{\top}\|_1\\
&= C\max_{0\le j\le J}\|\mathbf{\Psi}_j\mathbf{X}_{i_1}^{\top} - \mathbf{\Psi}_j\mathbf{X}_{i_2}^{\top}\|_2\,\text{UDEMD}(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}),
\end{aligned}$$

where C is a constant depending on J and β. It follows from Proposition 2.2 of ref. 60 that

$$\|\mathbf{\Psi}_j\mathbf{X}_{i_1}^{\top} - \mathbf{\Psi}_j\mathbf{X}_{i_2}^{\top}\|_2 \le C\,\|\mathbf{X}_{i_1}^{\top} - \mathbf{X}_{i_2}^{\top}\|_2,$$

where C is a constant depending on only the ratio between the maximal vertex degree and minimal vertex degree. (Reference 60 considers the wavelets on a weighted inner product space where vertices are

weighted by degree. Transferring this result to the unweighted ℓ^2 space induces dependence on the ratio between the maximal and minimal degrees.) Therefore, we have

$$\| \mathbf{X}_{i_1}\mathbf{W} - \mathbf{X}_{i_2}\mathbf{W}\|_2^2 \leq C \times \mathrm{UDEMD}\,(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}) \,\| \mathbf{X}_{i_1} - \mathbf{X}_{i_2}\|_2,$$

which in turn implies

$$\| \mathbf{X}_{i_1}\mathbf{W} - \mathbf{X}_{i_2}\mathbf{W}\|_2 \leq C \times \mathrm{UDEMD}\,(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}) \frac{\| \mathbf{X}_{i_1} - \mathbf{X}_{i_2}\|_2}{\| \mathbf{X}_{i_1}\mathbf{W} - \mathbf{X}_{i_2}\mathbf{W}\|_2}.$$

The lower bound in Proposition 2.2 of ref. 60 implies that $\frac{\|\mathbf{X}_{i_1}^T - \mathbf{X}_{i_2}^T\|_2}{\|\mathbf{X}_{i_1}\mathbf{W} - \mathbf{X}_{i_2}\mathbf{W}\|_2}$ is bounded above by a constant (depending on the ratio between the maximal and minimal vertex degrees). Therefore, we have

$$\| \mathbf{X}_{i_1}\mathbf{W} - \mathbf{X}_{i_2}\mathbf{W}\|_2 \leq C \times \mathrm{UDEMD}\,(\mathbf{X}_{i_1}, \mathbf{X}_{i_2})$$

as desired.

Noise robustness. Robustness to biological and technical noise is a key feature of diffusion-based single-cell analysis approaches[16]. Note that raising the diffusion operator $\mathbf{P}$ to the power t is equivalent to powering the eigenvalues of the diffusion operator by t, that is, $\mathbf{P} = \mathbf{\Sigma}\mathbf{\Lambda}\mathbf{\Sigma}^{-1}$, where the columns of $\mathbf{\Sigma}$ contain the (right) eigenvectors of $\mathbf{P}$ and $\mathbf{\Lambda}$ is a diagonal matrix whose entries are the corresponding eigenvalues. Thus, $\mathbf{P}^t = \mathbf{\Sigma}\mathbf{\Lambda}^t\mathbf{\Sigma}^{-1}$ and powering $\mathbf{P}$ effectively results in powering the eigenvalues contained in $\mathbf{\Lambda}$. The eigenvectors are decreasingly ordered by their 'frequency', a notion of how rapidly a signal oscillates over the graph. It is known that $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots$. Therefore, powering $\mathbf{P}$ preserves the lead eigenspace and suppresses the subsequent spaces by a factor of λ_i^t. Acting on a signal $\mathbf{X}_i$ by $\mathbf{P}^t$ preserves the portion of the signal aligned with the first eigenvector and depresses the portion of the signal corresponding to the other eigenvectors by a factor of λ_i^t. As t increases, the high-frequency (small eigenvalue) portion of the signal is suppressed. Naturally occurring signals tend to vary slowly and smoothly over the graph (and thus lie in the low-frequency eigenspaces), whereas noise is not related to the structure of the graph and therefore will often lie in the higher frequencies. In this manner, acting on the signal $\mathbf{X}_i$ by $\mathbf{P}^t$ has a denoising effect as it suppresses the high-frequency (noisy) portion of the signal. Therefore, we can restrict the dictionary to wavelets that decompose only the lower frequencies by initially multiplying each wavelet by $\mathbf{P}^t$. In addition, we note that the distance preservation result in Theorem 1 shows that the wavelet projection is continuous with respect to the UDEMD, which may be viewed as a form of noise robustness.

Flexibility to downstream tasks. We demonstrate flexibility through learning a low-dimensional representation generalizable for diverse downstream tasks, as represented in our case studies.

GSPA for multiple modalities, datasets and large graphs

GSPA-multimodal. The approach described in 'GSPA detailed overview' to construct cell–cell similarity graphs is useful for graphs derived from a single scRNA-seq dataset. However, in cases where we have datasets of the same datapoints with multiple modalities, we can construct a combined representation using integrated diffusion[45]. GSPA-multimodal accepts two or more modalities and constructs affinity graphs for each modality (for example, for two modalities, $\mathcal{G}_1$ and $\mathcal{G}_2$). Then, each graph has associated diffusion filters $\mathbf{P}_1^{t_1}$ and $\mathbf{P}_2^{t_2}$, where t_1 may not equal t_2 due to differing degrees of noise, and are thus calculated by spectral entropy[45]. Finally, the integrated diffusion operator is calculated by multiplying diffusion filters, that is $\mathbf{P}_{\text{integrated}} = \mathbf{P}_1^{t_1}\mathbf{P}_2^{t_2}$. The integrated diffusion operator allows us to construct an integrated wavelet dictionary and project gene-expression signals onto this dictionary for downstream analysis. Through the ability to flexibly define cell–cell affinity, GSPA-multimodal enables analysis of multimodal datasets including and beyond scRNA-seq.

GSPA for multiple datasets. For datasets consisting of cells from multiple datasets, GSPA can be used straightforwardly by concatenating all the datasets and constructing a graph of all the cells together (for example, as done in 'Results'). However, due to cells being sequenced in multiple runs, single-cell analysis is often affected by batch effects, where gene expression systematically differs between batches and confounds analysis of cell–cell variation. GSPA is also affected by batch effect if it is not corrected, where genes will separate based on association with batch in a way that confounds true gene–gene similarity. We show this with a simulated dataset generated with three (ground truth) clusters and two batches with batch effect (Extended Data Fig. 1a). Gene embeddings learned with GSPA+QR show clear separation of genes enriched in each cluster through ground-truth differential expression factor scores per cluster. However, these gene embeddings show clear separation of genes associated with each batch as well through coloring of ground-truth batch-effect factor scores per batch.

There exist a large number of batch-correction methods, generally falling into three categories: integration in the gene space (where the output is batch-corrected gene-expression measurements), the embedding space (where the output is batch-corrected cell embedding dimensions) or between graphs (where the output is a batch-corrected combined cell–cell graph). For batch correction in the gene space or the embedding space, the integrated output can be used to construct a cell–cell affinity graph and perform GSPA as above. For batch correction between graphs, the integrated graph can be used by GSPA, where we generate a wavelet dictionary based on the integrated graph. For example, we can use an MNN approach, based on the success of MNN-based correction introduced in ref. 17, to construct a batch-corrected graph. In our simulated example, the integrated graph clearly shows no more separation of batches (Extended Data Fig. 1b). GSPA+QR with this integrated graph still shows separation of genes based on enrichment per cluster, but now no more separation based on batch. We have implemented an extension of MNN-based correction to the adaptive decay kernel[62] to perform batch correction directly with GSPA.

GSPA for large graphs. For large graphs, GSPA utilizes diffusion condensation, a coarse-graining process that iteratively condenses datapoints toward local centers of gravity and is shown to approximate heat diffusion over the time-varying manifold[18]. Over the condensation time, the original coordinate functions are smoothed by a cascade of diffusion operators, which adaptively removes high-frequency variations. At each iteration, points closer than a given threshold ζ collapse to the same barycenter. This technique allows GSPA to summarize the underlying topology of the data manifold. We use a version of diffusion condensation designed for single-cell analysis, Multiscale PHATE[19], which uses the potential representation of datapoints from PHATE[3] as the initial features.

For graphs larger than threshold n_{condense}, we use multiscale PHATE to iteratively condense datapoints to a small number of nodes. GSPA then filters for iterations with n_{condense} or fewer nodes, where each node represents a condensation of one or more cells. Finally, GSPA selects the iteration with a node count closest to n_{condense} to balance coarse- and fine-grained information. This represents a smaller cell–cell graph representing the same underlying manifold as the initial (larger) dataset, and GSPA computes a wavelet dictionary based on this graph. Then, gene signals are defined on the nodes of the condensed graph as the mean expression of all the cells in each node. By default, $n_{\text{condense}} = 10{,}000$ cells. Owing to the smaller size of the graph, computation becomes much more tractable (100,000 cells in 33.17 min and 30.18 min with GSPA and GSPA+QR, respectively) with comparable

results, where pairwise distances between genes from exact versus GSPA showed high correlation ($R = 0.900$ for GSPA and $R = 0.713$ for GSPA+QR; Supplementary Fig. 1).

Computation of cell-type association

Beyond preserving relationships within the gene space, GSPA also preserves distances to other signals one can define on the shared cell–cell graph. This enables flexible ranking of genes based on distance to the synthetic signals. In systems where cells naturally can be distinguished by cell types, it is standard to characterize cell types by differentially expressed genes. GSPA naturally enables identification of cell-type-specific genes. Given a dataset where each cell is assigned a cluster (or annotated cluster, that is cell type), for each cell type C, we can define a set indicator signal $\mathbf{1}_C$ on all vertices of the cell–cell graph, where $\mathbf{1}_C(v) = 1$ if $v \in C$ and $\mathbf{1}_C(v) = 0$, otherwise. Then, we can rank genes $\mathbf{X}_i$ based on how close they are to the normalized indicator signal $\tilde{\mathbf{1}}_C = \frac{\mathbf{1}_C}{\|\mathbf{1}_C\|_2}$ in their dictionary representation, that is, how close $\mathbf{X}_i\hat{\mathbf{W}}$ is to $\tilde{\mathbf{1}}_C\hat{\mathbf{W}}$. Formally, we define cell-type association ranking of genes by the following score.

Definition 1. *Given normalized cell-type indicator signal $\tilde{\mathbf{1}}_C$ for cell type C and wavelet representation $\hat{\mathbf{W}}$, the cell-type association score, $c(i)$ for each gene signal $\mathbf{X}_i$ is defined as:*

$$c(i) := -\| \mathbf{X}_i\hat{\mathbf{W}} - \tilde{\mathbf{1}}_C\hat{\mathbf{W}}\|_2^2 \tag{10}$$

Unlike many differential expression tests, which compare mean expression between clusters, the cell-type association score ranks highly genes that are close to zero expression in all other cell types and close to uniform expression in the cell type of interest. This results in a ranking that prioritizes specificity to the cell type and achieves close results to ground-truth differential expression scores from simulated data (Extended Data Fig. 1c). In addition, conventional methods for detecting differential gene expression between clusters often return inflated P values because of the double use of gene-expression data, first to partition the data into clusters and then to define significance statistics along the same partitions; consequently, filtering based on P value alone results in an increased rate of false positives, and some pipelines have turned to gene rankings instead of cut-offs[23].

Computation of differential localization

Characterizing differentially expressed genes between clusters is not feasible for many biological systems. For example, for datasets that have trajectory-like structure, consist of subtypes within cell types or do not organize into discrete populations, there is utility in identifying genes localized to particular areas of the cellular manifold without prior cell-type identification. To this end, we naturally extend GSPA to a framework called differential localization. We calculate the specificity, termed gene localization score $l(i)$, of a given gene signal i by calculating the multiscale representation of a uniform signal $\mathbf{u}$ and computing the distance between this and each gene signal representation. Genes are then ranked, where those that are most differentially localized are farthest from the uniform signal representation.

The gene localization score, $l(i)$ for each gene $\mathbf{X}_i$, with normalized uniform signal $\mathbf{u} = \frac{1}{\sqrt{n}}\mathbf{1}$ and wavelet representation $\hat{\mathbf{W}}$, is defined as:

$$l(i) := \| \mathbf{X}_i\hat{\mathbf{W}} - \mathbf{u}\hat{\mathbf{W}}\|_2^2$$

Genes with a high localization score are considered more relevant for describing cell–cell variation and can be used for feature selection or characterization of gene programs and networks without the underlying assumption of discrete clusters.

GSPA-LR for cell–cell communication

Using known LR pairs, GSPA-LR first obtains ligand and receptor embeddings individually, then concatenates them into an LR pair representation. We do this because two communication patterns a and b should be represented similarly if ligand_a and ligand_b show similar expression profiles and receptor_a and receptor_b show similar expression profiles. So, mapping the concatenated representations identifies LR pairs with shared patterning on the cellular manifold across and within cell types and is more informative than aggregating the ligand and receptor signals (such as through summation or averaging the expression of both). If pathway attributes are available, LR–LR similarity allows us to map the pathway space. That is, we can build a kNN graph of the LR pair representations $\mathcal{G}_{\text{LR}} = (V_{\text{LR}}, E_{\text{LR}})$ and, for each pathway, define a set indicator signal $\mathbf{1}_{\text{pathway}}$ on all vertices of $\mathcal{G}_{\text{LR}}$, where $\mathbf{1}_{\text{pathway}}(v) = 1$ if $v \in$ pathway and $\mathbf{1}_{\text{pathway}}(v) = 0$, otherwise for $v \in V_{\text{LR}}$. Then, we can embed these pathway indicator signals using GSPA.

GSPA-multimodal for spatial analysis

To leverage GSPA-multimodal for analyzing spatial transcriptomic data, we constructed an integrated diffusion operator using the approach delineated in ref. 46, which defines two cell–cell affinity graphs $\mathcal{G}_{\text{exp}}$ and $\mathcal{G}_{\text{spatial}}$ based on expression similarity and spatial location, respectively, then constructs a diffusion operator that integrates these two graphs (see 'GSPA for multiple modalities, datasets and large graphs' above). Then, we build an integrated wavelet dictionary with this operator and learn gene embeddings as previously described.

GSPA-Pt for patient embeddings

In the GSPA-Pt framework, we first consider $\mathbf{X}_{\text{Pt}_p}$ as a single-cell dataset for patient p for $p \in 1, \ldots, P$. We then concatenate all samples to build a shared cell–cell graph $\mathcal{G}_{\text{cell}}$, which we use to build the wavelet dictionary $\hat{\mathbf{W}}$ as before. As each entry in $\hat{\mathbf{W}}$ is associated with a patient $p \in 1, \ldots, P$, we can split $\hat{\mathbf{W}}$ into patient-specific dictionaries $\hat{\mathbf{W}}_{\text{Pt}_1}$, $\hat{\mathbf{W}}_{\text{Pt}_2}$, …, $\hat{\mathbf{W}}_{\text{Pt}_P}$. Then, for each p, we project $\mathbf{X}_{\text{Pt}_p}$ onto $\hat{\mathbf{W}}_{\text{Pt}_p}$ and learn a reduced patient-specific gene representation. Each patient is represented by a gene embedding, which is flattened into a vector for downstream analysis.

Comparison with other gene-mapping strategies

Here we describe in detail each comparison for our experiments with GSPA:

- Raw measurements approach embeds $\mathbf{X}$
- $\text{GAE}_{\text{no-att}}(\mathcal{G}_{\text{gene}})$ (Graph Autoencoder without attention) embeds $\mathcal{G}_{\text{gene}}$, representing a gene–gene similarity graph based on the scRNA-seq data
- $\text{GAE}_{\text{att}}(\mathcal{G}_{\text{gene}})$ (Graph Autoencoder with attention) embeds $\mathcal{G}_{\text{gene}}$
- $\text{Node2Vec}(\mathcal{G}_{\text{gene}})$ is a shallow node embedding approach that embeds $\mathcal{G}_{\text{gene}}$
- MAGIC($\mathbf{X}$) (Markov Affinity-based Graph Imputation of Cells) embeds $\mathbf{X}$ after denoising with $\mathcal{G}_{\text{cell}}$, representing a cell–cell similarity graph based on the scRNA-seq data
- DiffusionEMD($\mathbf{X}$, $\mathcal{G}_{\text{cell}}$) (Diffusion Earth Mover's Distance) embeds $\mathbf{X}$ via optimal transport on $\mathcal{G}_{\text{cell}}$
- GFMMD($\mathbf{X}$, $\mathcal{G}_{\text{cell}}$) (Graph Fourier Maximal Mean Discrepancy) embeds $\mathbf{X}$ via MMD (maximal mean discrepancy) on $\mathcal{G}_{\text{cell}}$
- Eigenscore($\mathbf{X}$, $\mathcal{G}_{\text{cell}}$) is a diffusion-based signal selection approach that embeds $\mathbf{X}$ via alignment to Laplacian eigenvectors of $\mathcal{G}_{\text{cell}}$
- SIMBA (Single-Cell Embedding Along with Features) co-embeds $\mathbf{X}$ and $\mathbf{X}^\top$ via heterogeneous graph embedding
- siVAE (Scalable Interpretable Variational Autoencoder) co-embeds $\mathbf{X}$ and $\mathbf{X}^\top$ via jointly trained cell-wise and feature-wise VAEs

We summarize and diagram these comparisons in Extended Data Fig. 2.

Direct embedding of gene-expression measurements. The simplest and most intuitive approach to map the gene space is with the original measurements. $\mathbf{X}$ consists of values where each cell is measured as a vector of gene-expression counts, so we can consider the case where the genes are observations, and each gene is measured as a vector of expression counts in each cell. We use autoencoder $D \circ E$ to reduce the dimensionality, where $\mathbf{X} \approx D(E(\mathbf{X}))$ and $E(\mathbf{X})$ is the embedding.

Embedding constructed gene–gene graph. Another approach is to construct a gene–gene kNN graph $\mathcal{G}_{\text{gene}} = (V_{\text{gene}}, E_{\text{gene}})$ from $\mathbf{X}$, where each node in V_{gene} corresponds to a gene and each edge E_{ij} in E describes the similarity between gene i and gene j based on Euclidean distance. We can then leverage graph representation learning to propagate information between gene–gene relationships and learn node embeddings. We test one shallow embedding Node2Vec($\mathcal{G}_{\text{gene}}$) and two graph autoencoder embeddings. The graph autoencoder $D_{\text{no-att}} \circ E_{\text{no-att}}$ consists of graph convolutional layers, where $\mathcal{G}_{\text{gene}} \approx D_{\text{no-att}}(E_{\text{no-att}}(\mathcal{G}_{\text{gene}}))$. The graph autoencoder $D_{\text{att}} \circ E_{\text{att}}$ consists of graph attention layers, where $\mathcal{G}_{\text{gene}} \approx D_{\text{att}}(E_{\text{att}}(\mathcal{G}_{\text{gene}}))$. $E_{\text{no-att}}(\mathcal{G}_{\text{gene}})$ and $E_{\text{att}}(\mathcal{G}_{\text{gene}})$ correspond to the embeddings without and with attention, respectively.

Imputing gene signals with cell–cell graph. The above methods do not use information from the cell–cell graph for the computation of gene representations. On the basis of our desired properties ('Results'), we hypothesized that incorporating cellular affinities would enable the comparison of non-overlapping gene signals and local and global distances on the cellular manifold.

First, we compare against MAGIC[25], which imputes missing gene expression via data diffusion. MAGIC calculates a diffusion operator $\mathbf{M}$ powered to t, and left-multiplies $\mathbf{M}^t$ to $\mathbf{X}^\top$ as a low-pass filter. For comparison, we left-multiply $\mathbf{X}$ to $\mathbf{M}^t$, which practically denoises gene signals and performs comparatively to MAGIC (data not shown). We then employ an autoencoder $D \circ E$, where $\mathbf{X}\mathbf{M}^t \approx D(E(\mathbf{X}\mathbf{M}^t))$ and $E(\mathbf{X}\mathbf{M}^t)$ is the embedding.

Optimal transport distances between gene signals. Owing to the relationship between GSPA and Wasserstein distance (that is optimal transport), we compare GSPA against three approaches for fast optimal transport that have been developed and used for gene signals on the cellular graph.

DiffusionEMD[20] computes optimal transport based on multiscale diffusion kernels. This construction is related to UDEMD[21] described above. Between two genes $i_1, i_2 \in \mathbf{X}$, $\text{DiffusionEMD}_{\beta,J}(i_1, i_2) := \sum_{j=0}^{J} \left\| T_{\beta,j}(i_1) - T_{\beta,j}(i_2) \right\|_1$, where $0 < \beta < 1/2$ is a meta-parameter used to balance long- and short-range distances and J is the maximum scale considered here. $T_{\beta,j}(\mathbf{X}_i) := \begin{cases} 2^{-(J-j-1)\beta}\left(\boldsymbol{\mu}_i^{(2^{j+1})} - \boldsymbol{\mu}_i^{(2^j)}\right) & j < J \\ \boldsymbol{\mu}_i^{(2^j)} & j = J \end{cases}$, where $\boldsymbol{\mu}_i^{(t)} := \frac{1}{n_i}\mathbf{P}^t\mathbf{1}_{\mathbf{X}_i}$ is a kernel density estimate over $\mathcal{G}_{\text{cell}}$. GFMMD[28] is defined via an optimal witness function that is smooth on the graph and maximizes the difference in expectation between the pair of gene distributions. $\text{GFMMD}(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}) := \max_{f : f^\top \mathbf{L} f \le 1} \mathbb{E}_{\mathbf{X}_{i_1}}(f) - \mathbb{E}_{\mathbf{X}_{i_2}}(f)$, holding for any construction of a positive semi-definite Laplacian matrix $\mathbf{L}$ and chosen threshold $T = 1$.

For these approaches, multiscale signal features $\hat{\mathbf{X}}$ are computed before distance calculation. We reduce the dimensionality of these features via an autoencoder $D \circ E$, where $\hat{\mathbf{X}} \approx D(E(\hat{\mathbf{X}}))$ and $E(\hat{\mathbf{X}})$ is the embedding.

Computing eigenscores. Eigenscores were proposed as a topologically motivated mathematical method for feature selection, and they were also shown to be useful for mapping the gene space to distinguish cell types[29]. Eigenscores rank signals or genes based on their alignment to low-frequency patterns in the data, identified through spectral decomposition of the graph Laplacian. Specifically, given the first r left eigenvectors of the normalized Laplacian (where $r \ll n$ to preserve low-frequency patterning), $\text{Eigenscore}(i) := \text{concat}\left(\frac{\langle \mathbf{D}^{1/2}\mathbf{X}_i, e_1 \rangle}{\|\mathbf{D}^{1/2}\mathbf{X}_i\|}, \frac{\langle \mathbf{D}^{1/2}\mathbf{X}_i, e_2 \rangle}{\|\mathbf{D}^{1/2}\mathbf{X}_i\|}, \ldots, \frac{\langle \mathbf{D}^{1/2}\mathbf{X}_i, e_r \rangle}{\|\mathbf{D}^{1/2}\mathbf{X}_i\|}\right)$. We let Eigenscore($\mathbf{X}$), of shape $m \times r$ represent the eigenscores for each gene i in $\mathbf{X}$. We finally reduce the dimensionality for gene space mapping via an autoencoder $D \circ E$, where $\text{Eigenscore}(\mathbf{X}) \approx D(E(\text{Eigenscore}(\mathbf{X})))$ and $E(\text{Eigenscore}(\mathbf{X}))$ is the embedding.

Co-embedding of cells and genes. Finally, recent approaches incorporate cell–cell affinities through simultaneously learning embeddings for cells and genes. This methodology has the benefit of learning the pairwise similarities between cells, rather than constructing the cell–cell graph a priori, and training this module in tandem with gene–gene similarity training. siVAE[30] is a neural network consisting of cell-wise and gene-wise encoder–decoders. The cell-wise encoder takes each cell's measurement across all features and maps cell embeddings similarly to a classical VAE, which computes an approximate posterior distribution over the location of the cell. The gene-wise encoder takes a gene's measurement across all cells and maps gene embeddings. The decoders of both VAEs combine to output the expression level of each feature in each particular cell, ensuring that each mapping has semantic structure. SIMBA[31] constructs a heterogeneous graph, where the nodes are cells and genes, and edge type are determined through expression level. SIMBA first bins the continuous gene-expression values into a discrete distribution that preserves the shape of the original distribution, then encodes different bins as different relation types. A node embedding for each node in the graph is then learned via stochastic gradient descent optimization of a link prediction objective. For both procedures, we evaluated only the gene space embeddings in our comparisons.

Diffusion wavelets and comparison with MAGIC

Diffusion wavelets, compared with diffusion maps[7] and related approaches, such as MAGIC, perform multiscale analysis of graphs and functions on graphs. By representing the cell–cell graph at multiple scales, diffusion wavelets are able to decompose low-frequency and medium-frequency components of a signal, in addition to removing noise, whereas MAGIC acts as a low-pass filter and maintains only the low-frequency components (Supplementary Fig. 2).

In addition, diffusion wavelets represent the local and global geometry of the cell–cell graph, which enables the representation of the distance between signals that are far apart. We demonstrate this with an experiment using a linear simulated trajectory with noise (also used in Figs. 2 and 3; Supplementary Fig. 3a). This trajectory defines our cell–cell graph, and we simulate signals on this graph as diracs defined on each cell in the trajectory. Each signal naturally has a label associated with it—the pseudotime value of the cell it was defined on (Supplementary Fig. 3b). Notably, as each signal is not overlapping, approaches that do not use the cell–cell graph would not be able to capture meaningful distances between signals.

To test the signal embeddings, we regress the pseudotime label from the latent space (Supplementary Fig. 3c). We hypothesized that methods that locally smooth the signal would perform well when signals are close together, as local geometry sufficiently captures pseudotime, and poorly when the signals are far apart. Therefore, we evaluated the prediction with Spearman correlation and increased separation between signals. As expected, MAGIC worsens in the performance as signals are increasingly spread apart, whereas GSPA and GSPA+QR do not show a trend related to the separation between signals (Supplementary Fig. 3d). We also perform unsupervised evaluation of the embeddings at spacing 2^6 via the correlation between the Fiedler vector of the signal graph and the pseudotime label. GSPA and GSPA+QR show

high correlation versus MAGIC (Supplementary Fig. 3e), and also show a qualitative association with the pseudotime label when visualized versus MAGIC (Supplementary Fig. 3f).

Robustness to normalization and graph construction

On average across all hyperparameters and preprocessing choices, GSPA and GSPA+QR outperformed all other approaches (Extended Data Fig. 4b). Furthermore, despite potential sensitivity to graph construction, approaches that leveraged the cell–cell graph to calculate gene–gene relationships outranked approaches that used pointwise gene measurements on both experiments (Extended Data Fig. 4c). For the co-expression experiments, approaches with the cell–cell graph had an average rank of 2.929, and approaches without the cell–cell graph had an average rank of 8.071 (Extended Data Table 1a). For the localization experiments, approaches with the cell–cell graph had an average rank of 2.686, and approaches without the cell–cell graph had an average rank of 8.314 (Extended Data Table 1b). This result reinforces the desired distance preservation and noise robustness properties garnered from using the cell–cell graph and further supports our assertion that considering genes as signals on the cell–cell graph can improve analysis of gene–gene relationships. In addition, as most single-cell sequencing analysis tools and pipelines construct a cell–cell graph, including for visualization, clustering and trajectory inference[68], using the same graph can ensure consistent biological analysis with GSPA.

Training details

Default GSPA hyperparameter selection and training details. The cell–cell graph was built with PHATE using default parameters ($k = 5$ and α decay = 40) from the PCA space, as common for cell–cell graph construction. The power was set by default to 2 to mimic the dyadic scales in ref. 9 and J was set by default to $\log(n)$ based on ref. 20 (see Lemma 1 and surrounding discussion above). For GSPA+QR, the epsilon parameter was set to 1×10^{-3}. The data were first dimensionality reduced with PCA to 2,048 components (which captures the majority of variation), and then an autoencoder nonlinearly reduced the dimensionality further to latent dimension of 128. The autoencoder was designed with two layers with bias in the encoder and decoder, with a rectified linear unit (ReLU) activation function between layers. The models were trained for a mean squared error objective with an Adam optimizer with learning rate of 0.001 for 100 epochs, with early stopping (patience of 10) using the loss of a validation set 5% of the size of the training set. For all analyses, signals are first L2 normalized before projection.

Comparison hyperparameter and training details. For method comparisons in Figs. 2 and 3, we ran each method three times, including reconstructing the graph with new seeds. All signals were first L2 normalized, and, where applicable, dimensionality reduced using PCA with 2,048 components and an autoencoder (AE) with latent dimension of 128 (PCA+AE; same configuration as for GSPA). For raw measurements, we ran PCA+AE on $\mathbf{X}$. For MAGIC($\mathbf{X}$), we compute the diffusion operator with default parameters. We then project the signals onto this diffusion operator and run PCA+AE. We compute eigenscores based on the approach described in ref. 29, then dimensionality reduced with PCA+AE. We learned multiscale representations with DiffusionEMD and GFMMD, then dimensionality reduced with PCA+AE. For signal–signal graphs, kNN graphs were generated from the signals with $k = 5$. Node2Vec was run on this graph with latent dimensionality of 128, walk length of 80 and 10 walks. $\text{GAE}_{\text{no-att}}$ was run with graph convolutional layers, and GAE_{att} was run with graph attention layers on this graph. The GAE configuration matched the previous AE configuration. For SIMBA, we constructed a heterogeneous cell–gene graph using default parameters, without highly variable genes. We then trained the graph embedding with 128 dimensions, auto-estimating weight decay. For siVAE, we constructed the encoder–decoder architecture with the same number and size of layers as our GSPA autoencoder. We additionally used 2,000 iterations, mini-batch size (mb_{size}) of 0.2, L2 regularization strength ($l2_{\text{scale}}$) of 1×10^{-3}, learning rate of 1×10^{-4}, decay rate of 0.9, and early stopping with a patience of 100 iterations. We used ReLU activations in between layers.

Datasets and preprocessing

Simulated datasets with Splatter. Three datasets were simulated using Splatter[24] with one (linear) trajectory, two branches and three branches. All datasets were simulated with 10,000 cells and 10,000 genes, where cells were distributed equally between branches (where applicable). The dropout probability was set to 0.95 to generate 'noisy' datasets, and each dataset had associated 'true' noiseless counts from the same experiment. After simulation, genes expressed in fewer than 50 cells were removed, and the matrix was L1 normalized for library size and square-root transformed (or log-transformed for robustness analysis). This resulted in 8,821 genes in the linear simulation, 8,820 genes in the two-branch simulation and 8,823 genes in the three-branch simulation. Cells were then visualized with PHATE.

PBMC dataset. These data consisted of 2,638 cells and 1,838 genes, following the scanpy preprocessing workflow (https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html) to analyze 10x Genomics data acquired from 10x Genomics[33]. Cells with fewer than 200 genes expressed and genes expressed in fewer than 3 cells were removed. Cells with over 2,500 total counts or over 5% mitochondrial counts were removed. The data were L1 library size normalized and log-transformed, and highly variable genes were preserved and scaled.

Embryoid-body dataset. This data were derived from ref. 3 and capture the cellular populations within the embryoid-body differentiation process. We followed the preprocessing procedure from the original work, removing cells with library size higher than the 75% and lower than the 20% for each sample. Genes expressed in fewer than ten cells were removed, and the data was L1 library size normalized. The top 10% of cells with highest mitochondrial expression were removed, and the data were square-root transformed. This resulted in 16,821 cells and 17,845 genes.

Three-timepoint scRNA-seq dataset. Mice were infected with LCMV Armstrong (acute) and Clone 13 (chronic), and CD8^+ CD44^+ Tetramer^+ T cells were sorted by fluorescence-activated cell sorting before 10x Chromium 5p scRNA-seq at day 4, day 8 and day 40 (ref. 10). Three to five mice were infected for each timepoint/condition in a staggered manner to enable same day take down of each timepoint. Spleens from mice were pooled for each timepoint/condition and sorted before their loading on the Chromium instrument. Ten thousand cells were loaded into a lane of the instrument for each timepoint/condition. The resulting 10x libraries were sequenced on an Illumina NovaSeq with an approximate read depth of 20,000 reads per cell. We then processed the data using Cell Ranger before further filtering. Cells expressing fewer than 200 genes, with fewer than 500 counts or more than 25,000 counts, were removed. Genes expressed in fewer than three cells were removed. Cells with mitochondrial percentage greater than 6% were removed. We then L1 normalized for library size, log-transformed and clustered cells using Leiden clustering[64], removing contaminating populations enriched for non-CD8^+ T cell markers. The acute and chronic datasets were combined, and highly variable genes were detected as the top 10% of genes using scprep (https://scprep.readthedocs.io/en/stable/). This resulted in 14,152 genes and 39,704 cells detected across datasets, with 6,811 cells from acute day 4; 7,418 cells from acute day 8; 6,740 cells from acute day 40; 6,205 cells from chronic day 4; 7,553 cells from chronic day 8; and 4,977 cells from chronic day 40. The combined datasets were then visualized with PHATE, and key marker genes were visualized on the PHATE embedding with MAGIC. Graphs for PHATE, MAGIC and GSPA were built with default parameters, except k for the kNN graph construction was set to 30 due to the larger number of cells.

Transcription factor perturbation scRNA-seq dataset. The single-guide RNA (sgRNA) library was cloned into a murine stem cell virus retroviral backbone containing an expression cassette for human CD2 as a selectable marker[11]. The library targets 39 genes, the majority of which are transcription or epigenetic factors with 3 unique sgRNAs per target gene. sgRNA sequences were chosen via CHOPCHOP. Negative control sgRNAs were spiked in to make up 15% of the final library. Retrovirus production was performed using platinum-E cells. P14 Cas9 transgenic $CD8^+$ T cells were isolated (StemCell Mouse CD8 Selection Kit) and activated with anti-CD3/CD28 activation beads (Dynabeads Mouse T activator) for 1 day before infection with sgRNA library retrovirus using retronectin and protamine sulfate. The cells were then grown in human IL-2 (5 ng ml^{-1}) for 2 days before magnetic selection to enrich for transduced cells based on the human CD2 selection marker (StemCell Human CD2 positive selection kit). Then 100,000 cells were transferred into 7 day-1 LCMV-Armstrong infected mice. At day 8 of infection, mice were killed and P14 T cells were sorted from the spleens. Equal numbers of P14s were pooled from 7 mice before sorting and then loaded into 6 wells of a ChromiumX instrument to perform paired 5′ scRNA and CRISPR sequencing; 40,000 cells were loaded into each well to recover approximately 20,000 cells per well. The resulting 10x libraries were sequenced on an Illumina NovaSeq at a read depth of 20,000 reads per cell for RNA and 5,000 reads per cell for CRISPR feature barcode. The data were then processed using Cell Ranger before downstream analyses. We removed cells with fewer than 200 genes detected and cells with fewer than 500 counts and more than 20,000 counts. Cells with mitochondrial percentage greater than 5% were removed. To remove doublets, the data were log-transformed, scaled and centered, and doublets were identified with DoubletFinder[65] with parameterSweep_v3 and expected doublet percentage of 7.5%. We then clustered cells using shared-nearest-neighbors-based Louvain clustering[66] and removed contaminating populations enriched for non-$CD8^+$ T cell markers. We further filtered the cells to annotate their perturbation identity by assigning an sgRNA identity to a cell if there was >1 unique molecular identifier for that sgRNA and it made up >80% of the sgRNA reads in that cell. Finally, we integrated the replicates with canonical correlation analysis[67] using the top 2,000 highly variable genes and removing T cell receptor, mitochondrial, proliferation and immunoglobulin genes. This resulted in 23,206 cells and 1,795 genes across all perturbations.

Peripheral tolerance scRNA-seq dataset. We obtained scRNA-seq data from ref. 12 and pre-processed it as in the scRNA-seq section. There were 21,178 cells and 21,515 genes after preprocessing. This corresponds to 8,167 cells from Ag ON samples; 3,944 cells from Ag ON/CPI samples; and 9,067 cells from Ag OFF (that is, no AG) samples. Cells were visualized with PHATE and key genes were visualized on the PHATE embedding with MAGIC. The cell–cell graph for PHATE, MAGIC and GSPA were built with default parameters, except k for the kNN graph construction was set to 40 and α was set to 10 to match the analysis in the original work.

10x human lymph node spatial transcriptomics dataset. 10x Genomics data were obtained from the 10x website[13] and downloaded via the scanpy package[68]. According to their website, 10x obtained fresh frozen human lymph node tissue from BioIVT Asterand Human Tissue Specimens. The tissue was embedded and cryosectioned as described in the Visium Spatial Protocols Tissue Preparation Guide (Demonstrated Protocol CG000240). Tissue sections of 10 µm thickness were placed on Visium Gene Expression Slides. We removed spots with fewer than 5,000 counts, more than 35,000 counts and over 20% mitochondrial counts. We removed genes detected in fewer than ten cells, L1 normalized for library size, and log-transformed the data. After the above preprocessing, there were 3,861 spots and 19,685 genes detected. The top-2,000 highly variable genes were selected following the scanpy tutorial for this dataset.

Immunotherapy response in patients with melanoma scRNA-seq dataset. We obtained pre-processed scRNA-seq data with annotated cell types and other relevant metadata (for example, sample labels, patient response) from ref. 14 and the Single Cell Portal (https://singlecell.broadinstitute.org/single_cell). From these data, there were 48 samples, which corresponds to 19 pre-therapy samples and 29 post-therapy samples, as well as 31 non-responder samples and 17 responder samples. There were 15,300 cells and 12,364 genes detected across all samples, with 10,190 cells from non-responders and 5,110 from responders.

Computational details

Diffusion wavelets versus MAGIC. For Supplementary Fig. 2, the smooth signal was defined based on visualization coordinates to capture the low-dimensional axis of variation. The oscillating signal was defined by the sine of four times the smooth signal to define a medium-frequency component. The noise was generated based on random samples from a uniform distribution between [−1, 1). The aggregate signal was defined based on the sum of these three components. The denoised signal was computed with MAGIC and the wavelet dictionary was computed with GSPA and $J = 5$.

Dirac signals comparing GSPA and MAGIC. For the dirac experiment (Supplementary Fig. 3), we simulated and pre-processed a linear trajectory of cells as described above, and we defined signals as a dirac on each cell, where the signals naturally had a pseudotime label based on the cell it was defined on. We learned unsupervised embeddings for GSPA+QR, GSPA and MAGIC as described above. Then we increased the distance between signals by subsampling every other cell in the pseudotime-ordered trajectory, then every 2^2 cells, every 2^4 cells and so on, and we learned embeddings for the signals increasingly spread apart. We then evaluated the embeddings by repeated K-fold splits and ridge regression to predict the pseudotime labels, comparing methods based on Spearman correlation between prediction and true labels across ten runs. Gene embeddings were visualized with PHATE.

Simulated co-expression experimental details. For the co-expression experiment with a linear trajectory (Fig. 2) and two and three branches (Extended Data Fig. 3), we generated simulated data as described above, then defined signals as the gene features from the simulation experiment. Because of the simulation design, this meant we have both noisy $\mathbf{X}$ and noiseless $\mathbf{X}'$ versions of the same gene signals. This allows us to compute 'ground truth' co-expression as the Spearman correlation between all noiseless pairs of genes. Given the large number of genes and the nature of biological data, the large majority of gene–gene pairs had a near-zero correlation. The correlation also was associated with the library size of the genes in the pair. Therefore, we stratified the labels based on correlation and the mean library size of the pair within each correlation bin. We learned unsupervised gene embeddings for all comparisons as described above, then, for an equal number of pairs per stratification bin, we computed the distance between gene embedding pairs and the (anti-)correlation with the true co-expression.

To identify gene modules, we visualized the GSPA+QR embedding with PHATE and used Leiden clustering. To map these gene modules back to the cells most enriched for the modules, we leveraged a gene set enrichment approach from ref. 59 and implemented in ref. 68. This approach provides a score defined on all cells as the average expression of a set of genes subtracted with the average expression of a reference set of genes. Using 25% of the number of genes as the number of bins, we calculated a cell-enrichment score for each gene module and visualized this enrichment score versus pseudotime.

To analyze how the gene embeddings relate to peak over time (trajectory analysis), we binned cells into 100 bins based on pseudotime values and computed the mean expression of each gene over the binned timepoints. Then we annotated each gene based on which

bin it 'peaked', or had a maximum value. We colored all gene embeddings based on this score.

Finally, to perform archetypal analysis of the gene space, we ran Archetypal Analysis Network (AAnet)[69,70] on the gene embeddings outputted by each method with $n_{at} = 2$. Then, we identified the 50 genes nearest to each archetype and used the same gene set enrichment approach as above[59,68], now visualizing the cell-enrichment score on the embedding and over pseudotime for only archetypal genes, rather than all gene modules.

Simulated localization experimental details. *Generating simulated signals with known localization scores*. For the localization experiment with a linear trajectory (Fig. 3) and two and three branches (Extended Data Fig. 6), we generated simulated data as described above. However, instead of using the genes as signals, we designed signals with 'ground truth' localization labels (Extended Data Fig. 5). We intuited that more localized signals are not defined by where they are enriched in the trajectory, but rather by how spread out that enrichment is. Thus, we aimed to constrain the size of the region where each signal could be defined, termed 'window', where the window can be defined anywhere on the trajectory and is only defined by its size.

To generate signals and associated localization scores with these properties, we used the ground-truth pseudotime label (provided by Splatter) scaled to be between 0 and 1, and we defined window size δ. Then we randomly selected a timepoint t between $[\delta/2, 1-\delta/2]$ and defined a pseudotime window $[t-\delta/2, t+\delta/2]$. Next, we sampled 500 cells from all cells within this pseudotime window, and we let the signal equal 1 on these cells and 0 on all other cells (Extended Data Fig. 5).

For each of five window sizes $\delta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, we generated 50 signals, resulting in 250 signals total. As smaller δ corresponds to a higher localization score, we defined the true localization score for each signal to be $1-\delta$. This score is unrelated to where the signal is defined based on randomly selected t. Furthermore, all signals are defined on exactly 500 cells, so the localization score is not associated with the number of cells expressing the gene.

Predicting localization from uniform signal. We then defined a uniformly enriched gene as a signal equal to 1 on all cells. L2 normalizing all signals, we embedded the uniform gene with other signals and computed the (multiscale) distance of all signals to the uniform distribution. We intuited that the uniformly expressed gene should be closest to signals with a low localization score ($\delta = 1.0$) and farthest from signals with a high localization score ($\delta = 0.2$). That is, the distance to the uniform signal can be considered the predicted localization score (as described in 'Computation of differential localization' in Methods). We therefore predicted the localization score for all signals and evaluated methods based on Spearman correlation of predicted and true localization.

Computing localization for comparisons. For GSPA+QR, GSPA and MAGIC, this involved projecting the uniform signal onto the cell representation/dictionary and calculating the distance between the projected uniform signal and all other projected signals. Eigenscore and GFMMD defined a version of this localization based on the L2 norm of their embeddings, so we evaluated localization using this measure. For DiffusionEMD, we learned a multiscale representation of the uniform signal, and we computed the distance to all other signals before dimensionality reduction. For the raw measurements, we took the distance of the uniform signal to all other signals before dimensionality reduction. For Node2Vec and the GAE approaches, we built a signal–signal graph with the uniform signal and embedded these graphs, then computed the L2 distance between the uniform embedding and the other signals. For SIMBA and siVAE, which learn a low-dimensional representation of the genes directly, we learned a low-dimensional embedding of the uniform signal and computed the distance to all other signals in this latent space.

Robustness to transformation and graph construction details. To evaluate robustness to steps to process the cellular measurements before mapping the gene space, we ran our co-expression experiment and localization experiment for all comparisons over each combination of the following: 2 random seeds, 2 single-cell dataset transformations (log and sqrt), 4 choices for k in the construction of the kNN graph (5, 15, 25, 50) and 3 choices for construction of the nearest-neighbors graph (kNN, shared nearest neighbors and construction with an adaptive α-decaying kernel). Together this resulted in 48 runs for each method (Extended Data Fig. 4a).

Batch-effect experimental details. To analyze the effects of batch effect and batch-effect correction with GSPA, we simulated a dataset of 2,000 cells and 10,000 genes using Splatter[24], with 1,000 cells in each batch, 3 clusters with equal probability, 0.6 DE probability (probability that a gene will be selected to be differentially expressed), and batch factor location and scale of (0.1, 0.01) (where batches are specified by generating a small scaling factor for each gene in each batch from a log-normal distribution). These data were then processed to remove genes expressed in fewer than 50 cells, L1 normalized for library size and log-transformed. We then computed gene embeddings with default GSPA+QR for all genes, coloring by DE factor and batch-effect factor. To correct the graph for batch effect, we constructed an MNN graph, introduced in ref. 17, and reran and revisualized PHATE and GSPA+QR with the corrected graph. We then computed cell-type association scores for each of the annotated clusters and visualized those scores.

Characterization of PBMC gene embedding. For co-expression analysis (Fig. 2g), we embedded all highly variable genes using the default GSPA+QR approach. Then, to visualize cell-type-specific genes, we retrieved all annotated cell-type markers from PanglaoDB[34] for T cells, monocytes, B cells, natural killer cells, dendritic cells and megakaryocytes. We then subsetted this list to 'canonical markers' (defined by PanglaoDB) that were highly variable and had highest mean expression in the correct annotated cell type in our dataset.

For analysis of localized genes (Fig. 3d), we computed the localization score for all mapped genes and identified the genes with the top-25% localization score as 'predicted localized genes', and genes with the bottom-25% localization score as 'predicted non-localized genes'. We then constructed the cell–cell graph using only these genes and reran PHATE. Next we calculated the pairwise geodesic distances between all cells in the full graph and the feature-selected graph. As a proxy for the preservation of the cell–cell relationships with selected features versus all genes, we randomly chose 100,000 pairwise distances (in two runs) and computed Spearman correlation between the full graph distances and the feature-selected graph distances.

Characterization of embryoid-body gene embedding. For co-expression analysis (Fig. 2g), we embedded all measured genes except mitochondrial genes, as they showed a very different trend to the other genes and embedded distinctly. We used the default GSPA+QR approach. Then, to perform gene trajectory analysis, we computed the diffusion map for these genes. On the basis of the lineage analysis done in ref. 3, we identified diffusion map component 4 as associated with the hemangioblast lineage and colored the gene embedding by this component, annotating various key genes along this gene trajectory.

For analysis of localized genes (Fig. 3d), we repeated the procedure as for the PBMC data to compare the full graph and feature-selected graph distances and embedding.

Computing cluster rank for localization versus cluster rank comparison. For the comparison in Fig. 4d, we performed Leiden clustering on the cells, which identified nine cell clusters. Using scanpy, we ran a Wilcoxon rank sum test to identify genes differentially enriched in each cluster. This results in a z-score underlying the computation of a P value

for each gene for each group. For each gene, we stored the maximum score across clusters, and we ranked genes based on this maximum score, similar to a cluster-based ranking method described in ref. 23. This ranking reflects how enriched the gene is, which we compare against the computed localization score.

Computing gene module enrichment per condition. Genes were visualized with PHATE and modules were identified via Leiden clustering (Fig. 4b). Using the same gene module enrichment approach as in the simulated co-expression experiment, now using the highly variable genes as the reference set, we calculated a cell-enrichment score for each gene module. As positive values of this score indicate an enrichment over the reference set, we counted cells with a score >0 from each sample and normalized this count by the number of cells from each sample. This represents the gene module enrichment of cells per condition (Fig. 4e).

Computing and comparing type 1 interferon signaling signature. To determine the enrichment of the type 1 interferon signaling gene signature, we constructed gene embeddings for all approaches designed to map the gene space. Then we identified gene modules using Leiden clustering, chose the gene module containing canonical type 1 interferon marker *Irf7*, and selected the top-10% localized genes within the gene module. This allowed us to choose genes that were both related to type 1 interferon signaling, through similarity to *Irf7*, but were also unbiasedly selected based on the calculated gene modules and localization score. We next wanted to add additional comparisons to other canonical approaches for identifying gene signatures. To compare against analysis done by clustering cells and identifying differentially expressed genes, we selected the top-100 differentially expressed genes from each cell cluster (obtained as noted above). Finally, to compare against factor analysis approach cNMF, we extracted the gene program for which *Irf7* had the highest loading, then selected the genes with the highest-10% loading score to that program. To compare the biological relevance of selected genes from each comparison, we performed gene set enrichment analysis using Enrichr[71] and the BioPlanet gene set resource[72], and we visualized enrichment scores for two type 1 interferon-related gene sets.

Building module-specific gene co-expression networks. While gene modules group genes based on relatively similar expression profiles, the localization score determines how specific that expression profile is. For example, *Rps20* and *Tcf7* both belong to gene module 1, but, because *Rps20* shows high expression in other cells, whereas *Tcf7* shows almost no expression in other cells, *Tcf7* has a higher localization score. Therefore, to build module-specific gene co-expression networks, we identified the top-10% localized genes in each gene module, then built a *k*NN graph with $k = 5$ from the GSPA+QR gene representations. Networks were then visualized with Cytoscape[73]. We performed protein–protein interaction analysis with STRINGdb[39] by testing whether each module showed significantly higher interaction than expected for a random set of proteins of the same size and degree distribution.

Building perturbation-specific gene co-expression networks. First, we identified genes that were in the top-25% localized in both the negative control and the knockout. Then we built a *k*NN graph with $k = 5$ for the negative control genes, and a *k*NN graph with $k = 100$ for the knockout genes from the GSPA+QR representations. We subtracted the knockout adjacency matrix from the negative control adjacency matrix and built a new graph from the positive entries, visualizing this graph with Cytoscape. This effectively identifies co-expression edges that are in the negative control that are not in the knockout gene–gene graph. Notably, the difference in k was to emphasize connections that were very similar in the negative control and very different in the knockout. For visualization, we removed disconnected subgraphs consisting of two or fewer nodes.

Calculating cell-type-specific communication with CellPhoneDB. As a comparison with the canonical approach for cell–cell communication, we used the permutation test developed in ref. 44 and implemented in squidpy[74]. The test was run with default parameters and the cell-type-annotated labels from the original work[12]. We then visualized the results for the two interactions of interest (*Ccl5*–*Ccr5* and *Cd274*–*Pdcd1*).

Determining cells enriched with ligand and receptor in GSPA-LR analysis. In our analysis, we ignore cell-type labels and run the GSPA-LR communication pipeline described in Fig. 5a using the CellChatDB[75] intercellular communication database, which contains triplets (ligand, receptor, pathway), where the ligand is the source gene, receptor is the target gene and pathway is an attribute defining to which communication pathway the interaction belongs. LR pairs were embedded with PHATE, and modules were identified via Leiden clustering (Fig. 5e). Then, to map these gene modules back to the cells most enriched for the modules, we leveraged the gene set enrichment approach from ref. 59 to identify cells enriched for the ligands and the receptors for each module (Supplementary Fig. 7). We visualize the enrichment score for the ligands and receptors in each module and calculate the condition-specific score as above.

Calculating LR module enrichment scores. For each module, we convert the LR pairs into a list of all unique genes within the module and compute gene set enrichment scores with this list using Enrichr[71] and the BioPlanet database[72]. Gene sets enriched for each module are ranked based on the combined score (as determined by Enrichr), and the top-5 gene sets are visualized for modules 5 and 19.

Calculating spatial variability with SpatialDE. We compared GSPA-multimodal localization on spatial transcriptomic data with SpatialDE[47], run using default parameters. Spatially variable genes were determined as genes with adjusted P value (q) < 0.001 and FSV > 0.2 (where FSV is the fraction of variance explained by spatial variation). We visualized these spatially variable genes on our gene embedding and colored them by FSV. We then compared the localization score of spatially variable and non-spatially variable genes with a one-sided Wilcoxon rank sums test, where $P = 9.47 \times 10^{-7}$. All genes were visualized with spARC denoised counts, but GSPA-multimodal and SpatialDE were run with the original data.

Building module-specific cell-state communication networks from spatial transcriptomic data. We visualize gene embeddings with PHATE and identify gene modules via Leiden clustering (Fig. 6c). We built a *k*NN graph ($k = 5$) and kept only edges that existed in OmniPathDB, adding directionality based on the OmniPathDB annotation, resulting in a gene signalling network. We then mapped the gene signaling network to the cell-type signaling networks. We repeat the following for each gene module graph: for each directed edge (gene_s, gene_t), for all pairs of cell states (celltype_a, celltype_b), if gene_s is differentially expressed in celltype_a, and gene_t is differentially expressed in celltype_b, we add a directed edge from celltype_a to celltype_b. We finally use Cytoscape to visualize intercellular communication edges in blue, and intracellular communication edges within the same cell type (that is (gene_s, gene_t) is intracellular and $\text{celltype}_a = \text{celltype}_b$) in red.

Learning patient embeddings and immunotherapy response. We performed PCA with 5 components and flattened these gene representations into a single vector of size $1 \times 5m$ to represent the patient. We used the first five PCs to represent the patient rather than the autoencoder embedding (as in previous analysis) because the PCs allowed for more interpretable analysis of the coefficients of the classifier. A single dimension of the latent space of the autoencoder may not necessarily capture the major axes of variation for a gene, but the first dimension of the gene PC definitionally captures the major (linear) axis of variation.

For comparison, we performed GSPA using the patient indicator signals on the cell–cell graph, then ran PCA+AE as described above. We also computed the mean expression across all genes for each patient. Finally, we computed the proportion of all clusters (representing immune cell types) and all CD8 clusters (representing CD8 cell types). Using these as unsupervised patient representations, we then classified response using a ridge classifier, comparing based on the area under the receiver operating characteristic curve of classification. Given that the ridge classifier is a linear model, the coefficients represent the features of the patient representation most important for prediction. The features correspond to five components for each gene, so we can map the coefficients to genes relevant for prediction (Supplementary Table 4). We visualized all patient embeddings with PHATE.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The scRNA-seq data of $CD8^+$ T cells during acute and chronic LCMV infection, and the Perturb-seq data during acute LCMV infection, are available on the Dryad repository[10,11]. scRNA-seq datasets for peripheral tolerance in the skin are available from the Gene Expression Omnibus (GEO) database under GEO Series accession number GSE228586, originally from ref. 12. Visium spatial gene-expression data for the human lymph node are from 10x Genomics[13]. The scRNA-seq data for melanoma samples pre- and post-therapy are accessible from the GEO database through GEO Series accession number GSE120575, originally from ref. 14. We additionally used the following data resources for our case studies: OmniPath (https://omnipathdb.org)[50], STRINGdb (https://string-db.org)[39], Enrichr (https://maayanlab.cloud/Enrichr/)[71] and CellChatDB (http://www.cellchat.org)[75].

Code availability

The source code for the Python package is available on The Python Package Index at https://pypi.org/project/gspa/ and on GitHub at https://github.com/KrishnaswamyLab/Gene-Signal-Pattern-Analysis (ref. 15). Notebooks to generate figures presented here are available at https://github.com/KrishnaswamyLab/GSPA-manuscript-analyses (ref. 76).

References

1. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Phil. Mag. J. Sci.* **2**, 559–572 (1901).
2. Becht, E. et al. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.* **37**, 38–44 (2018).
3. Moon, K. R. et al. Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.* **37**, 1482–1492 (2019).
4. Grün, D., Kester, L. & van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nat. Methods* **11**, 637–640 (2014).
5. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nat. Methods* **11**, 740–742 (2014).
6. Ortega, A., Frossard, P., Kovačević, J., Moura, J. M. & Vandergheynst, P. Graph signal processing: overview, challenges, and applications. *Proc. IEEE* **106**, 808–828 (2018).
7. Coifman, R. R. & Lafon, S. Diffusion maps. *Appl. Comput. Harmon. Anal.* **21**, 5–30 (2006).
8. Mallat, S. *A Wavelet Tour of Signal Processing* (Elsevier, 1999).
9. Coifman, R. R. & Maggioni, M. Diffusion wavelets. *Appl. Comput. Harmon. Anal.* **21**, 53–94 (2006).
10. Data from: KLF2 maintains lineage fidelity and suppresses CD8 T cell exhaustion during acute LCMV infection (LCMV DSM scRNA data and ATAC-seq). *Dryad* https://doi.org/10.5061/dryad.dv41ns27h (2024).
11. Data from: KLF2 maintains lineage fidelity and suppresses CD8 T cell exhaustion during acute LCMV infection (PerturbSeq data). *Dryad* https://doi.org/10.5061/dryad.s7h44j1gr (2024).
12. Damo, M. et al. PD-1 maintains CD8 T cell tolerance towards cutaneous neoantigens. *Nature* **619**, 151–159 (2023).
13. V1 human lymph node, spatial gene expression dataset by Space Ranger 1.1.0. *10x Genomics* https://support.10xgenomics.com/spatial-gene-expression/datasets/1.1.0/V1_Human_Lymph_Node (2023).
14. Sade-Feldman, M. et al. Defining T cell states associated with response to checkpoint immunotherapy in melanoma. *Cell* **175**, 998–1013.e20 (2018).
15. xingzhis KrishnaswamyLab/Gene-Signal-Pattern-Analysis: GSPA v1.1. *Zenodo* https://doi.org/10.5281/zenodo.13953555 (2024).
16. Moon, K. R. et al. Manifold learning-based methods for analyzing single-cell RNA-sequencing data. *Curr. Opin. Syst. Biol.* **7**, 36–46 (2018).
17. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* **36**, 421–427 (2018).
18. Brugnone, N. et al. Coarse graining of data via inhomogeneous diffusion condensation. In *2019 IEEE International Conference on Big Data* 2624–2633 (IEEE, 2019).
19. Kuchroo, M. et al. Multiscale PHATE identifies multimodal signatures of COVID-19. *Nat. Biotechnol.* **40**, 681–691 (2022).
20. Tong, A. Y. et al. Diffusion earth mover's distance and distribution embeddings. In *Proc. 38th International Conference on Machine Learning* (eds Meila, M. & Zhang, T.) *Proc. Machine Learning Research* Vol. 139, 10336–10346 (PMLR, 2021).
21. Tong, A. et al. Embedding signals on graphs with unbalanced diffusion earth mover's distance. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing* 5647–5651 (IEEE, 2022).
22. Kiselev, V. Y., Andrews, T. S. & Hemberg, M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat. Rev. Genet.* **20**, 273–282 (2019).
23. Vandenbon, A. & Diez, D. A clustering-independent method for finding differentially expressed genes in single-cell transcriptome data. *Nat. Commun.* **11**, 4318 (2020).
24. Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18**, 174 (2017).
25. van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729.e27 (2018).
26. Kipf, T. N. & Welling, M. Variational graph auto-encoders. In *Advances in Neural Information Processing Systems Bayesian Deep Learning Workshop* (NIPS, 2016).
27. Grover, A. & Leskovec, J. node2vec: scalable feature learning for networks. In *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 855–864 (ACM, 2016).
28. Leone, S. et al. Graph Fourier MMD for signals on graphs. In *2023 International Conference on Sampling Theory and Applications* 1–6 (IEEE, 2023).
29. Hoekzema, R. S. et al. Multiscale methods for signal selection in single-cell data. *Entropy* **24**, 1116 (2022).
30. Choi, Y., Li, R. & Quon, G. siVAE: interpretable deep generative models for single-cell transcriptomes. *Genome Biol.* **24**, 29 (2023).
31. Chen, H., Ryu, J., Vinyard, M. E., Lerer, A. & Pinello, L. SIMBA: single-cell embedding along with features. *Nat. Methods* **21**, 1003–1013 (2024).
32. Kotliar, D. et al. Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-seq. *eLife* **8**, e43803 (2019).

33. 3k PBMCs from a healthy donor, single cell gene expression dataset by Cell Ranger 1.1.0. *10x Genomics* https://www.10xgenomics.com/datasets/3-k-pbm-cs-from-a-healthy-donor-1-standard-1-1-0 (2016).
34. Franzén, O., Gan, L.-M. & Björkegren, J. L. M. PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database* **2019**, baz046 (2019).
35. Giles, J. R. et al. Shared and distinct biological circuits in effector, memory and exhausted CD8$^+$ T cells revealed by temporal single-cell transcriptomics and epigenetics. *Nat. Immunol.* **23**, 1600–1613 (2022).
36. Grayson, J. M., Zajac, A. J., Altman, J. D. & Ahmed, R. Cutting edge: increased expression of *bcl-2* in antigen-specific memory CD8$^+$ T cells. *J. Immunol.* **164**, 3950–3954 (2000).
37. Wu, T. et al. The TCF1–Bcl6 axis counteracts type I interferon to repress exhaustion and maintain T cell stemness. *Sci. Immunol.* **1**, eaai8593 (2016).
38. McNab, F., Mayer-Barber, K., Sher, A., Wack, A. & O'Garra, A. Type I interferons in infectious disease. *Nat. Rev. Immunol.* **15**, 87–103 (2015).
39. Szklarczyk, D. et al. STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* **43**, D447–D452 (2014).
40. Yang, C. Y. et al. The transcriptional regulators Id2 and Id3 control the formation of distinct memory CD8$^+$ T cell subsets. *Nat. Immunol.* **12**, 1221–1229 (2011).
41. Sidwell, T. & Kallies, A. Bach2 is required for B cell and T cell memory differentiation. *Nat. Immunol.* **17**, 744–745 (2016).
42. Joshi, N. S. et al. Inflammation directs memory precursor and short-lived effector CD8$^+$ T cell fates via the graded expression of T-bet transcription factor. *Immunity* **27**, 281–295 (2007).
43. Armingol, E., Officer, A., Harismendy, O. & Lewis, N. E. Deciphering cell–cell interactions and communication from gene expression. *Nat. Rev. Genet.* **22**, 71–88 (2021).
44. Efremova, M., Vento-Tormo, M., Teichmann, S. A. & Vento-Tormo, R. CellPhoneDB: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes. *Nat. Protoc.* **15**, 1484–1506 (2020).
45. Kuchroo, M., Godavarthi, A., Tong, A., Wolf, G. & Krishnaswamy, S. Multimodal data visualization and denoising with integrated diffusion. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing* 1–6 (IEEE, 2021).
46. Kuchroo, M. et al. spARC recovers human glioma spatial signaling networks with graph filtering. Preprint at *bioRxiv* https://doi.org/10.1101/2022.08.24.505139 (2022).
47. Svensson, V., Teichmann, S. A. & Stegle, O. SpatialDE: identification of spatially variable genes. *Nat. Methods* **15**, 343–346 (2018).
48. Grasso, C. et al. Identification and mapping of human lymph node stromal cell subsets by combining single-cell RNA sequencing with spatial transcriptomics. Preprint at *bioRxiv* https://doi.org/10.1101/2023.08.18.553530 (2023).
49. Kleshchevnikov, V. et al. cell2location maps fine-grained cell types in spatial transcriptomics. *Nat. Biotechnol.* **40**, 661–671 (2022).
50. Türei, D. et al. Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. *Mol. Syst. Biol.* **17**, e9923 (2021).
51. Pardoll, D. M. The blockade of immune checkpoints in cancer immunotherapy. *Nat. Rev. Cancer* **12**, 252–264 (2012).
52. Fuertes Marraco, S. A., Neubert, N. J., Verdeil, G. & Speiser, D. E. Inhibitory receptors beyond T cell exhaustion. *Front. Immunol.* **6**, 310 (2015).
53. Tirosh, I. et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196 (2016).
54. Connolly, K. A. et al. A reservoir of stem-like CD8$^+$ T cells in the tumor-draining lymph node preserves the ongoing antitumor immune response. *Sci. Immunol.* **6**, eabg7836 (2021).
55. Kurtulus, S. et al. Checkpoint blockade immunotherapy induces dynamic changes in PD-1–CD8$^+$ tumor-infiltrating T cells. *Immunity* **50**, 181–194.e6 (2019).
56. Zehn, D., Thimme, R., Lugli, E., de Almeida, G. P. & Oxenius, A. 'Stem-like' precursors are the fount to sustain persistent CD8$^+$ T cell responses. *Nat. Immunol.* **23**, 836–847 (2022).
57. Sade-Feldman, M. et al. Resistance to checkpoint blockade therapy through inactivation of antigen presentation. *Nat. Commun.* **8**, 1136 (2017).
58. Morinaga, T. et al. Mixed response to cancer immunotherapy is driven by intratumor heterogeneity and differential interlesion immune infiltration. *Cancer Res. Commun.* **2**, 739–753 (2022).
59. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33**, 495–502 (2015).
60. Perlmutter, M., Tong, A., Gao, F., Wolf, G. & Hirn, M. Understanding graph neural networks with generalized geometric scattering transforms. *SIAM J. Math. Data Sci.* **5**, 873–898 (2023).
61. Chew, J. et al. Geometric scattering on measure spaces. *Appl. Comput. Harmon. Anal.* **70**, 101635 (2024).
62. Burkhardt, D. B. et al. Quantifying the effect of experimental perturbations at single-cell resolution. *Nat. Biotechnol.* **39**, 619–629 (2021).
63. Beylkin, G., Coifman, R. & Rokhlin, V. Fast wavelet transforms and numerical algorithms I. *Commun. Pure Appl. Math.* **44**, 141–183 (1991).
64. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019).
65. McGinnis, C. S., Murrow, L. M. & Gartner, Z. J. DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. *Cell Syst.* **8**, 329–337.e4 (2019).
66. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902.e21 (2019).
67. Knapp, T. R. Canonical correlation analysis: a general parametric significance-testing system. *Psychol. Bull.* **85**, 410–416 (1978).
68. Wolf, F. A., Angerer, P. & Theis, F. J. scanpy: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
69. van Dijk, D. et al. Finding archetypal spaces using neural networks. In *2019 IEEE International Conference on Big Data* 2634–2643 (IEEE, 2019).
70. Venkat, A. et al. AAnet resolves a continuum of spatially-localized cell states to unveil tumor complexity. Preprint at *bioRxiv* https://doi.org/10.1101/2024.05.11.593705 (2024).
71. Xie, Z. et al. Gene set knowledge discovery with enrichr. *Curr. Protoc.* **1**, e90 (2021).
72. Huang, R. et al. The NCATS BioPlanet—an integrated platform for exploring the universe of cellular signaling pathways for toxicology, systems biology, and chemical genomics. *Front. Pharmacol.* **10**, 445 (2019).
73. Shannon, P. et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).
74. Palla, G. et al. Squidpy: a scalable framework for spatial omics analysis. *Nat. Methods* **19**, 171–178 (2022).
75. Jin, S. et al. Inference and analysis of cell–cell communication using CellChat. *Nat. Commun.* **12**, 1088 (2021).
76. Venkat, A. KrishnaswamyLab/GSPA-manuscript-analyses: GSPA v.0.0. *Zenodo* https://doi.org/10.5281/zenodo.13953558 (2024).

Acknowledgements

We thank the Krishnaswamy Lab members who provided feedback on this paper. A.V. acknowledges funding from the Gruber Foundation

Science Fellowship. M.P. is funded by NSF DMS grant number 2327211 and NSF OIA grant number 2242769. S.K. is funded by NSF Career Grant number 2047856, NSF DMS grant 2327211 and NSF CISE grant 2403317. E.F. is funded by NIAID/NIH grant number T32 AI155387. N.S.J. is funded by the Mark Foundation Emerging Leader Award.

Author contributions

A.V., S.L., S.E.Y. and S.K. developed and implemented GSPA. A.V. designed and performed computational analyses, overseen by S.K. E.F. and J.A. performed scRNA-seq and perturbation experiments, overseen by N.S.J. A.V., M.P. and S.K. wrote the paper, with substantial contributions from all authors.

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s43588-024-00734-0.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-024-00734-0.

Correspondence and requests for materials should be addressed to Smita Krishnaswamy.

Peer review information *Nature Computational Science* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editor: Ananya Rastogi, in collaboration with the *Nature Computational Science* team.

https://doi.org/10.1038/s43588-024-00734-0

Extended Data Table 1 | Ranking for each coexpression and localization test across comparisons

1a_coexpression													
dataset	**graph**	**GSPA**	**Eigen**	**GSPA+QR**	**MAGIC**	**Raw**	**DEMD**	**GFMMD**	**siVAE**	**GAEatt**	**SIMBA**	**Node2Vec**	**GAEno-att**
linear	knn_5_SNN	0	1	2	3	4	5	6	7	8	9	10	11
linear	knn_25_kNN	0	2	1	3	4	7	5	6	8	9	10	11
linear	knn_25_adaptive	1	0	2	3	4	6	5	7	8	9	10	11
linear	knn_15_SNN	1	0	2	3	4	7	5	6	8	9	10	11
linear	knn_15_adaptive_sqrt	2	1	0	3	4	5	6	7	8	10	9	11
linear	knn_5_adaptive	1	2	0	3	4	7	5	6	8	9	10	11
linear	knn_50_SNN_sqrt	1	3	0	2	4	6	5	7	8	10	9	11
linear	knn_5_adaptive_sqrt	2	1	0	3	4	5	6	7	8	10	9	11
linear	knn_15_kNN	1	0	2	3	4	7	5	6	8	9	10	11
linear	knn_50_kNN_sqrt	1	2	0	3	4	6	5	7	8	10	9	11
linear	knn_15_SNN_sqrt	2	1	0	3	4	6	5	7	8	10	9	11
linear	knn_15_kNN_sqrt	1	2	0	3	4	6	5	7	8	10	9	11
linear	knn_5_kNN	0	1	2	3	4	5	6	7	8	9	10	11
linear	knn_50_adaptive_sqrt	2	1	0	3	4	6	5	7	8	10	9	11
linear	knn_5_kNN_sqrt	2	1	0	3	4	5	6	7	8	10	9	11
linear	knn_25_SNN	0	2	1	3	4	7	5	6	8	9	10	11
linear	knn_25_adaptive_sqrt	2	1	0	3	4	5	6	7	8	10	9	11
linear	knn_50_SNN	1	3	0	2	4	7	5	6	8	9	10	11
linear	knn_50_kNN	1	2	0	3	4	7	5	6	8	9	10	11
linear	knn_25_SNN_sqrt	1	2	0	3	4	6	5	7	8	10	9	11
linear	knn_15_adaptive	1	0	2	3	4	6	5	7	8	9	10	11
linear	knn_25_kNN_sqrt	1	2	0	3	4	6	5	7	8	10	9	11
linear	knn_5_SNN_sqrt	2	1	0	3	4	5	6	7	8	10	9	11
linear	knn_50_adaptive	2	0	1	3	4	7	5	6	8	9	10	11
two_branches	knn_5_adaptive_sqrt	1	2	0	3	4	5	8	6	9	10	11	7
three_branches	knn_5_adaptive_sqrt	1	2	0	3	4	5	11	8	7	6	9	10
1b_localization													
dataset	**graph**	**GSPA**	**Eigen**	**GSPA+QR**	**MAGIC**	**Raw**	**DEMD**	**GFMMD**	**siVAE**	**GAEatt**	**SIMBA**	**Node2Vec**	**GAEno-att**
linear	knn_25_SNN	0	1	2	3	4	5	6	7	8	9	10	11
linear	knn_5_adaptive_sqrt	0	1	4	3	2	6	8	7	5	9	10	11
linear	knn_15_adaptive	3	1	2	4	0	5	6	8	7	9	10	11
linear	knn_15_SNN	0	1	4	3	2	5	6	8	7	9	10	11
linear	knn_50_SNN	1	0	2	3	4	5	7	8	6	9	10	11
linear	knn_15_adaptive_sqrt	0	1	4	3	2	6	8	7	5	9	10	11
linear	knn_5_kNN_sqrt	1	2	4	3	0	6	8	7	5	9	10	11
linear	knn_5_adaptive	1	0	2	3	4	5	6	8	7	9	10	11
linear	knn_50_kNN	2	1	3	4	0	5	7	8	6	9	10	11
linear	knn_25_kNN	1	2	3	4	0	5	6	7	8	9	10	11
linear	knn_5_SNN_sqrt	1	2	4	3	0	6	8	7	5	9	10	11
linear	knn_25_adaptive_sqrt	0	1	3	4	2	6	8	7	5	9	10	11
linear	knn_15_SNN_sqrt	0	1	4	3	2	6	8	7	5	9	10	11
linear	knn_15_kNN	1	2	3	4	0	5	6	8	7	9	10	11
linear	knn_25_SNN_sqrt	0	1	4	3	2	5	8	7	6	9	10	11
linear	knn_50_SNN_sqrt	0	1	2	3	4	5	8	7	6	9	10	11
linear	knn_25_kNN_sqrt	1	2	4	3	0	5	8	7	6	9	10	11
linear	knn_5_kNN	0	2	4	3	1	5	7	8	6	9	10	11
linear	knn_50_kNN_sqrt	1	2	3	4	0	5	8	7	6	9	10	11

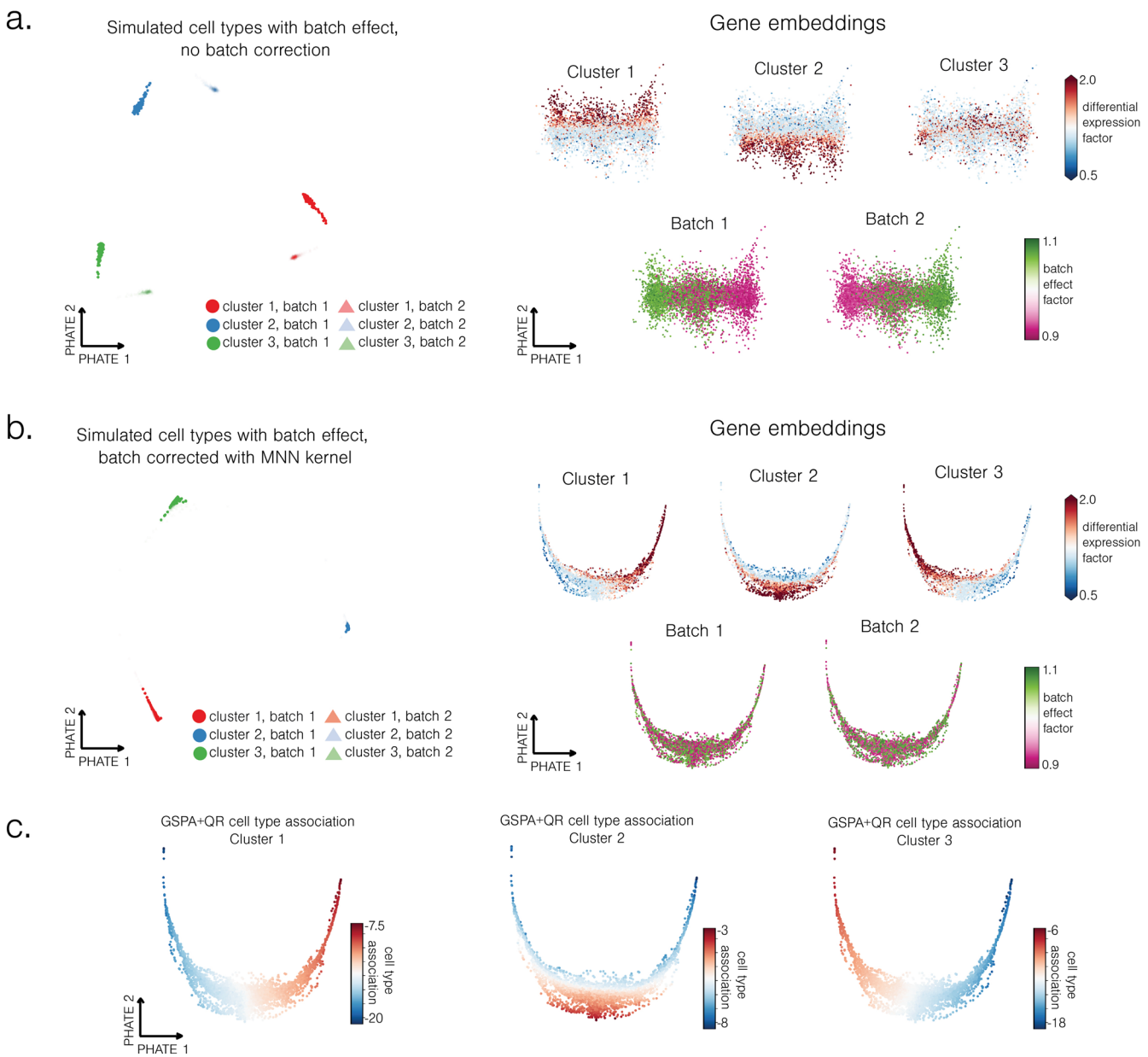
https://doi.org/10.1038/s43588-024-00734-0

Extended Data Table 1 (continued) | Ranking for each coexpression and localization test across comparisons

1b_localization													
dataset	**graph**	**GSPA**	**Eigen**	**GSPA+QR**	**MAGIC**	**Raw**	**DEMD**	**GFMMD**	**siVAE**	**GAEatt**	**SIMBA**	**Node2Vec**	**GAEno-att**
linear	knn_25_adaptive	4	1	2	3	0	5	6	7	8	9	10	11
linear	knn_15_kNN_sqrt	1	2	4	3	0	6	8	7	5	9	10	11
linear	knn_50_adaptive_sqrt	0	2	3	4	1	6	8	7	5	9	10	11
linear	knn_5_SNN	0	2	4	3	1	5	7	8	6	9	10	11
linear	knn_50_adaptive	3	1	2	4	0	5	6	7	8	9	10	11
two_branches	knn_5_adaptive_sqrt	0	1	5	2	3	8	6	7	4	10	9	11
three_branches	knn_5_adaptive_sqrt	0	2	4	3	6	7	5	8	1	10	9	11

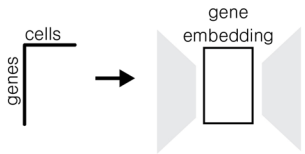
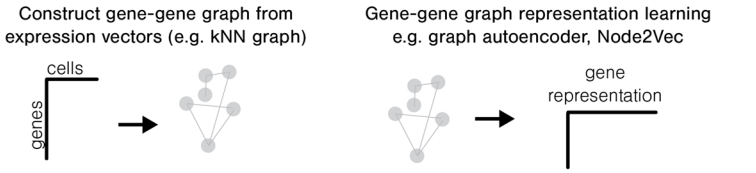
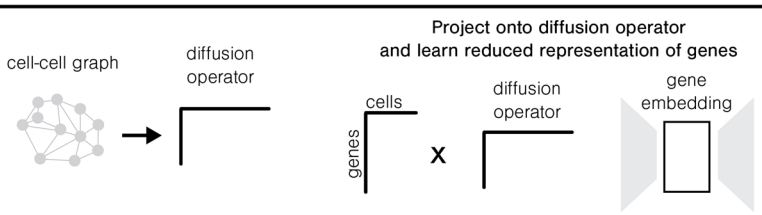
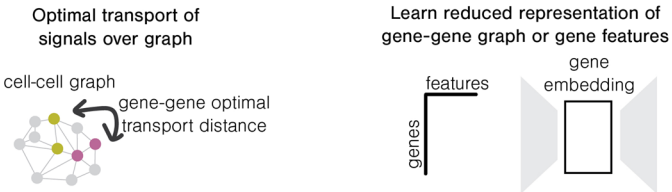
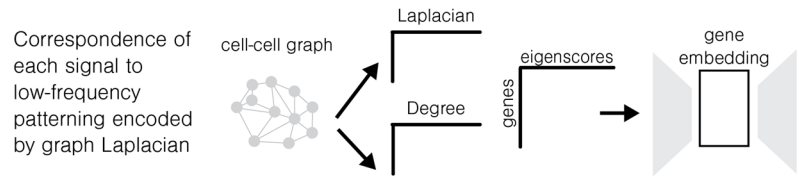
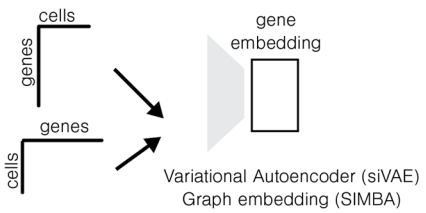
a. Ranking of approaches for each coexpression prediction test on linear, two branch, and three branch simulated data. **b**. Ranking of approaches for each localization prediction test on linear, two branch, and three branch simulated data.

https://doi.org/10.1038/s43588-024-00734-0



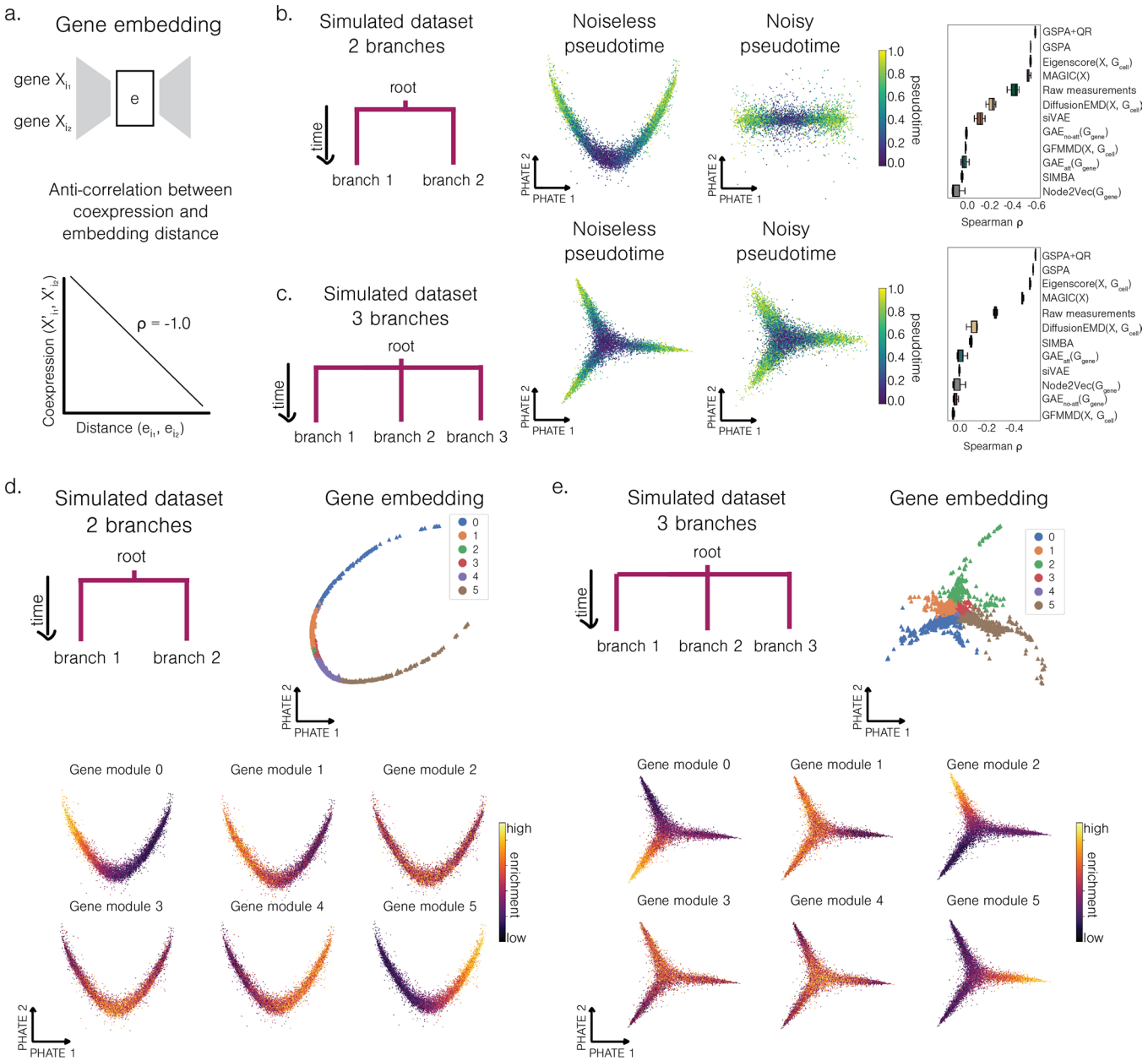
Extended Data Fig. 1 | Batch effect robustness in GSPA. a. Dataset simulated with 3 clusters and 2 batches with batch effect. Gene embeddings colored by ground truth cluster association (differential expression factor) and batch effect association (batch effect factor) show separation by both. **b.** Dataset with batch effect corrected. Gene embeddings separate by cluster, but not batch effect. **c.** GSPA cell type association score correctly identifies relationship between genes and each cluster.

https://doi.org/10.1038/s43588-024-00734-0

Comparison names	Method	Diagram of methodology	Uses cell-cell graph?
X (i.e. Raw measure-ments)	Embedding gene signals		No
$GAE_{att}(G_{gene})$ $GAE_{no\text{-}att}(G_{gene})$ $Node2Vec(G_{gene})$	Embedding constructed gene-gene graph		No
MAGIC(X)	Imputing gene signals with cell-cell graph		Yes
$DiffusionEMD(X, G_{cell})$ $GFMMD(X, G_{cell})$	Optimally transporting gene signals		Yes
$Eigenscore(X,G_{cell})$	$\frac{\langle D^{1/2} f, e^i \rangle}{\lvert\lvert D^{1/2} f \rvert\rvert}$ where e^i is the ith eigenvector of graph Laplacian and f is each signal in X		Yes
SIMBA siVAE	Co-embedding cells and genes		No

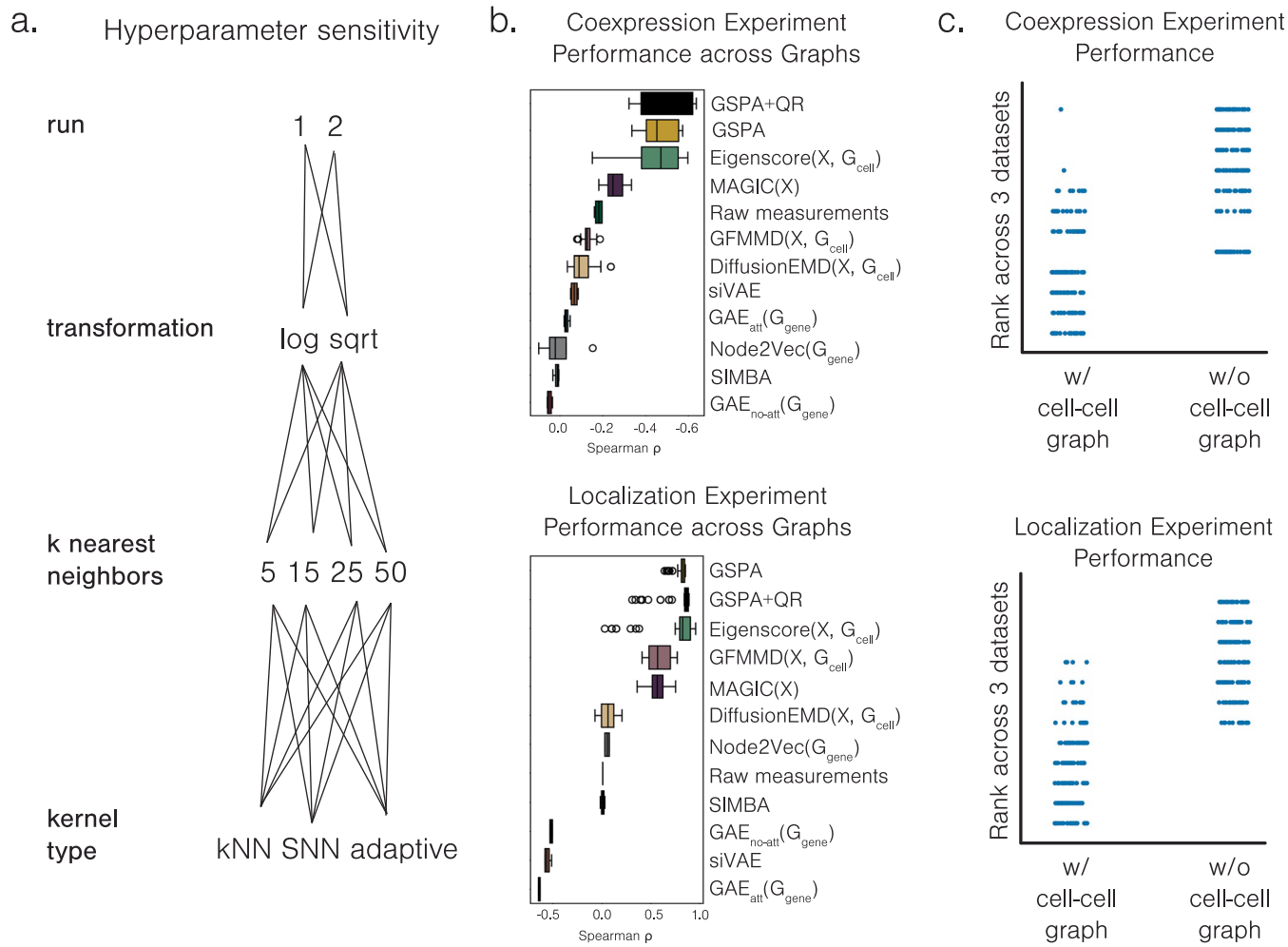
Extended Data Fig. 2 | Overview of Gene Signal Pattern Analysis Comparisons. Comparison names, methodology in text and diagram, and use of cell-cell graph based on shared properties of comparison.

https://doi.org/10.1038/s43588-024-00734-0



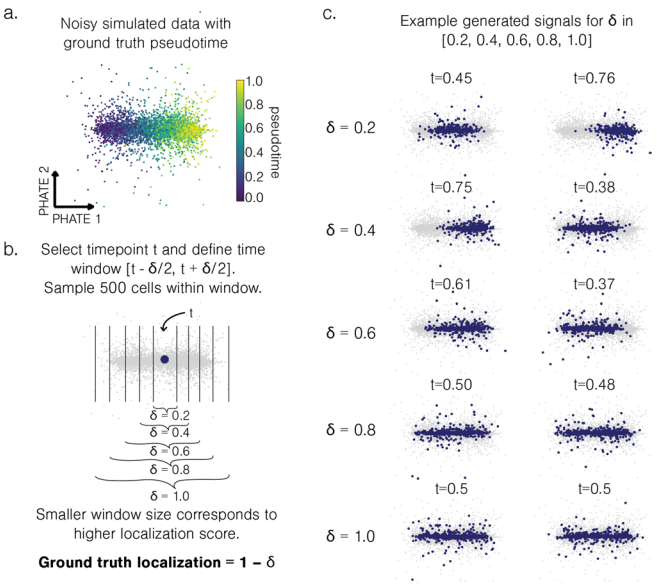
Extended Data Fig. 3 | Coexpression preservation in two-branch, three-branch single-cell simulations. a. Experimental setup. **b.** Simulated dataset with two branches schematic. PHATE embedding of cells from noiseless simulation and noisy simulation, colored by pseudotime. Spearman correlation evaluating performance for all comparisons across 3 runs. **c.** Simulated dataset with three branches schematic. PHATE embedding of cells from noiseless simulation and noisy simulation, colored by pseudotime. Spearman correlation evaluating performance for all comparisons across 3 runs. **d.** PHATE embedding of genes from two branch simulation, colored by gene module assignments. Cells colored by gene module enrichment score. **e.** PHATE embedding of genes from three branch simulation, colored by gene module assignments. Cells colored by gene module enrichment score.

https://doi.org/10.1038/s43588-024-00734-0



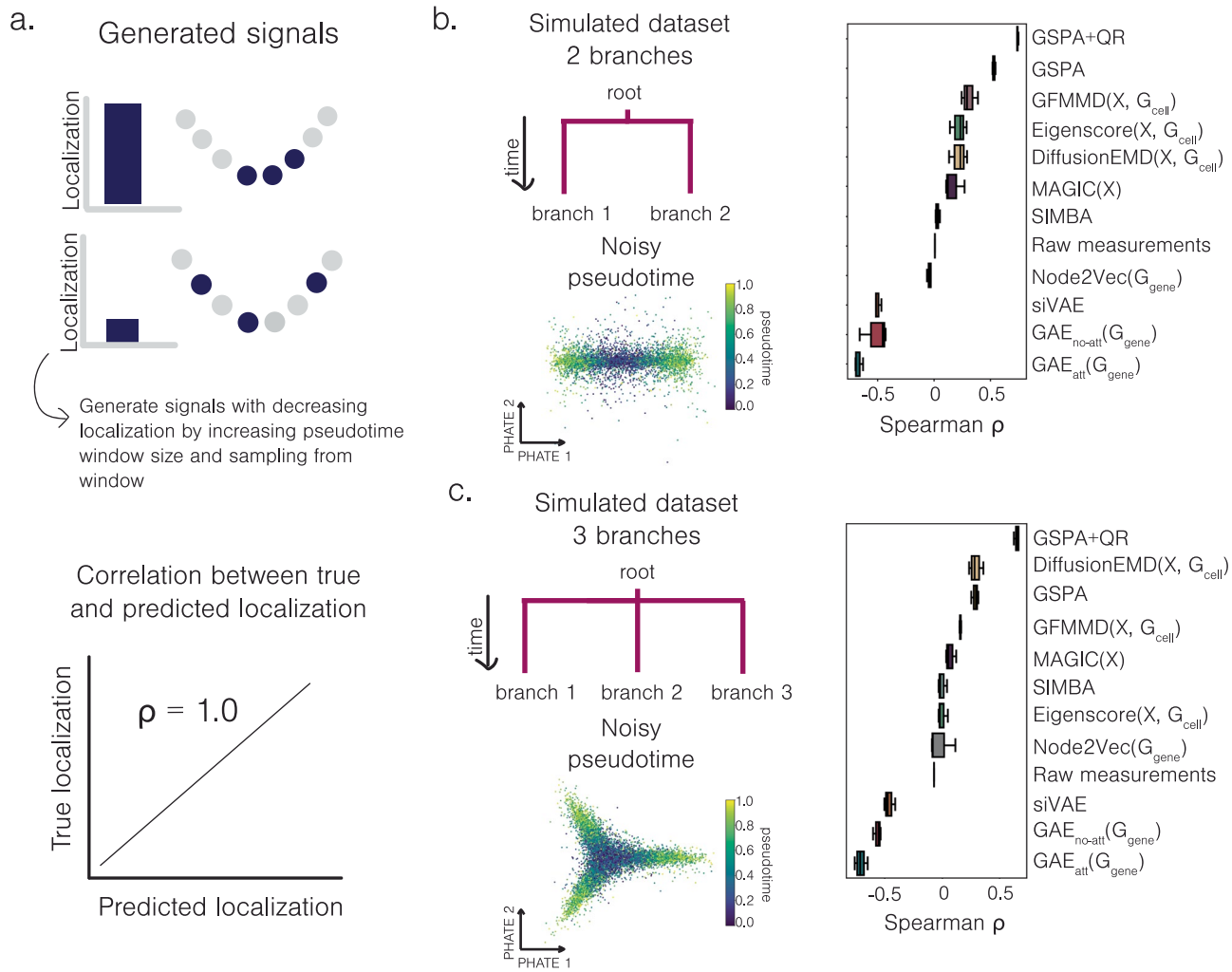
Extended Data Fig. 4 | Transformation and graph construction robustness in GSPA. a. Schematic of grid search of 2 transformations, 4 kNN choices, 3 kernels, and 2 replicates (48 runs total). **b**. Coexpression and localization experiment performance across all runs. **c**. Comparison of performance rank of methods that use cell-cell graph versus without cell-cell graph.

https://doi.org/10.1038/s43588-024-00734-0



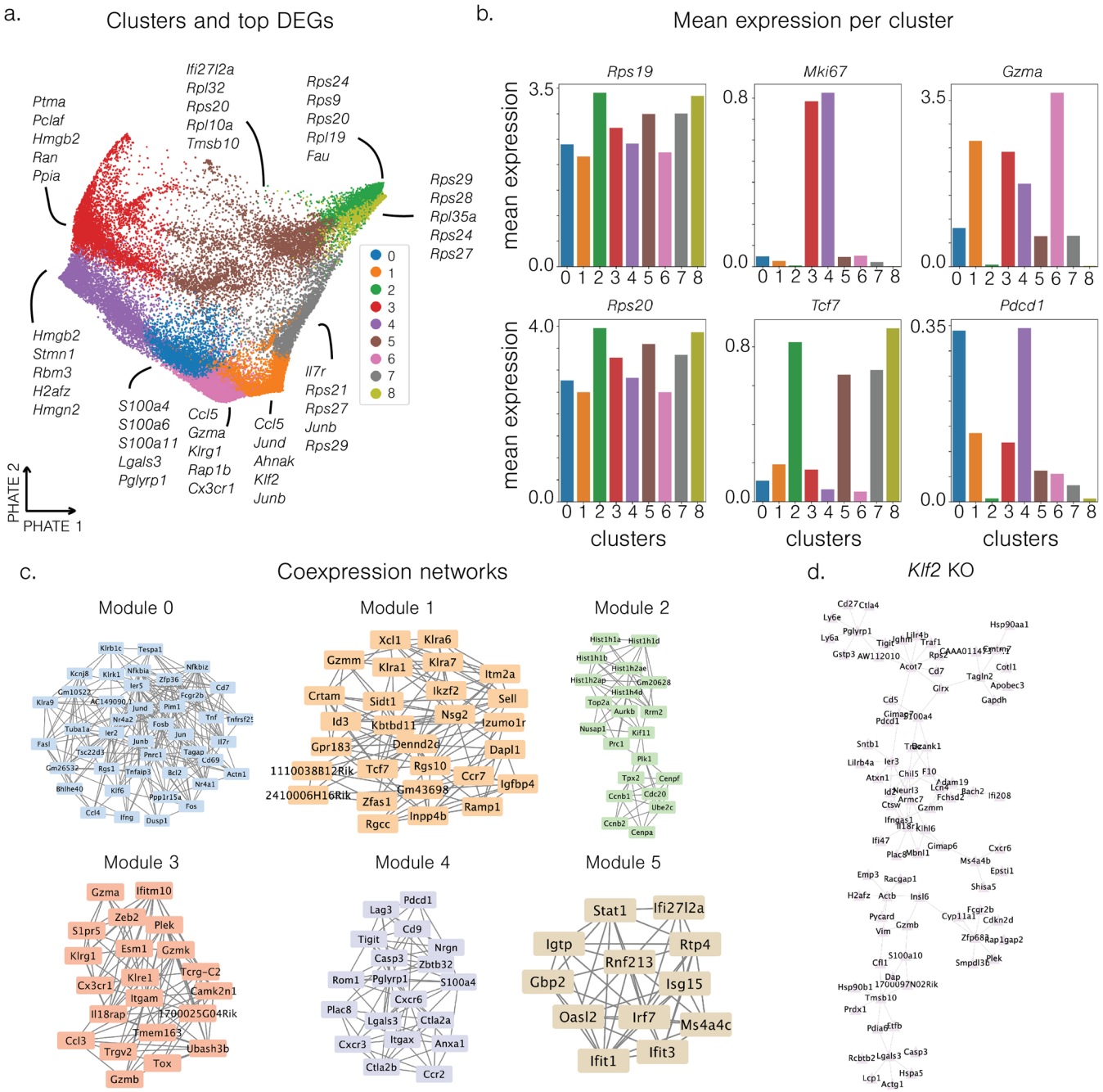
Extended Data Fig. 5 | Schematic of generation of signals for localization experiment. a. Noisy simulated data with pseudotime. **b.** Selection of windows of size δ where ground truth localization is $1-\delta$. **c.** Examples of generated signals of different δ.

https://doi.org/10.1038/s43588-024-00734-0



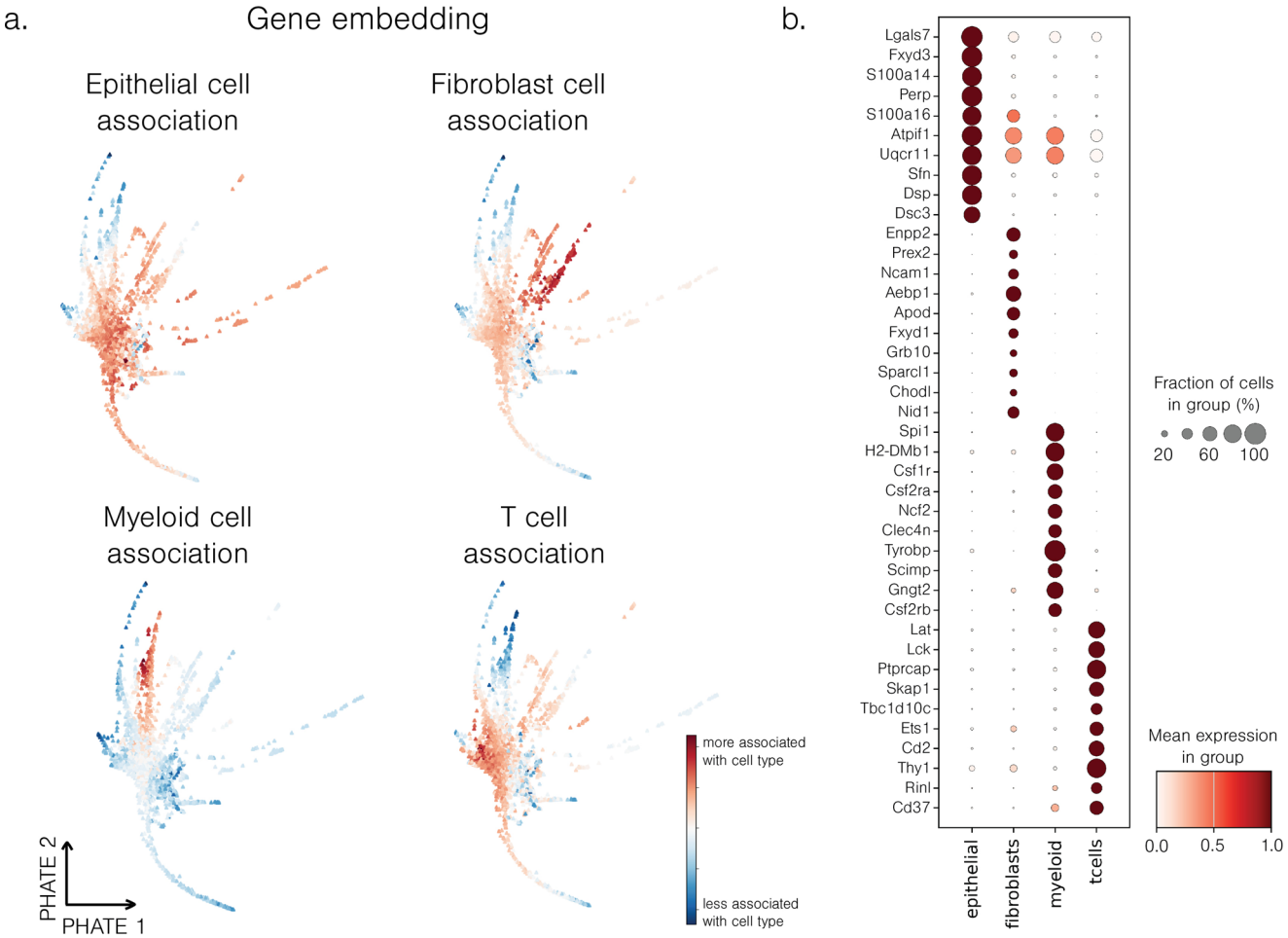
Extended Data Fig. 6 | Differential localization in two-branch, three branch single-cell simulations. a. Diagram of generated signals based on pseudotime window and anti-correlation between window size and localization. **b.** Two branch noisy simulated dataset, visualized with PHATE and colored by pseudotime. Spearman correlation evaluating performance for all comparisons across 3 runs. **c.** Three branch noisy simulated dataset, visualized with PHATE and colored by pseudotime. Spearman correlation evaluating performance for all comparisons across 3 runs.

https://doi.org/10.1038/s43588-024-00734-0



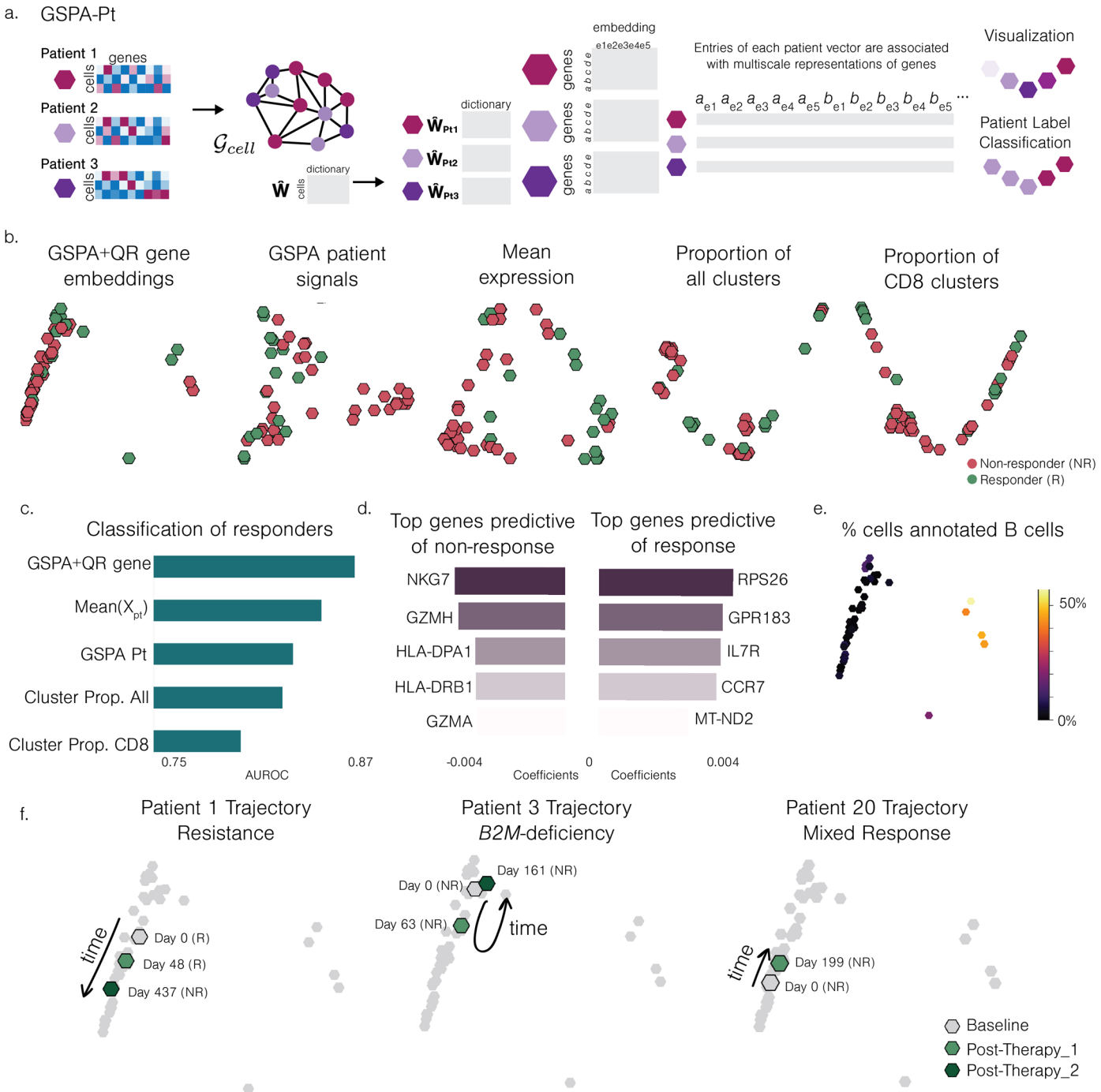
Extended Data Fig. 7 | Extended GSPA and comparison analysis for CD8+ T cells. a. Cells clustered with top DEGs identified. **b**. Mean expression of top DEGs *Rps19* and Rps20 and key CD8+ T cell marker genes per cluster. **c**. Coexpression networks of top localized genes in each gene cluster. **d**. *Klf2* KO network.

https://doi.org/10.1038/s43588-024-00734-0



Extended Data Fig. 8 | Cell type association scores for peripheral tolerance model. a. Gene embedding colored by cell type association ranking. **b.** Dot plot with top 10 genes associated with each cell type.

https://doi.org/10.1038/s43588-024-00734-0



Extended Data Fig. 9 | Response trajectories and biomarkers revealed by multiscale GSPA patient manifold. a. Schematic of GSPA-Pt. **b.** PHATE visualization of patient embeddings based on GSPA+QR gene embeddings and comparisons. **c.** AUROC evaluation of response classification (logistic regression). **d.** Top genes predictive of response and non-response based on highest and lowest logistic regression coefficients. **e.** Patient embedding colored by percent of total cells annotated as B cells. **f.** Patient embedding with three samples from patient 1, 3, and 20 highlighted, corresponding to samples obtained from patients over time (pre-therapy baseline, post-therapy-1, and post-therapy-2). Trajectories of samples visualized.

nature portfolio

Corresponding author(s): Smita Krishnaswamy

Last updated by author(s): Oct 21, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Simulated data was generated with Splatter implemented in Python via scprep 1.2.3. https://github.com/KrishnaswamyLab/Gene-Signal-Pattern-Analysis (DOI: 10.5281/zenodo.13953555); https://github.com/KrishnaswamyLab/GSPA-manuscript-analyses (DOI: 10.5281/zenodo.13953559)
Data analysis	Python 3.8.18 with graphtools 1.5.3, tensorflow 2.13.0, keras 2.13.1, numpy 1.22.4, sklearn 1.3.2, scipy 1.10.1, tqdm 4.66.4, scanpy 1.9.3, phate 1.0.11; https://github.com/KrishnaswamyLab/Gene-Signal-Pattern-Analysis (DOI: 10.5281/zenodo.13953555); https://github.com/KrishnaswamyLab/GSPA-manuscript-analyses (DOI: 10.5281/zenodo.13953559)

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The scRNA-seq data of CD8+ T cells during acute and chronic LCMV infection, and the Perturb-seq data during acute LCMV infection, are available on the Dryad repository at [10] and [11], respectively. scRNA-seq datasets for peripheral tolerance in the skin are available from the Gene Expression Omnibus (GEO) database under GEO Series Accession number GSE228586 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE228586), originally from [12]. Visium spatial gene expression data for the human lymph node is from 10x Genomics [13]. The scRNA-seq data for melanoma samples pre- and post- therapy are accessible from the GEO database through GEO Series Accession number GSE120575 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120575), originally from [14]. We additionally used the following data resources for our case studies: OmniPath (https://omnipathdb.org) [50], STRINGdb (https://string-db.org) [39], Enrichr (https://maayanlab.cloud/Enrichr/) [71], and CellChatDB (http://www.cellchat.org) [75].

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	N/A
Reporting on race, ethnicity, or other socially relevant groupings	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	3-5 mice were infected for each timepoint and condition in a staggered manner for single-cell RNA sequencing of CD8+CD44+Tetramer+ T cells during acute and chronic LCMV infection. Equal numbers of P14s were pooled from 7 mice prior to sorting for CRISPR sequencing. For in silico analysis, simulated datasets were generated with 10,000 cells and 10,000 genes to represent a single lane of the 10X instrument and high-dimensional measurements. Each comparison was tested with three random seeds for the coexpression and localization tests.
Data exclusions	Low quality cell and gene measurements were excluded from in silico analysis based on standard protocols for single-cell analysis. This has been detailed in the Methods section.
Replication	For scRNA-seq data, cells were pooled from 3-5 mice (biological replicates) prior to sorting, and for the CRISPR data, cells were pooled in equal numbers from 7 mice prior to sorting. The full experimental protocol has been provided in the manuscript, ensuring replication across mice. For in silico analysis, coexpression and localization experiments were repeated three times per comparison for 10 comparisons, and robustness analysis was performed on different data normalizations and graph constructions (50 runs total per method). Additionally, we have provided the codebase for GSPA and GSPA manuscript analyses.
Randomization	All mice were biological replicates and randomly assigned to groups before infection. For in silico analysis, all methods were run with randomly chosen seed.
Blinding	All cells were pooled before sequencing, so data collection after sequencing was blinded.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks

Report on the source of all seed stocks or other plant material used. If applicable, state the seed stock centre and catalogue number. If plant specimens were collected from the field, describe the collection location, date and sampling procedures.

Novel plant genotypes

Describe the methods by which all novel plant genotypes were produced. This includes those generated by transgenic approaches, gene editing, chemical/radiation-based mutagenesis and hybridization. For transgenic lines, describe the transformation method, the number of independent lines analyzed and the generation upon which experiments were performed. For gene-edited lines, describe the editor used, the endogenous sequence targeted for editing, the targeting guide RNA sequence (if applicable) and how the editor was applied.

Authentication

Describe any authentication procedures for each seed stock used or novel genotype generated. Describe any experiments used to assess the effect of a mutation and, where applicable, how potential secondary effects (e.g. second site T-DNA insertions, mosiacism, off-target gene editing) were examined.