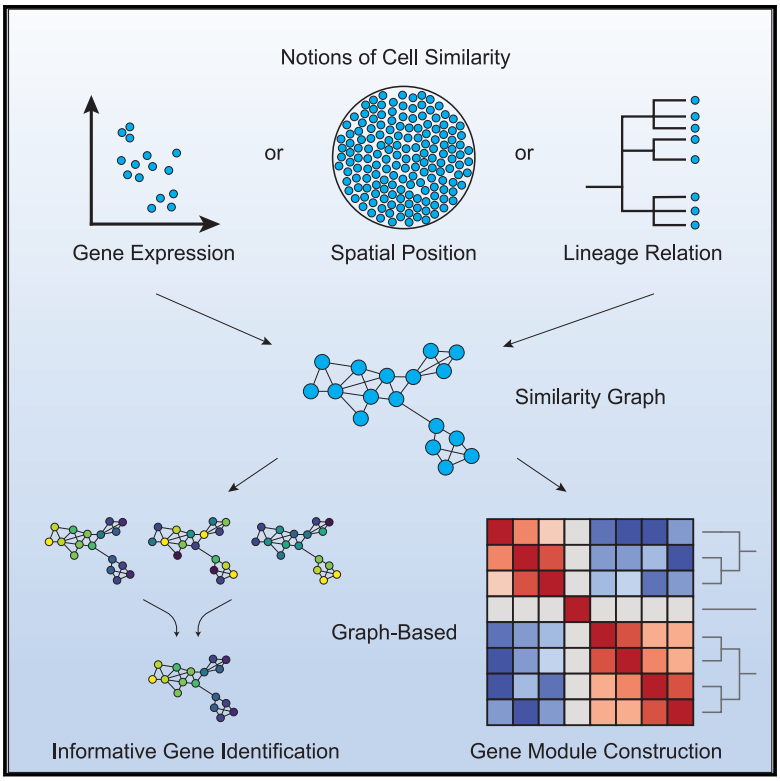
Cell Systems

Report

Hotspot identifies informative gene modules across modalities of single-cell genomics

Graphical abstract



Authors

David DeTomaso, Nir Yosef

Correspondence

niryosef@berkeley.edu

In brief

When analyzing single-cell RNA-seq data, it is often instructive to identify genes that exhibit patterns of variation indicative of the underlying milieu of cell types and states, and to further organize these genes into coordinated modules. DeTomaso and Yosef propose a novel, graph-based method for this purpose with improved sensitivity under sparse conditions. They further show how this procedure can be used to detect informative genes in multimodal settings, with applications for detecting spatially regulated gene modules and heritable expression programs.

Highlights

- Hotspot is a graph-based procedure to identify informative genes and gene modules
- Information from similar cells is shared to improve sensitivity in sparse data
- The method can be applied downstream of most dimensionality reduction procedures
- It can also be used to detect gene modules in multimodal settings (RNA + x)

DeTomaso & Yosef, 2021, Cell Systems *12*, 446–456
May 19, 2021
https://doi.org/10.1016/j.cels.2021.04.005

CellPress

CellPress
OPEN ACCESS

Cell Systems

Report

Hotspot identifies informative gene modules across modalities of single-cell genomics

David DeTomaso[1] and Nir Yosef[2,3,4,5,*]
[1]Center for Computational Biology, University of California Berkeley, Berkeley, CA, USA
[2]Department of Electrical Engineering and Computer Science and Center for Computational Biology, University of California Berkeley, Berkeley, CA, USA
[3]Ragon Institute of Massachusetts General Hospital, MIT and Harvard, Cambridge, MA, USA
[4]Chan Zuckerberg Biohub, San Francisco, CA, USA
[5]Lead contact
*Correspondence: niryosef@berkeley.edu
https://doi.org/10.1016/j.cels.2021.04.005

SUMMARY

Two fundamental aims that emerge when analyzing single-cell RNA-seq data are identifying which genes vary in an informative manner and determining how these genes organize into modules. Here, we propose a general approach to these problems, called "Hotspot," that operates directly on a given metric of cell-cell similarity, allowing for its integration with any method (linear or non-linear) for identifying the primary axes of transcriptional variation between cells. In addition, we show that when using multimodal data, Hotspot can be used to identify genes whose expression reflects alternative notions of similarity between cells, such as physical proximity in a tissue or clonal relatedness in a cell lineage tree. In this manner, we demonstrate that while Hotspot is capable of identifying genes that reflect nuanced transcriptional variability between T helper cells, it can also identify spatially dependent patterns of gene expression in the cerebellum as well as developmentally heritable expression programs during embryogenesis. Hotspot is implemented as an open-source Python package and is available for use at http://www.github.com/yoseflab/hotspot. A record of this paper's transparent peer review process is included in the supplemental information.

INTRODUCTION

Transcriptome-scale profiling at a single-cell resolution has enabled comprehensive categorization of cell types and states in diverse tissues (Grün et al., 2015; Macosko et al., 2015; Zeisel et al., 2015), investigation of developmental transitions (Trapnell et al., 2014; Tusi et al., 2018), and in-depth characterization of disease processes (Gaublomme et al., 2015; Tirosh et al., 2016). While initial studies focused on the estimation of the transcriptomes in small numbers of cells, improvements in technology are rapidly increasing the number of cells per sample (Svensson et al., 2018b). Furthermore, the emergence of multimodal single-cell technologies now enables simultaneous profiling of cell transcriptomes along with cellular DNA (Macaulay et al., 2015), cell surface proteomes (Stoeckius et al., 2017), spatial position (Ståhl et al., 2016; Rodriques et al., 2019), epigenetic properties (Angermueller et al., 2016; Buenrostro et al., 2015), or cell lineages (Chan et al., 2019).

Two primary problems that emerge in the analysis of any high-dimensional gene expression data are the identification of genes that vary in an informative manner and the organization of these genes into co-varying groups (modules). In the context of single-cell genomics, these tasks are complicated in two critical ways. First, the data tend to be noisy and suffer from a limited level of sensitivity, whereby expressed genes often remain undetected (Satija et al., 2015; Lopez et al., 2018). Second, there is a growing need to account for additional data modalities (when available) when highlighting the most informative gene expression programs. For instance, when provided with the spatial location of each cell, a natural question is "which genes' transcription may be influenced by a cell's location in a tissue?" When provided with lineage information, one may ask "which gene expression programs reflect the clonal relatedness between cells, and are thus possibly heritable?"

Although the tasks of identifying informative genes and gene modules are not new, here we present a fundamentally different approach with properties that help address the two challenges mentioned earlier. Our method, called "Hotspot," is based on the notion that if some measure of biological proximity can be evaluated between cells, then genes that are expressed at similar levels by proximal cells must be varying in a non-random and possibly informative manner. The specific interpretation of such genes depends on the property used for evaluating cell-cell proximity. For instance, if proximity is defined by the transcriptional state of the cell, then Hotspot will detect genes that are indicative of cell types, sub-types, and primary phenotypic gradients. If it reflects the position of a cell in a tissue, then the genes returned by Hotspot may reflect the composition of cell

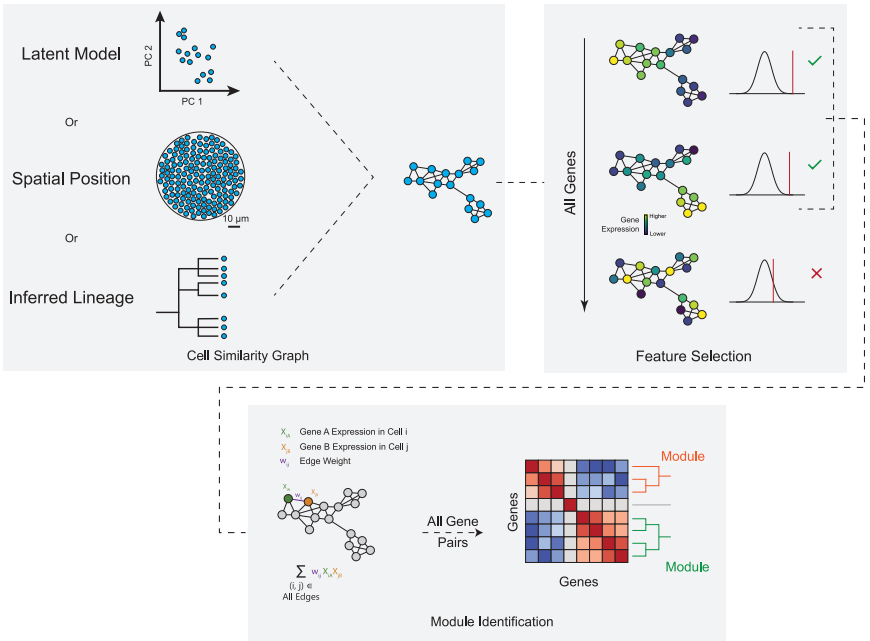


Figure 1. Schematic representation of the Hotspot procedure

The three main components of the Hotspot procedure are visualized. First, a cell similarity graph is created based on a provided input metric. Second, feature selection is used to filter the full list of genes to those which exhibit highly non-random expression patterns in the similarity graph. Finally, pairwise evaluations are conducted between this reduced set of genes in order to construct a *Z* score matrix (analogous to a correlation matrix), which is then clustered to form gene modules.

types in different spatial niches. Since the evaluation of cell-cell proximity is decoupled from the algorithm itself, Hotspot can be applied to a variety of other similarity measures (e.g., reflecting the cell's genome, epigenome, or its position in a cell lineage) with the only requirement for these measures to be abstracted as a weighted graph (connecting proximal cells; Figure 1A). While this helps address the second challenge mentioned earlier, Hotspot also leverages the graph structure to more robustly evaluate when a gene is being expressed at similar levels between proximal cells, thus mitigating some of the effects of low sensitivity (Figure 1B). Finally, we extend our method to identify *pairs* of genes that have similar expression profiles and use this measure to organize the informative genes into modules (Figure 1C).

When applied to a metric of transcriptome similarity, Hotspot effectively addresses the well-studied problems of selection and clustering of genes in single-cell RNA sequencing (RNA-seq) data. Many of the popular methods to the selection problem rank genes based on the marginal distribution of their expression (such as the highly variable genes procedure in Seurat [Satija et al., 2015] or NBDisp [Andrews and Hemberg, 2019]). A caveat of this approach is that it may neglect genes that do not exhibit overdispersion (or some other desirable property of their marginal distribution) and yet vary in an informative manner (e.g., by co-varying with many other genes). One alternative to this approach is to take a transcriptome-wide approach and select genes that explain the primary axes of variation in linear models of cell-cell similarity (e.g., with the principal component analysis [PCA], factor analysis [FA], or independent component analysis [ICA]; Gaublomme et al., 2015). However, most studies now utilize more expressive, non-linear models (e.g., scVI [Lopez et al., 2018], DCA [Eraslan et al., 2019], SIMLR [Wang et al., 2017], Seurat [Satija et al., 2015], Harmony [Korsunsky et al., 2019], or MNN [Haghverdi et al., 2018]), for which a direct association between genes and model components is not readily available. Hotspot addresses these caveats as it is applicable to any graph of cell-cell similarity, thus providing a way to account for the whole transcriptome and to interpret both linear and non-linear models of similarity. We demonstrate its advantages using a combination of simulated data and publicly available single-cell human CD4$^+$ profiles. We also use these case studies to demonstrate its ability to group the selected gene into modules, compared with popular methods such as WGCNA (Langfelder and Horvath, 2008) and GRNBoost (Aibar et al., 2017).

In cases where spatial information is available, the similarity metric can be defined as the physical distance between cells, and Hotspot can identify spatially varying genes and group them into patterns of gene expression *in situ*. We demonstrate this use for the analysis of expression in mouse cerebellum (Rodriques et al., 2019) and benchmark against a leading method, SpatialDE (Svensson et al., 2018a). While the two methods are comparable in terms of gene identification, we demonstrate that Hotspot requires orders of magnitude less computation time, making it more readily applicable for realistically sized datasets. Finally, if the developmental relationship between cells is known, Hotspot is able to identify developmentally associated gene modules. We demonstrate this using available data from a CRISPR-Cas9-based lineage-tracing system in mouse embryogenesis (Chan et al., 2019). To the best of our knowledge, Hotspot is the first method proposed for this purpose.

Hotspot is implemented as an open-source Python package and is available for use at http://www.github.com/Yoseflab/Hotspot.

RESULTS

Leveraging similarity maps for feature selection and module identification

The Hotspot procedure is divided into three main steps. In the first step, a similarity map is computed between cells. In the second step, a feature selection process is performed to isolate potentially informative genes by selecting those which exhibit non-random patterns of expression within the similarity map. The final step groups these genes into modules by using a modified type of correlation, which is conditioned on the structure of the similarity map. A representative schematic of the method is shown in Figure 1.

To compute the similarity map, some notion of "similarity" between cells is needed. By leveraging notions of similarity that derive from information other than gene expression, Hotspot is able to connect changes in expression with other types of single-cell data. If identifying spatial patterns, similar cells would be nearby cells in 2- or 3-dimensional space. For lineage data, similar cells would be more highly related cells within an inferred lineage tree (Chan et al., 2019). The similarity map can also come from the expression data itself for the case of identifying transcriptional modules in a cluster-free manner. Here, similar cells are cells with similar overall transcriptional profiles. Due to the high dimensionality of the data and the abundance of noise, cell-cell similarities are often based on a reduced dimensional space generated by procedures such as PCA (Gaublomme et al., 2015), diffusion maps (Haghverdi et al., 2015), other factor models such as ZINB-WaVE (Risso et al., 2018), scVI (Lopez et al., 2018), or DCA (Eraslan et al., 2019), and non-linear methods for sample integration such as Harmony (Korsunsky et al., 2019) or MNN (Haghverdi et al., 2018).

Once a notion of similarity is selected, a similarity graph can be computed between cells as a K-nearest-neighbors graph. In this graph representation, each cell is a node, and edges connect each cell to the K (configurable, but typically around 30) most similar cells. This structure is already frequently used in the single-cell analysis for clustering (Levine et al., 2015; Traag et al., 2019) and visualization (Becht et al., 2018).

For the feature selection step, we seek genes whose expression is well represented by the similarity graph—genes for which a cell's expression is highly predictable by its local neighborhood in the graph. To quantify this, we define a test statistic, which is then evaluated on each gene to test its association with the similarity graph. The results of these, per-gene statistical tests can then be used to rank and filter genes both to aid in exploratory analysis and to limit the scope of the downstream module identification problem.

For a gene x of interest, let x_i be its standardized expression value (such that $E[x_i] = 0$ and $E[x_i^2] = 1$) in cell i. Furthermore, let weight w_{ij} be a positive value if cells i and j are K-nearest neighbors in the similarity graph; otherwise $w_{ij} = 0$. Among a cell's neighbors, higher weight values are used for more highly similar neighbors and weights are normalized to sum to 1 for each cell (see STAR methods for more details on weight calculation).

We then define a test statistic, H_x as follows:

$$H_x = \sum_i \sum_{j \neq i} w_{ij} x_i x_j$$

We refer to this statistic, which sums pairwise products of nearby cells in the similarity map as "local autocorrelation" due to its conceptual similarity to the use of "autocorrelation" in signal processing and "spatial autocorrelation" in demographic analysis (Getis, 2008). Here, H_x is dependent on the organization of gene expression values in the similarity graph with higher values arising from genes, which are highly expressed in a particular region of the graph or genes whose expression exhibits a smooth gradient across the single-cell manifold that is depicted by the graph (Moon et al., 2018).

Our approach extends previous work in demographic analysis (Geary, 1954) and machine learning (He et al., 2006) by incorporating a parametric null model for the dependence of each gene on the graph structure. For each gene, we define a null model in which the expression values are drawn independently from some assumed distribution. Before evaluating H_x, the selected null model is used to standardize expression values (such that $E[x] = 0$ and $E[x^2] = 1$), and this is then used to compute expectations of H_x and transform the resulting H_x values into Z scores for significance calculation. Hotspot provides two options for the null distribution model of gene expression, depending on the depth of the transcriptomic data. The first option is a negative binomial with the mean dependent on library size (as in NBDisp; Andrews and Hemberg, 2019), which can be used with most transcriptomic data sets. In cases of unusually sparse datasets, we limit Hotspot to a more qualitative analysis using the Bernoulli distribution (STAR methods).

For the final module identification step, the genes with significant local autocorrelation are grouped into modules based on co-expression between nearby cells in the similarity map.

To accomplish this, we extend the test statistic defined previously to instead operate on pairs of genes and quantify the degree to which two genes have correlated expression in the same regions of the similarity graph. In a similar manner to local autocorrelation, we define a "local correlation" statistic H_{xy} between gene x and gene y as follows:

$$H_{xy} = \sum_i \sum_{j \neq i} w_{ij} \left(x_i y_j + y_i x_j \right)$$

Here, the i,j indices represent individual cells and x_i and y_i are the standardized expression values in cell i for genes x and y, respectively. Weights w_{ij} are defined as before based on the cell similarity map. As with H_x, the null model is again used to convert H_{xy} values into Z scores.

This quantifies, for example, whether cells expressing high levels of one gene tend to be similar to cells expressing high levels of another gene. It should be emphasized that this is distinct from Pearson's correlation coefficient in that it involves products of expression values between different cells rather than products of expression values within the same cell. As a result, it is more sensitive in identifying pairs of co-expressed genes that have low detection rates. This is because while such a gene pair is less likely to be coincidentally detected within the same cell (for technical reasons), the two genes will still exhibit higher expression rates in similar regions of the cell similarity map (for a direct comparison between pairwise Hotspot Z scores and Pearson correlation coefficients, see Figure S7).

After this statistic is evaluated on all pairs of genes, a hierarchical clustering procedure is used to group genes into modules. The process begins with every module represented by a single gene and proceeds by merging modules with the highest pairwise Z scores using the UPGMA procedure to derive updated Z scores. Modules cease merging once the Z scores fall below an input threshold. Additionally, modules above a certain gene count (configurable) remain unmerged to help preserve the hierarchical relationship between gene modules. For full details on this procedure as well as the feature selection method, see the STAR methods section.

Uncovering informative gene modules in single-cell RNA-seq with application to CD4$^+$ T cells

We start by focusing on the task of identifying informative genes and grouping them in modules, provided only with data of gene expression. As the first evaluation for Hotspot in this context, we simulated single-cell RNA-seq libraries with SymSim (Zhang et al., 2019) so that we could compare Hotspot's prediction against a known, ground truth. SymSim is able to simulate gene expression based on an input hierarchical structure of cell types in which differences in expression are correlated with distances in the hierarchical tree. Expression is simulated by first starting from a kinetic model of transcription (using the stationary distribution of a 2-state promoter, which fits the Beta Poisson; Kim and Marioni, 2013), and later adding noise and bias to simulate steps of the sample preparation and sequencing protocol. To simulate coordinated biological variation, a limited set of extrinsic variation factors (EVFs) are simulated based on the population structure, and these factors are then used to control variation in the per-gene kinetic parameters. For this test, we simulated a dataset of 3,000 cells, based on a tree structure of five cell sub-populations (Figure S1A). We define 500 biologically varying genes, each of which couples to one of the five EVFs, providing an assignment of genes to gene modules. We additionally simulate a set of 4,500 genes which vary independently and serve as negative controls.

We first evaluated the feature selection procedure to determine how well Hotspot could detect the 500 genes which participated in the simulated modules and distinguish these from the remaining 4,500 genes that vary independently. Here, Hotspot is run using the latent space as inferred by scVI (Lopez et al., 2018) to construct the nearest-neighbors graph. We choose scVI both as a way of demonstrating the flexibility of Hotspot with regard to the source of cell-cell distances (here, Hotspot computes nearest neighbors using Euclidean distance in the low-dimensional space inferred by scVI) and because of the benefits of scVI over PCA. Like PCA, scVI learns a low-dimensional representation of the expression dataset in an unsupervised manner. However, scVI is a Bayesian model, which represents expression values directly with a negative binomial distribution (as opposed to requiring a transformation to assume normal residuals) and can infer non-linear relationships between expression values and latent component coordinates through the use of a deep neural network.

We define the null model for the Hotspot statistics H_x and H_{xy} using the negative binomial distribution, which is commonly used for single-cell RNA-seq (note that this distribution is different from the one used to simulate the data; see STAR methods). Compared with the highly variable genes procedure ("HVG," as implemented in Seurat; Satija et al., 2015), and the NBDisp procedure from Andrews and Hemberg (2019), Hotspot is able to attain significantly improved performance. It also compared favorably to the PCA-based feature selection described in Andrews and Hemberg (2019), as demonstrated by precision-recall curves shown in Figure S1B.

We additionally sought to compare Hotspot's ability to group genes into correlated modules against the results of alternate module-detection methods. We first sought to compare the pairwise H_{xy} scores computed by Hotspot to the more traditional Pearson correlation coefficients, which support many existing module identification procedures (such as WGCNA; Langfelder and Horvath, 2008). Using the simulated transcriptional data from before, we took Pearson correlation coefficients evaluated between the total cellular counts (representing putative total mRNA molecule counts in a cell), as ground truth. Then, we compared these, using Spearman's correlation, against either Pearson's correlations or Hotspot pairwise Z scores on the generated single-cell expression profiles at various sequencing depths (Figure S7A). To separate this evaluation from that of feature selection, only the ground truth genes were used. We found that Hotspot pairwise Z scores better replicate the true correlations than Pearson correlation coefficients and that this performance difference improves as sequencing depth is lowered.

Next, to evaluate the accuracy of the inferred modules, we selected several top-performing methods from a recent benchmarking study (Saelens et al., 2018), including WGCNA (Langfelder and Horvath, 2008), ICA with an FDR gene-component thresholding procedure (as described in Rotival et al., 2011) and Grnboost (Aibar et al., 2017) (which was not included in the benchmark, but is a modern evolution of GENIE3 (Huynh-Thu et al., 2010) optimized for single-cell expression datasets). We first ran these procedures on our simulated data and evaluated gene-module assignment accuracy against the known gene-component associations (Figure S1C). Here, we observed improved performance in comparison to each alternate method except for ICA.

To further demonstrate our procedure, we utilized a set of 1,500 CD4$^+$ T cells filtered from a sample of human peripheral blood mononuclear cells (PBMCs) made available by 10x Genomics. We ran Hotspot with its negative binomial null model to identify relevant genes within the latent space as modeled by scVI. An initial investigation revealed that Hotspot emphasizes key genes involved in various aspects of T cell biology (cell state determination, trafficking, suppressive signaling, co-stimulation), which are not overdispersed and are therefore not emphasized by common procedures for gene selection (Figure S6). In addition, as expected, we observed that the genes selected by Hotspot exhibit higher local autocorrelation in the scVI latent space than those selected by the alternate feature selection methods (Figure 2C) and generated clearer gene modules and visual patterns when represented on a UMAP projection (Figure S12).

To more comprehensively evaluate feature selection on these cells, we utilized two approaches. We first evaluated our method's ability to prioritize biologically relevant genes. To quantify biological relevance, we computed a "gene relevance" (GR) score for every gene, as the number of CD4$^+$ T cell-related gene sets from MSigDB (Liberzon et al., 2011) within which the gene is found (STAR methods). With this metric, we evaluated the relevance of a set of genes as the average GR score for genes in the set. Under this evaluation, Hotspot outperformed the commonly used highly variable genes selection procedure (across all thresholds) and the PCA-based procedure (for the top 1,800 genes; Figure 2A). When comparing against the NBDisp (Andrews and Hemberg, 2019) procedure, the performance is equivalent for the top few hundred genes after which the genes reported by Hotspot have consistently higher GR scores.

To evaluate the feature selection procedure in an unsupervised manner, we attempted to quantify the quality of the similarity map generated by scVI when retrained only on the selected genes. Here, we reasoned that a more informative set of genes will result in a similarity map that more accurately represents the true cell states. To compare the quality of different similarity maps, we made use of the surface protein abundance data, which accompanies this single-cell mRNA-seq dataset as an independent indicator of cell state. We reasoned that cells that are from a similar state will express their proteins at similar levels. Thus, we used the measure of autocorrelation to evaluate the extent to which every protein in the data is expressed similarly by nearby cells, where the notion of "nearby" depends on the mRNA data. Better mRNA-based similarity maps should lead to higher protein-level autocorrelations. The results of this evaluation (Figure 2B) show that employing any feature selection procedure tended to increase the level of protein autocorrelation in general. Furthermore, the procedure utilized by Hotspot showed greater increases than the other methods compared, though differences between the HVG, PCA, and Hotspot feature selection procedures were not statistically significant (rank-sums test, $p > 0.05$).

After the selection of the top 500 significant genes by Hotspot, we ran the module identification procedure on these genes (Figure 2D). Analysis of the genes in each module revealed expression programs that are consistent with known T cell biology. Module 1 (includes CCR7 and SELL/CD62L) and module 3 (which includes IL7R and S100A4) appeared to distinguish the naive and activated T cell subsets, whereas other modules illustrated transcriptional differences within the activated group. This includes module 0, which highlights a subset of these cells expressing Th1 associated genes (including IFNG and TBX21) and module 2, which highlights another subset expressing higher levels of genes associated with cytotoxicity (PRF1, GZMB). Module 6, on the other hand, gathered genes associated with regulatory T cell activity (FOXP3, IL2RA, and inhibitory receptors CTLA4 and TIGIT). In this manner, Hotspot was able to isolate distinct patterns of transcription even when they exhibited a nested, hierarchical structure.

We next sought to compare Hotspot's pairwise local correlations against the Pearson correlation on real data, where true correlations are not known. For this, we reasoned that true correlations would tend to be preserved in a replicate experiment (while spurious correlations, i.e., those due to noise, would not replicate). We generated a pseudoreplicate by taking the $CD4^+$ T cells described in this section and by splitting the sample in half, by cells. Using the first half of the dataset, we evaluated Hotspot Z scores and Pearson correlation coefficients and compared these (again using a Spearman correlation) to Pearson correlations evaluated on the held-out half. We showed that Hotspot Z scores better correspond to correlations in the held-out dataset and that this further replicates by repeating the procedure in the $CD14^+$ monocyte subset of the same PBMC dataset (Figures S7B and S7C).

We then sought to evaluate the module assignments of Hotspot using actual transcriptional profiles for which ground truth is unknown. As with the previous comparisons to Pearson's correlation, we reasoned that better-performing methods should arrive at consistent modules when comparing between replicates. As before, we generated pseudoreplicates using the $CD4^+$ T cell dataset, and evaluated, for each method, the proportion of gene pairs which, when assigned to the same module in one replicate, are also assigned to the same module using the other replicate. We noted that this task increases in difficulty when more modules are detected (i.e., it is trivial to achieve perfect consistency by assigning all genes to a single module), and we further noted the number of modules each method output as well as the proportion of the input genes assigned to a module. To uncouple module identification from any benefits in feature selection, all methods were evaluated using the same input gene list—the top 1,000 highly variable genes. In this comparison, we found that Hotspot generates gene pairs that reproduce at a higher rate across replicates while detecting a greater number of modules and leaving less genes unassigned (Figure 2E and replicated in the $CD14^+$ monocyte subset, Figure S11B).

Identifying spatially dependent gene expression programs

In the growing field of spatial transcriptomics (Lee et al., 2014; Shah et al., 2016; Ståhl et al., 2016; Rodriques et al., 2019) new experimental methods have been developed that assay transcriptional profiles of single cells or small groups of cells, while also retaining information on their spatial coordinates. By using these spatial positions to define the cell-cell similarity metric, Hotspot can be used to identify genes that drive spatial features such as spatially dependent patterns of activation or non-random distributions of cell types.

To demonstrate this, we applied Hotspot to a sample consisting of 32,000 spatially indexed transcriptional libraries from the mouse cerebellum (Rodriques et al., 2019), with each library consisting of a small group of spatially adjacent cells. As these libraries are sequenced at a low depth (median UMIs/bar code is 45), we ran Hotspot in "bernoulli" mode, where only the binary detection of a gene at each position is modeled (see Figure S9 for comparison to Hotspot run using the negative binomial model). The nearest-neighbor graph was constructed by taking each barcode's $K = 300$ nearest neighbors in two-dimensional space. Based on their spatial distributions, 560 genes were identified with significant spatial autocorrelation ($FDR<0.05$; Figure 3A). This set is enriched in marker genes for cerebellar cell types (Figure 3B) and are distinct from genes selected on the basis of high variability (Figure S2A).

To demonstrate the utility of using the spatial distances as a metric for feature selection, we compared the top genes identified by Hotspot against those identified using the highly variable genes method (as implemented in Seurat; Satija et al., 2015). Here, we found that the top genes highlighted by Hotspot included examples such as Plp1 (a marker for oligodendrocytes) and Car8 (expressed by Purkinje neurons), which exhibit clear spatial patterns in the sample (Figure S8). Interestingly, these genes do not exhibit overdispersion and would be missed by a selection procedure based on this statistic. More globally, we found that selection of genes based on overdispersion (as in the highly variable genes, or NBDisp procedures) resulted in sets that are largely different from the one prioritized by Hotspot (Figure S13). As expected, the genes selected by Hotspot tended to have much higher spatial autocorrelation values (i.e., H_x computed with spatial distances as cell-cell similarities;

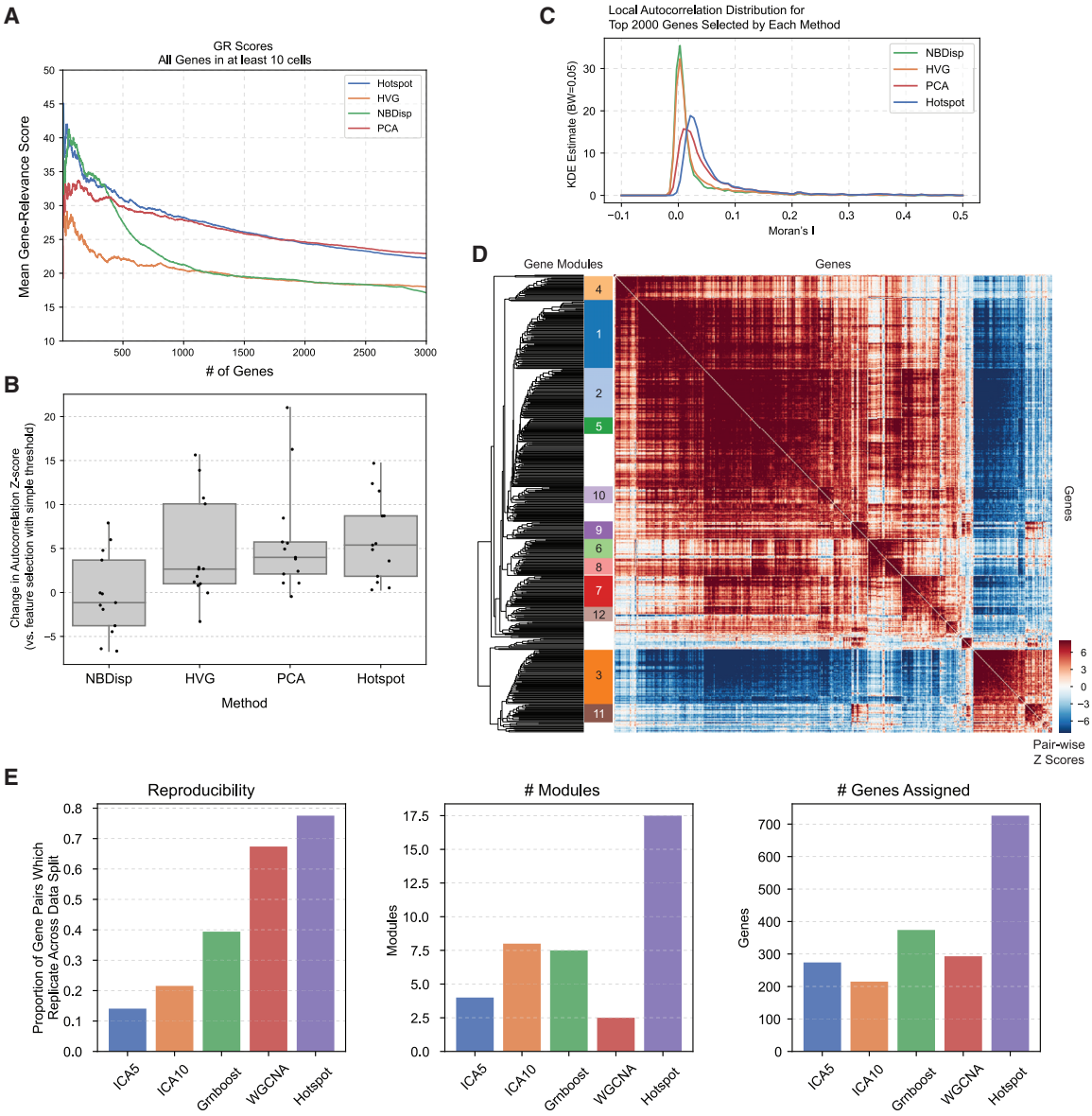


Figure 2. Evaluating Hotspot on a transcriptional dataset of CD4$^+$ T cells

(A) Comparing feature selection for four methods—Hotspot, HVG (highly variable genes, as implemented in Seurat; Satija et al., 2015), NBDisp (from Andrews and Hemberg, 2019), and a PCA-based procedure (as described in Andrews and Hemberg, 2019). For each method, the average gene relevance score (computed as the number of CD4$^+$ T cell-relevant gene sets from MSigDB [Liberzon et al., 2011] in which a gene is found) is computed on the output set of genes as the selection threshold is varied.

(B) For each method in (A) the top 1,000 genes were used to build a latent space model in scVI (Lopez et al., 2018). The local autocorrelation *Z* scores of 15 surface protein expressions (not included in the model building) are used to evaluate estimates of cell state. Shown is the difference (change in *Z* score) for each of the 15 proteins when comparing the scVI models from the selected top 1,000 features to a model built from all genes above an expression threshold (12,000).

(C) Comparing the local autocorrelation distribution for the top 2,000 genes selected by each method.

(D) Top 500 genes selected by Hotspot are grouped into 12 modules on the basis of pairwise local correlation.

(E) Evaluation of modules detected by different methods—the dataset was split in half (by cells) and each method was run on each half. Reproducibility—the proportion of genes assigned to the same module (in one half), which were also assigned to the same module in the other. # Modules—number of modules detected by each method (average across both halves). # Genes assigned—of the 1,000 genes used for module analysis, the number which were assigned a module by each method (average across both halves).

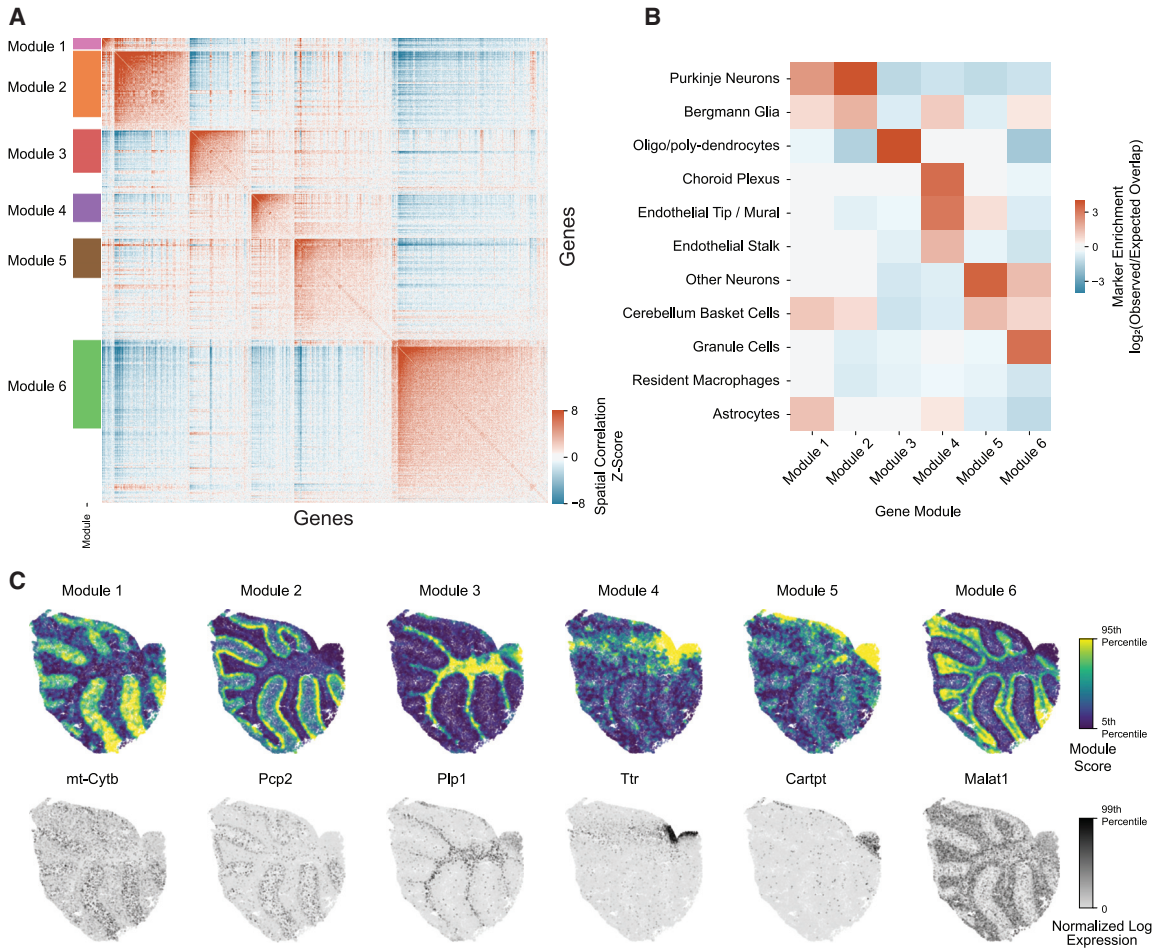


Figure 3. Spatial gene signatures in mouse cerebellum

Hotspot is used to identify spatially relevant genes within a spatial single-cell expression sample from mouse cerebellum.

(A) Genes with significant spatial autocorrelation (845 genes, FDR < 0.05) are grouped into 6 gene modules on the basis of pairwise spatial correlations.

(B) Spatial modules are associated with specific cerebellar cell types as shown by enrichment in cell-type-specific marker genes.

(C) Spatial gene modules are visualized with their summary, per bar code, module scores (top row). Beneath each plot of the module score, the expression of the gene with the highest spatial autocorrelation is visualized for comparison.

Figure S2A). Accordingly, genes prioritized by Hotspot but not by overdispersion demonstrate clearer spatial patterns and form clearer modules (Figure S13). Interestingly, we observed a similar trend when selecting genes based on their loadings in the top principal components (see STAR methods for details on this procedure).

To gain a more global view of the different spatial expression programs in the cerebellum, we used Hotspot to group the genes into modules. This resulted in six transcriptional modules, which correspond to known cerebellar cell types (Figure 3A) and are reproducible between different cerebellar samples (Figure S3). These modules reflect the primary structure of the cerebellum, with module-1 capturing the Purkinje layer of neurons adjacent to the granular layer (module-6) followed by the oligodendrocytes of the white matter (module-3). By computing summary module scores, the structure of the cerebellum becomes clearly visible (Figure 3C).

We compared our findings with an existing method for identifying expression patterns in spatial transcriptomics—SpatialDE (Svensson et al., 2018a). A key difference between our method and SpatialDE is that the latter uses a comprehensive Gaussian process model, which requires the inversion of a large matrix as part of its optimization procedure. As a result, Hotspot was able to run much more quickly as the number of cells/positions increased (Figure S2C). To compare the results produced by each algorithm, we evaluated each for its ability to detect known positives and for its reproducibility between the four cerebellum samples produced by Rodriques et al. (2019). In using a set of marker genes for cell types in the mouse cerebellum (derived from Saunders et al., 2018), we show that both methods achieve similar performance (Figure S2B). To evaluate reproducibility, we used the irreproducible discovery rate (IDR) metric (Li et al., 2011) to compare results between pairs of cerebellum samples and show that both methods achieve a similar level of performance, typically highlighting several hundred genes at an IDR level of 0.1 (Figure S2D).

In a similar manner to Hotspot, spatialDE was also able to identify modules of genes with similar spatial distributions of

expression. In comparing to the gene modules output by Hotspot, we observed that both methods were able to uncover similar patterns of variation (Figure S4). In conclusion, we found that in the task of identifying spatially dependent transcriptional programs, Hotspot achieves performance that is similar to that of a more complex model that was designed specifically for this purpose, while requiring a significantly lower processing time (e.g., 93 s versus 6.2 days for a subset of 10,000 spatially indexed transcriptional libraries; Figure S2C).

Identifying developmentally associated expression in mouse embryogenesis

To demonstrate Hotspot on other measures of relatedness, we turned to a dataset of lineage-traced embryogenesis (Chan et al., 2019). In this system, mouse embryos were engineered with a CRISPR Cas9 lineage-tracing system in which irreversible mutations are generated randomly throughout development at specified cut cites. The embryos then underwent single-cell sequencing on day 8.5 of development. Using the induced mutations, a cell's developmental relationship to other cells can be assessed in the form of a lineage tree (Jones et al., 2020). Here, we use this relationship to compute the cell-cell similarity metric when running Hotspot (the cells most closely related to a cell in the inferred lineage tree are used as its nearest neighbors—see STAR methods for details). In this way, Hotspot can be used to extract genes whose expression is similar among clonally related cells and derive modules of the gene whose expression is associated with developmental changes.

We ran this procedure on 1,756 cells from Chan et al. (2019) and identified 2,554 developmentally associated genes ($FDR < 0.05$) using the local autocorrelation test. We then applied the second step of Hotspot to group these genes into 5 modules on the basis of expression changes with related cells (Figure 4A). To interpret these modules, we made use of the annotated developmental cluster profiles from a separate dataset from Chan et al. (2019) (Figure 4C). From this comparison, it is clear that module 1 (which includes markers Cubn, Amot, Amn, and Slc39a8) describes expression associated with the visceral and definitive endoderm, module 3 (with Hbb-bh1, Gata1, Klf1, and Nfe2) is associated with primitive blood, module 4 (with Srgn) is associated with the development of the parietal endoderm, and module 5 (with Plac1 and Ascl2) is associated with the differentiation of trophoblasts.

To compare the expression signatures identified using the lineage information and those identified with gene expression alone, we additionally ran Hotspot using the transcriptional differences between cells to define their similarity graph (based on the proximity of PCA projections of the expression profiles; STAR methods and Figure 4D). Notably, an expression signature associated with the emergence of angioblasts was identifiable when analyzing the expression data alone (Figures 4D and 4E) with marker genes such as Pecam1 having high local autocorrelation in the expression space. This same signature, however, was not detectable based on lineage similarity, implying that either emerging angioblasts are less related than cells of other cell types at this developmental stage, or the number of cells or frequency of lineage-tracing mutations were insufficient to capture this effect. In this way, Hotspot can distinguish between genes and gene modules that have significant variation due to gene expression correlations and those that arise from the inferred developmental tree.

DISCUSSION

Here, we described Hotspot as a general framework for highlighting informative genes and grouping these genes into coherent expression programs in multimodal (expression + X) or standard (expression only) single-cell datasets. We demonstrate its application using several such scenarios.

In the context of gene expression only, we use a data set of T helper cells to demonstrate that Hotspot is able to identify informative gene programs in a cluster-independent manner, thus capturing gradual changes in cell phenotype, which do not naturally fit into distinct clusters. Another key advantage of Hotspot is that it provides a natural way to interpret complex, non-linear models of the manifold of gene expression states. Such models (including BBKNN, Scanorama, scVI, DCA) are becoming standard in the analysis of single-cell RNA-seq, particularly due to their performance in integrating samples. Finally, we use experimental and simulated data to show that our approach is able to better identify relevant features than leading approaches (such as the commonly used highly variable genes procedure) and that our metric of local correlation is able to better detect gene-gene correlations when compared with standard Pearson's correlation.

In the context of multi-modality, we show our approach may be used to identify spatially varying gene expression and derive spatial gene modules using a single-cell mouse cerebellar dataset (Rodriques et al., 2019). We compare against a leading method designed specifically for this purpose, SpatialDE (Svensson et al., 2018a), and demonstrate comparable performance despite orders of magnitude of improvement in run time. In a dataset consisting of a combination of gene expression and lineage-tracing data (Chan et al., 2019), we demonstrate Hotspot as the first approach, to the best of our knowledge, for identifying genes which exhibit lineage-dependent expression patterns. It should be noted that the objective of Hotspot is distinct from that of multimodal integration methods such as LIGER (Welch et al., 2019) or MOFA+ (Argelaguet et al., 2020). These methods seek to construct a similarity metric or latent space jointly informed by the combination of multiple modalities. Our procedure does not construct a similarity metric but rather seeks to identify genes that are consistent with a provided metric and group these genes into modules.

The core of our method is the definition of two key test statistics. We defined a statistic for local autocorrelation within a KNN similarity graph that takes inspiration from the Geary's C (Geary, 1954) and the Laplacian Score (He et al., 2006), which have been proposed for similar purposes. Notably, our statistic differs from these by using a parametric null model for a gene's expression to avoid the need for time-consuming permutation tests to determine the null distribution. Furthermore, we propose a novel pairwise local correlation statistic as an extension to this approach so that features (genes) may be grouped into modules on the basis of local expression similarity within the graph.

Beyond the examples shown here, the flexibility of our approach allows for additional applications. Here, we have varied the data used in the cell-cell metric (spatial, lineage, or

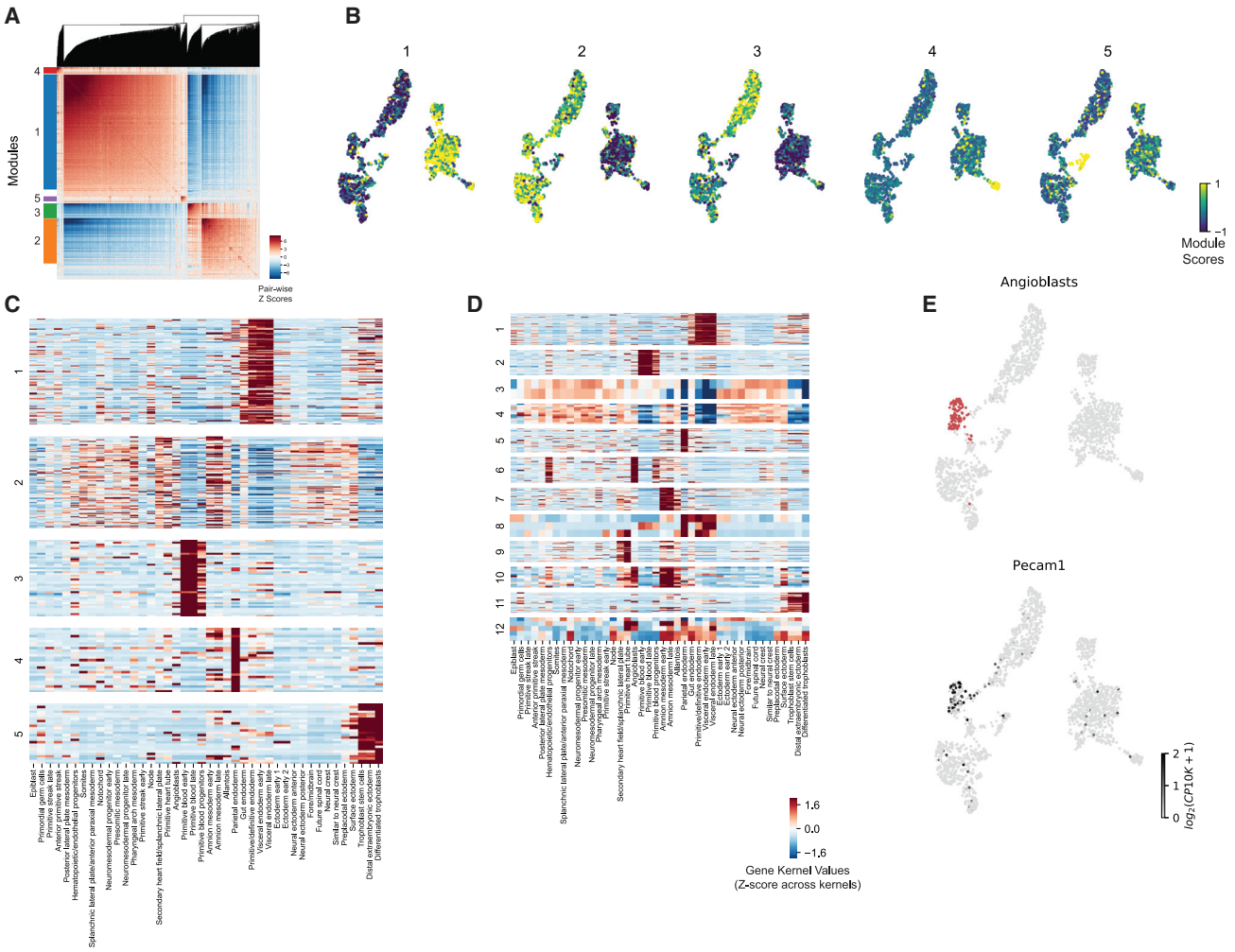


Figure 4. Lineage gene signatures during embryogenesis

(A) 2,554 genes with significant (FDR < 0.05) lineage autocorrelation are grouped into 5 gene modules on the basis of pairwise lineage correlation.

(B) Module summary scores are plotted against a UMAP of the expression data.

(C) A set of 42 manually annotated kernels from Chan et al. (2019) representing cell types in mouse embryogenesis is used to identify putative labels for lineage-derived gene modules. For each module, the kernel values of the module genes are visualized.

(D) Similar to (C), only modules derived from gene expression (not lineage) are shown instead.

(E) Annotating cells that are most similar to the angioblast kernel (top) along with an angioblast marker gene, Pecam1 (bottom). CP10K: gene counts per ten thousand UMIs. (C–E) show that Hotspot is able to distinguish between gene modules with transcriptional support (such as that of angioblasts) and those with lineage support.

transcriptional) and used Hotspot to identify genes with associated expression. However, the inverse is also possible in which the cell-cell metric is computed from gene expression alone and Hotspot is used to identify features in alternate modalities that associate in expression space. For example, if the values under test represent the chromatin accessibility of a particular genomic region (as is available with combined single-cell expression and ATAC-seq; Angermueller et al., 2016), a high local autocorrelation score would indicate that the accessibility of this region is highly connected with a cell's expression state. By repeating this test over many regions, a rank-ordering is produced to subset regions of interest for follow-up and downstream analysis. Similarly, if input values represent the (binary) presence or absence of individual CRISPR guide RNAs (e.g., through Perturb-seq; Dixit et al., 2016), then high local autocorrelation would indicate guides which induce stronger expression changes.

We have made the software behind the Hotspot procedure available at http://www.github.com/Yoseflab/Hotspot so that as single-cell technology evolves and additional modalities are incorporated, this framework can continue to be used to extract signals across different classes of biological measurements.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - A test statistic for feature selection
  - Null models for gene expression
  - Negative binomial
  - Deriving gene modules
  - Evaluating pair-wise local correlation
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - Analysis of mouse cerebellum samples
  - Analysis of mouse embryogenesis lineage-tracing data
  - Analysis of CD4 T cell transcriptomes
  - Simulated transcriptional profiles
  - Comparison of local correlation with pearson correlation
  - Alternate methods for feature selection
  - Comparison to alternate module identification methods
  - Sensitivity analysis of K (number of neighbors)

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.cels.2021.04.005.

ACKNOWLEDGMENTS

This work was supported by the Chan Zuckerberg Initiative 2018-184034.

AUTHOR CONTRIBUTIONS

D.D. and N.Y. conceived the Hotspot algorithm, interpreted analysis results, and wrote the manuscript.

DECLARATION OF INTERESTS

D.D. is an employee of ArsenalBio. N.Y. is an adviser and/or has equity in Cellarity, Celsius Therapeutics, and Rheos Medicines.

Received: March 2, 2020
Revised: December 22, 2020
Accepted: April 9, 2021
Published: May 4, 2021

REFERENCES

Aibar, S., González-Blas, C.B., Moerman, T., Huynh-Thu, V.A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.C., Geurts, P., Aerts, J., et al. (2017). SCENIC: single-cell regulatory network inference and clustering. Nat. Methods *14*, 1083–1086.

Andrews, T.S., and Hemberg, M. (2019). M3Drop: dropout-based feature selection for scRNASeq. Bioinformatics *35*, 2865–2867.

Angermueller, C., Clark, S.J., Lee, H.J., Macaulay, I.C., Teng, M.J., Hu, T.X., Krueger, F., Smallwood, S.A., Ponting, C.P., Voet, T., et al. (2016). Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. Nat. Methods *13*, 229–232.

Argelaguet, R., Arnol, D., Bredikhin, D., Deloro, Y., Velten, B., Marioni, J.C., and Stegle, O. (2020). MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol *21*, 111.

Becht, E., McInnes, L., Healy, J., Dutertre, C.A., Kwok, I.W.H., Ng, L.G., Ginhoux, F., and Newell, E.W. (2018). Dimensionality reduction for visualizing single-cell data using UMAP. Nat. Biotechnol. *37*, 38–47.

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: Series B (Methodological) *57*, 289–300.

Buenrostro, J.D., Wu, B., Litzenburger, U.M., Ruff, D., Gonzales, M.L., Snyder, M.P., Chang, H.Y., and Greenleaf, W.J. (2015). Single-cell chromatin accessibility reveals principles of regulatory variation. Nature *523*, 486–490.

Chan, M.M., Smith, Z.D., Grosswendt, S., Kretzmer, H., Norman, T.M., Adamson, B., Jost, M., Quinn, J.J., Yang, D., Jones, M.G., et al. (2019). Molecular recording of mammalian embryogenesis. Nature *570*, 77–82.

DeTomaso, D., Jones, M.G., Subramaniam, M., Ashuach, T., Ye, C.J., and Yosef, N. (2019). Functional interpretation of single cell similarity maps. Nat. Commun. *10*, 4376.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C.P., Jerby-Arnon, L., Marjanovic, N.D., Dionne, D., Burks, T., Raychowdhury, R., et al. (2016). Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. Cell *167*, 1853–1866.e17.

Eraslan, G., Simon, L.M., Mircea, M., Mueller, N.S., and Theis, F.J. (2019). Single-cell RNA-seq denoising using a deep count autoencoder. Nat. Commun. *10*, 390.

Gaublomme, J.T., Yosef, N., Lee, Y., Gertner, R.S., Yang, L.V., Wu, C., Pandolfi, P.P., Mak, T., Satija, R., Shalek, A.K., et al. (2015). Single-cell genomics unveils critical regulators of Th17 cell pathogenicity. Cell *163*, 1400–1412.

Geary, R.C. (1954). The contiguity ratio and statistical mapping. Inc. Stat. *5*, 115–146.

Getis, A. (2008). A history of the concept of spatial autocorrelation: a geographer's perspective. Geogr. Anal. *40*, 297–309.

Grün, D., Kester, L., and van Oudenaarden, A. (2014). Validation of noise models for single-cell transcriptomics. Nat. Methods *11*, 637–640.

Grün, D., Lyubimova, A., Kester, L., Wiebrands, K., Basak, O., Sasaki, N., Clevers, H., and Van Oudenaarden, A. (2015). Single-cell messenger RNA sequencing reveals rare intestinal cell types. Nature *525*, 251–255.

Haghverdi, L., Buettner, F., and Theis, F.J. (2015). Diffusion maps for high-dimensional single-cell analysis of differentiation data. Bioinformatics *31*, 2989–2998.

Haghverdi, L., Lun, A.T.L., Morgan, M.D., and Marioni, J.C. (2018). Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat. Biotechnol. *36*, 421–427.

He, X., Cai, D., and Niyogi, P. (2006). Laplacian score for feature selection. Adv. Neural Inf. Process. Syst. *18*, 121–128.

Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010). Inferring regulatory networks from expression data using tree-based methods. PLoS One *5*, e12776.

Jones, M.G., Khodaverdian, A., Quinn, J.J., Chan, M.M., Hussmann, J.A., Wang, R., Xu, C., Weissman, J.S., and Yosef, N. (2020). Inference of single-cell phylogenies from lineage tracing data using Cassiopeia. Genome Biol *21*, 92.

Kim, J.K., and Marioni, J.C. (2013). Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. Genome Biol *14*, R7.

Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.R., and Raychaudhuri, S. (2019). Fast, sensitive and accurate integration of single-cell data with Harmony. Nat. Methods *16*, 1289–1296.

Langfelder, P., and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics *9*, 559.

Lee, J.H., Daugharthy, E.R., Scheiman, J., Kalhor, R., Yang, J.L., Ferrante, T.C., Terry, R., Jeanty, S.S.F., Li, C., Amamoto, R., et al. (2014). Highly multiplexed subcellular RNA sequencing in situ. Science *343*, 1360–1363.

Levine, J.H., Simonds, E.F., Bendall, S.C., Davis, K.L., Amir, el-A.D., Tadmor, M.D., Litvin, O., Fienberg, H.G., Jager, A., Zunder, E.R., et al. (2015). Data-

driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell *162*, 184–197.

Li, Q., Brown, J.B., Huang, H., and Bickel, P.J. (2011). Measuring reproducibility of high-throughput experiments. Ann. Appl. Stat. *5*, 1752–1779.

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdóttir, H., Tamayo, P., and Mesirov, J.P. (2011). Molecular signatures database (msigdb) 3.0. Bioinformatics *27*, 1739–1740.

Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. Nat. Methods *15*, 1053–1058.

Macaulay, I.C., Haerty, W., Kumar, P., Li, Y.I., Hu, T.X., Teng, M.J., Goolam, M., Saurat, N., Coupland, P., Shirley, L.M., et al. (2015). G&T-seq: parallel sequencing of single-cell genomes and transcriptomes. Nat. Methods *12*, 519–522.

Macosko, E.Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A.R., Kamitaki, N., Martersteck, E.M., et al. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell *161*, 1202–1214.

Moon, K.R., Stanley, J.S., Burkhardt, D., van Dijk, D.v., Wolf, G., and Krishnaswamy, S. (2018). Manifold learning-based methods for analyzing single-cell RNA-sequencing data. Curr. Opin. Syst. Biol. *7*, 36–46.

Moran, P.A.P. (1950). Notes on continuous stochastic phenomena. Biometrika *37*, 17–23.

Munsky, B., Neuert, G., and Van Oudenaarden, A. (2012). Using gene expression noise to understand gene regulation. Science *336*, 183–187.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.P. (2018). A general and flexible method for signal extraction from single-cell rna-seq data. Nat. Commun. *9*, 284.

Rodriques, S.G., Stickels, R.R., Goeva, A., Martin, C.A., Murray, E., Vanderburg, C.R., Welch, J., Chen, L.M., Chen, F., and Macosko, E.Z. (2019). Slide-seq: a scalable technology for measuring genome-wide expression at high spatial resolution. Science *363*, 1463–1467.

Rotival, M., Zeller, T., Wild, P.S., Maouche, S., Szymczak, S., Schillert, A., Castagné, R., Deiseroth, A., Proust, C., Brocheton, J., et al. (2011). Integrating genome-wide genetic variations and monocyte expression data reveals trans-regulated gene modules in humans. PLoS Genetics *7*, e1002367.

Saelens, W., Cannoodt, R., and Saeys, Y. (2018). A comprehensive evaluation of module detection methods for gene expression data. Nat. Commun. *9*, 1090.

Satija, R., Farrell, J.A., Gennert, D., Schier, A.F., and Regev, A. (2015). Spatial reconstruction of single-cell gene expression data. Nat. Biotechnol. *33*, 495–502.

Saunders, A., Macosko, E.Z., Wysoker, A., Goldman, M., Krienen, F.M., de Rivera, H., Bien, E., Baum, M., Bortolin, L., Wang, S., et al. (2018). Molecular diversity and specializations among the cells of the adult mouse brain. Cell *174*, 1015–1030.e16.

Shah, S., Lubeck, E., Zhou, W., and Cai, L. (2016). In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. Neuron *92*, 342–357.

Ståhl, P.L., Salmén, F., Vickovic, S., Lundmark, A., Navarro, J.F., Magnusson, J., Giacomello, S., Asp, M., Westholm, J.O., Huss, M., et al. (2016). Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. Science *353*, 78–82.

Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P.K., Swerdlow, H., Satija, R., and Smibert, P. (2017). Simultaneous epitope and transcriptome measurement in single cells. Nat. Methods *14*, 865–868.

Svensson, V., Teichmann, S.A., and Stegle, O. (2018a). SpatialDE: identification of spatially variable genes. Nat. Methods *15*, 343–346.

Svensson, V., Vento-Tormo, R., and Teichmann, S.A. (2018b). Exponential scaling of single-cell RNA-seq in the past decade. Nat. Protoc. *13*, 599–604.

Tirosh, I., Izar, B., Prakadan, S.M., Wadsworth, M.H., Treacy, D., Trombetta, J.J., Rotem, A., Rodman, C., Lian, C., Murphy, G., et al. (2016). Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. Science *352*, 189–196.

Traag, V.A., Waltman, L., and van Eck, N.J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. Sci. Rep. *9*, 5233.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N.J., Livak, K.J., Mikkelsen, T.S., and Rinn, J.L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nat. Biotechnol. *32*, 381–386.

Tusi, B.K., Wolock, S.L., Weinreb, C., Hwang, Y., Hidalgo, D., Zilionis, R., Waisman, A., Huh, J.R., Klein, A.M., and Socolovsky, M. (2018). Population snapshots predict early haematopoietic and erythroid hierarchies. Nature *555*, 54–60.

van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. J. Mach. Learn. Res. *9*, 2579–2605.

Wang, B., Zhu, J., Pierson, E., Ramazzotti, D., and Batzoglou, S. (2017). Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. Nat. Methods *14*, 414–416.

Welch, J.D., Kozareva, V., Ferreira, A., Vanderburg, C., Martin, C., and Macosko, E.Z. (2019). Single-cell multi-omic integration compares and contrasts features of brain cell identity. Cell *177*, 1873–1887.e17.

Zeisel, A., Muñoz-Manchado, A.B., Codeluppi, S., Lönnerberg, P., La Manno, G.L., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science *347*, 1138–1142.

Zhang, X., Xu, C., and Yosef, N. (2019). Simulating multiple faceted variability in single cell RNA sequencing. Nat. Commun. *10*, 2611.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
PBMCs	10x Genomics	https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.1.0/5k_pbmc_protein_v3
Slide-Seq Mouse Cerebellum Samples	Rodriques et al., 2019	https://www.cell.com/cell-systems/fulltext/S2405-4712(20)30464-6
Mouse Embryogenesis Lineage-Tracing Data	Chan et al., 2019	GEO: GSE117542
Software and algorithms		
Python	https://www.python.org/	v3.6.7
R	https://www.r-project.org/	v3.5.1
SymSim	Zhang et al., 2019	v0.0.0.9000
scVI	Lopez et al., 2018	v0.2.4
Scikit-learn	https://scikit-learn.org	v0.21.2
Seurat	Satija et al, 2015	v2.3.4
M3Drops	Andrews and Hemberg, 2019	v3.10.4
WGCNA	Langfelder and Horvath, 2008	v1.69
arboreto	Aibar et al., 2017	v0.1.5
SpatialDE	Svensson et al., 2018a	v1.1.3
Hotspot	This paper	https://github.com/yoseflab/hotspot v0.9.1

RESOURCE AVAILABILITY

Lead contact

Requests for further information should be directed to and will be fulfilled by the lead contact, Nir Yosef (niryosef@berkeley.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- This paper analyzes existing, publicly available data. These datasets' accession numbers are provided in the key resource table.
- All original code is available at https://www.github.com/yoseflab/hotspot and archived at https://doi.org/10.5281/zenodo.4604128
- The scripts used to generate the figures reported in this paper are available at https://www.github.com/deto/hotspot_analysis and archived at https://doi.org/10.5281/zenodo.4604132
- Any additional information required to reproduce this work is available from the lead contact.

METHOD DETAILS

A test statistic for feature selection

In analyzing single-cell RNA-seq with Hotspot, the first step consists of selecting genes which are informative, given a cell-cell similarity metric. Intuitively this can be thought of as identifying genes whose expression, if plotted onto a visualization of the cell manifold (such as those produced by tSNE or UMAP), produce non-random visual patterns and are therefore likely of interest to the analyst. In our previous work (DeTomaso et al., 2019), we made use of the Geary's C (Geary, 1954) as a test statistic to select gene signatures whose aggregate signature scores exhibited this property. One drawback with this statistic, however, is the lack of a well-defined null distribution, necessitating the use of permutations for significance testing. To eliminate the computational burden this produces

when operating at the level of individual genes, we modify the Geary's C, removing terms which do not explicitly depend on the interaction between neighboring samples, and define our statistic for local autocorrelation as:

$$H = \sum_i \sum_j w_{ij} x_i x_j$$

This is evaluated on a single gene at a time with x_i representing the expression of the gene in cell i, and w_{ij} represents the weight between cells I and j. Weights are strictly non-negative and defined such that larger values are assigned to cells with higher similarity, with values decaying to 0 for highly dissimilar cells. Notably this formulation of the local autocorrelation statistic is proportional to the Moran's I (Moran, 1950), a spatial autocorrelation statistic commonly used in demographic analysis.

By default, weights are set to decay using a gaussian kernel with a per-cell bandwidth set to the distance to the $K/3$-neighbor (similar to the approach used in tSNE (van der Maaten and Hinton, 2008)). This results in $w_{ij} = e^{-d_{ij}^2/\sigma_{ij}^2}$ where d_{ij} represent the distance between cells i and j and σ_i represents the bandwidth for cell i. Weights are then further scaled such that $\sum_j w_{ij} = 1$ for each cell. Alternately, an 'unweighted' approach can be selected where $w_{ij} = 1$ if cells (i,j) are neighbors and 0 otherwise.

For computational efficiency, we assign weights using a K-nearest-neighbors graph, such that w_{ij} is only non-zero if cells i and j are neighbors and there are no self-edges. As a result, the double summation can be re-expressed as a sum over edges, E in the resulting sparse graph:

$$H = \sum_{(i,j) \in E} w_{ij} x_i x_j$$

To evaluate expectations of H (for significance testing), a null model is needed. For our null model, we assume that expression values are drawn independently from some underlying distribution for which we can compute $\mathrm{E}[x_i]$ and $\mathrm{E}[x_i^2]$ for each cell. Notably, values for $\mathrm{E}[x_i]$ and $\mathrm{E}[x_i^2]$ in the null model are estimated on a per-cell basis so that the effect of varying sequencing depths can be incorporated. Then expectations of H can be expressed as:

$$\mathrm{E}[H] = \sum_{(i,j) \in E} w_{ij} \mathrm{E}[x_i] \mathrm{E}[x_j]$$

$$\mathrm{E}\left[H^2\right] = \sum_{(i,j) \in E} \sum_{(k,l) \in E} w_{ij} w_{kl} \mathrm{E}[x_i x_j x_k x_l]$$

$$var(H) = \mathrm{E}\left[H^2\right] - \mathrm{E}[H]^2$$

In computing $\mathrm{E}[H^2]$, the lack of self edges implies that $i \neq j$ and $k \neq l$, but the inner expectation may not be split as edge (i,j) can share nodes with (k,l). Additionally, computing $\mathrm{E}[H^2]$ in this manner is difficult as a double summation over edges involves many terms ($O(N^2K^2)$ for N cells and K neighbors). However, an alternate summation may be used that can be evaluated in $O(NK)$ steps which first computes $\mathrm{E}[H^2]$ assuming no edges share nodes and then corrects the (much fewer) terms for which this is not true:

$$\mathrm{E}\left[H^2\right] = \left(\sum_{(i,j) \in E} w_{ij} \mathrm{E}[x_i] \mathrm{E}[x_j] \right)^2 +$$

$$\sum_{(i,j) \in E} w_{ij}^2 \left(\mathrm{E}\left[x_i^2\right] \mathrm{E}\left[x_j^2\right] - \mathrm{E}[x_i]^2 \mathrm{E}[x_j]^2 \right) +$$

$$\sum_i \left(\mathrm{E}\left[x_i^2\right] - \mathrm{E}[x_i]^2 \right) \left(\left(\sum_{j \in N(i)} w_{ij} \mathrm{E}[x_j] \right)^2 - \sum_{j \in N(i)} w_{ij}^2 \mathrm{E}[x_j]^2 \right)$$

Finally, we can simplify further by standardizing each cell's null model prior to computing H. Here we define:

$$\widehat{x}_i = \frac{x_i - \mathrm{E}[x_i]}{var(x_i)^{\frac{1}{2}}}$$

$$\widehat{H} = \sum_i \sum_j w_{ij} \widehat{x}_i \widehat{x}_j$$

Computing the null model for $\widehat{H}$ is then simplified as $E[\widehat{x}_i] = 0$ and $E[\widehat{x}_i^2] = 1$ for all i. The resulting expression is then

$$E\left[\widehat{H}\right] = 0$$

$$E\left[\widehat{H}^2\right] = \sum_i \sum_j w_{ij}^2$$

$$\widehat{Z} = \frac{\widehat{H} - E\left[\widehat{H}\right]}{var\left(\widehat{H}\right)^{\frac{1}{2}}} = \frac{\sum_i \sum_j w_{ij} \widehat{x}_i \widehat{x}_j}{\left(\sum_i \sum_j w_{ij}^2\right)^{\frac{1}{2}}}$$

To compute p values the resulting $\widehat{Z}$ values are compared to the normal distribution. Q-Q plots evaluating this approach using shuffled data are shown in Figure S5. When testing multiple genes, the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) is used to estimate the FDR.

Null models for gene expression

In computing local autocorrelation, a model is needed for each gene and cell describing the expected distribution of expression values under the null hypothesis that each value is drawn independently. One utility of the approach described above is that only the first and second moments of this distribution are needed, allowing for flexibility in the choice of null model. In this work, two different models were used and are described here. Importantly, both explicitly account for the per-barcode library size and adjust expected expression levels accordingly. This is necessary so that genes are not incorrectly flagged as significant due to local autocorrelation in the library size.

Negative binomial

The negative binomial distribution is a common choice to model the counts arising in single-cell RNA-seq experiments with UMIs (Grün et al., 2014). Here we utilize the **NBDisp** model proposed in Andrews and Hemberg (2019) in which the mean of the distribution for every gene is assumed to vary linearly with the library size for each cell. The model is defined as:

$$l_j = \sum_g x_{gj} \Big/ \sum_g \sum_j x_{gj}$$

$$\widehat{\mu}_{gj} = \sum_j x_{gj} l_j$$

$$t_g = \sum_j x_{gj}$$

$$\widehat{r}_g = \frac{t_g^2 \sum_j l_j^2}{(\eta_c - 1) \sum_j \left(x_{gj} - \widehat{\mu}_{gj}\right)^2 - t_g}$$

Here x_{gj} represents the UMI count for gene g in cell j. t_g represents the total number of UMI counts for gene g, η_c is the total number of cells, and $\widehat{\mu}_{gj}$ and $\widehat{r}_g$ represent the negative binomial mean and dispersion. Moments of expression are then estimated as:

$$E\left[x_{gj}\right] = \widehat{\mu}_{gj}$$

$$var\left(x_{gj}\right) = \widehat{\mu}_{gj} + \frac{\widehat{\mu}_{gj}^2}{\widehat{r}_g}$$

Bernoulli

As an alternative to the negative binomial model, we define a "Bernoulli" model where only the *detection* of a gene (defined as a UMI count greater than 0) is estimated. This model may be a better choice for extremely sparse data where detecting more than one count per gene in a cell is rare (such as the Slide-Seq (Rodriques et al., 2019) analyzed in this article.) The model for gene g is formulated as:

$$x_{gj} \in \{0, 1\}$$

$$\text{logit}\left(P\left(x_{gj} = 1\right)\right) = a_g + b_g * log_{10}(n_j)$$

where n_j is the total number of UMI in cell j. Model parameters a_g and b_g are fit on a per-gene basis. First cells are aggregated into 30 bins based on UMI, with bin center's $\hat{n}$ and the average probability of detection per bin is computed as $\hat{p}$. Then linear regression is used to estimate model coefficients as $logit(\hat{p}) = a + b * log_{10}\hat{n}$. We considered the use of logistic regression directly on the sample values, but due to performance issues decided to utilize this binned approximation. A comparison of Hotspot results under different bin numbers than 30 (Figure S11A) showed that values between 10 and 50 bins produced highly similar results.

Deriving gene modules

To compute gene modules, Hotspot uses a three-step procedure:

1. Find informative genes with high local autocorrelation
2. Evaluate pair-wise local correlations between genes
3. Cluster the resulting gene-gene affinity matrix

For step (1), the feature selection procedure described above is applied and a cutoff is used to select the most highly-informative genes. In the step (2), we modify our procedure for feature selection to evaluate correlations between genes in a manner that also leverages the global cell-cell similarity map. We denote this 'local correlation'. In this way, gene pairs which tend to be sparsely expressed in the same regions of the similarity map can be detected as correlated even if they are infrequently detected in the same cell. Finally, step (3) involves in clustering the genes by genes local correlation matrix computed in step (2). Here we found good performance in running a modified hierarchical clustering procedure (details follow later in this section).

Evaluating pair-wise local correlation

We define the following to evaluate *local correlations* between genes x and y:

$$\hat{H}_{xy} = \sum_{(i,j)\in E} w_{ij}\left(\hat{x}_i\hat{y}_j + \hat{y}_i\hat{x}_j\right)$$

For the purpose of clustering, we transform this local correlation into a Z-score by comparing it to its expected first and second moments under a null model. We initially considered a null model that assumes expression values for genes x and y are all independent. However, this formulation tends to significantly underestimate the variance of the test statistic if at least one gene has high *local autocorrelation* - which is a guarantee since we pre-select these genes. Instead, we compare against a null model where one gene's values are fixed and the other are assumed to be independent. In other words "given the observed of gene x, how extreme is H_{xy} compared with independent values of y". Since the test statistic is symmetric with respect to the choice of x and y, we compute Z-scores using both $P(H_{xy}|x)$ and $P(H_{xy}|y)$ and conservatively retain the least-significant (closest to zero) result.

The moments of $\hat{H}_{xy}$ under $P(\hat{H}_{xy}|\hat{x})$ are computed as:

$$\text{E}\left[\hat{y}_i\right] = 0 \quad \text{E}\left[\hat{y}_i^2\right] = 1$$

$$\text{E}\left[\hat{H}_{xy}\right] = \text{E}\left[\sum_{(i,j)\in E} w_{ij}\left(\hat{x}_i\hat{y}_j + \hat{y}_i\hat{x}_j\right)\right]$$

$$= \sum_{(i,j)\in E} w_{ij}\left(\hat{x}_i\text{E}\left[\hat{y}_j\right] + \text{E}\left[\hat{y}_i\right]\hat{x}_j\right)$$

$$= 0$$

Where $\hat{x}$, $\hat{y}$ represent expression values of genes x, y that have been standardized with respect to one of the previously described gene models. Notably, as we are evaluating $P(\hat{H}_{xy}|\hat{x})$, values of x are held fixed (i.e., they are not treated as random variables).

Computing $\mathrm{E}[H_{xy}^2]$ is more involved. First we note that the value can be expressed as a sum of edge pairs:

$$\text{let } E_{ij} = w_{ij}\left(\hat{x}_i\hat{y}_j + \hat{y}_i\hat{x}_j\right)$$

$$\mathrm{E}\left[H_{xy}^2\right] = \mathrm{E}\left[\left(\sum_{(i,j)\in E} E_{ij}\right)^2\right]$$

$$\mathrm{E}\left[H_{xy}^2\right] = \mathrm{E}\left[\left(\sum_{(i,j)\in E} E_{ij}\right)\left(\sum_{(k,l)\in E} E_{kl}\right)\right]$$

$$\mathrm{E}\left[H_{xy}^2\right] = \sum_{(i,j)\in E}\sum_{(k,l)\in E}\mathrm{E}[E_{ij}E_{kl}]$$

To compute this efficiently, we make note that when evaluating expectations of pairs of edges, $E[E_{ij}E_{kl}]$, there are three possible situations:

(1): edge pair shares no nodes (i, j, k, l all distinct):

$$\mathrm{E}[E_{ij}E_{kl}]$$

$$= w_{ij}w_{kl}\mathrm{E}\left[\hat{x}_i\hat{y}_j\hat{x}_k\hat{y}_l + \hat{x}_i\hat{y}_j\hat{x}_l\hat{y}_k + \hat{x}_j\hat{y}_i\hat{x}_k\hat{y}_l + \hat{x}_j\hat{y}_i\hat{x}_l\hat{y}_k\right]$$

$$= 0$$

(2): edge pair shares one node (for example, $i = k$):

$$\mathrm{E}[E_{ij}E_{il}]$$

$$= w_{ij}wil\mathrm{E}\left[\hat{x}_i^2\hat{y}_j\hat{y}_l + \hat{x}_i\hat{y}_j\hat{x}_l\hat{y}_i + \hat{x}_j\hat{y}_i\hat{x}_i\hat{y}_l + \hat{x}_j\hat{y}_i^2\hat{x}_l\right]$$

$$= w_{ij}w_{il}\hat{x}_j\hat{x}_l$$

(3): edge pair shares two nodes (for example, $i = k$ and $j = l$):

$$\mathrm{E}[E_{ij}E_{ij}]$$

$$= w_{ij}^2\mathrm{E}\left[\hat{x}_i^2\hat{y}_j^2 + \hat{x}_i\hat{y}_j\hat{x}_j\hat{y}_i + \hat{x}_j\hat{y}_i\hat{x}_i\hat{y}_j + \hat{x}_j^2\hat{y}_i^2\right]$$

$$= w_{ij}^2\left(\hat{x}_i^2 + \hat{x}_j^2\right)$$

Since only products of edges that share neighbors need to be considered, we can compute the expectation in $O(E)$ time as:

$$\mathrm{E}\left[H_{xy}^2\right] = \sum_{i}^{N}\left(\sum_{j\in N(i)} w_{ij}\hat{x}_j\right)^2$$

where N is the number of nodes and $N(i)$ are nodes which share an edge with node i.

Clustering the gene-gene affinity matrix

Once the gene by gene matrix of Z-scores has been computed, we apply a bottom-up clustering procedure with two parameters: *min_cluster_genes*, and *fdr_threshold*. As the algorithm proceeds, the two genes/modules with the highest pair-wise Z-score are merged, using the UPGMA procedure to derive updated Z-scores between the resulting module and the remaining modules and/or genes. If a module accumulates more than *min_cluster_genes*, then it is assigned a label. To preserve hierarchical structure between modules, if two labeled modules are merged, a new label is not assigned and genes merged into the resulting composite module remain unlabeled. The *fdr_threshold* parameter is used to set a minimum significant Z-score by applying the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) to the associated p values of all Z-scores and selecting the minimal such Z-score which is below the FDR threshold. If at any point in the above merging procedure the maximal Z-score falls below this threshold, the procedure halts as further gene assignments fall below the significance threshold and are therefore ambiguous.

Computing per-cell module scores

To visualize gene modules, it is useful to evaluate per-cell module scores which can then be plotted onto UMAPs, spatial plots, dendrograms, etc. When computing module scores, Hotspot uses the following procedure: first counts are centered using the selected null model (e.g., 'bernoulli' or 'Negative Binomial'). Then the resulting values are smoothed using the KNN graph - the smoothed expression for each cell is computed as the weighted average of its neighbors in the graph. These smoothed values are then modeled with PCA using a single component, and the cell-loadings are reported as the model scores (potentially with a sign inversion so that gene coefficients are positive).

QUANTIFICATION AND STATISTICAL ANALYSIS

Analysis of mouse cerebellum samples

Barcode expression and position information was downloaded for four mouse cerebellum samples (Puck_180819_9, Puck_180819_10, Puck_180819_11, and Puck_180819_12) from https://portals.broadinstitute.org/single_cell/study/slide-seq-study as directed in (Rodriques et al., 2019). For analyses involving a single sample, Puck_180819_12 was used.

When running Hotspot, genes were initially prefiltered to remove those detected in less than 50 cells. For feature selection, a neighborhood size of 300 was used and the 'bernoulli' model was used to model gene detection probabilities. For pair-wise correlation and module identification, genes were selected with an FDR <.05 and the neighborhood size was reduced to 30 to identify spatial modules at a finer resolution.

When running spatialDE, genes were again prefiltered to remove those detected in less than 50 cells. Initial runs of spatialDE tended to select an unreasonably small length scale with poor results, and so we recomputed the length-scale values with a minimum of 50 prior to fitting the model. When computing modules with spatialDE, we selected a length scale of 350 (following the tools guidelines to select a length scale slightly larger than the average per-gene optimal length scale) and 10 components as an initial run with 5 components did not appear to capture all spatial patterns. Additionally, only a representative subset of the data (12,000 / 32,000 barcodes) was used here due to the prohibitive computational runtime of processessing the entire dataset.

To evaluate the performance of feature selection with the cerebellum data, we sought to create a list of 'true' positives. We reasoned that the spatial patterns in the cerebellum would be largely influenced by changes in the composition of cell types and so marker genes for cerebellum cell types could be used a positive gene list. To create this list, we downloaded raw expression data from the DropViz (Saunders et al., 2018) database and computed marker gene sets for each of the 11 annotated mouse cerebellum clusters. For each cluster, we evaluating a 1 vs. all ranksums test for differential expression on every gene and retained the top 100 genes above a significance threshold (FDR < 0.1). The eleven lists were then merged to create a true positive set for the precision-recall curves. These same marker gene lists were also used when characterizing Hotspot modules.

In addition to this marker-gene based approach, we evaluated performance based on reproducibility between pairs of mouse cerebellum samples. For the four samples, Hotspot and spatialDE were both run in the same manner as described above and for each of the six sample pairs, the reproducibility of the gene ordering was evaluating using the Irreproducible Discovery Rate (IDR) metric (Li et al., 2011).

Analysis of mouse embryogenesis lineage-tracing data

Expression data from Chan et al. (Chan et al., 2019) was downloaded from NCBI GEO: GSE117542. For this analysis, the sample corresponding to Embryo 3 was used. The developmental lineage was inferred by running the Cassiopeia-Greedy algorithm (Jones et al., 2020) with priors determined from the frequency of indels across all observed embryos (as done in the original study (Chan et al., 2019)). Hotspot was run using two different metrics to compare outputs: the 'tree' metric in which the KNN graph was formed from the 30 nearest neighbors according to the inferred lineage and the 'transcription' metric in which PCA was run (on $log_2(x + 1)$-transformed counts-per-10,000 expression values) and the KNN graph was formed from the 30 nearest neighbors

(euclidean distance) in the reduced (top 20 component) transformed space. For both cases, genes were pre-filtered to remove those expressed in less than 10 cells, and the negative binomial model was used (with adjustment for library size). To evaluate local correlations between genes, for the tree-metric all genes with local autocorrelation passing the 0.05 FDR threshold were selected. For the transcription metric, a large number of genes exhibited statistically significant local autocorrelation and so the top 2000 (based on Z-score) were selected for evaluating pairwise local correlations. When extracting modules, a local correlation FDR threshold of 0.05 was used and *min_cluster_genes* settings of 30 and 50 were used (for tree-metric and transcription-metric respectively).

In evaluating the modules computed from Hotspot, the expression data from Chan et al. (2019) was used. Specifically the Cell State Kernels were downloaded from NCBI GEO (accession GSE122187) and for each module returned by Hotspot, the kernel values were plotted (after standardizing across cell types) for the intersection of the 712 kernel genes and the genes in each module.

UMAP projections were generated by first running PCA (as described above) and then running UMAP (Becht et al., 2018) on the transformed space (top 20 components) with n_neighbors=30.

Analysis of CD4 T cell transcriptomes

For the CD4 T cells, we made use of the public "5k_pbmc_protein_v3" available from 10x Genomics. Digital gene expression was downloaded from 10x and the full set of PBMCs was filtered based on surface protein abundance to extract the CD4+ T cell and CD14+ Monocyte subsets. We initially discarded cells with low recovered mRNA UMI counts (UMI < 3000) or with a high proportion of counts from mitochondrial genes (>16%). Then, the protein abundance measurements were log-transformed ($log(x + 1)$) and then each cell was individually mean-centered. Cells were then retained based on manual thresholding of bimodal markers. CD4+ T cells were selected using the criteria, CD4 > 3, CD3 > 2.5 and CD14 < 2. CD14+ monocytes were selected on the criteria of CD14 > 2.

When running the Highly Variable Genes procedure, Seurat's *FindVariableGenes* method was used. First counts were log-normalized with a scale factor of 10,000. Then *FindVariableGenes* was run with a x.low.cutoff =.01, x.high.cutoff = 3.04 (corresponding to a high-end cut-off of 20 UMI counts/10,000), and y.cutoff =.5. The resulting genes were then ordered in descending order by the reported 'gene.dispersion.scaled'.

For the NBDisp procedure, the NBumiFitModel function as implemented in the M3Drop (Andrews and Hemberg, 2019) package was used with default settings on raw gene counts. Additionally, for consistency with the HVG procdure, a high-end cutoff of 20 counts/10,000 was used. Output genes were ordered by the reported 'q.value' in ascending order.

For the PCA procedure, the feature proposed by Andrews and Hemberg (2019) was used. Specifically, principal components analysis was run with 5 components on the log-transformed, scaled counts/10,000. A genes score was reported as the sum of the absolute value of its PCA coefficients in the top 5 components. Genes were then ordered in descending order based on this score.

To compute gene relevance (GR) scores, we downloaded the c7 immune signature set from MSigDB (Liberzon et al., 2011). Gene sets were filtered to retain sets describing comparisons between CD4 T cells, specifically sets with one of the following terms (Th1, Th2, Tfh, Treg, Tconv, T, Th17, Tcell, NKTcell, CD4, Thymocyte) and none of these terms (DC, PDC, Macrophage, BMDM, Monocyte, Neutrophil, Mast, B, BCell, NKCell, NK) resulting in 706 CD4-relevant gene sets. For every gene, its GR score was computed as the number of CD4-relevant gene sets in which it appears. When evaluating a set of genes (selected by the Hotspot, HVG, NBDisp, or PCA procedure), the resulting metric was reported as the average GR score for the set.

To evaluate a set of genes using protein local autocorrelation, the raw counts were subset to include only the genes under evaluation and then scVI (Lopez et al., 2018) was run with 10 latent components. Hotspot was then used to evaluate the local autocorrelation of the surface protein measurements. This is predicated on the assumption that a more informative set of genes leads to a latent space model that more accurately reflects true cell state, and that both mRNA expression and protein surface abundance are derived from this cell state. Therefore, we would expect a more informative set of genes to result in increased local autocorrelation values for surface proteins as well. The resulting protein Z-scores for each of the four methods were then compared with the Z-scores from a fifth run where genes were selected with a simple thresholding procedure (all genes expressed in at least 10 cells).

Other implementation details are as follows: Within Hotspot, protein abundance was modeled using a depth-adjusted normal model. First protein counts were log-transformed, and then the log of the total protein counts (per cell) were linearly regressed out of the transformed protein counts as we observed significant correlation between individual protein counts and the total counts in other proteins, per cell. This normalized protein abundance matrix was then standardized on a per-protein basis. In selecting genes with Hotspot, PCA was first used with 20 components to create the latent space for cell-cell similarities. For all procedures, the top 1000 reported genes were used. To avoid evaluating on surface proteins which were not expressed in CD4 T cells (and whose detection therefore represents background noise), we filtered out proteins in which the 95th percentile of expression (across all cells) was below 100 counts per 10,000.

In identifying expression modules in the CD4 T cells, first scVI was run on the top 1000 highly variable genes with 10 components. This latent space was then used as input to Hotspot and modules were created from top 500 genes based on local autocorrelation. For module creation, an *fdr_threshold* of 0.05 was used along with a *min_cluster_genes* setting of 15.

Simulated transcriptional profiles

Simulated data were used to evaluate the performance of Hotspot when run on transcriptional data alone (i.e., when the cell-cell similarity graph is also constructed based on transcriptional similarity). To generate simulated single-cell profiles, we used the SymSim (Zhang et al., 2019) framework. This framework utilizes a three-parameter kinetic model of transcription to model extrinsic cell-variability (differences in biological state), intrinsic cell variability due to the stochastic nature of transcription, and technical

sources of variation from the incomplete sampling of cellular transcriptomes and non-uniform library amplification. A low-rank matrix of extrinsic variability factors (EVFs) are used to inform variations in the kinetic parameters (on-rate, off-rate, and gene synthesis rate), which are then distribution-matched to kinetic parameters inferred from a reference single-cell dataset. From these kinetic parameters, then, the Master Equation (Munsky et al., 2012) is used to sample from the steady state distribution of expected mRNA molecule counts and generate synthetic 'full' cell transcriptomes. Finally, the single-cell sequencing process is simulated by down-sampling these counts (to account for incompletely mRNA capture) and further accounting for the effects of PCR amplification and variable reads/cell. Notably this is a different and more complex model of transcription than the depth-adjust negative binomial model (default) assumed by Hotspot.

For the specific simulations in this study we used the 'phylogenetic tree' input of SymSim to generate 5 cellular populations of varying transcriptional differences. This mode uses a tree to inform a covariance matrix which is then used to generate per-cell EVFs via by sampling from a multivariate normal distribution. In this manner, populations which are closer in the tree definition will contain cells with more similar EVFs (and ultimately more similar transcriptional profiles), and the EVFs themselves model distinct, yet correlated, axes of biological variation. The tree definition was encoded with the Newick string ''(((A:1,B:1):1,(C:0.5,D:0.5):1.5):1,E:3);'' which is visualized in Figure S1A.

For each of 10 simulation replicates, UMI count profiles were generated for 3000 cells consisting of 5000 genes. 5 EVFs were created to vary along with this population structure, each EVF randomly coupling into 2% (approximately 100) of the genes. Genes coupling into EVFs in this manner were taken as the 'positive' set when evaluating feature selection and the assignment of gene to EVF provides a definition of 'true' gene modules within the context of the simulated data. EVFs were used to vary the synthesis parameter only, allowing the on-rate and off-rate parameters to be assigned randomly. In addition a random per-gene offset was added for each kinetic parameter. For full implementation details, see accompanying code.

When evaluating feature selection on this data, methods were run as described here. Seurat's (Satija et al., 2015) 'FindVariableGenes' function was used as the Highly Variable Genes (HVG) method with x.low.cutoff = 0.1, x.high.cutoff = 100, and y.cutoff = 0.5. The NBDisp method from Andrews and Hemberg (2019) was run with default settings. For the PCA procedure, as described in Andrews and Hemberg (2019), we first ran PCA on the $log(x+1)$ transformed scaled counts per 10,000 values retaining the top 5 components. Genes were then ordered using the sum of the absolute value of their component weights in descending order. When running Hotspot, we used the cell-components of this same PCA procedure as the latent space to construct the KNN graph (300 neighbors), and the NBDisp model (from Andrews and Hemberg, 2019) as the expression null model.

Comparison of local correlation with pearson correlation

We compared gene-gene local correlation with Pearson's correlation using both simulated transcriptional profiles and subsets of the 10x PBMC dataset (Figure S7). The simulated transcriptional profiles were generated as described in the previous section, and the PBMC subsets (CD4+ and CD14+) were defined as previously described.

For evaluation with simulated data, we utilized the feature of SymSim in which 'full' transcriptional profiles are first simulated (representing putative actual mRNA counts in a cell), prior to the generation of actual, 'observed' profiles (where the effects of capture rate, sequencing depth, PCR, etc. are incorporated). Accordingly, we computed correlations between genes in the 'full' profiles as reference, 'true' correlations, and then compared with those computed in the 'observed' profiles (using either Pearson's correlation, or local correlation). As local correlation Z-scores are not on the same scale as Pearson correlation coefficients, we compared to 'true' correlations by first unraveling gene by gene correlations into a single vector, and then evaluating the Spearman correlation between vectors. In this way, higher Spearman coefficients between local correlations and true correlations indicate that the rank of the local correlation Z scores more closely resembled the rank of the true correlations. This comparison was further repeated as the simulated sequencing depth for the 'observed' profiles was decreased. In all cases, local correlations were computed using a cell-cell similarity graph constructed from a latent space inferred by scVI (using 10 components). Pairwise correlations were only evaluated on genes which coupled to an EVF.

When using the $CD4^+$ and $CD14^+$ transcriptional profiles from 10x, we could not directly evaluate using any ground truth correlations. Instead, we reasoned that correlations due to real biological factors (and not technical noise) should better agree between replicate samples. Accordingly, we split each dataset in half (by cells), and evaluated the degree to which Pearson or local correlations on one half corresponded with Pearson correlations computed on the held-out, reference half. In the same manner as with the simulated data, we made this comparison using a Spearman correlation between the vectors of pair-wise correlation coefficients. We further repeated this comparison after downsampling at various proportions (by UMIs), on the non-reference half. All pairwise gene evaluations were made using the set of genes returned by the highly variable genes procedure (using parameters x.low.cutoff = 0.1 and y.cutoff = 0.5), and not the Hotspot gene selection procedure to avoid biasing the results in favor of local correlation.

Alternate methods for feature selection

Unless otherwise specified, alternate methods for feature selection were run in the following manner:

Highly-variable genes (HVG)

The *FindVariableGenes* function was run in the Seurat 2.3 R package using the parameters mean.function=ExpMean, dispersion.function=LogVMR, x.low.cutoff=0.1, and y.cutoff=0.5. Genes were ordered using the resulting gene.dispersion.scaled value in descending order.

NBDisp

The R M3Drop package (version 3.10) was used by running the *NBumiFitModel* function on the gene count matrix followed by the *NBumiFeatureSelectionCombinedDrop* function on the resulting model with parameter method set to 'fdr'. For ordering the resulting genes, the resulting q.value was used in ascending order.

PCA

The *irlbaPcaFS* function of the R M3Drop package (version 3.10) was used for PCA-based feature selection. This function performs PCA on the log-transformed, scaled, count matrix and orders genes by the sum of the absolute value of their loadings in the top N principal components. Unless otherwise specified, the top 5 components were used.

Comparison to alternate module identification methods

Alternative methods for module identification were run in the following manner:

WGCNA

The *power* hyperparameter was selected as advised in the R package documentation by running the pickSoftThreshold function across a range of values and selecting the minimal value in which the R^2 value exceeds 0.9. Then, gene modules were constructed using the blockwiseModules function with minModuleSize set to 15, and other hyperparameters (mergeCutHeight = 0.25, reassignThreshold = 0, pamRespectsDendro = FALSE) set as indicated in the "Automatic, one-step network construction and module detection" tutorial on the WGCNA documentation website.

Grnboost2

The *arboreto* Python package was used to run Grnboost2. As Grnboost2 does not create modules, but rather reports a test statistic of 'importance' between gene-pairs, we constructed modules from its output by performing hierarchical clustering on the rows of the resulting gene-by-gene matrix, cutting the tree at the 90th percentile of importance values, and discarding gene modules with less than the minimum module size of 15 genes.

ICA

As Independent Components Analysis (ICA) was one of the top performing methods from the Saelens et al (2018) review (Saelens et al., 2018), we sought to run this procedure for module detection as described in that study. To this end, we ran the FastICA procedure (as implemented in the Python Scikit-Learn package), on log-transformed, scaled, and centered expression data. FDR values were then evaluated on gene-component loadings using the fdrtool function of the fdrtool R package. Gene-component loadings were then transformed into crisp modules by removing associations with FDR<0.001 and associating each gene with the component in which its gene-component loading value is maximal. As ICA has the component number as a hyperparameter, we ran the method with both 5 and 10 components. The FDR-thresholding, in some instances, removed components from the output in which there were no significant or maximal gene-component associations. When running on simulated profiles, an FDR<.1 threshold was used instead as the 0.001 threshold resulted in very few associations.

Pearsons

In this method, Pearson's correlation is computed between genes, and correlation scores are grouped into modules using the same hierarchical clustering procedure as implemented in Hotspot with a minimum cluster size of 15 and an FDR threshold of 0.05. For FDR thresholding, Pearson correlation coefficients were converted into FDR values by transforming the coefficient into its corresponding T statistic, computing the associated p value, and then performing the Benjamini-Hochberg FDR correction.

When comparing module identification with Hotspot on actual transcriptional profiles, each method was run with the same set of input genes - those passing the highly variable genes filter. For simulated profiles, only the genes coupling into EVFs (e.g. biologically-varying genes) were used. On simulated profiles, accuracy was evaluated directly by using the set of genes coupling to a particular EVF as a 'true' module. For the PBMC-derived transcriptional profiles for which ground truth was unavailable, we split the dataset in half (by cells), and evaluated the reproducibility of the results as the proportion of genes assigned to the same module (in one half) which were similarly assigned to the same module in the other half. We additionally reported the number of modules reported by each method, and the number of genes assigned to a module (since all methods allow for some genes to remain unassigned) to root out high reproducibility due simply to reporting few genes or few modules.

Sensitivity analysis of K (number of neighbors)

The primary hyperparameter involved in graph construction is the number of neighbors, K used when constructing the K nearest neighbors graph. To evaluate our method's sensitivity to this parameter, we conducted a sensitivity analysis (Figure S10) using several scenarios and datasets for which evaluation was possible. In general we find performance tends to increase as K increases, but optimal performance is typically achieved in the range of $100 \leq K \leq 300$.