nature methods

BRIEF COMMUNICATION

https://doi.org/10.1038/s41592-021-01286-1

An analytical framework for interpretable and generalizable single-cell data analysis

Jian Zhou [1] ✉ and Olga G. Troyanskaya [2,3,4] ✉

The scaling of single-cell data exploratory analysis with the rapidly growing diversity and quantity of single-cell omics datasets demands more interpretable and robust data representation that is generalizable across datasets. Here, we have developed a 'linearly interpretable' framework that combines the interpretability and transferability of linear methods with the representational power of non-linear methods. Within this framework we introduce a data representation and visualization method, GraphDR, and a structure discovery method, StructDR, that unifies cluster, trajectory and surface estimation and enables their confidence set inference.

Single-cell exploratory analysis methods, including visualization methods and approaches for trajectory estimation, rely on either linear or non-linear data representation, each of which currently presents important limitations to single-cell data analysis. Linear dimensionality reduction methods, including principal component analysis (PCA), provide high interpretability via linear maps. We define linear interpretability as the properties of linear representations that enable intuitive understanding and comparison of data in the representation space. Specifically, linear interpretability means that any position, direction, or distance in the representation space has a clear meaning in the original data space. For example, the position of a cell may represent a linear combination of expression scores of multiple genes, and such mapping is invariant regardless of position in the representation space. Such invariance further enables comparison of different subregions of a representation, such as the comparison of cell states in different developmental stages in the same dataset. Moreover, linear interpretability enables the application of the same low-dimensional projection to different datasets to produce comparable representations. However, linear dimensionality reduction methods typically cannot efficiently represent cell identities in single-cell data: spatial adjacency in low-dimensional representations is not as good a predictor of similarity in overall expression state compared with non-linear methods.

These limitations have led to wide use of non-linear representations on single-cell omics, including t-distributed stochastic neighbor embedding (t-SNE)[1], uniform manifold approximation and projection (UMAP)[2] and others[3–5]. Trajectory estimation methods[6–10] can also often be considered specialized non-linear data representation methods, and they generally lack the desirable linear interpretability properties of linear methods. These limitations present a practical barrier to the comparison or integration of datasets at scale. Moreover, for most methods it is infeasible to apply statistical inference to analyze uncertainties of extracted structures, making the drawing of robust conclusions more difficult.

We propose that the difficulty of linear dimensionality reduction for single-cell data arises from a high level of noise: high dimensionality is necessary to capture similarities between cells when each individual dimension is noisy, and this renders low-dimensional linear representations less appealing. Indeed, all popular non-linear methods for single-cell omics data use high-dimensional information, which is often represented by distances between cells from high-dimensional input, thus effectively reducing the effect of noise. We reasoned that allowing information sharing across cells leveraging high-dimensional information could improve the quality of cell state representation while preserving the linear space and its interpretability.

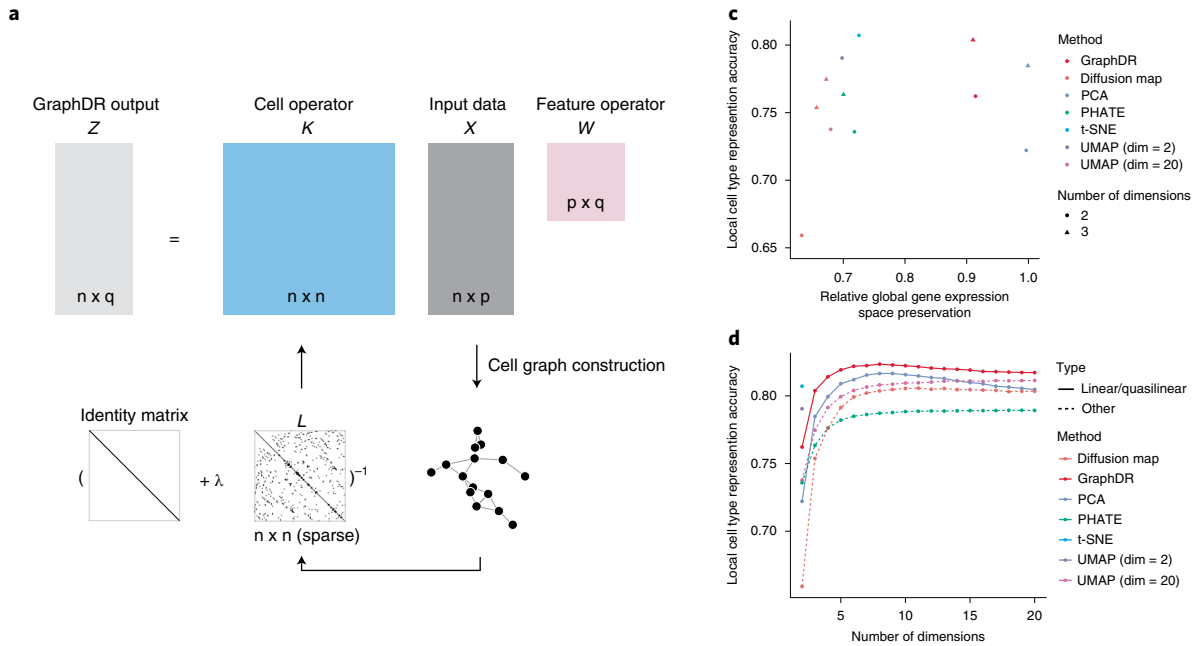
We therefore developed a 'linearly interpretable' framework for exploratory analysis of single-cell omics data, which includes methods that exactly or approximately preserve the linear interpretability but improve upon linear methods on cell state representation quality or other desired properties. Specifically we developed two methods that complement each other: a dimensionality reduction and visualization method, GraphDR, and a general structure extraction method, StructDR, that unify cluster, trajectory and surface estimation under the same framework and enable inference of confidence sets for these structures.

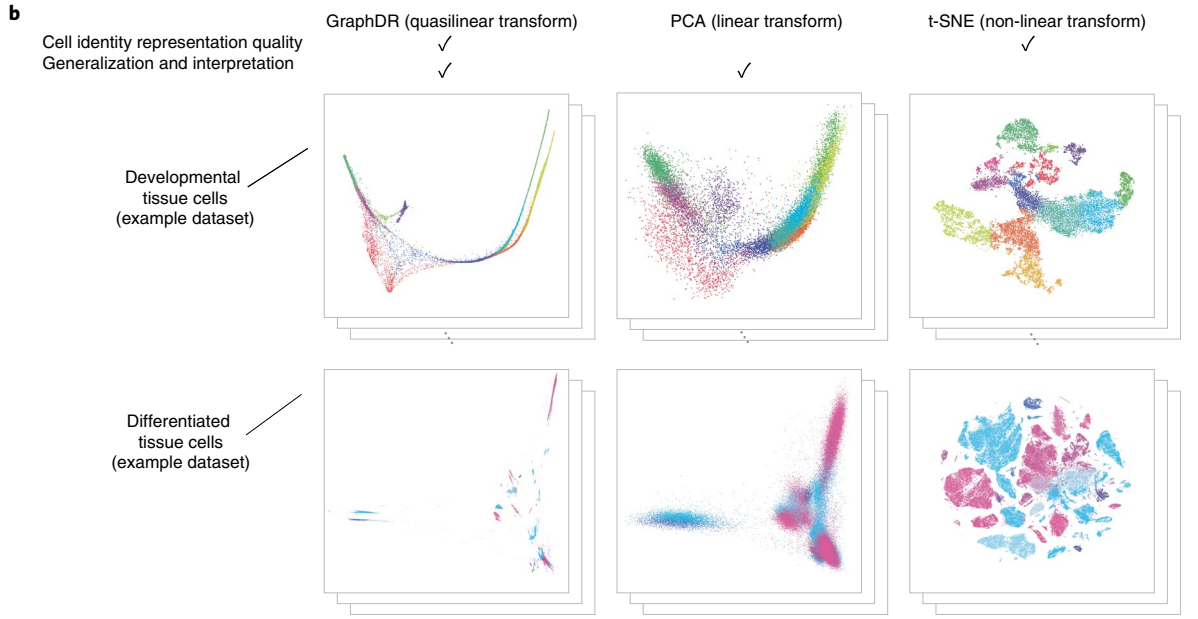
We designed GraphDR to be a graph-based linearly interpretable data representation and visualization method that addresses the limitations of linear representations in single-cell data (Fig. 1a and Methods). We achieved these desired properties by considering a flexible class of methods that improves over linear methods but which maintains interpretability by introducing non-linearity specifically for information sharing across cells. Specifically, GraphDR applies, first, a feature (for example, gene) space transformation W, as in linear methods, and second, an interpretability-preserving cell space transformation K that introduces non-linearity and improves cell state representation.

GraphDR applies a cell space transformation derived from the analytical solution of a graph-based optimization problem that allows information sharing among cells connected in the graph (Fig. 1a). The graph can be constructed with cell state similarities in high-dimensional input data and can incorporate experimental design information when appropriate. GraphDR then simultaneously optimizes the reconstruction of the input data and the consistency with the graph. The existence of a closed-form solution makes GraphDR analytically tractable and allows ultrafast computation scalable to very large datasets.

We first demonstrated GraphDR, PCA, and t-SNE using two distinct single-cell RNA sequencing (RNA-seq) datasets, representing developing mouse hippocampus cell types[11] and mature mouse brain cell types[12], respectively (Fig. 1a). GraphDR generated representations that preserved the linear interpretability, like PCA, and resolved different cell types, like t-SNE (Fig. 1b and Extended Data

[1]Lyda Hill Department of Bioinformatics, University of Texas Southwestern Medical Center, Dallas, TX, USA. [2]Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, NJ, USA. [3]Flatiron Institute, Simons Foundation, New York, NY, USA. [4]Department of Computer Science, Princeton University, Princeton, NJ, USA. ✉e-mail: jian.zhou@utsouthwestern.edu; ogt@cs.princeton.edu


a
GraphDR output
Z
Cell operator
K
Input data
X
Feature operator
W
n x q
=
n x n
n x p
p x q
Cell graph construction
Identity matrix
L
(+ λ)^-1
n x n (sparse)
c
Local cell type representation accuracy
0.80
0.75
0.70
0.65
0.7
0.8
0.9
1.0
Relative global gene expression space preservation
Method
GraphDR
Diffusion map
PCA
PHATE
t-SNE
UMAP (dim = 2)
UMAP (dim = 20)
Number of dimensions
2
3
d
Local cell type representation accuracy
0.80
0.75
0.70
5
10
15
20
Number of dimensions
Type
Linear/quasilinear
Other
Method
Diffusion map
GraphDR
PCA
PHATE
t-SNE
UMAP (dim = 2)
UMAP (dim = 20)


b
Cell identity representation quality
Generalization and interpretation
GraphDR (quasilinear transform)
✓
✓
PCA (linear transform)
✓
t-SNE (non-linear transform)
✓
Developmental tissue cells (example dataset)
Differentiated tissue cells (example dataset)

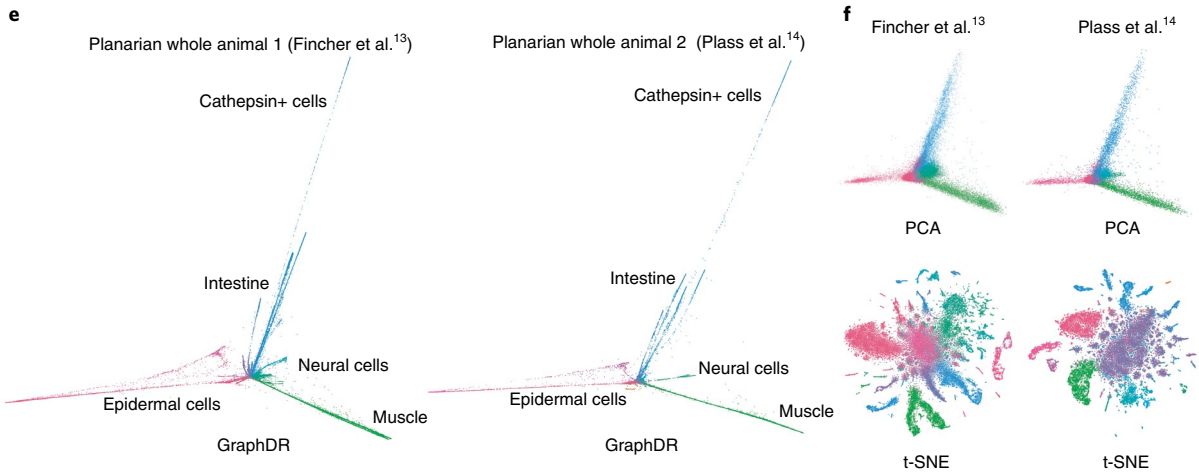

e
Planarian whole animal 1 (Fincher et al.[13])
Cathepsin+ cells
Intestine
Neural cells
Epidermal cells
Muscle
GraphDR
Planarian whole animal 2 (Plass et al.[14])
Cathepsin+ cells
Intestine
Neural cells
Epidermal cells
Muscle
GraphDR
f
Fincher et al.[13]
Plass et al.[14]
PCA
PCA
t-SNE
t-SNE

Fig. 1 | A linearly interpretable data representation method that captures the structure of single-cell data while preserving interpretability and transferability. a, Schematic overview of the linearly interpretable data representation method, GraphDR, for single-cell omics data representation and visualization. GraphDR approximately preserves the structure and interpretability of a corresponding linear transform. **b**, Visualization of two example datasets of developmental trajectory[11] (top) and mature cell types[18] (bottom) using GraphDR and representative linear and non-linear methods, PCA and t-SNE. GraphDR is applied without rotation relative to PCA (Methods). **c**, Comparison of single-cell data dimensionality reduction methods in recovering cell type information from its nearest neighbor in the representation, and preservation of the input linear space measured in correlation between the pairwise distances of all cells in the input and the pairwise distances in the representation (the relative global space preservation score is further divided by the correlation of PCA at the same number of dimensions). Both two-dimensional (filled circle) and three-dimensional (filled triangle) representations are compared. **d**, Cell type identity representation accuracy for multiple numbers of dimensions for single-cell data dimensionality reduction methods. **e**,**f**, Facilitation of comparison across datasets by the linearly interpretable transform, which balances the advantages of linear and non-linear transforms. Two planarian single-cell datasets, Fincher et al. 2018 (ref. [13]) and Plass et al. 2018 (ref. [14]), were processed with a linear transform PCA, a non-linear transform t-SNE and GraphDR.

Fig. 1). Importantly, this gain of interpretability was achieved without a loss of accuracy: in a benchmark using 339 single-cell datasets with cell type annotations[10], GraphDR distinguished cell types as accurately as several state-of-the-art non-linear methods, measured by consistency of nearest neighbors in dimensionality-reduced embedding with literature-based cell type identities (Fig. 1c,d).

GraphDR also facilitates direct comparisons across datasets. To demonstrate, we used GraphDR to analyze two planarian *Schmidtea mediterranea* whole-animal single-cell RNA-seq datasets by two different laboratories[13,14]. GraphDR generated representations that could both distinguish all cell types and be compared across datasets (Fig. 1e and Extended Data Figs. 2 and 3). By contrast, PCA representations were comparable across the two datasets but did not resolve specific cell types, whereas t-SNE representations resolved cell types but were not comparable across datasets (Fig. 1f).

GraphDR can incorporate experimental design information into the analysis by encoding the information into graph construction (Extended Data Fig. 3). To illustrate this, we applied GraphDR to single-cell RNA-seq datasets from developing zebrafish embryos (time-series design; Extended Data Fig. 4) and *Xenopus* embryos (batch + time-series design; Extended Data Fig. 5). Interestingly, the visualization of each developmental landscape showed branches of lineages extruding from a continuum of cell states, suggesting a more complex paradigm than the traditional branch view of cell fate specification.

Although visualization methods provide an intuitive and flexible representation of the structure of the data, quantitatively defined structures such as clusters and trajectories often need to be extracted to perform a detailed analysis of cell types and developmental trajectories. Existing single-cell analysis methods[6–10] are still limited in the types of structures that they can represent and, for example, do not allow for unsupervised detection of two-dimensional surface structure (Fig. 2a). Furthermore, state-of-the-art methods do not allow statistical inference of the uncertainties of the trajectories such as through the construction of confidence sets, which is essential for assessing the robustness of the inferred trajectories.

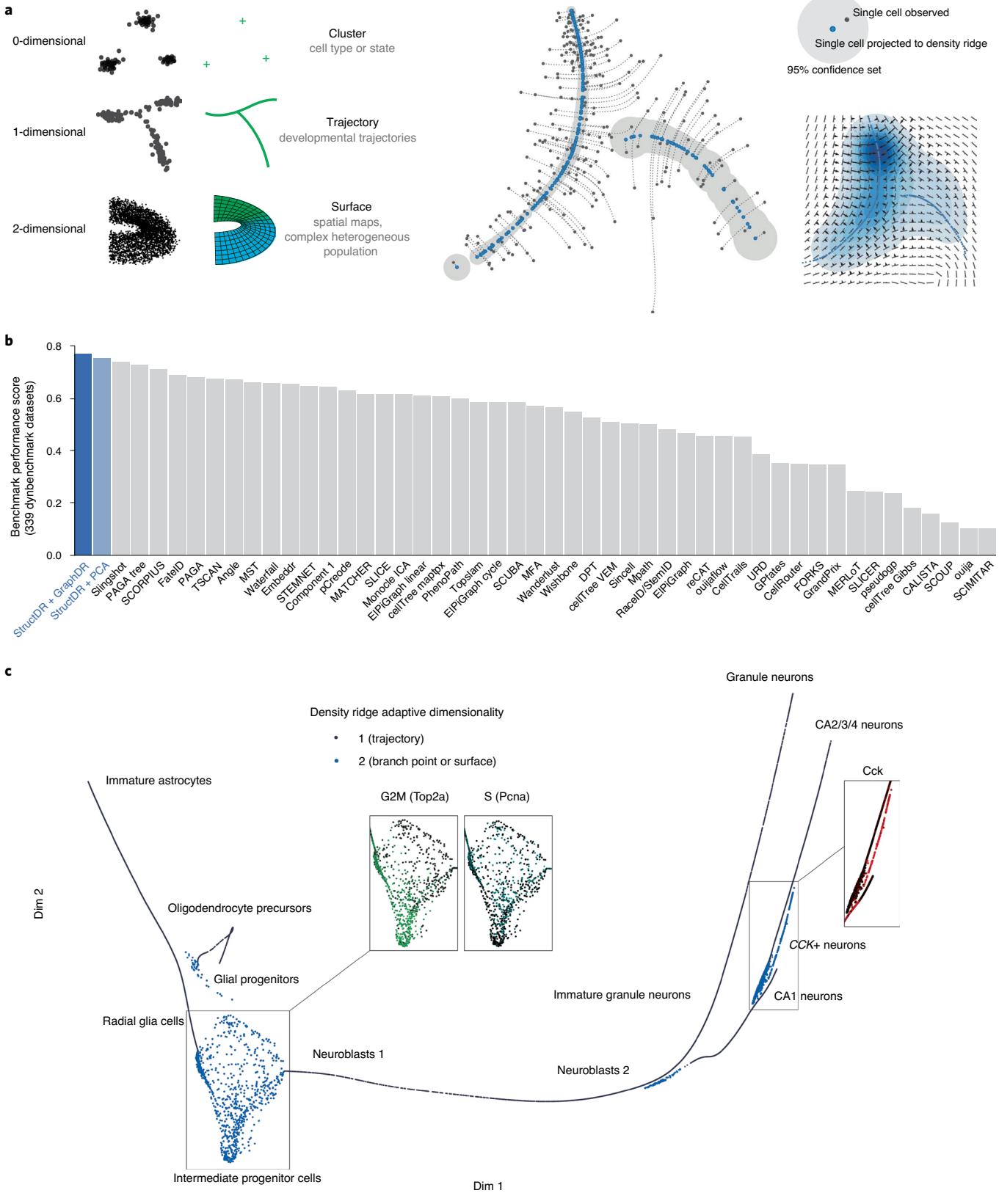
We thus developed StructDR, a unified framework for single-cell cluster, trajectory and surface structure discovery based on the non-parametric density ridge estimation (NRE) method[15–17] (Methods). NRE also enables estimation of statistical confidence sets of these structures (Fig. 2a). More specifically, StructDR casts the problem of structure discovery in single-cell data as the estimation of a smooth density function of cells and the subsequent projection of the cells to their corresponding positions on a set of mathematical structures called density ridges (Extended Data Figs. 6 and 7).

In StructDR, we map the problems of discovering clusters, trajectories and surfaces to the identification of zero-, one- and two-dimensional density ridges of the density function, which are, geometrically, points, curves and surfaces, respectively. Density ridges are generalizations of local maxima[15] and are uniquely defined given any smooth density function of cells estimated from single-cell data (Extended Data Fig. 7). Trajectory or one-dimensional density ridge analysis is appropriate for representing cell populations that have one dominant direction of variation or when the top one principal direction of variation is of interest, such as in terminally differentiating cells. By contrast, surface or two-dimensional density ridge analysis is appropriate for representing cell populations with two dominant directions of variation or when the top two principal directions of variation are of interest, such as differentiating cells that are also in cell cycle. StructDR can further connect density ridges via graph construction to represent the global topological structure of the data. Thus, in the context of StructDR, we consider trajectory estimation as the inference of one-dimensional density ridges or a connected graph of one-dimensional density ridges.

Fig. 2 | Density-based generalized trajectory estimation and inference. a, Schematic overview of the StructDR framework. The left panel shows zero-, one-, and two-dimensional density ridges and examples of corresponding biological structures. The middle panel shows an example of trajectory estimation (a one-dimensional density ridge) based on myoblast single-cell RNA-seq data[6]. The original cell positions are shown as black dots, the projected positions are shown in blue, and the projection lines are shown as dotted lines. The gray shading represents the confidence sets of trajectory positions. In the right panel, the top plot shows an annotated example of confidence set estimation. The bottom plot illustrates the elements of the SCMS algorithm[16]: the arrows indicate the gradient vectors of the probability density function, the bars indicate the direction of the first eigenvectors of the Hessians of the log probability density function, the contour plot represents the kernel density estimator-based density function, and the blue dots represent the estimated trajectory positions. **b**, Performance of StructDR + GraphDR and StructDR + PCA on a published large-scale benchmark of 339 datasets. The performance scores are computed based on Saelens et al. 2019[10]. StructDR is applied with one-dimensional density ridge estimation and automated graph construction for cells projected onto the density ridge. **c**, Density ridge identification in a hippocampus developmental trajectory single-cell dataset[11] as an example of adaptive dimensionality. Cells projected to one-dimensional (black dots) and two-dimensional density ridges (blue dots) are shown, where the dimensionality of the density ridge is adaptively determined based on the data. Insets show the gene expression levels of the indicated genes in subregions of the representation. CCK, cholecystokinin; PCNA, proliferating cell nuclear antigen; TOP2A, DNA topoisomerase II alpha.

We evaluated the trajectory estimation performance with a large benchmark dataset created by Saelens et al.[10] consisting of 339 single-cell datasets. StructDR had the best overall performance across all datasets (Fig. 2b and Extended Data Fig. 8).

In addition to capturing the complexity of single-cell data with zero-, one- and two-dimensional density ridge representations (Fig. 2a,c and Extended Data Fig. 9), StructDR can adaptively select density ridge dimensionality for each cell based on the data. For example, when analyzing single-cell RNA-seq data from hippocampus development, StructDR captured the cellular heterogeneity of neuronal progenitor cells going through the cell cycle by a two-dimensional surface instead of arbitrarily mapping these cells



to one-dimensional trajectories (Fig. 2c). Furthermore, this analysis identified a population of *CCK*-positive neurons between the CA1 and CA2/3/4 branches within the hippocampus (Fig. 2c), which was not reported in the previous analysis of this dataset with standard methods[11].

StructDR can estimate confidence sets of ridge positions when linear representations such as PCA are used as the input (Fig. 2a). We demonstrate that StructDR-inferred confidence sets can effectively control the coverage probability of ground truth trajectory positions in trajectory estimation (Extended Data Fig. 10) as expected from theoretical results.

Taken together, our work presents a linearly interpretable single-cell data analysis framework that provides interpretable, scalable and robust representations and which facilitates dataset comparison and integration. With the rapid growth of single-cell datasets, we expect linear interpretability to become increasingly important. We have also developed a feature-rich interactive analysis interface that supports 3D visualization, Trenti (Supplementary Fig. 1), to facilitate exploratory analysis and make our tools broadly accessible. We also envision that these methods will be potentially applicable to other high-dimensional data beyond single-cell omics data applications.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-021-01286-1.

Received: 19 March 2020; Accepted: 31 August 2021;
Published online: 1 November 2021

References

1. van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
2. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: uniform manifold approximation and projection. *J. Open Source Softw.* **3**(29), 861 (2018).
3. Haghverdi, L., Buettner, F. & Theis, F. J. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics* **31**, 2989–2998 (2015).
4. Moon, K. R. et al. Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.* **37**, 1482–1492 (2019).
5. Weinreb, C., Wolock, S. & Klein, A. M. SPRING: a kinetic interface for visualizing high dimensional single-cell expression data. *Bioinformatics* **34**, 1246–1248 (2018).
6. Trapnell, C. et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32**, 381–386 (2014).
7. Bendall, S. C. et al. Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell* **157**, 714–725 (2014).
8. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
9. Farrell, J. A. et al. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science* **360**(6392), eaar3131 (2018).
10. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* **37**, 547–554 (2019).
11. Hochgerner, H., Zeisel, A., Lönnerberg, P. & Linnarsson, S. Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing. *Nat. Neurosci.* **21**, 290–299 (2018).
12. Marques, S. et al. Oligodendrocyte heterogeneity in the mouse juvenile and adult central nervous system. *Science* **352**, 1326–1329 (2016).
13. Fincher, C. T., Wurtzel, O., de Hoog, T., Kravarik, K. M. & Reddien, P. W. Cell type transcriptome atlas for the planarian *Schmidtea mediterranea*. *Science* **360**(6391), eaaq1736 (2018).
14. Plass, M. et al. Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics. *Science* **360**(6391), eaaq1723 (2018).
15. Genovese, C. R., Perone-Pacifico, M., Verdinelli, I. & Wasserman, L. Nonparametric ridge estimation. *Ann. Stat.* **42**, 1511–1545 (2014).
16. Ozertem, U. & Erdogmus, D. Locally defined principal curves and surfaces. *J. Mach. Learn. Res.* **12**, 1249–1286 (2011).
17. Chen, Y. C., Genovese, C. R. & Wasserman, L. Asymptotic theory for density ridges. *Ann. Stat.* **43**, 1896–1928 (2015).
18. Zeisel, A. et al. Molecular architecture of the mouse nervous system. *Cell* **174**, 999–1014 (2018).

Methods

GraphDR: a linearly interpretable data representation method. We propose a class of linearly interpretable dimensionality reduction methods, which are non-linear methods that produce representations that aim to maximally preserve the interpretability of a corresponding linear subspace, while enabling other desired properties that are unachievable by linear methods, such as information sharing across cells. We believe it is beneficial to provide a unified viewpoint for this class of methods sharing the same properties.

To design a linearly interpretable representation method, we first propose the form $Z=KXW$, analogous to the linear dimensionality reduction $Z=XW$, to enable fast computation and analytical tractability. Z represents the data representation output matrix $n\times d$, where n is the number of samples and d is the number of output dimensions, and X is the input data $n\times c$, where n is the number of samples and c is the number of input dimensions. W and K are matrices that apply feature (for example, gene) space and cell space linear transformations that are of shape $d\times c$ and $n\times n$, respectively. In other words, we apply both a linear projection on feature space W, like linear methods, and an additional linear transform on cell space K, which is also derived from X. The addition of the cell space operator K enables much greater flexibility in the transformation, which can be exploited to improve the quality of the representation. For example, setting K to a block-diagonal matrix with all entries within a block equal to 1/block-size can move all cells within one block to their average position, leading to clustering-like behavior. In theory, an ideal K can move all cells within the same ground truth state to the same position asymptotically in the limit of a large number of cells.

For the design of K in GraphDR, we use $K=(I+\lambda L)^{-1}$, which is motivated by the solution to the loss function

$$\underset{W,\,Z}{\text{minimize}} \quad ||X - ZW^{\mathrm{T}}||_2^2 + \lambda \sum_{\{i,j\}\in G} G_{ij}||Z_i - Z_j||_2^2,\ s.t.\ W^{\mathrm{T}}W = I,$$

where the first term is the typical PCA loss, and the second term is a graph-based regularization term that encourages cells connected in the graph to be close to each other. L is the graph Laplacian matrix of graph G. The second loss term is also shared by a related non-linear representation method, Laplacian eigenmaps. Compared with Laplacian eigenmaps, the first loss term enables a linear interpretation not available to Laplacian eigenmaps and allows the graph to contain disconnected components. The analytical solution to the optimization problem is $Z=(I+\lambda L)^{-1}XW$, where W represents the top d eigenvectors of $X^{\mathrm{T}}(I+\lambda L)^{-1}X$, where d is the dimensionality of Z. The existence of an analytical solution makes it much easier to be analyzed, modified, and incorporated in downstream analyses compared with methods that do not.

For graph G, a practical and empirically well-performing choice for GraphDR is the nearest-neighbors graph. The graph construction process can also incorporate experimental design or prior knowledge information (see the next section for more details).

GraphDR can also be applied with a predefined W matrix or applied without reducing the dimensionality. If W is fixed to be an identity matrix to preserve the original input space, the problem becomes

$$\underset{Z}{\text{minimize}} \quad \|X - Z\|_2^2 + \lambda \sum_{\{i,j\}\in G} G_{ij} \left\|Z_i - Z_j\right\|_2^2.$$

The solution is also simplified as $Z=KX$, while K remains unchanged as $(I+\lambda L)^{-1}$. Notably, preservation of the input data dimensionality enables preservation of the ability to choose a linear subspace to visualize after applying the transformation, to enable more flexible comparison of datasets processed separately. For example, a user can construct G from PCA-transformed data and apply GraphDR to the full gene by cell matrix to obtain Z without reducing dimensionality. Applying any linear transform W to this representation Z is equivalent to applying GraphDR with W being the predefined feature space linear transformation matrix.

GraphDR reduces to linear representation when the regularization parameter λ is 0, and the value of λ controls the tradeoff between non-linear regularization, which contributes to better cell type representation, and preservation of the interpretation of the corresponding linear representation. Note that choosing very high values of λ may lead to some distortion and compression of the span of the data compared with the linear representation. Smooth interpolation between any GraphDR representation and the corresponding linear representation can be obtained by generating a series of representations by gradually reducing λ to 0.

For large-scale evaluation of the GraphDR method, we used the data and cell type annotations from the Saelens et al. 339-dataset benchmark[10]. Specifically, we measured the local cell type representation quality score by the average accuracy of predicting the cell type identity of each cell by the cell type of its nearest neighbor by Euclidean distance in the representation. We measured the global gene expression space preservation score by computing the Pearson correlation between the pairwise distances in the input data space and the pairwise distances in the representation space. Furthermore, the relative global gene expression space preservation score is computed by dividing the global gene expression space preservation score by the score of the PCA representation at the same dimensionality.

GraphDR graph construction incorporating experimental design. The graph construction step in GraphDR can flexibly incorporate specific experimental design information, such as batch and time-series design (Extended Data Fig. 3), for better representation of the biological variations of interest. Here, we provide a few examples that can be generalized to other scenarios: for experiments performed in two or multiple batches, nearest-neighbor connections (if the batches contain the same cell types) or mutual nearest-neighbor connections[19] (if the batches may contain different cell types) between all pairs of batches can be introduced in addition to the nearest-neighbor graphs constructed for each batch; for experiments with a time-series design, the graph can be constructed by combining the nearest-neighbor graphs for each time point and the k-nearest-neighbor connections between all adjacent time points; for experiments with both batch and time-series design, in addition to constructing a graph for each batch in the same way as the time-series design, nearest-neighbor connections between two batches in the same or adjacent time points can be added.

Computational efficiency optimization for GraphDR. With the constant growth in single-cell dataset size, it is important to design fast algorithms that scale with the dataset size. We have optimized the performance of GraphDR, resulting in a very fast method that takes only 1.5 min for a dataset containing 1.3 million cells. With only CPU computing, GraphDR takes 5 min to analyze a large 1.3-million-cell dataset on a typical modern server machine (2x Xeon Gold 6148), which is 10-fold faster than UMAP (52 min), currently one of the fastest non-linear dimensionality reduction methods. For very large datasets that will probably become available in the near future, we have developed a graphics processing unit (GPU)-accelerated version of GraphDR, which takes only 1.5 min to analyze a 1.3-million-cell dataset and 18 min to analyze a 10-million-cell simulated dataset (1x Tesla V100). To achieve this speed, each major step of computation was optimized to use fast algorithms and implementations.

The two major steps in the GraphDR algorithms are the graph construction and the solving of the output Z. To speed up the graph construction step, we leveraged recent progress in fast approximate KNN (k-nearest neighbor) algorithms (ANN). Exact KNN algorithms based on a ball tree or a KD tree fit the need for small- to medium-sized datasets but do not scale to a very large number of cells. For ANN algorithms, we support both the HNSW (hierarchical navigable small world) method[20] from NMSlib written in C++ with Python binding, and an alternative pure Python implementation of the nearest neighbor (NN)-descent method[21] built into the package (originally implemented by the UMAP package[2]). The HNSW option is faster and was used for our performance test.

In the final step of computing Z, in the case of a large number of cells it is much faster to avoid explicit computation of K and to solve Z with a linear solver. This is because the inverse of K, $I+\lambda G$, is sparse, which enables fast computation. To implement the linear solver efficiently with modern multicore architecture we used libraries with highly optimized linear algebra routines, including taking advantage of CUDA-based GPU computation, which gives the best performance.

StructDR: unified framework for structure discovery. We propose to unify cluster, trajectory and surface estimation by formulating it as an NRE problem. The NRE problem can be solved via the subspace-constrained mean shift (SCMS) algorithm[16,22]. The statistical theory of NRE has been described in detail previously[15,17].

In brief, density ridges represent the local maxima in probability density functions, and whereas zero-dimensional density ridges are local maxima of the density functions, one-dimensional or two-dimensional ridges are curves or surfaces that have maximum density locally, except for one or two orthogonal directions with the least negative curvature (a formal mathematical definition can be found below). Gradient-based algorithms can be generalized to identify density ridges of arbitrary dimensionality. Specifically, the mean shift algorithm, which projects any points to zero-dimensional density ridges or local maxima, can be generalized to the SCMS algorithm, which projects points to density ridges of arbitrary dimensionality[16]. Therefore, zero-dimensional ridge estimation is equivalent to clustering with the mean shift algorithm, and one- and two-dimensional ridge estimation can be considered trajectory and surface identification algorithms, respectively.

More formally, in an N-dimensional space in which the positions of cells in this space represent cell states from single-cell data, NRE identifies positions that satisfy the condition $R = \{x : \|G_d(x)\| = 0,\ \lambda_{d+1}(x) < 0\}$,[15] where d is the dimensionality of the density ridge. $\|G_d(x)\|=0$ is the key condition that requires the projected gradient of the density function at any position on the d-dimensional density ridges to be zero. The projected gradient at any position x can be computed from the gradient by setting the values on the directions of the top d eigenvectors of the Hessian of the log density function at x (eigenvalues ranked in descending order) to zero. A Gaussian kernel is used in the kernel density estimation of the probability density function because it provides a smooth density function that also enables the fast computation of derivatives. $\lambda_{d+1}(x) < 0$ is a stability condition that requires the trajectory to include points that are local maxima instead of local

minima in the probability density function, where $\lambda_{d+1}(x)$ is the $(d+1)$th largest eigenvalue of the Hessian matrix of the probability density function. This condition is automatically satisfied with the SCMS algorithm.

We use the SCMS algorithm (Supplementary Information) to simultaneously solve the problems of identifying the density ridges and projecting individual cells to the trajectory. Notably, an additional advantage of this approach is that not only can all estimated trajectory positions be directly interpreted as cell states in the input space, but all possible cell states, including cell states not observed in the input set, can be mapped to their corresponding positions on the density ridges.

The algorithm iteratively moves any point toward the projected gradient direction, until it converges to a point at which the projected gradient is zero. To allow fast convergence, the step size of each update along the projected gradient direction is decided based on the step size of the mean shift algorithm, which is a fixed-point iteration algorithm with good empirical convergence properties. To integrate over the projected gradient curve and project single cells more accurately, we prefer to use a smaller step size than the standard mean shift and introduce a multiplication factor a for step size, which is usually set to values <1. StructDR can be applied with arbitrary density ridge dimensionality d, and $d=0$ (cluster), 1 (trajectory), or 2 (surface) are the most interpretable. As d increases, the projected cell positions on density ridges approach the input data point.

StructDR can in principle be used with any data representation, but we recommend using it with GraphDR or linear representations such as PCA, because they enable clear interpretation of the output by allowing any position and direction on the trajectory to have a clear meaning in the original data space. GraphDR synergizes with StructDR by addressing the limitation of applying the original NRE method to single-cell omics data. Specifically, the original NRE method becomes less statistically efficient when applied to single-cell data with higher dimensionality (for example, when the number of principal components used is >6), therefore the method is unable to utilize all available information. To address this challenge, we use GraphDR to generate a linearly interpretable representation of the data, which reduces high-dimensional noise and enables StructDR to effectively use all informative principal components for density ridge structure estimation.

Even though the GraphDR and StructDR methods aim to preserve linear interpretability, they provide non-linear representations that do not replace linear representations in all cases. For example, the representations for individual cells in GraphDR or StructDR representations are not independent, and statistical methods that require such an assumption should not be applied.

Estimation of confidence set of density ridge with StructDR. As described by Chen et al.[17], the bootstrap confidence set can be constructed using the following procedure. In the first step, generate N bootstrap samples via sampling with replacement, then estimate density ridges for each bootstrap sample. In the second step, for each position in the density ridge set estimated from the original sample, calculate the distance to its nearest position in each bootstrap sample density ridge set. And in the last step, take the a-upper quantile of the distances t_a, and then the a-confidence set of each estimated ridge position is constructed as a sphere of radius t_a centered at the estimated ridge position.

Regarding the interpretation of the constructed confidence sets, two properties of this approach to constructing confidence sets for density ridge positions should be noted. First, the true density ridges are considered to be the density ridges of the smoothed true data distribution after applying the same kernel density estimation kernel. Second, the theoretical asymptotic properties of bootstrap statistical inference are valid only for linear representations for which no cell-to-cell dependencies have been introduced. Most non-linear representations, including GraphDR, introduce dependencies across cells, and generalization of the bootstrap procedure to methods such as GraphDR awaits further work. Finally, even though resampling-based methods can be widely applied, and have indeed been applied for the analysis of the stability of trajectory estimation methods[10], not all resampling estimates can be used to construct confidence sets. In fact, in most cases they do not correspond to confidence sets, and StructDR is specifically motivated by the statistical works on NRE[15,17], which describe the theoretical properties of bootstrap-based inference of confidence sets with these algorithms.

Adaptive density ridge dimensionality. To enable the flexible representation of data containing complex structures with different dimensionalities, we propose mixed-dimensionality representation, which adaptively determines ridge dimensionality. Empirically, mixed one- and two-dimensional density ridges are found to be a robust and informative representation of data structure. In this mode, cells with one dominant direction of variation based on the curvature of the density function are projected onto a one-dimensional density ridge, while the rest are projected onto two-dimensional density ridges. Specifically, we modified the SCMS method to adaptively determine, for any position in the space, the ridge dimensionality between $d=1$ (trajectory) mode and $d=2$ (branch point or surface) mode at every iteration. This decision is based on the eigengap between the first and second eigenvalues of the Hessian matrix of the log probability density function: if $\frac{\lambda_1-\lambda_2}{\lambda_1-\lambda_{-1}}$ is above the specified threshold then $d=1$ is used, otherwise $d=2$ is used, where λ_1, λ_2, and λ_{-1} represent the first, second, and last eigenvalues, respectively.

Simulation study for evaluation of confidence set. We used the inferred trajectory from a real dataset[11] as the ground truth to generate synthetic datasets. One hundred simulated datasets are generated by adding independent Gaussian noise to the samples from the ground truth trajectory density function. For each simulated dataset, 20 bootstraps were used to construct a confidence set of trajectory positions based on the distance from bootstrapped trajectories to the estimated trajectory[17]. The estimated confidence sets with a coverage probability were then compared with the ground truth to determine the true number of times that any point on the ground truth trajectory is covered by the confidence set constructed.

Graph construction from density ridge positions. In StructDR output, density ridges are represented by the positions of data points projected to these ridges. To enable more flexible applications we construct graph representations from these projected points. To do so, we first construct a candidate graph connecting k-nearest neighbors in both the projected cell space and in the input cell space. The candidate graph is then simplified by choosing only the one nearest neighbor in 2^d orthogonal on opposite directions in the projected cell space, where d is the density ridge dimensionality (for example, two nearest neighbors are chosen for $d=1$ or trajectories, and four are chosen for $d=2$ or surfaces). We chose the directions based on first d eigenvectors of the Hessian, leveraging the observation that density ridges typically extend in the same directions as these eigenvectors. Optional filters can be applied to remove edges based on edge length and direction. The output of this step is a graph representation of density ridges, without imposing prior assumptions on its structure type, and which does not require all cells to be connected, and we call this algorithm SimpleNNG and implemented it in our Python package. To construct a minimal graph representation that is guaranteed to connect every cell, we construct a minimum spanning tree (MST) graph based on SimpleNNG output with two additional steps: first, we add edges to connect every connected component to its nearest neighbor in each of the other connected components; and second, we extract an MST of the whole graph. The MST algorithm is robust but makes the assumption of a tree structure. The SimpleNNG algorithm does not make such an assumption and is thus more suitable for cyclic trajectory types or disconnected graphs. For simplicity, the MST algorithm is used in all analyses in this paper unless otherwise indicated. For use with the dynbenchmark package, we further convert a graph to a dynbenchmark-compatible graph format with backbone cells assigned based on betweenness centralities. Cells that pass a betweenness centrality threshold of 10-fold the total number of cells are assigned as backbones of the graph.

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

Data availability

The 339-dataset benchmark dataset published by Saelens et al.[10] was downloaded from https://zenodo.org/record/1443566. The unnormalized performance scores were extracted from https://github.com/dynverse/dynbenchmark_results/blob/1ac55e6c54a950890208b1f7730092d39783dfd2/06-benchmark/benchmark_results_unnormalised.rds. The normalized scores were computed as in ref. [10], with the scaling factors kept to the same values as the original methods benchmarked. Other singe-cell datasets analyzed in this paper were from refs. [6,9,13,14,18,23–25]. The Scanpy package[8] was used for preprocessing steps when needed, as described previously[26]. We created a Zenodo record, https://zenodo.org/record/3710980 (ref. [27]), that contains all the input data used in this paper.

Code availability

All methods described in this paper are implemented in an open-source Python package, quasildr (https://github.com/jzthree/quasildr). A Code Ocean capsule of the package is provided (https://doi.org/10.24433/CO.9410876.v1)[28].

References

19. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* **36**, 421–427 (2018).
20. Malkov, Yu. A. & Yashunin, D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**, 824–836 (2020).
21. Dong, W., Charikar, M. & Li, K. Efficient K-nearest neighbor graph construction for generic similarity measures. In *WWW 2011: Proceedings of the 20th International Conference on World Wide Web* 577–586 (https://doi.org/10.1145/1963405.1963487, 2011).
22. Saragih, J. M., Lucey, S. & Cohn, J. F. Face alignment through subspace constrained mean-shifts. In *2009 IEEE 12th International Conference on Computer Vision* 1034–1041 (https://doi.org/10.1109/ICCV.2009.5459377, 2009).
23. Briggs, J. A. et al. The dynamics of gene expression in vertebrate embryogenesis at single-cell resolution. *Science* **360**(6392), eaar5780 (2018).

24. Nestorowa, S. et al. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood* **128**(8), e20–31 (2016).
25. Paul, F. et al. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell* **163**, 1663–1677 (2015).
26. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
27. Zhou J. & Troyanskaya, O. An analytical framework for interpretable and generalizable single-cell data analysis (Dataset). *Zenodo* https://doi.org/10.5281/zenodo.3710980 (2020).
28. Zhou, J. & Troyanskaya, O. An analytical framework for interpretable and generalizable single-cell data analysis. *Code Ocean* https://doi.org/10.24433/CO.9410876.v1 (2021).

Acknowledgements

The authors acknowledge all members of the Troyanskaya laboratory and Zhou laboratory for helpful discussions. This work was performed using the high-performance computing resources (supported by the Scientific Computing Core) at the Flatiron Institute and the BioHPC at UT Southwestern Medical Center. J.Z. is supported by the Cancer Prevention and Research Institute of Texas grant (RR190071) and the UT Southwestern Endowed Scholars program. O.G.T. is supported by National Institutes of Health grant nos. R01HG005998, U54HL117798 and R01GM071966, US Department of Health and Human Services grant no. HHSN272201000054C and Simons Foundation grant no. 395506. O.G.T. is a senior fellow of the Genetic Networks program of the Canadian Institute for Advanced Research.

Author contributions

J.Z. conceived the framework, developed the computational methods, and performed the analyses. J.Z. and O.G.T. wrote the paper.

Competing interests

The authors declare no competing interests.

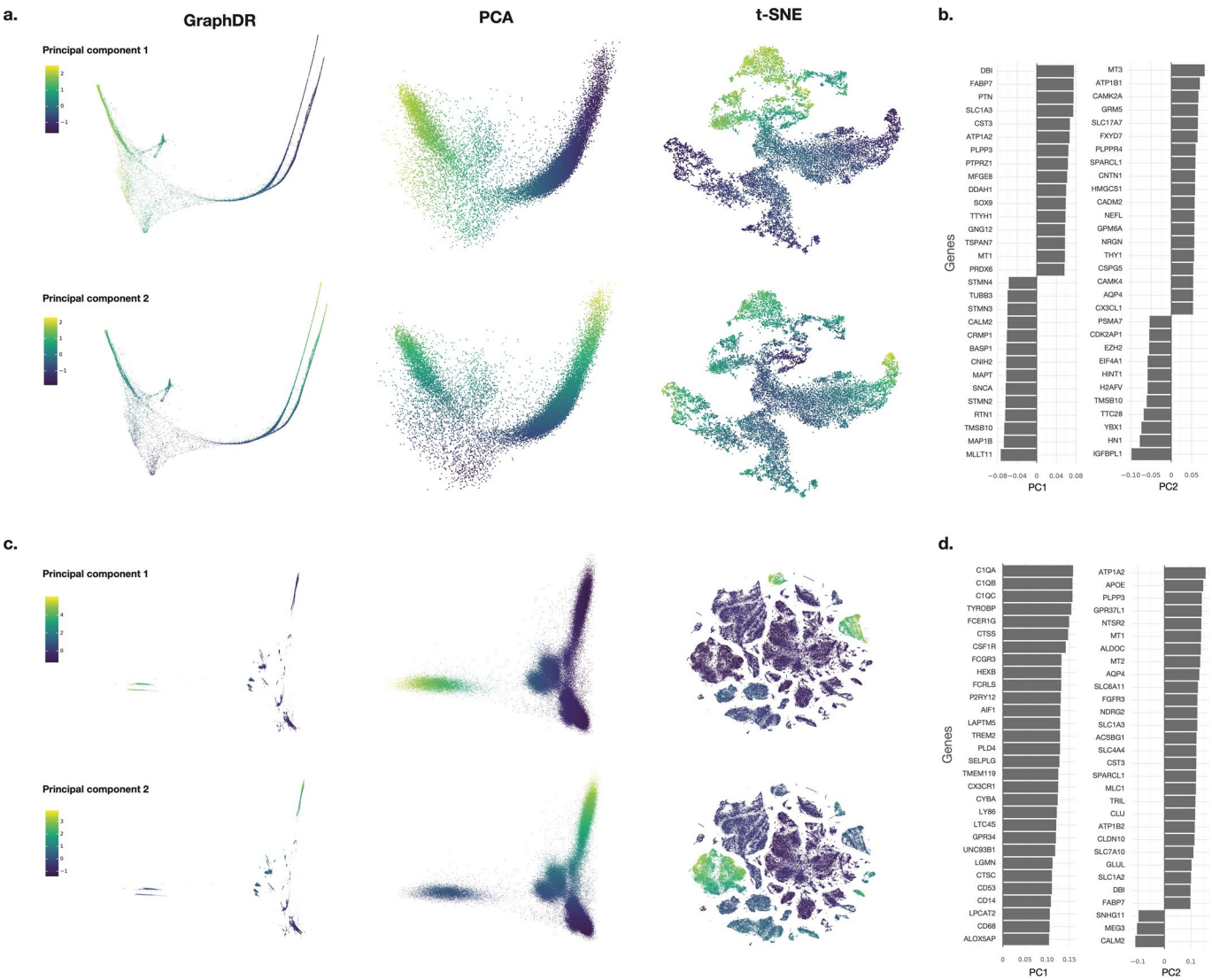
Additional information

Extended data are available for this paper at https://doi.org/10.1038/s41592-021-01286-1.

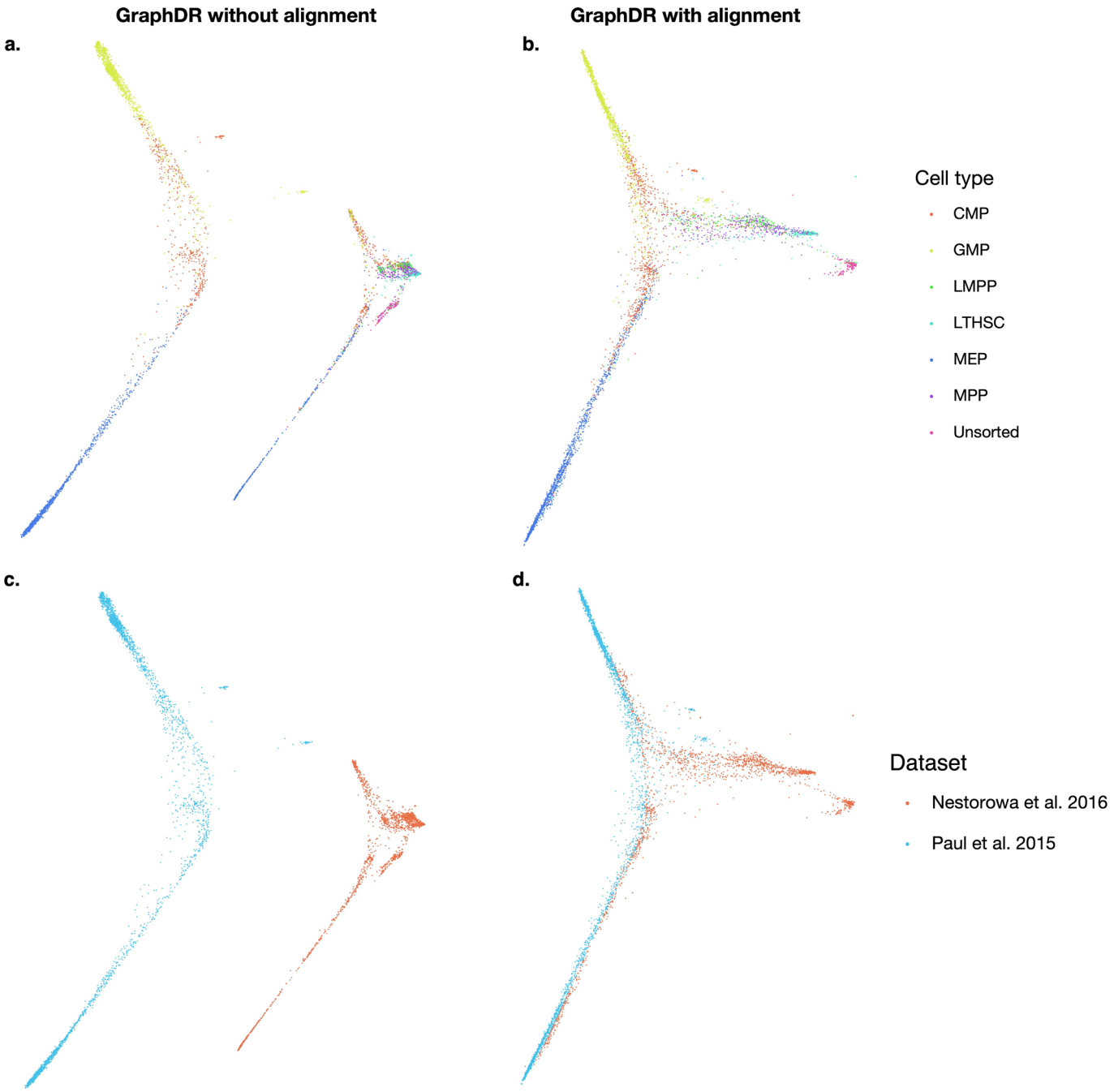
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-021-01286-1.

Correspondence and requests for materials should be addressed to Jian Zhou or Olga G. Troyanskaya.

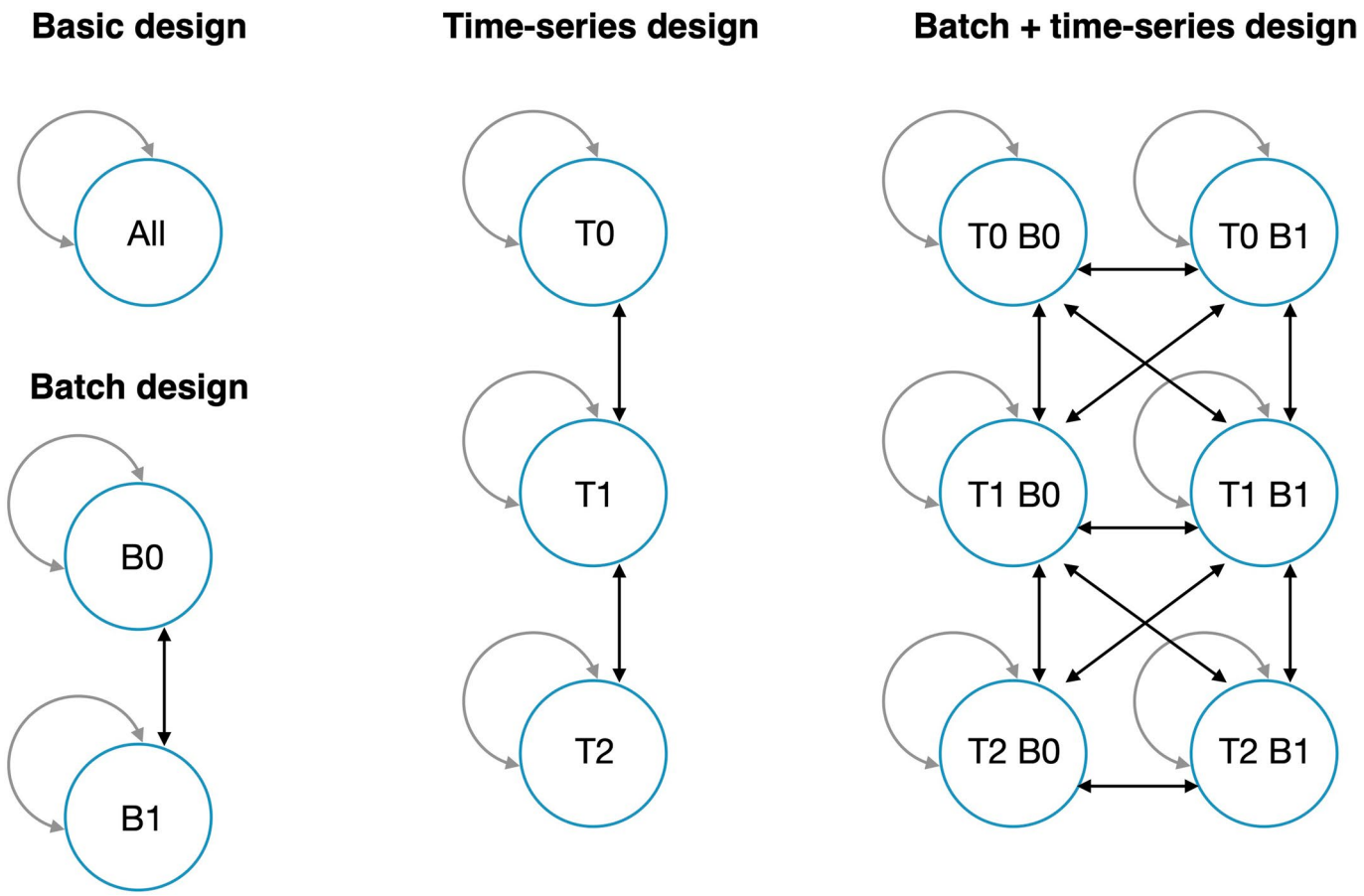
Peer review information *Nature Methods* thanks Yvan Saeys and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Lin Tang was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.



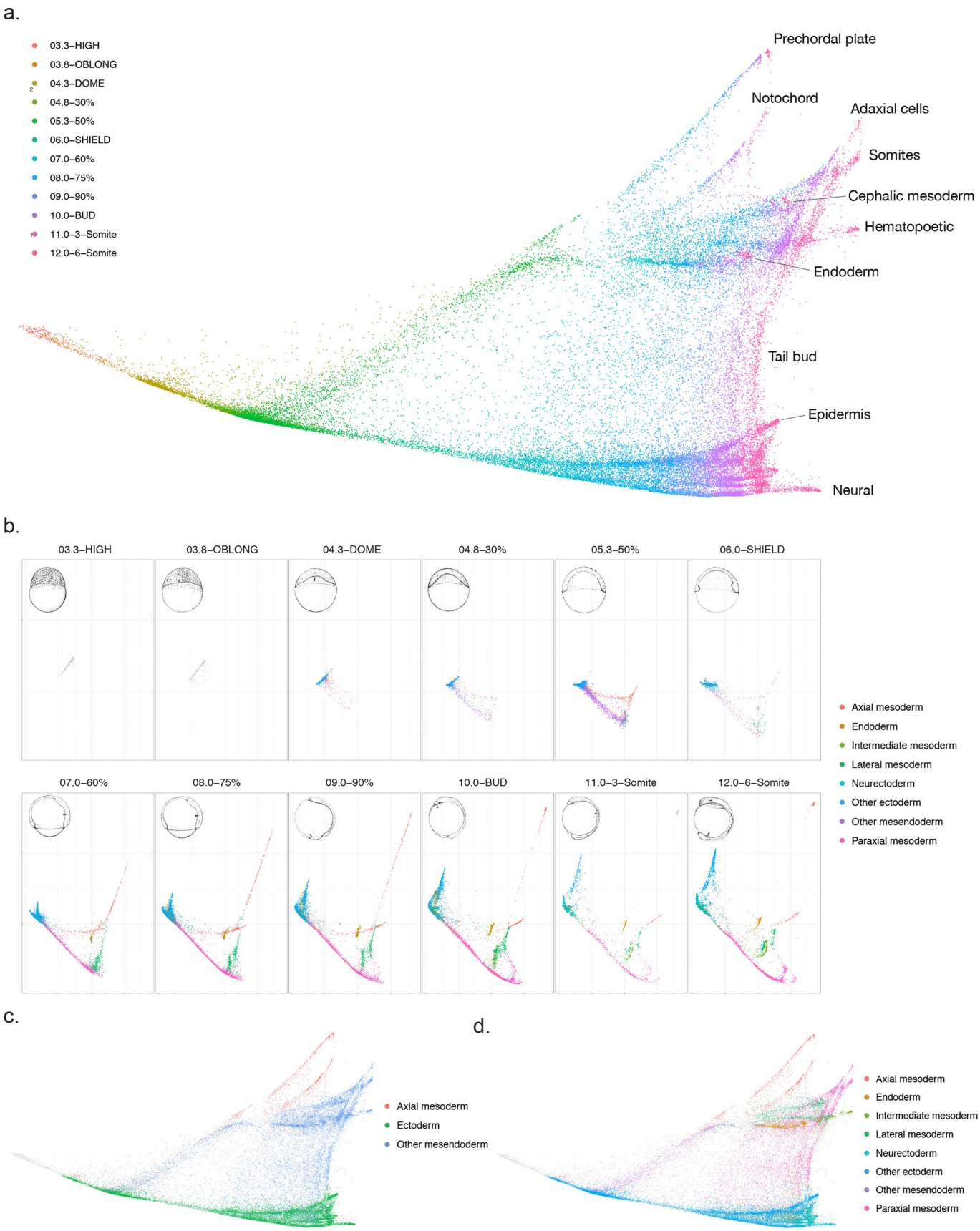
Extended Data Fig. 1 | Visualization of first two principal components in PCA, GraphDR, and tSNE visualizations. We compared the PCA, GraphDR, and tSNE representations by the values of first two principal components (PCs, shown by color) on a developing mouse hippocampus dataset (**a,b**) (Hochgerner et al. 2018[11]) and a mature mouse brain dataset (**c,d**) (Zeisel et al. 2018[18]). The top weighted genes by absolute values for the first two PCs are also shown (**b, d**).



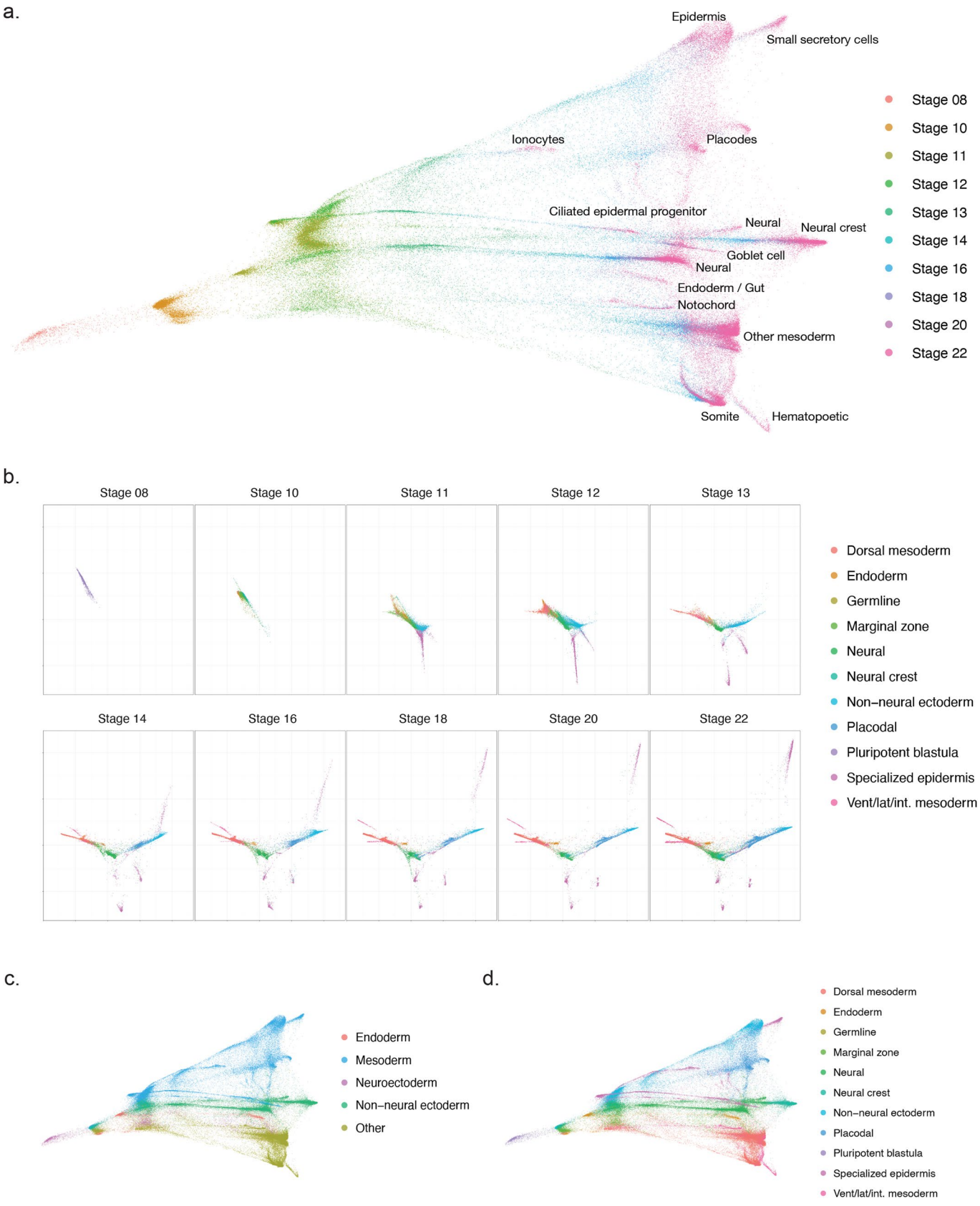
Extended Data Fig. 2 | Dataset alignment with GraphDR further improves dataset comparison. Comparison with applying GraphDR without (**a, c**) and with (**b, d**) graph-based dataset alignment on two hematopoietic datasets (Nestorowa et al. 2016[24] and Paul et al. 2015[25]). The GraphDR visualizations are colored by cell types (**a, b**) and by datasets (**c, d**). The cell types are common myeloid progenitors (CMPs), granulocyte-monocyte progenitors (GMPs), lymphoid multipotent progenitors (LMPPs), long-term HSCs (LTHSC), megakaryocyte-erythrocyte progenitors (MEPs), multipotent progenitors (MPPs). Specifically, GraphDR with graph-based dataset alignment constructs a joint graph that also connects the nearest neighbors between datasets (see batch design in Extended Data Fig. 3).



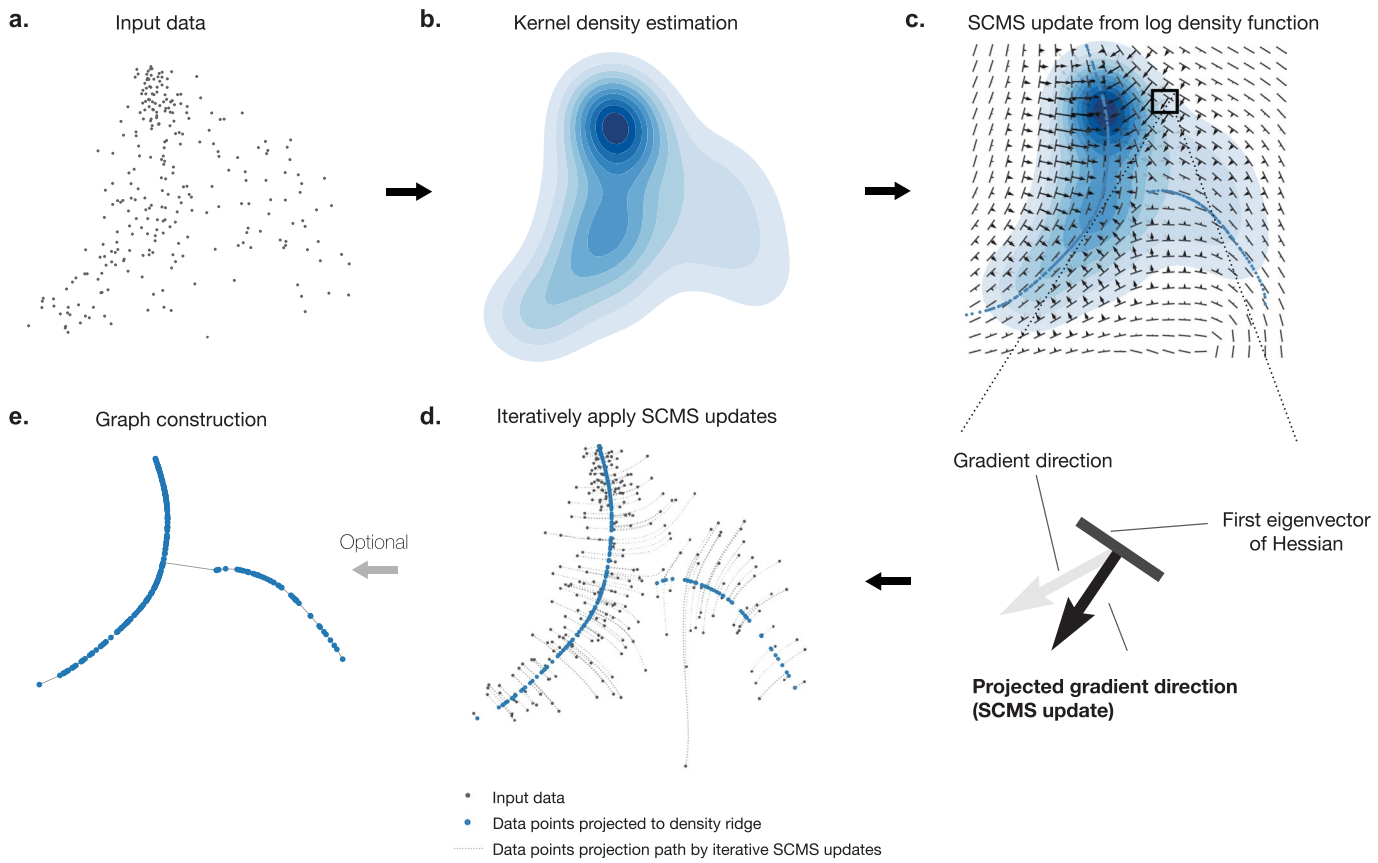
Extended Data Fig. 3 | Experimental design encoding through graph construction. Experimental design information can be encoded through graph construction in GraphDR. Each arrow indicates that nearest-neighbor connections are established between the two groups, where two connected cells are in the two different groups. Self-loop indicates nearest-neighbor connections from cells within a group. Basic design constructs a nearest neighbor graph using all cells, which is suitable for single-batch experiments or experiments with minimal batch effects. Batch design addresses batch effects by introducing nearest-neighbor connections between all pairs of batches, in addition to with-in batch nearest-neighbor connections. Time-series design extends basic design by only allowing connections between the same and adjacent time points. Batch + time series design introduces nearest neighbor connections between two batches in the same or adjacent time points.



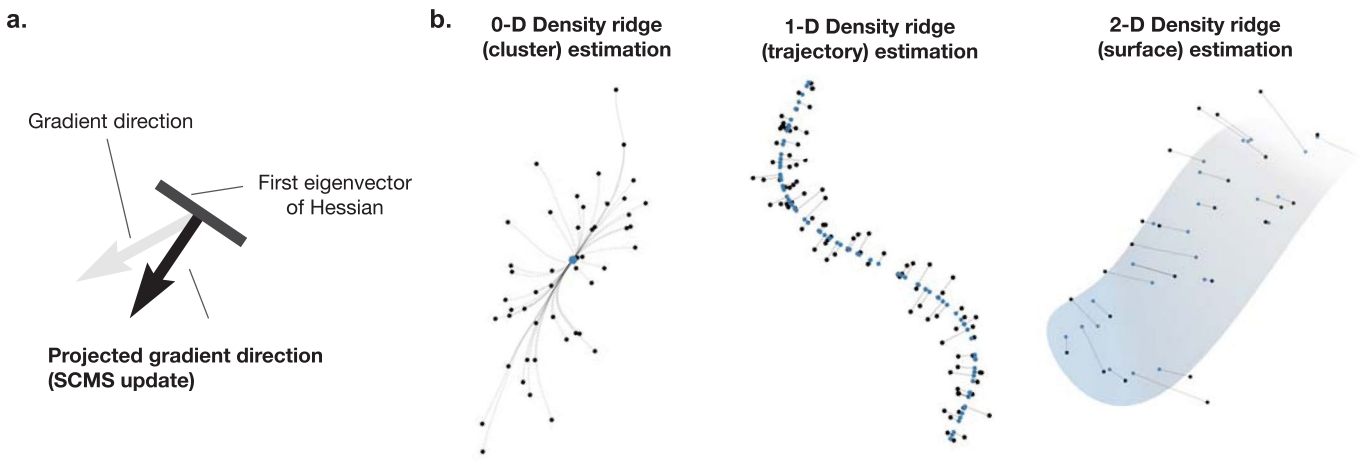
Extended Data Fig. 4 | Visualization of zebrafish whole embryo single-cell developmental landscape with GraphDR. Application of GraphDR to a single-cell dataset (Farrell et al. 2018[9]) with a time-series design. **a**. Single-cell visualization by GraphDR, colored by developmental stages. **b**. Comparative visualization of developmental stages. This shows the 'cross-section' view by visualizing the second and third dimensions. **c,d**. Single-cell visualization by GraphDR, colored by cell origins.



Extended Data Fig. 5 | Visualization of Xenopus tropicalis whole embryo single-cell developmental landscape with GraphDR. This is an example of applying GraphDR to a single-cell dataset with a batch + time-series design (Briggs et al. 2018[23]). **a**. Single-cell visualization by GraphDR, colored by developmental stages. **b**. Comparative visualization of developmental stages. This shows the 'cross-section' view by visualizing the second and third dimensions. **c**,**d**. Single-cell visualization by GraphDR, colored by cell origins.



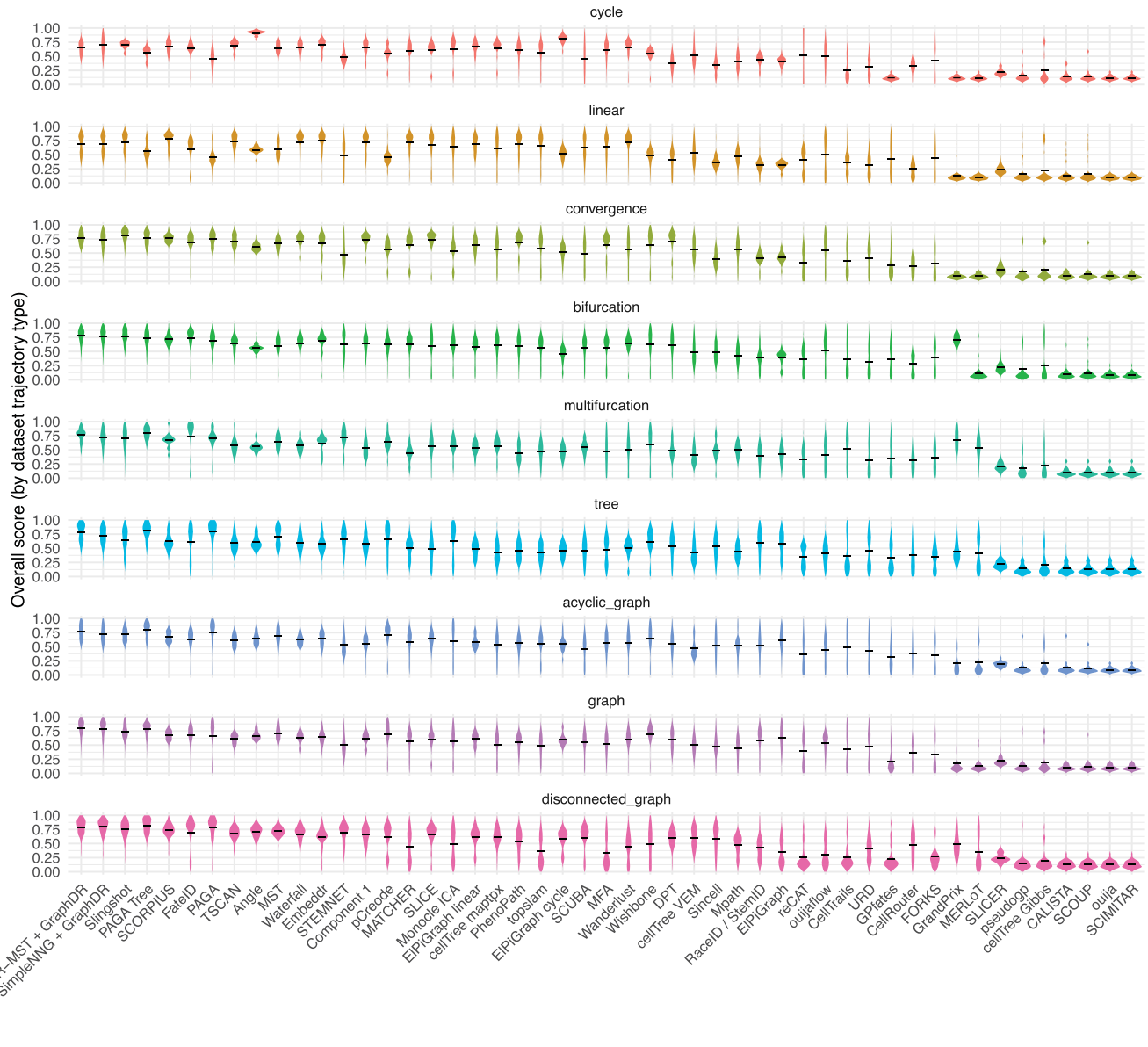
Extended Data Fig. 6 | Schematic overview of StructDR density ridge estimation procedures with the SCMS algorithm. (**a**,**b**) StructDR starts from performing kernel density estimation with Gaussian kernel on the input cells. (**c**) Based on the estimated density function, and a selected density ridge dimensionality d (d = 1 in this example), the SCMS update can be derived for any position in the space from the gradient and Hessian of the log density function. For any data point or position of interest, iteratively updating the position with the SCMS update will project the data point or position to density ridges of chosen dimensionality. (**d**). Optional step: construct graph connecting points on the density ridges with one of two optional methods (Methods). The backbone of the graph can be specified based on a betweenness centrality threshold.



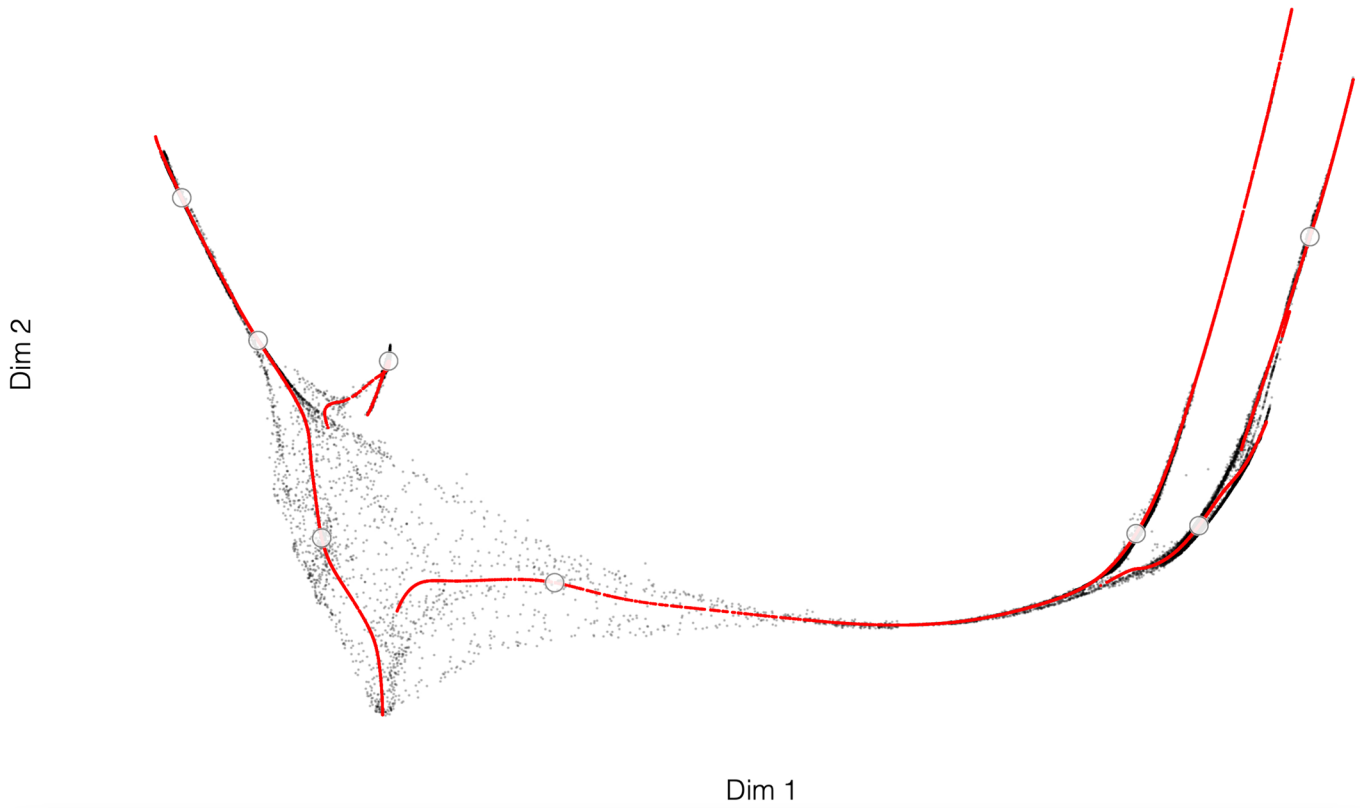
c.

Density ridge dimensionality	Density ridge mathematical property	Subspace contraint for SCMS update	Example related single-cell concepts
0 (Cluster)	Gradient is 0	No projection	Cell type
1 (Trajectory)	Gradient is 0 in directions other than the 1st eigenvector of Hessian	Project onto subspace spanned by 2nd to last eigenvectors of Hessian	Developmental trajectory
2 (Surface)	Gradient is 0 in directions other than the 1st and 2nd eigenvectors of Hessian	Project onto subspace spanned by 3rd to last eigenvectors of Hessian	Spatial map, complex heterogeneous cell population
k	Gradient is 0 in directions other than the top k eigenvectors of Hessian	Project onto subspace spanned by k+1-th to last eigenvectors of Hessian	

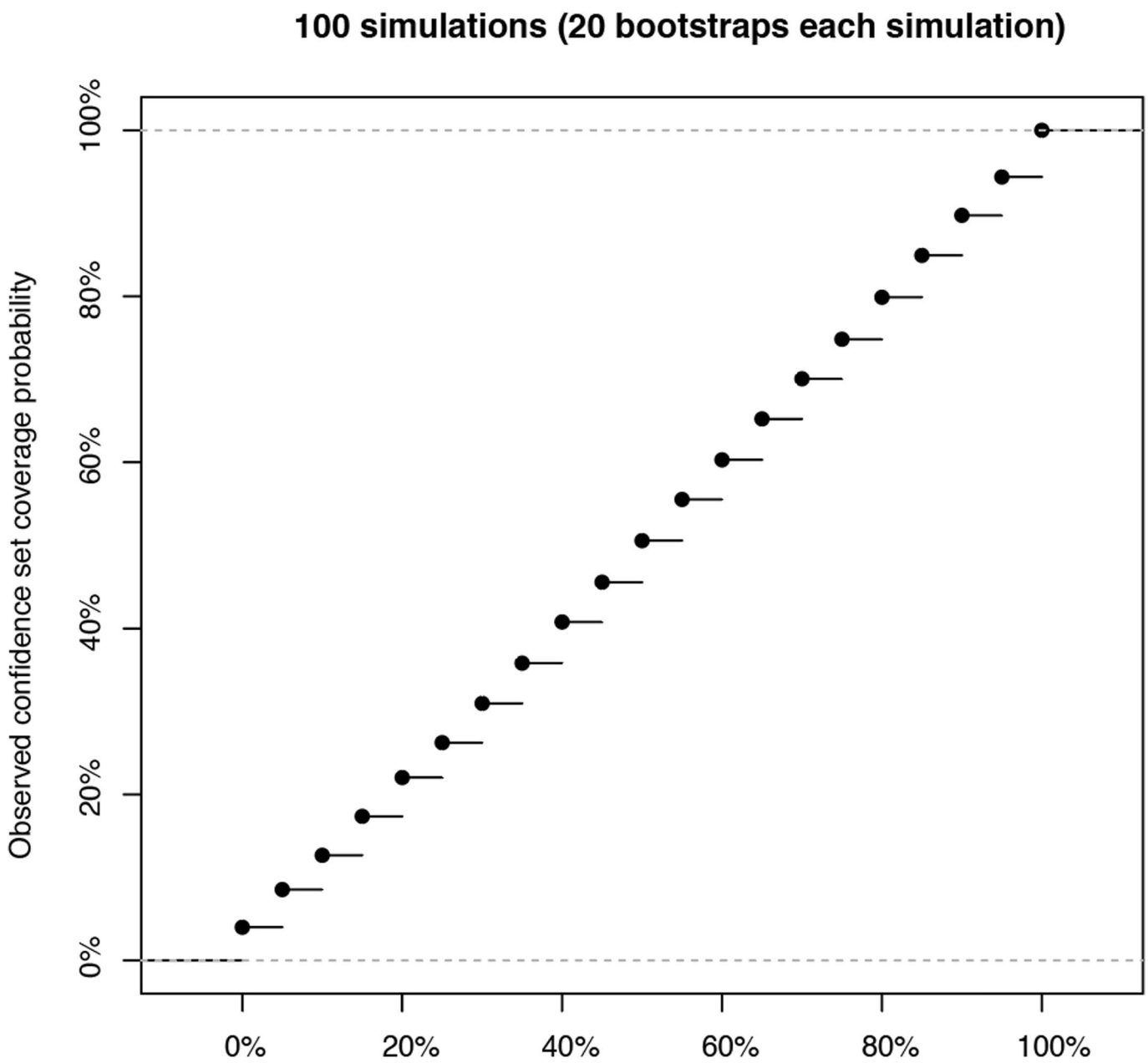
Extended Data Fig. 7 | Overview of the unified framework of cluster, trajectory, and surface analysis with StructDR. (**a**) StructDR uses the SCMS update for the estimation of clusters, trajectories, and surfaces, which can all be derived based on gradient and Hessian of log density function. (**b**) Examples of projection paths by SCMS updates for zero, one, and two-dimensional density ridges. (**c**). Comparisons of SCMS algorithms for 0, 1, 2, or k-dimensional density ridges. The SCMS update can identify any k-dimensional density ridges, by projecting a gradient-based update onto subspace spanned by the k+1 th to last eigenvector of the Hessian of log density function.



Extended Data Fig. 8 | Performance score distributions on the 339-dataset benchmark shown by dataset type. Per-dataset performance scores are computed based on Saelens et al. 2019. The performance score distributions are shown with violin plots, separated into panels by dataset types. The performance of applying StructDR + GraphDR with two graph construction algorithms, MST and SimpleNNG, are shown along with the performance of other algorithms benchmarked in Saelens et al. 2019[10].



Extended Data Fig. 9 | Trajectory identification with zero, one, and two-dimensional density ridges example on a developmental hippocampus single-cell dataset. The circle symbols indicate zero-dimensional density ridge positions (local maxima of density function). The red dots indicate one-dimensional density ridge positions (trajectory). The black dots indicate two-dimensional density ridge positions.



Extended Data Fig. 10 | Simulation studies of confidence sets construction with nonparametric ridge estimation. 100 simulation datasets were generated. For each dataset the confidence sets for each estimated trajectory were estimated with 20 bootstraps. x-axis shows the expected coverage probabilities of the constructed confidence sets. y-axis shows the observed proportion that the true trajectory position is covered by the confidence set.

nature portfolio

Corresponding author(s): Jian Zhou, Olga Troyanskaya

Last updated by author(s): Jul 6, 2021

Reporting Summary

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

Statistics

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

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

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

Software and code

Policy information about availability of computer code

Data collection	No software was used
Data analysis	The code is available at Github: https://github.com/jzthree/quasildr Additional softwares used: Single cell data preprocessing and analysis methods: scanpy(1.4.2) Nearest neighbor graph construction: nmslib(1.8)

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

Data

Policy information about availability of data

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

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

The 339-dataset benchmark dataset published by Saelens et al.10 was downloaded from https://zenodo.org/record/1443566. The unnormalized performance scores were extracted from https://github.com/dynverse/dynbenchmark_results/blob/1ac55e6c54a950890208b1f7730092d39783dfd2/06-benchmark/benchmark_results_unnormalised.rds. The normalized scores were computed as in 10, with the scaling factors kept to the same values as the original methods

benchmarked. Other singe-cell datasets analyzed in this manuscript were from the following publications6,13,14,18,23–26. Scanpy package8 was used for preprocessing steps as described in 27 when needed. We created a Zenodo record for https://zenodo.org/record/3710980 that contains all the input data used in this manuscript.

Field-specific reporting

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

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

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

Life sciences study design

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

Sample size	This study uses only published datasets from prior studies. No sample size calculation is involved in this study.
Data exclusions	No data is excluded from the datasets used.
Replication	Replication is not relevant to this study. In this study we developed computational methods and evaluated these methods on a diverse set of datasets.
Randomization	Randomization is not relevant to this study and no new data is collected.
Blinding	Blinding is not relevant to this study and no new data is collected.

Reporting for specific materials, systems and methods

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

Materials & experimental systems

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

Methods

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