Article

TopOMetry systematically learns and evaluates the latent dimensions of single-cell atlases

Davi Sidarta-Oliveira[1,2] and Licio A. Velloso[2]

1. Department of Physiology, Anatomy and Genetics, University of Oxford, United Kingdom
2. Laboratory of Cell Signaling, Obesity and Comorbidities Research Center, School of Medical Sciences, University of Campinas, Brazil

Correspondence:

L. A. Velloso - lavellos@unicamp.br

Obesity and Comorbidities Research Center, University of Campinas

13083-864, Campinas, SP, Brazil

Abstract

A core task in single-cell data analysis is recovering the latent dimensions encoding the genetic and epigenetic landscapes inhabited by cell types and lineages. However, consensus is lacking for optimal modeling and visualization approaches. Here, we propose these landscapes are ideally modeled as Riemannian manifolds and present TopOMetry, a computational toolkit based on Laplacian-type operators to learn these manifolds. TopOMetry systematically learns and evaluates dozens of possible representations, eliminating the need to choose a single dimensional reduction method *a priori*. The learned visualizations preserve more original information than current PCA-based standards across single-cell and non-biological datasets. TopOMetry allows users to estimate intrinsic dimensionalities and visualize distortions with the Riemannian metric, among other challenging tasks. Illustrating its hypothesis generation power, TopOMetry suggests the existence of dozens of novel T cell subpopulations consistently found across public datasets that correspond to specific clonotypes. TopOMetry is available at https://github.com/davisidarta/topometry.

Keywords: dimensional reduction, knowledge-representation, Laplace-Beltrami, genomics, latent models

Introduction

Analyzing high-dimensional data (i.e., data in which the number of variables per observation exceeds hundreds or thousands) is a challenge shared among fields as diverse as signal processing, economics, and structural chemistry - and, more recently, the life sciences, particularly with the development of single-cell experiments. Single-cell assays have inaugurated a new era in the unbiased molecular sampling of the cellular diversity in biological systems, from cell lines to whole organisms, leading to an ongoing revolution across the life sciences[1,2]. In such high-dimensional data (hd-data), information regarding cellular phenotype (e.g., gene expression, chromatin accessibility, etc.) is individually gathered for every cell, resulting in high-dimensional matrices containing up to millions of cells and dozens of thousands of variables for each. This property creates challenges in data analysis, collectively known as the *curse of dimensionality,* and requires sophisticated statistical and geometric methods to allow meaningful exploration and understanding of the latent features underlying the data. These are broadly referred to as *dimensionality reduction* techniques and include matrix decomposition methods, autoencoders, and graph-layout optimization algorithms.

Many dimensionality reduction approaches have been developed and recently reviewed elsewhere[3–5]; here, we will briefly describe the strategies behind these algorithms and outline their current use standards in sc-data. Briefly, there are two main ways to perform dimensionality reduction (DR): i) decomposing matrices; and ii) optimizing graph layouts. Matrix decomposition DR methods embed data into an arbitrary number of latent dimensions and include the classic Principal Component Analysis (PCA)[6,7], a linear method that involves the eigendecomposition of the covariance matrix. More recent non-linear dimensionality reduction (NLDR) methods are based on manifold-learning techniques and assume that the input data lie on or near a low-dimensional manifold, which is almost always the case in real-world datasets. In particular, spectral methods (i.e., based on the decomposition of Laplacian-type operators), such as Laplacian Eigenmaps (LE)[8] and Diffusion Maps (DM)[9,10], have arisen as robust non-linear alternatives, in which the eigendecomposition of Laplacian-type operators[8,9,11–13] yield orthonormal components that encode the latent graph structure of the manifold. The other broad class of DR techniques consists of graph-layout optimization methods. Their approach involves learning a similarity graph from samples, then minimizing a loss function to optimize the layout of this graph in 2 or 3-D for visualization. Graph-layout optimization (GLO) methods include the

popular t-Stochastic Neighborhood Embedding (t-SNE)[14], Uniform Manifold Approximation and Projection (UMAP)[15,16], and Pairwise Manifold Approximation and Projection (PaCMAP)[17], Minimum Distortion Embedding (MDE)[18]. Approaches based on autoencoders[19] are a recent exception to these two categories.

All these methods attempt to find adequate low-dimensional mappings to high-dimensional data, which are crucial in single-cell analyses. Theoretically, finding latent signals representative of cellular identity is intrinsically related to defining a finite number of cell lineages and types within a given dataset[20,21]. Ideally, in such mappings, moving through continuous topologies in the form of connected populations often correspond to differentiation trajectories or stimulus-response processes[21,22], while the space between discrete topologies should indicate the correspondence to different main cell types. The connection between this concept and an empirical view of the Waddington epigenetic landscape is direct and has been recently explored[2,20,21,23]. Such mappings are indispensable for downstream tasks in sc-data analysis[24], such as clustering, RNA velocity projection[25,26], batch-correction[27,28], trajectory inference[29], and pseudotime estimation[21,30]. Current practice involves using the first few principal components (a hyperparameter commonly set by the user) to compute neighborhood graphs for downstream tasks, such as clustering and graph-layout optimization (GLO) algorithm (i.e., t-SNE or UMAP) for visualization. The GLO procedure is rarely performed on the full set of genes/features (or highly-variable features); instead, the default approach performs it with either the top principal components or other latent dimensions learned by an autoencoder as with *scvi-tools*[31]. Despite their popularity, little evidence supports these currently held standards.

Defining which dimensionality reductions are the best, from both theoretical and practical perspectives, has remained an active challenge when analyzing sc-data, as generating and evaluating the reliability of these mappings is an extremely challenging task. No known ground truth exists, and real-world data are immensely more complex than any toy datasets that could be generated to investigate the suitability of DR techniques. Some attempts to compare different DR methods have been made[5,32,33]; however, they often rely on metrics associated with prior clustering results (e.g., the Adjusted Rand Index), thus holding no guarantees that a good-performing method preserves the underlying geometry of the data. Qualitatively assessing the representation of known biological ground truths as visualization landmarks is also challenging, as only a few are known (e.g., the cell cycle as a closed loop), and even when present, they are prone to promote unconscious confirmatory biases and cannot be extended to

the entire representation. The current gold standard of experimentally validating some hypotheses drawn from the learned representations cannot be considered a validation of the whole representation *per se* either. Moreover, each algorithm's assumptions (often implicit and by construction) regarding data's statistical and geometrical properties are also disturbingly challenging to assess in practice. For example, the universally used PCA holds the strong assumption that data is uniformly distributed and lies approximately in a series of hyperplanes (the principal components) by definition, which generates misleading results when faced with non-uniform or non-linear structure by missing local information and introducing linear distortions[6,14,15,34–37]. The popular UMAP[15], whereas more extensive by considering that the data is dispersed over or close to a locally connected Riemannian manifold, assumes this distribution is uniform across a manifold, with a locally constant metric. As such, representations of data with non-uniform data sampling could be filled with artifacts without any warning to users. As most downstream analysis and biological conclusions drawn from sc-data rely on the lower-dimensional mappings learned by DR, it is crucial for them to be formally guaranteed to represent latent biology. These should be independent of the data distribution or the underlying geometry, which should be considered unknown even for extensively investigated systems. Regardless, this is not the case when using the current standard approach (i.e., using UMAP or t-SNE on the top Principal Components); in fact, its assumptions likely do not hold true for all datasets, in which case artifactual distortions can be introduced in the learned representations. Any inferences made from these representations are critical and will impact downstream analyses, thus raising concerns about the credibility of the vast literature in which this practice was adopted.

Despite the importance of minimizing assumptions regarding the latent geometry underlying high-dimensional data, an approach that fully yields the power of topological Laplacian-type operators is yet to be proposed. These operators have been widely and successfully used in many applications, from computer vision to natural language processing, but not in single-cell biology. They are particularly advantageous because they approximate the Laplace-Beltrami Operator (LBO) at the limit of large data (i.e., when the number of samples is very large, as in sc-data). We consider the extremely loose assumption that data lies on a Riemannian manifold. It is a fundamental result in mathematics that the LBO encodes all the local geometric information of a Riemannian manifold with an unknown shape[8,9,11,36,38]. The top eigenfunctions of this operator can then be used to approximate the original data[9] so that each eigenfunction

carries a fundamental harmonic of the data[8,9,13]. In addition, because current theoretical biology points to an underlying Waddington landscape within the sampled biological system[20], these can be naturally modeled as a Riemannian manifold. We refer to such theoretical manifolds as *phenotypic manifolds*. We reasoned that recovering such manifolds by approximating the LBO would thus render optimal sc-data representations while holding no prior assumptions regarding the underlying data geometry. Here, we show how combining LBO eigenfunctions with existing projection (GLO) techniques yields representations of high-dimensional data that preserve more original information than those learned with current standards (PCA-based) or with GLO alone. This strategy makes the Riemannian metric[36] a natural metric for evaluating the distortion of these visualizations, a most critical feature that the field currently lacks. We present TopOMetry, a python toolkit to combinatorially obtain numerous LBO approximations using different computational approaches and systematically evaluate them from a quantitative and qualitative perspective. We show this strategy is remarkably superior to the currently held PCA-based approach or to using stand-alone GLO implementations when analyzing single-cell data.

Results

A framework to systematically learn and evaluate manifolds in single-cell data

To achieve the task of approximating the LBO to recover the phenotypic manifolds underlying sc-data, we developed TopOMetry (Topologically-Optimized geoMetry), a toolkit that combines and enhances state-of-the-art techniques to approximate the LBO through weighted graph Laplacian-type operators. TopOMetry efficiently obtains the eigenspectra of these operators and uses them to approximate the LBO once again. Users can then visualize its graph representation with any existing graph-layout optimization (GLO) algorithms. This modular approach allows combinatorially evaluation of different options of kernels, decomposition, post-processing, and GLO algorithms. It eliminates the need to choose a single approach *a priori* and allows users to decide which of the possible representations better describes each dataset objectively.

TopOMetry takes as input a data matrix of variables per observation (e.g., genes per cell; Figure 1A). It then computes a *k*-nearest-neighbors graph to build a kernel matrix with

state-of-the-art similarity functions (the default adaptive bandwidth estimation is related to the local intrinsic dimensionality). TopOMetry includes a new family of manifold-adaptive kernels closely related to the intrinsic dimensionalities of the underlying manifold[39]. These kernels are then used to learn Laplacian-type operators, which asymptotically approximate the LBO (Figure 1B). Next, the eigenspectra of the learned Laplacian-type operators are obtained, allowing the estimation of the data's global intrinsic dimensionality (Figure 1C). The resulting orthonormal eigenbasis is related to a Fourier basis and encodes all latent signals present in the original data[8,9,13]. Such eigenbasis can then be used to compute a new weighted kernel (referred to as *graph kernel*) to obtain a second LBO approximation encoded as a similarity graph (the associated *topological graph;* Figure 1D), which can then be visualized with existing GLO algorithms (Figure 1E). These graphs can also be used for downstream tasks such as clustering, signal filtering, and pseudotime estimation (Figure 1F). In TopOMetry, dozens of visualizations obtained from the graphs or directly from the eigenbasis can be obtained with a single line of code. These representations' preservation of local[40] and global structure is then systematically evaluated (Figure 1G), alleviating practitioners' choice of a single similarity-learning or GLO method. The top-performing projections can then be qualitatively evaluated using the Riemannian metric to assess the distortion induced by the mapping (Figure 1G).

The outlined approach has numerous advantages in comparison to the current standards. First, it holds an extremely loose assumption regarding the underlying data geometry compared to PCA or UMAP (the data lies in or close to a Riemannian manifold). Second, such an assumption is mathematically solid and intrinsically related to the core biological goal in sc-data analysis (i.e., to recover the Waddington landscape or an extended version thereof). Third, it involves the estimation of the manifold local and global dimensionality, which alleviates the similarity-learning process and eliminates the need to guess a number of latent dimensions to keep for downstream analysis (as needed when using PCA or autoencoders). Another major advantage is using the Riemannian metric[36] as a “distortion ruler,” allowing qualitative assessment of the distortion present in two-dimensional mappings and the robust estimation of the global and local dimensionality of the underlying manifold. Finally, TopOMetry allows users to obtain and evaluate dozens of possible representations with a single command, eliminating the need to choose a pre-defined method (e.g., UMAP on PCA) and allowing users to pick which performs best for each case. This is the toolkit's main advantage and represents a

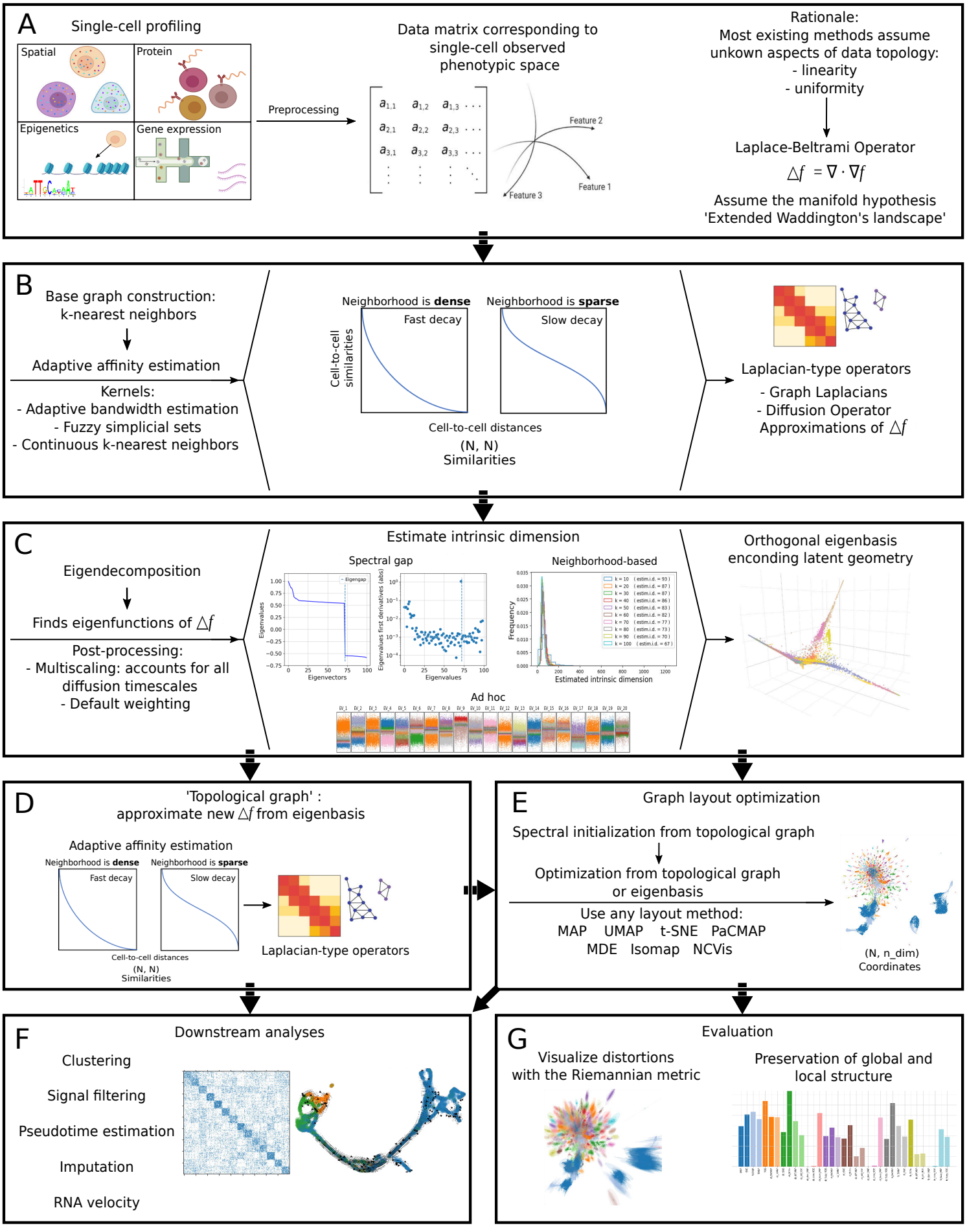

A
Single-cell profiling
Spatial
Protein
Epigenetics
Gene expression
Preprocessing
Data matrix corresponding to single-cell observed phenotypic space
Feature 2
Feature 1
Feature 3
Rationale:
Most existing methods assume unkown aspects of data topology:
- linearity
- uniformity
Laplace-Beltrami Operator
$\Delta f = \nabla \cdot \nabla f$
Assume the manifold hypothesis
'Extended Waddington's landscape'
B
Base graph construction:
k-nearest neighbors
Adaptive affinity estimation
Kernels:
- Adaptive bandwidth estimation
- Fuzzy simplicial sets
- Continuous k-nearest neighbors
Neighborhood is dense
Fast decay
Neighborhood is sparse
Slow decay
Cell-to-cell similarities
Cell-to-cell distances
(N, N)
Similarities
Laplacian-type operators
- Graph Laplacians
- Diffusion Operator
Approximations of Δf
C
Eigendecomposition
Finds eigenfunctions of Δf
Post-processing:
- Multiscaling: accounts for all diffusion timescales
- Default weighting
Estimate intrinsic dimension
Spectral gap
Eigenvalues
Eigenvectors
Eigenvalues first derivatives (abs)
Neighborhood-based
Frequency
Estimated intrinsic dimension
Ad hoc
Orthogonal eigenbasis enconding latent geometry
D
'Topological graph' :
approximate new Δf from eigenbasis
Adaptive affinity estimation
Neighborhood is dense
Fast decay
Neighborhood is sparse
Slow decay
Cell-to-cell distances
(N, N)
Similarities
Laplacian-type operators
E
Graph layout optimization
Spectral initialization from topological graph
Optimization from topological graph or eigenbasis
Use any layout method:
MAP UMAP t-SNE PaCMAP
MDE Isomap NCVis
(N, n_dim)
Coordinates
F
Downstream analyses
Clustering
Signal filtering
Pseudotime estimation
Imputation
RNA velocity
G
Evaluation
Visualize distortions with the Riemannian metric
Preservation of global and local structure

Figure 1. Schematic overview of single-cell data analysis with TopOMetry. (A) Single-cell experiments are preprocessed into high-dimensional matrices of cells and measured variables (e.g., gene expression, protein content, chromatin accessibility) and require dimensional reduction tools for their analysis. Most tools assume unknown aspects of the underlying geometry (e.g., linearity) or distribution (e.g., uniformity). Instead, relying on the Laplace-Beltrami Operator (LBO) and its eigenfunctions allows learning such geometry assuming only the manifold hypothesis. (B) Approximating the LBO involves constructing a k-nearest-neighbors (kNN) graph using adaptive affinity estimation, which can be done through several kernels so to make the affinity graph insensitive to neighborhood densities. The Laplacian-type operators (particularly the anisotropic diffusion operator) are approximations of the LBO. The resulting matrices can be used for several tasks, such as imputation of missing data, signal filtering and interpolation, and graph sparsification and coarsening. (C) The eigendecompositions of the LBO approximations yield eigenvectors and eigenvalues that are weighted or multiscaled to form a new orthogonal eigenbasis. The eigenvalues can be used to estimate an eigengap or spectral gap to estimate the intrinsic dimensionality of the data. The intrinsic dimensionality can also be estimated using neighborhood-based methods (e.g., FSA and MLE) or *ad-hoc* inspection of eigenvectors for discriminative potential. (D) A second neighborhood graph is learned from the eigenbasis, and its Laplacian-type operators are used to obtain a new LBO approximation, rendering 'topological graphs'. (E) From a spectral initialization obtained from the topological graph, any graph layout optimization (GLO) can be used on the topological graph or the eigenbasis for visualization. (F) Downstream tasks such as clustering, RNA velocity estimation, and imputation can be performed with the learned topological graphs and layouts. (G) The learned eigenbases and visualizations can be evaluated regarding the preservation of global and local structure, and the Riemannian metric can be used to visualize distortions in layouts to aid biological interpretation.

fundamental change of paradigm in sc-data analysis: instead of *a priori* electing a single method to be used at all times across all datasets, it allows users to discover which particular model works best for representing each individual dataset before other downstream tasks. Computationally, TopOMetry is easy-to-use and centered around a single python class, the *TopOGraph* object, which orchestrates the analyses (Supplementary Figure 1).

Representations learned with TopOMetry preserve local and global structures across single-cell datasets

We first performed a qualitative sanity test on TopOMetry by running it on synthetic toy data (Supplementary Figure 2A). The eigenfunctions of the Laplacian-type operators used in the toolkit pass negative control checks with uniformly distributed noise and square and also positive control checks with normally distributed classes, as visualized by coloring sample points by their first eigencomponent or principal component (Supplementary Figure 2B). The representation of uniformly distributed classes as points or straight lines by the first two non-trivial eigenfunctions is related by their encoding of manifold structure (Supplementary Figure 2B), which further highlights the power of this approach to discover the true geometry of the data regardless of the sampling distribution. We also reinforce that this approach can unfold curved manifolds, such as the classic 'S'-shape and the swiss-roll examples (Supplementary Figure 2B-C), and stress the inadequacy of trying to use PCA to represent non-linear geometries (Supplementary Figure 2B-C). For an additional test on high-dimensional toy data with ground-truth labels, we obtained embeddings from the MNIST handwritten digits dataset with TopOMetry and existing methods (Supplementary Figure 2D).

To determine TopOMetry's performance in real-world sc-data, we curated a diverse set of 20 single-cell datasets from different biological systems with varying numbers of cells and variables per cell across diseases, species, and technologies (Table 1). The datasets were then processed following current standards in scRNA-seq analysis[24]: i) library size normalization; ii) logarithmic transformation; iii) selection of highly variable genes (HVG), and iv) scaling (Z-score normalization, i.e., mean centering and variance clipping). For each dataset, we computed the first 100 principal components and obtained two-dimensional (2-D) projections with three existing DR methods: i) t-SNE[14]; ii) UMAP[15]; iii) PaCMAP[17]. Projections were computed using

the principal components (the current standard) or the full data matrices. We then used TopOMetry to obtain 2-D projections using similar graph-layout optimization (GLO) methods. Each visualization was then scored regarding the preservation of the global and local structure. The Mean Reconstruction Error (MRE) was used for quantifying the preservation of the global structure as recently proposed[41], with PCA being a reference by construction (Figure 3A). It has been shown that the preservation of global structure is mainly driven by the initialization given to GLO algorithms[42], thus making the preservation of local structure a more important factor to consider when choosing between options. The trustworthiness score[40] was used to assess the preservation of local structure (Figure 3B). In addition, we also propose a score based on the correlation of geodesic distances in the high- and low-dimensional spaces (Figure 3C), drawing from previous approaches[33,43]; however, its high computational cost makes its application impractical for most real-world datasets.

Across all datasets, the eigenbases (Figure 2D) and 2-D representations (Figure 2E) that yielded the highest local scores (LS) for all datasets were generated with TopOMetry. Global scores (GS) were as high as PCA for TopOMetry's eigenbases (Supplementary Figure 3A) and discretely lower for projections when using TopOMetry compared to GLO on data or top principal components on some datasets, with a maximum difference of 0.06 for the *AgeingMouseBrain* dataset (Supplementary Figure 3B), further suggesting that the preservation of local structure is a more robust metric to discriminate against different visualizations. We found that PCA yielded lower LS for all datasets (Figure 2D), as expected from a global and linear method. Strikingly, we also found that projections derived from using GLO methods on the top principal components had lower LS when compared to using GLO alone (Figure 2E), further suggesting that this practice could be inadequate despite being currently held as a default approach. The opposite happens when using TopOMetry's Laplacian-type eigenbases to build the neighborhood graphs projected with PaCMAP or the diffusion potential from these eigenbases with MAP, as these approaches achieve higher LS than using GLO alone (Figure 2E). Within TopOMetry, the variation of LS differed mostly across eigenmap methods, with LE scoring lower than DM and msDM, and little to no variation across different kernel methods (Figure 2E). These results show the superiority of TopOMetry over the standard workflow or GLO methods alone and highlight the value of critically evaluating methods before electing one for the downstream analysis approach.

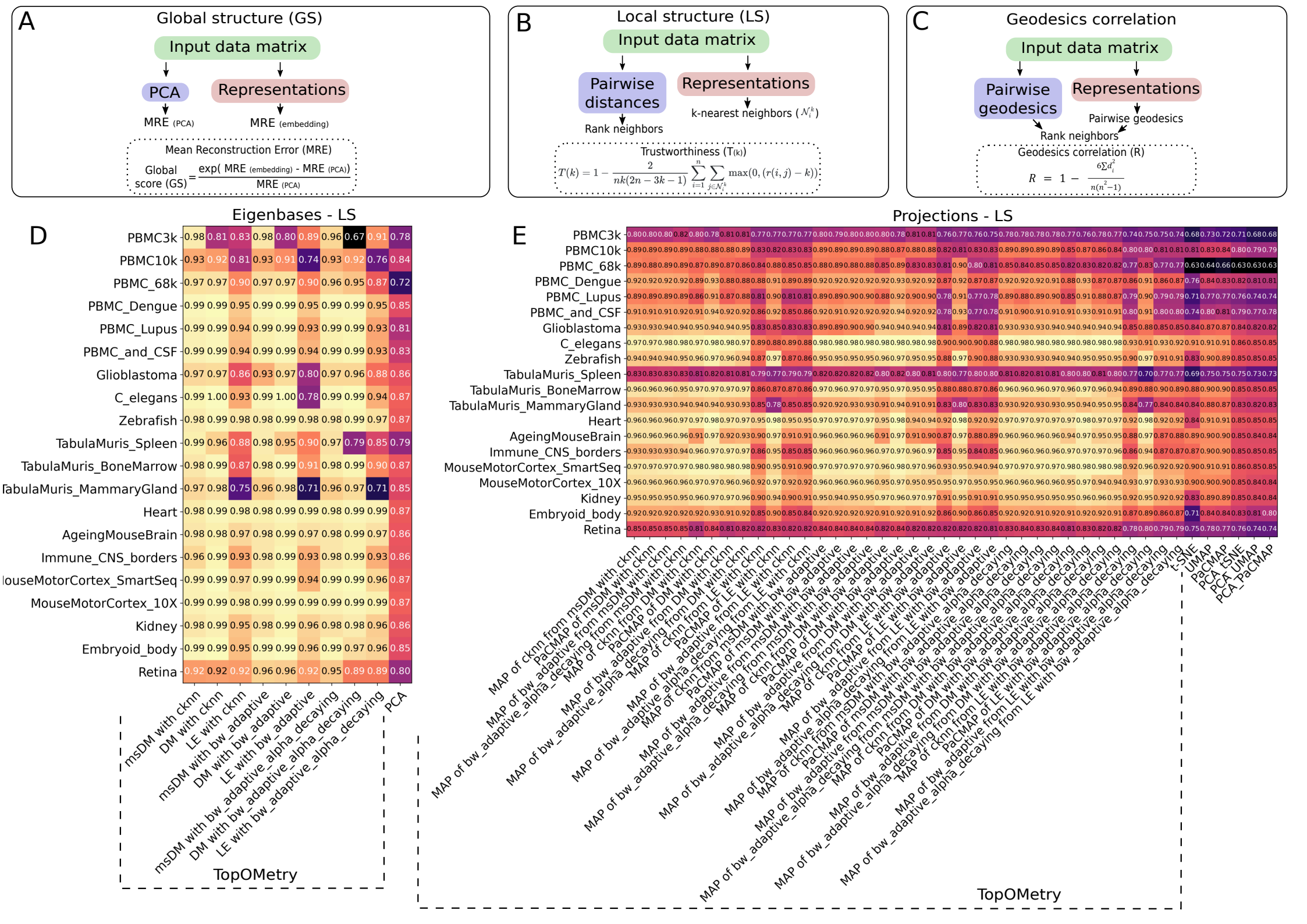

A
Global structure (GS)
Input data matrix
PCA
Representations
MRE (PCA)
MRE (embedding)
Mean Reconstruction Error (MRE)
Global score (GS) = exp(MRE (embedding) - MRE (PCA)) / MRE (PCA)
B
Local structure (LS)
Input data matrix
Pairwise distances
Representations
k-nearest neighbors ($\mathcal{N}_i^k$)
Rank neighbors
Trustworthiness (T(k))
$T(k) = 1 - \frac{2}{nk(2n-3k-1)} \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i^k} \max(0, (r(i,j) - k))$
C
Geodesics correlation
Input data matrix
Pairwise geodesics
Representations
Pairwise geodesics
Rank neighbors
Geodesics correlation (R)
$R = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$
D
Eigenbases - LS
PBMC3k
PBMC10k
PBMC_68k
PBMC_Dengue
PBMC_Lupus
PBMC_and_CSF
Glioblastoma
C_elegans
Zebrafish
TabulaMuris_Spleen
TabulaMuris_BoneMarrow
TabulaMuris_MammaryGland
Heart
AgeingMouseBrain
Immune_CNS_borders
MouseMotorCortex_SmartSeq
MouseMotorCortex_10X
Kidney
Embryoid_body
Retina
msDM with cknn
DM with cknn
LE with cknn
msDM with bw_adaptive
DM with bw_adaptive
LE with bw_adaptive
msDM with bw_adaptive_alpha_decaying
DM with bw_adaptive_alpha_decaying
LE with bw_adaptive_alpha_decaying
PCA
TopOMetry
E
Projections - LS
MAP of cknn from msDM with cknn
PaCMAP of msDM with cknn
MAP of bw_adaptive from msDM with cknn
MAP of bw_adaptive_alpha_decaying from msDM with cknn
MAP of cknn from DM with cknn
PaCMAP of DM with cknn
MAP of bw_adaptive from DM with cknn
MAP of bw_adaptive_alpha_decaying from DM with cknn
MAP of cknn from LE with cknn
PaCMAP of LE with cknn
MAP of bw_adaptive from LE with cknn
MAP of bw_adaptive_alpha_decaying from LE with cknn
t-SNE
UMAP
PaCMAP
PCA_tSNE
PCA_UMAP
PCA_PaCMAP
TopOMetry

Figure 2. TopOMetry eigenbases and projections preserve more original structure than the current standards. (A) Schematic overview of the assessment of global structure in TopOMetry - the global score is calculated by taking the exponential of the difference between an embedding MRE and PCA's MRE, divided by PCA's MRE. (B) Schematic overview of the assessment of local structure in TopOMetry with the trustworthiness score - the score penalizes embeddings in which cells that are neighbors in the low- but not in the high-dimensional space. (C) Schematic overview of the assessment of distances preservation in manifolds - the pairwise geodesic distances in the high- and low-dimensional spaces are computed, and Spearman R correlation between the rank of neighbors for each cell is obtained as a score. (D) Annotated heatmap of local scores for TopOMetry's eigenbases and PCA for 20 single-cell datasets (higher is better). (E) Annotated heatmap of local scores for projections learned using expression data, the first 100 principal components, and TopOMetry's eigenbases or topological graphs for 20 single-cell datasets (higher is better).

Estimating and visualizing intrinsic dimensionalities with TopOMetry

Intrinsic dimensionality (i.d.) can be loosely defined as the minimum number of parameters needed to describe a multiparameter (high-dimensional) system accurately[44]. Many definitions of i.d. have been proposed, including local and global i.d.[39,45], and its accurate estimation remains an active field of research. Estimating the global i.d. of a given dataset is intrinsically related to estimating the dimensionality of its underlying manifold[46], which is crucial when performing dimensional reduction (i.e., to estimate how many components to use in matrix decomposition). Furthermore, local i.d. is used in many signal processing applications as a separate variable to describe the data and provide meaningful information regarding the manifold structure. Noticeably, the similarity kernels employed in TopOMetry are related to the FSA (Farahmand-Szepesvári-Audibert) method, a recently adapted estimator of intrinsic dimensionality[39], as they harness the ratio between the distances to median nearest-neighbors (i.e., $k/2$) and to k-nearest-neighbors as a measure of local sampling density. To our knowledge, no tool has ever been proposed to evaluate the i.d. of sc-data specifically, despite its centrality for learning accurate representations. Likewise, we know of no other similarity kernels straightforwardly related to local i.d. estimates.

TopOMetry uses three existing strategies to estimate the i.d. of the data's underlying manifold: i) analyzing the eigenspectrum of its Laplacian-type operators; ii) inspecting the eigenfunctions to evaluate the separation of known classes; and iii) using two independent state-of-the-art algorithms for i.d. estimation, FSA[39], and MLE[47]. The two first strategies estimate the global i.d., and the latter estimates the local i.d. and weights them to estimate the global i.d. We first demonstrate these strategies using the MNIST dataset, comprising images of handwritten digits. In this dataset, TopOMetry found an eigengap at 19 dimensions using the eigenvalues' first derivatives (Figure 3A). The corresponding eigenfunctions separate the ground-truth classes well, in contrast to the principal components (Figure 3B). Local i.d. estimates with both FSA and MLE presented nearly identical distributions across a range of possible kNN (Figure 3C), and the global estimates obtained with these algorithms are similar to the global i.d. predicted by the eigengap (FSA: 17-18, MLE: 12-14). Furthermore, the estimates obtained with these algorithms are highly correlated (Figure 3D), reinforcing their reliability. To interrogate whether i.d. estimates translated to actual insights regarding the data, we used a subsample of the dataset and visualized the digits with the highest and lowest i.d. Interestingly, the 100 digits with the

highest local i.d. were predominantly 3's, 5's, and 8's written with varying calligraphy, indicating an association between local i.d. and intra-cluster heterogeneity (Figure 3E, left). Conversely, the 100 digits with the lowest local i.d. were predominantly 2's, 6's, and 1's written with consistent calligraphy, further suggesting this association (Figure 3E, right).

We next examined the PBMC 3k dataset following the same approach to test whether these results translated to real-world sc-data. This dataset comprises ~2,700 peripheral blood mononuclear cells (PBMC) from a healthy human donor. Once again, the global estimate obtained with the eigengap method (72) (Figure 3F) was reasonably similar to those obtained with MLE (122-59) and FSA (122-74) (Figure 3G). Similar to the MNIST dataset, the estimates of local i.d. obtained with MLE and FSA were also highly correlated (Figure 3H) but strikingly varied across different regions of the represented manifold (Figure 3I), which is inhabited by discrete cell types (Figure 3J, K). In this dataset, we found that megakaryocytes, natural killer (NK) T cells, and dendritic cells (DC) presented the highest local i.d. However, different from the MNIST dataset, in which the images can be directly visualized, the biological implications of higher local i.d. for these cell types in the PBMC 3k dataset remain unclear, requiring further investigative strategies. Nevertheless, estimating the local i.d. during exploratory sc-data analysis mains remarkably advantageous, as it: i) allows discovering that distinct cell types inhabiting discrete regions of the phenotypic manifold present strikingly different dimensionalities; ii) indicates that these cell types will present different distortions in representations learned with dimensional reduction; and iii) estimates the number of latent dimensions needed to fully describe the data, which should be between the average and the upper limit of the local i.d. distribution. Defining such a number is a particularly critical issue in sc-data analysis, as most existing approaches try to denoise the data into an arbitrary number of latent dimensions prior to downstream tasks. Within TopOMetry, such estimates guide the number of eigencomponents users should compute for each specific dataset. We expect that estimating the i.d. of sc-data will greatly assist practitioners during its analysis, allowing more accurate representations and providing additional insights regarding its geometry across various methods and tasks.

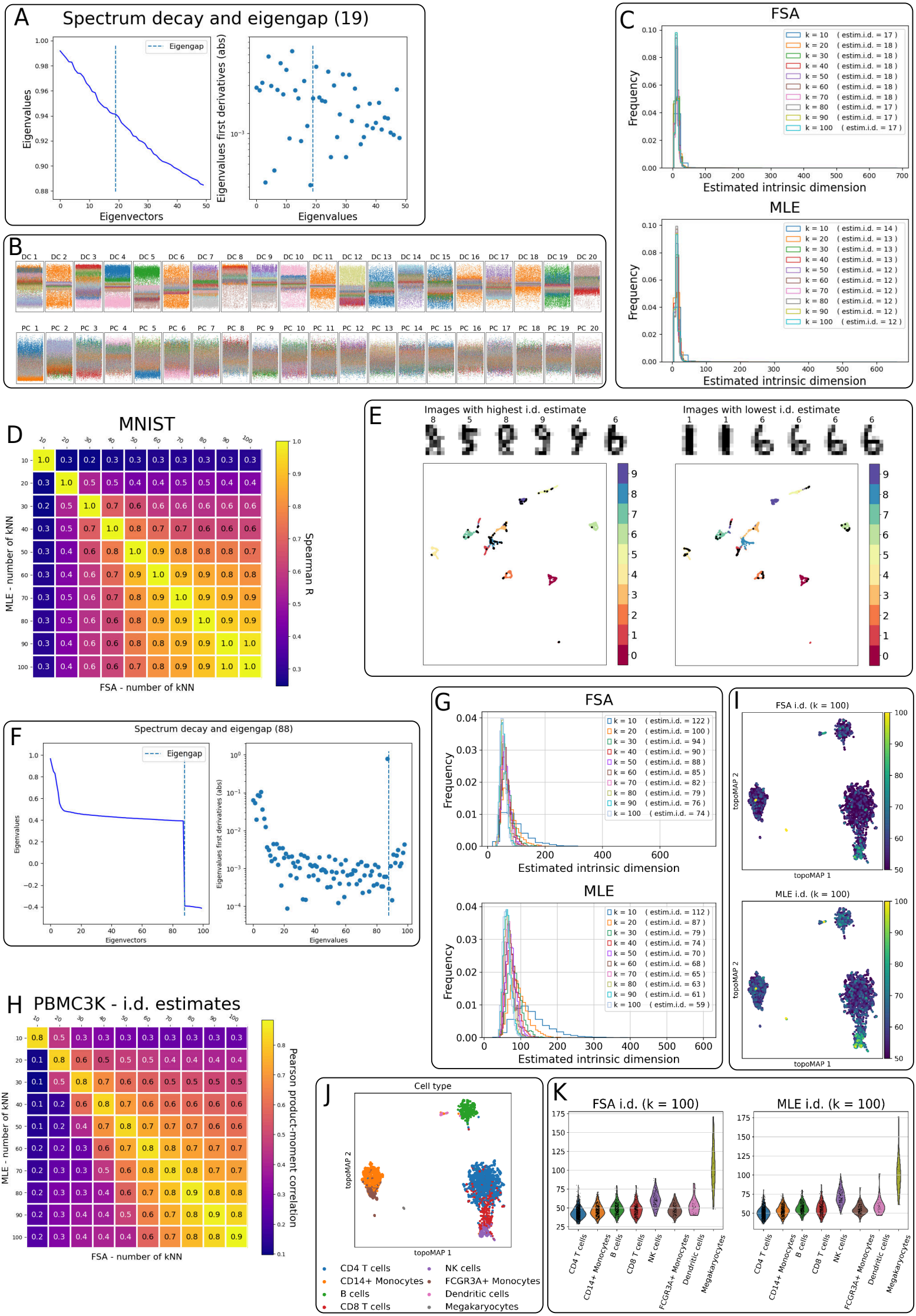

A
Spectrum decay and eigengap (19)
Eigengap
Eigenvalues
Eigenvectors
Eigenvalues first derivatives (abs)
Eigenvalues
B
DC 1 DC 2 DC 3 DC 4 DC 5 DC 6 DC 7 DC 8 DC 9 DC 10 DC 11 DC 12 DC 13 DC 14 DC 15 DC 16 DC 17 DC 18 DC 19 DC 20
PC 1 PC 2 PC 3 PC 4 PC 5 PC 6 PC 7 PC 8 PC 9 PC 10 PC 11 PC 12 PC 13 PC 14 PC 15 PC 16 PC 17 PC 18 PC 19 PC 20
C
FSA
Frequency
Estimated intrinsic dimension
k = 10 (estim.i.d. = 17)
k = 20 (estim.i.d. = 18)
k = 30 (estim.i.d. = 18)
k = 40 (estim.i.d. = 18)
k = 50 (estim.i.d. = 18)
k = 60 (estim.i.d. = 18)
k = 70 (estim.i.d. = 18)
k = 80 (estim.i.d. = 17)
k = 90 (estim.i.d. = 17)
k = 100 (estim.i.d. = 17)
MLE
Frequency
Estimated intrinsic dimension
k = 10 (estim.i.d. = 14)
k = 20 (estim.i.d. = 13)
k = 30 (estim.i.d. = 13)
k = 40 (estim.i.d. = 13)
k = 50 (estim.i.d. = 12)
k = 60 (estim.i.d. = 12)
k = 70 (estim.i.d. = 12)
k = 80 (estim.i.d. = 12)
k = 90 (estim.i.d. = 12)
k = 100 (estim.i.d. = 12)
D
MNIST
MLE - number of kNN
FSA - number of kNN
Spearman R
E
Images with highest i.d. estimate
Images with lowest i.d. estimate
F
Spectrum decay and eigengap (88)
Eigengap
Eigenvalues
Eigenvectors
Eigenvalues first derivatives (abs)
Eigenvalues
G
FSA
Frequency
Estimated intrinsic dimension
k = 10 (estim.i.d. = 122)
k = 20 (estim.i.d. = 100)
k = 30 (estim.i.d. = 94)
k = 40 (estim.i.d. = 90)
k = 50 (estim.i.d. = 88)
k = 60 (estim.i.d. = 85)
k = 70 (estim.i.d. = 82)
k = 80 (estim.i.d. = 79)
k = 90 (estim.i.d. = 76)
k = 100 (estim.i.d. = 74)
MLE
Frequency
Estimated intrinsic dimension
k = 10 (estim.i.d. = 112)
k = 20 (estim.i.d. = 87)
k = 30 (estim.i.d. = 79)
k = 40 (estim.i.d. = 74)
k = 50 (estim.i.d. = 70)
k = 60 (estim.i.d. = 68)
k = 70 (estim.i.d. = 65)
k = 80 (estim.i.d. = 63)
k = 90 (estim.i.d. = 61)
k = 100 (estim.i.d. = 59)
I
FSA i.d. (k = 100)
topoMAP 1
topoMAP 2
MLE i.d. (k = 100)
topoMAP 1
topoMAP 2
H
PBMC3K - i.d. estimates
MLE - number of kNN
FSA - number of kNN
Pearson product-moment correlation
J
Cell type
topoMAP 1
topoMAP 2
CD4 T cells
CD14+ Monocytes
B cells
CD8 T cells
NK cells
FCGR3A+ Monocytes
Dendritic cells
Megakaryocytes
K
FSA i.d. (k = 100)
MLE i.d. (k = 100)
CD4 T cells
CD14+ Monocytes
B cells
CD8 T cells
NK cells
FCGR3A+ Monocytes
Dendritic cells
Megakaryocytes

Figure 3. Estimating intrinsic dimensionalities in toy and single-cell data with TopOMetry. (A) Eigenspectrum of the eigenvalues for each diffusion component learned for the MNIST dataset of handwritten digits with TopOMetry (left) and the absolute value of their first derivatives (right). (B) Distribution of MNIST samples (individual images) across different classes (digits) across diffusion components (top) and principal components (bottom). (C) Histograms of intrinsic dimensionality estimates for each sample in the MNIST dataset, with varying numbers of k-nearest-neighbors, with the FSA (top) and MLE (bottom) methods. (D) Heatmap of Spearman R correlation between FSA and MLE estimates of intrinsic dimensionalities of MNIST images for varying numbers of k-nearest-neighbors. (E) topoMAP projections of a subset of the MNIST handwritten dataset, colored by classes (numbers). The 100 images with the highest (left) or lowest (right) estimates of intrinsic dimensionality are colored black in the projections and shown on the top of each projection. (F) Eigenspectrum of the eigenvalues for each diffusion component learned for the *PBMC3k* dataset of peripheral blood mononuclear cells (10X Genomics) with TopOMetry (left) and the absolute value of their first derivatives (right). (G) Histograms of intrinsic dimensionality estimates for each cell in the *PBMC3k* dataset, with varying numbers of k-nearest-neighbors, with the FSA (top) and MLE (bottom) methods. (H) Heatmap of Spearman R correlation between FSA and MLE estimates of intrinsic dimensionalities of *PBMC3k* cells for varying numbers of k-nearest-neighbors. (I) topoMAP projections of the *PBMC3k* dataset, colored by the estimates obtained with FSA (top) and MLE (bottom) with 100 nearest-neighbors. (J) topoMAP projection of the *PBMC3k* dataset, colored by annotated cell types. (K) Violin plots of estimates of intrinsic dimensionalities of *PBMC3k* cells with FSA (left) and MLE (right) with 100 nearest-neighbors.

TopOMetry allows precise manifold learning of small and large-scale datasets

We next aimed to explore how TopOMetry can be useful in inferring cellular development trajectories. For a first demonstration, we obtained scRNAseq data from the developing murine pancreas[48] popularly used in RNA velocity estimation (Figure 3A) and analyzed it with TopOMetry's overall top-performing model (Figure 3B, topoMAP for brevity). The representation learned with this model yielded finer-grained lineages when compared to the default PCA-based UMAP. When scoring cells by their predicted cell cycle phase, our model effectively mapped mitotic cells to a closed-loop structure (Figure 3C), which provides a better representation of the cell cycle than the homogenous cloud found with the default approach (Figure 3D). A detailed inspection revealed that the default approach places cells with high S and G2/M scores in an intermediate position of the differentiation lineage apart from the remaining cells undergoing the cell cycle (Figure 3D, marker). Conversely, the same cells are placed close to other cycling cells in the representation learned with TopOMetry (Figure 3C, marker), providing an optimal representation of the global structure of the underlying manifold. These results and the fact that the cells are indeed cycling are confirmed by visualizing RNA velocity vectors on the embeddings (Figure 4E-F). The superiority of our approach over the current standard is further reinforced by the fact that the RNA velocity vector field based on the default approach suggests that epsilon cells give rise to beta cells (Figure 3E). Lineage tracing studies have shown that this is the exception rather than the norm[49]: instead, delta cells represent a transient state that later gives rise to alpha cells, which is faithfully represented by the RNA velocity vector field based on the TopOMetry model (Figure 3F). Finally, it is also shown that the latent dimensions learned by TopOMetry (the multiscaled diffusion components) are associated with particular regions of the underlying manifold (Figure 4G) corresponding to specific cellular states. This effectively reconstructs the manifold's topology one region at a time (i.e., one identity per component). These components can also be used for downstream tasks, such as inferring changes in gene signatures along several developmental axes.

Next, we show how TopOMetry can be used for lineage inference in organism-wide cell atlases to learn detailed cellular hierarchies. For this, we harnessed 1.3 million high-quality pre-filtered single-cell transcriptomes from the mouse embryo during organogenesis[50]. Standard processing was performed, and gene expression data was then represented with PCA (Figure 3H) or UMAP on the top 300 PCs (Figure 3I and 3J - annotations are from the original study).

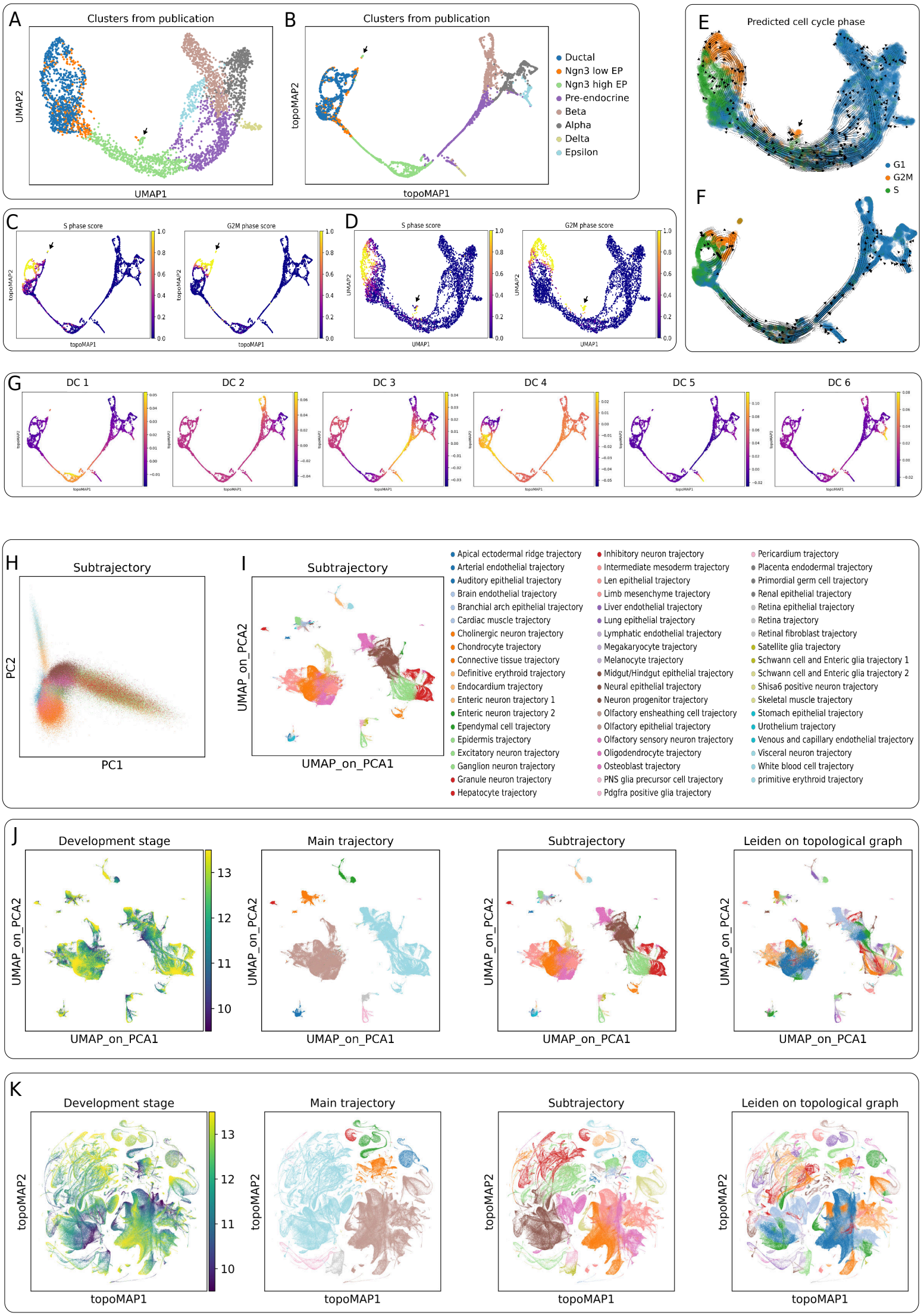

A
Clusters from publication
UMAP2
UMAP1
B
Clusters from publication
topoMAP2
topoMAP1
Ductal
Ngn3 low EP
Ngn3 high EP
Pre-endocrine
Beta
Alpha
Delta
Epsilon
E
Predicted cell cycle phase
G1
G2M
S
F
C
S phase score
G2M phase score
topoMAP2
topoMAP1
D
S phase score
G2M phase score
UMAP2
UMAP1
G
DC 1
DC 2
DC 3
DC 4
DC 5
DC 6
H
Subtrajectory
PC2
PC1
I
Subtrajectory
UMAP_on_PCA2
UMAP_on_PCA1
Apical ectodermal ridge trajectory
Arterial endothelial trajectory
Auditory epithelial trajectory
Brain endothelial trajectory
Branchial arch epithelial trajectory
Cardiac muscle trajectory
Cholinergic neuron trajectory
Chondrocyte trajectory
Connective tissue trajectory
Definitive erythroid trajectory
Endocardium trajectory
Enteric neuron trajectory 1
Enteric neuron trajectory 2
Ependymal cell trajectory
Epidermis trajectory
Excitatory neuron trajectory
Ganglion neuron trajectory
Granule neuron trajectory
Hepatocyte trajectory
Inhibitory neuron trajectory
Intermediate mesoderm trajectory
Len epithelial trajectory
Limb mesenchyme trajectory
Liver endothelial trajectory
Lung epithelial trajectory
Lymphatic endothelial trajectory
Megakaryocyte trajectory
Melanocyte trajectory
Midgut/Hindgut epithelial trajectory
Neural epithelial trajectory
Neuron progenitor trajectory
Olfactory ensheathing cell trajectory
Olfactory epithelial trajectory
Olfactory sensory neuron trajectory
Oligodendrocyte trajectory
Osteoblast trajectory
PNS glia precursor cell trajectory
Pdgfra positive glia trajectory
Pericardium trajectory
Placenta endodermal trajectory
Primordial germ cell trajectory
Renal epithelial trajectory
Retina epithelial trajectory
Retina trajectory
Retinal fibroblast trajectory
Satellite glia trajectory
Schwann cell and Enteric glia trajectory 1
Schwann cell and Enteric glia trajectory 2
Shisa6 positive neuron trajectory
Skeletal muscle trajectory
Stomach epithelial trajectory
Urothelium trajectory
Venous and capillary endothelial trajectory
Visceral neuron trajectory
White blood cell trajectory
primitive erythroid trajectory
J
Development stage
Main trajectory
Subtrajectory
Leiden on topological graph
UMAP_on_PCA2
UMAP_on_PCA1
13
12
11
10
K
Development stage
Main trajectory
Subtrajectory
Leiden on topological graph
topoMAP2
topoMAP1
13
12
11
10

Figure 4. Inferring cellular lineages in small and large datasets with TopOMetry. (A) UMAP and (B) topoMAP projections of the *Pancreas* dataset showing cellular developmental trajectories in the murine pancreas, colored by annotated cell type. (C) topoMAP and (D) UMAP projections of the same data, colored by inferred scores of different cell-cycle phases. The same projections are also colored by the predicted cell-cycle phase of each cell in (E) and (F), and by the first five diffusion components in (G). Arrows indicate a population of cells that are classified as mitotic but were misplaced in the middle of the differentiation trajectory by the default PCA-based UMAP projection. (H) PCA and (I) PCA-based UMAP projections of the *Mouse Organogenesis Cell Atlas* (MOCA) showing cellular developmental trajectories of whole mouse embryos, colored by annotated subtrajectories from the original study. (J) PCA-based UMAP and (K) topoMAP projections of the MOCA dataset, colored by development stage, annotated trajectories and subtrajectories, and TopOMetry clustering results using the standard Leiden community-detection algorithm. Note how the topoMAP projection and the clustering results from TopOMetry uncover dozens of neuronal subtrajectories that match the developmental stage from which cells were sampled.

TopOMetry analysis was then performed using its default model (Figure 3K, topoMAP for brevity), and the Leiden community detection algorithm was employed on its diffusion potential graph. Both approaches successfully mapped major cell types as distinct regions of the manifold without apparent connections between them. However, TopOMetry revealed considerably finer-grained cellular lineages (Figure 3K) than those represented with UMAP on PCA (Figure 3I). In total, we found approximately 380 subpopulations, while only 56 were originally described (and labeled as subtrajectories). Most of these populations were neuronal populations, which is consistent with the expected diversity of the central system. Some of the populations found when clustering with TopOMetry are also suggested by the UMAP on PCA (Figure 3I) representation, albeit with less clearly defined trajectories. Hence, it is shown that TopOMetry excels at faithfully representing vast datasets comprising hundreds of coexisting lineages, being ideal for representing whole-organism data and performing lineage inference due to its intrinsic relation with the Waddington epigenetic landscape.

TopOMetry augments representations with the Riemannian metric to visualize distortions

A central issue when learning low-dimensional representations from sc-data is the introduction of distortions. Distortion is inevitably introduced when a high-dimensional manifold is embedded into a lower-dimensional space. As an illustrative example, consider the Earth and the planar projections used for making two-dimensional maps (e.g., the Mercator projection): in these maps, there is always some distortion, which can involve distances between points or shapes, angles, and areas within the projection. However, while Earth's true geometry can be physically visualized from space, the epigenetic manifolds[20] of phenotypic identities from which sc-data is sampled are unknown and cannot be visually assessed, which makes it extremely challenging to qualitatively evaluate how the low-dimensional mapping distorted them. The theoretical model of cellular diversity as Riemannian manifolds proposed here allows a direct connection with a unique method for evaluating these distortions: expressing the Riemannian metric of the original manifold[36] in any given 2-D mapping. The Riemannian metric (RM) is expressed in 2-D coordinates so that it makes the represented distances and angles approximately equal to their original manifold counterparts. In practice, the RM can be visualized as ellipses whose axes indicate the magnitude and directions in which the manifold was distorted. Visually, these

ellipses will have eccentricity equal to one if the distortion is uniform in all directions, rendering a circle with a radius approaching zero in the special case of no distortion (i.e., the representation of a given point fully preserves its geometric properties on the manifold). Such an approach allows users to consider how the representation distorted different regions of the manifold and to qualitatively compare them for hypothesis generation, effectively working as a "distortion ruler."

To illustrate these concepts, we used a toy dataset with three randomly distributed clusters with distinct variances (Figure 4A) and learned their representations with PCA, DM (within TopOMetry), UMAP, and a TopOMetry model (MAP on the diffusion potential of a diffusion map eigenbasis). The RM was then estimated for each embedding and visualized using ellipses. These ellipses present high eccentricity when using the first two principal components, indicating distortion towards the periphery of the embedding and reasonably high eccentricities across all three clusters when using UMAP. In contrast, when using the first two diffusion components learned with TopOMetry, the distortion was negligible for one of the clusters and happened along with the components themselves for the other two. This was expected, as each cluster is encoded by a separate diffusion component in this scenario. When using the TopOMetry MAP on DM potential model, we found a distortion pattern similar to that observed within UMAP.

Next, to demonstrate its usability in sc-data analysis, we estimated the RM for representations of the classic PBMC 3k dataset learned with PCA, UMAP, UMAP on PCA, and a TopOMetry model (Figure 4B). As represented by the ellipses, each representation differently distorts specific regions of the manifold. To allow for a more uniform comparison, we color-coded the eccentricity of ellipses corresponding to each cell across methods and visualized them within the TopOMetry layout (Figure 4C) and with violin plots (Figure 4D). As observed, the populations associated with the greatest distortion were reasonably consistent across representations; CD4 T cells, CD14+ monocytes, B lymphocytes, and NK cells. These findings suggest that all representations will preferentially misrepresent some cell types, often at different fashions when comparing different methods and that this could be related to the intrinsic geometry of the underlying manifold. Albeit illustrative, this example is limited by the small number of cells available. As CD4+ T cells were more favorably represented within the TopOMetry model and are known for their extreme diversity, we asked whether analyzing a larger amount of samples (cells) could harbor additional insights into their biology.

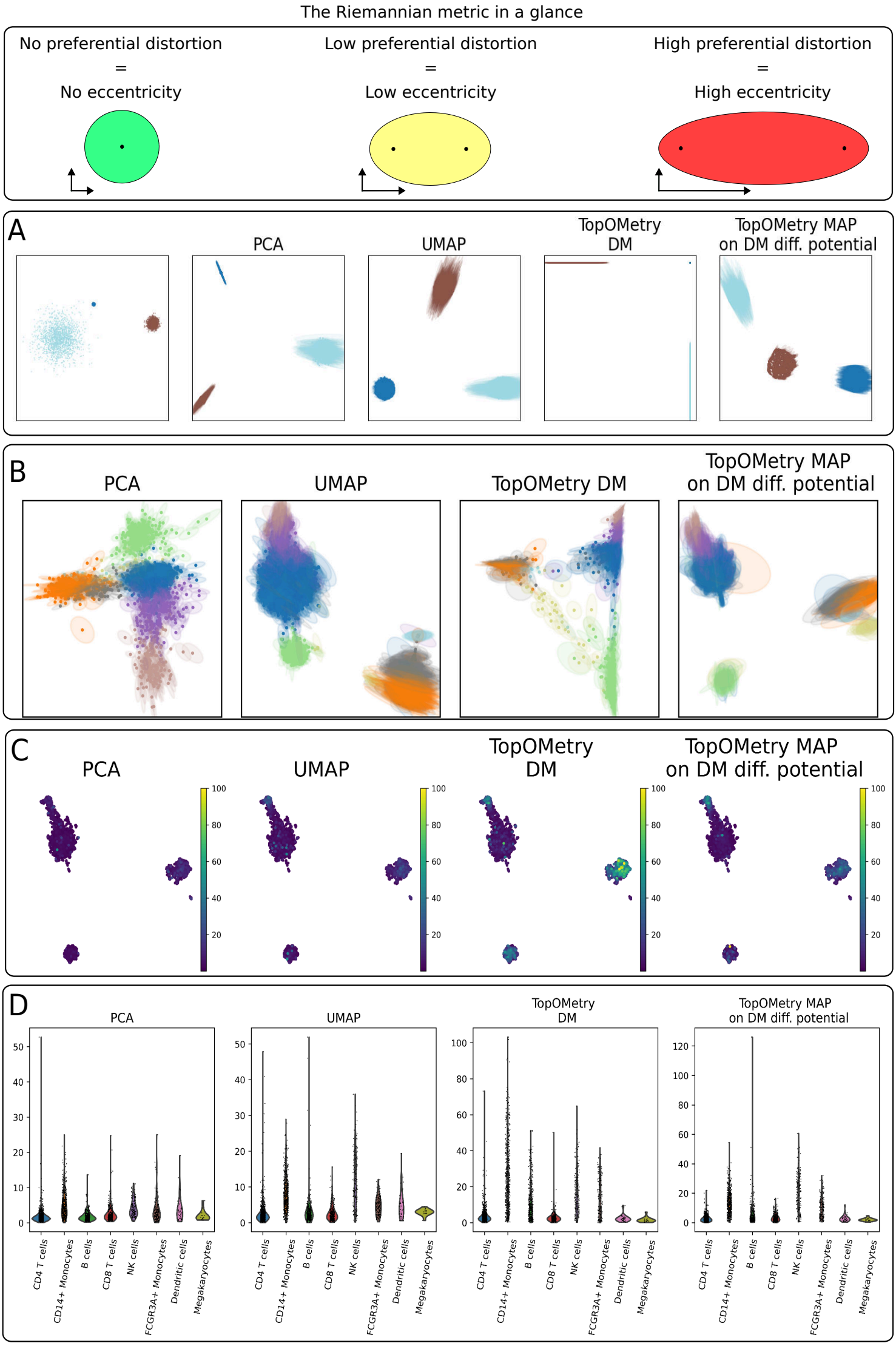

The Riemannian metric in a glance
No preferential distortion
=
No eccentricity
Low preferential distortion
=
Low eccentricity
High preferential distortion
=
High eccentricity
A
PCA
UMAP
TopOMetry DM
TopOMetry MAP on DM diff. potential
B
PCA
UMAP
TopOMetry DM
TopOMetry MAP on DM diff. potential
C
PCA
UMAP
TopOMetry DM
TopOMetry MAP on DM diff. potential
D
PCA
UMAP
TopOMetry DM
TopOMetry MAP on DM diff. potential
CD4 T cells
CD14+ Monocytes
B cells
CD8 T cells
NK cells
FCGR3A+ Monocytes
Dendritic cells
Megakaryocytes

Figure 5. Evaluating distortions in two-dimensional projections with the Riemannian metric. The Riemannian metric can be used to estimate distortions in two-dimensional visualizations and can be represented by ellipses. If no distortion is present in any preferential direction, the ellipses will have zero eccentricities and correspond to circles. If distortion is present in a preferential direction, the ellipse will be aligned in that direction, and its eccentricity indicates the degree of distortion. (A) Synthetic data comprised of three normally distributed clusters with varying degrees of variance (leftmost), and visualization of the Riemannian metric on two-dimensional visualizations of the data obtained with PCA, UMAP, DM (within TopOMetry), and topoMAP (TopOMetry MAP on DM diffusion potential). (B) Visualization of the Riemannian metric on two-dimensional visualizations of the *PBMC3k* dataset obtained with PCA, UMAP, DM (within TopOMetry), and topoMAP (TopOMetry MAP on DM diffusion potential). (C) topoMAP projections of the *PBMC3k* dataset, colored by the eccentricities of ellipses representing the Riemannian metric across the aforementioned visualizations for each cell. (D) Violin plots of the eccentricities of ellipses representing the Riemannian metric on the aforementioned visualizations of the *PBMC3k* dataset, for each cell type annotation.

TopOMetry reveals novel transcriptionally defined T cell populations

To further investigate CD4+ T cells and distortions of the underlying manifold, we harnessed a public dataset comprising approximately 68,000 PBMCs from a healthy donor (10X Genomics, *PBMC68K* dataset). Representations were obtained with UMAP (using either the top 50 or 300 PCs or the highly-variable genes) and TopOMetry models. Cells were then clustered by the Leiden community detection algorithm using weighted neighborhood graphs learned from; i) the top 50 PCs; ii) the top 300 PCs; and iii) TopOMetry's diffusion potential. We then used CellTypist[51] to predict major cell types and subtypes. Results obtained using 50 PCs were essentially identical to previous reports[52] (Figure 6A), rendering 24 clusters, 16 being T cells, which have been classically described with poor marker genes across different single-cell studies[51,53–58] (Figure 6B). However, these findings were inconsistent with what we observed when using 300 PCs, particularly for CD4+ T cells (Supplementary Figures 4A-B), further suggesting that these putative clusters could be linear distortions. A similar discrepancy was found when using stand-alone UMAP, which suggested the existence of additional T cell clusters (Supplementary Figure 4C). Strikingly, the results obtained with TopOMetry suggest the existence of 123 clusters in this dataset (Figure 6C), 116 of them being of T cells (Figure 6C, Supplementary Figure 4D), and each presenting specific marker genes (Figure 6D), in sharp contrast to the classic PCA-based approach. The agreement between clustering results obtained with TopOMetry models or with the HVG data and the disparity of these results in confrontation with those found using the PCA-based workflow was quantitatively confirmed using the adjusted mutual information (AMI) score[59] (Supplementary Figure 4E).

Regarding the identity of the additional clusters, the majority of the discrepant ones were annotated by CellTypist as T central memory (T_{CM}) cells (Supplementary Figure 4F), although additional clusters corresponding to NK (natural killer) and MAIT (mucosal-associated invariant T) cells were also identified. T regulatory cells were the only T cell subpopulation found across all approaches (Supplementary Figures 4G-I), in contrast to four other putative subpopulations we identified (marked by expression of *MDN1, C11orf68, IL11RA,* and *IL21R*), which were randomly scattered in the PCA-based UMAP embedding, marginally detectable using stand-alone UMAP and clearly identified using TopOMetry (Supplementary Figures 4G-I) .

To elucidate whether these findings were reproducible across different sets of data and not restricted to this particular example, we harnessed three additional independent scRNAseq datasets of human PBMCs, including healthy controls and patients suffering from dengue virus (DenV, *Dengue* dataset) infection, multiple sclerosis (MS, *MS_CSF* dataset) and systemic lupus erythematosus (SLE, *Lupus* dataset). The analyses performed on the *PBMC68K* dataset were then applied to these three datasets with identical parameters. We found strikingly consistent results across all three datasets, in which clustering and UMAP results obtained with 50 PCs were inconsistent with those obtained using the original data (HVGs) or TopOMetry (Supplementary Figures 5A-C). Likewise, specific marker genes were found for the clusters identified with TopOMetry, in contrast to the poorly-defined signatures observed for PCA-derived clusters (Supplementary Figures 5D-F). On average, 18 clusters of T cells were found using 50PC, in contrast to a mean of 73 identified using the HVGs and of 137 identified using TopOMetry (Figure 6E). The average number of cells in these clusters was expressively smaller using HVGs (608) and TopOMetry (320) in comparison to using 50 PCs (2489). In addition, the adjusted mutual information score for different clustering strategies followed a similar pattern to that observed with the *PBMC68K* dataset, in which clustering results using the expression data are more similar to those found with TopOMetry than with the standard PCA-based workflow (Supplementary Figure 5G). Unexpectedly, different TopOMetry models presented a high AMI score, demonstrating their successful convergence towards the LBO to describe the underlying phenotypic manifold.

To explore whether or not the identified T cell subpopulations were functionally distinct from a T cell receptor perspective, we harnessed an additional dataset - the T cell compartment of the Tissue Immune Cell Atlas[51], comprising scRNA-seq and scVDJ-seq from T cells obtained from multiple human donors and tissues, including blood, that were jointly analyzed using the standard PCA-based workflow (Figure 6F). When analyzing this data with TopOMetry, we found a mismatch between the clusters identified with it and the clusters described in the publication, particularly around CD4+ cells (Figure 6G, Supplementary Figure 6A and 6B), in a similar fashion to the previous datasets. T cell clonotypes were identified based on amino-acid sequence similarity using ScirPy[60], and were found to match the clusters found with TopOMetry (Supplementary Figure 6C) better than the originally described clusters (Supplementary Figure 6D). When analyzing TCR clonal expansion, we found that subpopulations with larger expansion corresponded to NK and CD8+ cells, as expected and previously described[51] (Figure

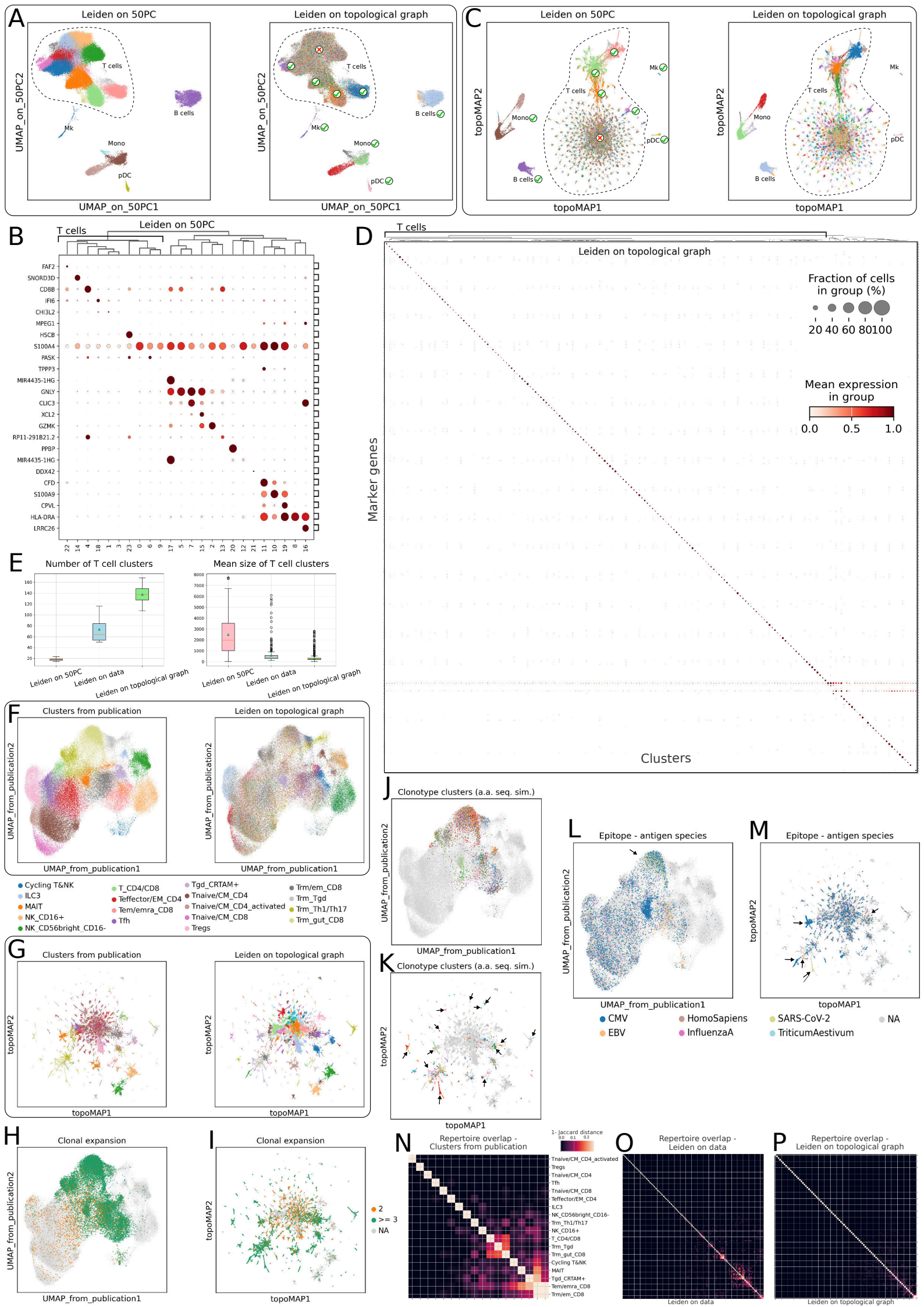

A
Leiden on 50PC
Leiden on topological graph
UMAP_on_50PC2
UMAP_on_50PC1
T cells
B cells
Mk
Mono
pDC
C
topoMAP2
topoMAP1
B
T cells
Leiden on 50PC
FAF2
SNORD3D
CD8B
IFI6
CHI3L2
MPEG1
HSCB
S100A4
PASK
TPPP3
MIR4435-1HG
GNLY
CLIC3
XCL2
GZMK
RP11-291B21.2
PPBP
MIR4435-1HG
DDX42
CFD
S100A9
CPVL
HLA-DRA
LRRC26
22 14 4 18 1 3 23 0 6 9 17 5 7 15 2 13 20 12 21 11 10 19 8 16
D
T cells
Leiden on topological graph
Fraction of cells in group (%)
20 40 60 80 100
Mean expression in group
0.0 0.5 1.0
Marker genes
Clusters
E
Number of T cell clusters
Mean size of T cell clusters
Leiden on 50PC
Leiden on data
Leiden on topological graph
F
Clusters from publication
Leiden on topological graph
UMAP_from_publication2
UMAP_from_publication1
Cycling T&NK
ILC3
MAIT
NK_CD16+
NK_CD56bright_CD16-
T_CD4/CD8
Teffector/EM_CD4
Tem/emra_CD8
Tfh
Tgd_CRTAM+
Tnaive/CM_CD4
Tnaive/CM_CD4_activated
Tnaive/CM_CD8
Tregs
Trm/em_CD8
Trm_Tgd
Trm_Th1/Th17
Trm_gut_CD8
J
Clonotype clusters (a.a. seq. sim.)
L
Epitope - antigen species
M
Epitope - antigen species
topoMAP2
topoMAP1
CMV
EBV
HomoSapiens
InfluenzaA
SARS-CoV-2
TriticumAestivum
NA
G
Clusters from publication
Leiden on topological graph
K
Clonotype clusters (a.a. seq. sim.)
H
Clonal expansion
I
Clonal expansion
2
>= 3
NA
N
Repertoire overlap - Clusters from publication
1- Jaccard distance
O
Repertoire overlap - Leiden on data
P
Repertoire overlap - Leiden on topological graph
Leiden on data
Leiden on topological graph

Figure 6. Uncovering the transcriptional diversity of T cells with TopOMetry. (A) PCA-based UMAP projections of the *PBMC68k* dataset, comprising approximately 68,000 peripheral blood mononuclear cells from a healthy donor (10X Genomics), colored by clustering results obtained with the standard PCA-based approach (left) or with the diffusion potential of TopOMetry msDM eigenbasis (right). The main cell types and the populations for which the two approaches yield similar clustering results are indicated in the figure. Note that the two approaches only reach disagreeing results for T cells. (B) Dotplot of the top marker gene for each cluster found with the standard PCA-based approach, ranked by logistic regression. Note how clusters corresponding to T cells present poor marker genes. (C) topoMAP projection of the *PBMC68k* dataset, colored by clustering results obtained with the standard PCA-based approach (left) or with the diffusion potential of TopOMetry msDM eigenbasis (right). Note that the clusters of T cells found with the former are randomly spread across the dozens of clusters found with the latter. (D) Dotplot of the top marker gene for each cluster found when clustering with TopOMetry, ranked by logistic regression with identical parameters. Note how clusters corresponding to T cells present highly-specific marker genes. (E) Boxplots of the number of T cell clusters (left) found with different clustering strategies (standard PCA-based, using expression data and with TopOMetry) and their mean size (right), across four datasets of peripheral blood mononuclear cells from human donors: *PBMC68k, Dengue*, *Lupus,* and *MS_CSF*. (F) PCA-based UMAP and (G) topoMAP projections of lymphoid cells from the *Tissue Immune Cell Atlas* (TICA) dataset, comprised of lymphoid cells across different organs and donors, colored by annotations from the original study (left) and clustering results obtained with TopOMetry (right). Note that the two clustering results reach disagreeing results mostly for CD4+ cell clusters. (H) PCA-based UMAP and (I) topoMAP projections of the TICA dataset, colored by clonal expansion thresholds. (J) PCA-based UMAP and (K) topoMAP projections of the TICA dataset, colored by the 30 clonotypes with the highest amount of cells. The arrows indicate the precise match of these clonotypes to the manifold structure uncovered by TopOMetry. (L) PCA-based UMAP and (M) topoMAP projections of the TICA dataset, colored by the epitopes recognized by each T cell. The arrows indicate populations recognizing specific epitopes that precisely match the manifold structure uncovered by TopOMetry. (N) Heatmap of T cell repertoire overlap between the clusters found in the original study, (O) using expression data and (P) TopOMetry. Note how the original clusters present high repertoire overlap, which is less evident in clusters found using expression data and nearly absent from clusters found with TopOMetry.

6H). However, populations with modest expansion corresponded to the CD4+ T cell clusters identified with TopOMetry (Figure 6H), suggesting these clusters correspond to specific clonal identities. To further explore this hypothesis, the 30 clonotypes with the most cells were visualized on the original UMAP embedding (Figure 6J), on which they are sparsely and randomly distributed, and on the TopOMetry embedding (topoMAP, Figure 6K), on which they are remarkably localized alongside the clusters found with TopOMetry. Consistent with these findings, the clonotype modularity (how densely connected the mRNA-derived neighborhood graph underlying the cells in a given clonotype is) was found to be higher when using TopOMetry in comparison to the standard approach (Supplementary Figure 6E and 6F). Next, we cross-referenced the TCR structure to VDJdb, a public database of antigen epitopes[60,61], and identified two transcriptionally defined clusters that present TCR that bind to SARS-CoV-2 and EBV epitopes (Figures 6L and 6M); these populations are well-defined in the TopOMetry analyses but are unidentifiable in the original UMAP. These results suggest that the clusters of T cells identified by TopOMetry correspond to specific clonal populations. This hypothesis is confirmed by analysis of TCR repertoire overlap across clustering results - while the clusters defined in the publication present a high repertoire overlap (Figure 6N), those identified using HVGs are expressively more well-separated (Figure 6O), and those obtained with TopOMetry present no overlap at all (Figure 6P). Indeed, we found that cells belonging to specific clonotypes defined by amino-acid sequence similarity also expressed the transcripts for the TCR chains identified by VDJ-seq (Supplementary Figures 6G-J). These results confirm that TopOMetry is able to identify clonal populations from mRNA expression alone and strongly suggest that the T cell clusters found in the other analyzed datasets (for which VDJ data is not available) also correspond to specific clones.

Throughout the analyzed datasets containing T cells, it was noticeable that the major cell populations (CD4+ and CD8+ T, NK, and B cells, monocytes, and megakaryocytes) were similarly represented by the default workflow using 50 or 300 PCs, stand-alone UMAP, and clustering on HVG and TopOMetry. Two main reasons could underlie such discrepancy in the underlying manifold of different cell types: i) high intrinsic dimensionalities (i.d.) and ii) highly non-linear geometry at a local level. To test the former, we used the FSA and MLE i.d. estimation methods and found that T cells did not present particularly high i.d. in any of the tested datasets (Supplementary Figure 7A). To test the latter, we computed PCA with up to 300 PCs and calculated the total explained covariance and found that 50 and 300 PCs explained an

average of only 20.09% and 41.89% of the total covariance for these datasets when using the default parameters in *scanpy*[52], with an *ad hoc* 'elbow point' around 30 PCs (Supplementary Figures 7B-F), at which only 17.7% of covariance was explained, in average. Remarkably, no HVG selection approach yielded orthogonal bases that explained more than 55% of covariance at 300 PCs. Such low values of explained covariance are generally found in highly non-linear systems. Taken together, these findings and the lack of specific marker genes for clusters learned from a PCA-based neighborhood graph strongly suggest that the phenotypic manifolds underlying T cell identity are highly non-linear and poorly approximated by hyperplanes of maximum covariance and point to the introduction of linear distortion by PCA preprocessing. Accordingly, we found that the distortion was significantly higher in the PCA-based UMAP representations compared to the topoMAP representation across all datasets in T cells ($p < 10^{-90}$, two-sided Wilcoxon test), as represented by ellipse eccentricities when projecting the Riemannian metric (Supplementary Figure G). Thus, it is clear that the CD4+ T cell clusters found with the standard workflow are PCA-derived artifacts rather than real biological subpopulations and that the clusters found using HVGs or TopOMetry better correspond to the actual biological subpopulations.

Discussion

Finding low-dimensional spaces encoding the phenotypic manifolds underlying single-cell data is a fundamental task in single-cell analysis and a prerequisite for the performance of most existing algorithms used in downstream analyses (e.g., clustering, marker genes, RNA velocity). However, how to properly find and evaluate such spaces has remained elusive. Here, we draw from recent advances in spectral graph theory and high-dimensional data analysis to show that such spaces are best modeled as Riemannian manifolds in a comprehensive theoretical framework and present TopOMetry, a robust, user-friendly, and efficient toolkit that allows users to fully explore such manifolds in single-cell atlases with minimal assumptions about the data (i.e., the manifold hypothesis[62]). A first-of-its-kind, the toolkit addresses numerous gaps in high-dimensional and single-cell data analysis, providing means to estimate intrinsic dimensionalities, learn consistent topological representations[13] and evaluate subspaces from a global and local perspective. The ability to learn and evaluate

dozens of representations with a single line of code empowers single-cell analysis, as one cannot be sure whether a particular computational algorithm will perform well in any given set of data: extending the conclusions obtained from even dozens of datasets to the vast and exponentially growing corpora of single-cell data would be a dangerous assumption. Instead of relying on the presumed supremacy of any given algorithm or assuming data structures, users can test different approaches that pursue the same unbiased mathematical goal (LBO approximation) and decide which to adopt based on quantitative and qualitative evaluations. Remarkably, it allows assessing distortion in visualizations for the first time by harnessing the Riemannian metric, addressing current criticism[63] on the biological interpretation of embeddings from single-cell data. Its modular design, extensive documentation, and its interoperability with the larger python ecosystem for single-cell analyses ought to make its adoption easy and seamless for the majority of users. Similar to any other tool, it presents limitations, the major one being that, in its current form, it does not allow online or inverse learning, although that may be bypassed by the use of recently developed geometric regularized autoencoders[64]. Another limitation is the high computational cost of evaluating subspaces, a shared feature of existing tools[33,43], as it involves operating on dense matrices pairwise distances on memory. We expect that our results will inspire the development of numerous other approaches harnessing the power of Laplacian-type operators and consistent manifold learning for the analysis of sc-data.

The power of the proposed theoretical framework and toolkit are demonstrated throughout the presented examples, identifying developmental trajectories in the murine pancreas that better match existing knowledge, representing the cell cycle as a closed loop, and dissecting numerous neuronal lineages throughout embryonic development. In particular, the identification of approximately 130 novel subpopulations of T cells across datasets sheds unprecedented light on T cell biology, alongside the match between these populations and TCR clones. Diversity is a known hallmark of T cells, but such diversity has remained obscured in scRNA-seq data, leading to a common belief that T cells are not transcriptionally heterogeneous. The results shown herein suggest the opposite, forming a new model in which different clones of T cells are transcriptionally distinct and that clonality can be inferred from transcriptional information alone. Moreover, our findings point to the widespread use of PCA as a universal preprocessing step without any consideration for possible non-linear geometries as the reason previous studies failed to detect them, and these results warrant the community to further assess the extent to which published results have been corrupted by PCA, and to

carefully reconsider its use in single-cell analyses. However, this methodological study falls short of further experimental investigation of these subpopulations. Thus, the mechanisms and functions underlying such diversity are yet unclear, and future studies targeting specific T cell subtypes are warranted to further elucidate these populations.

Methods

Public data acquisition and availability

Public single-cell RNA-seq datasets (Table 1) were obtained from various sources and processed following the standard data analysis. The used sources and processing parameters are available in Table 2, alongside additional information for each dataset.

Code availability

TopOMetry is freely accessible as a python library. The source code is available at https://github.com/davisidarta/topometry, and can be easily installed through the python package index (PyPI) at https://pypi.org/project/topometry/ . The library is extensively documented at https://topometry.readthedocs.io . Code used to generate the exposed examples is available at https://github.com/davisidarta/topometry-manuscript. Code used for the benchmark shown in Figure 2 and Supplementary Figure 3 is available at https://github.com/davisidarta/topometry-dr-benchmark. Any additional information can be requested from the authors and will be promptly made available.

Standard data analysis

Count matrices were used to create AnnData objects within *scanpy*[52], which were filtered to recommended thresholds. Data were then processed with the following workflow: i) data were library-size normalized to a factor of 10,000 and variance stabilized by log normalization; ii) highly variable genes (HVGs) were identified and kept for downstream analysis- selection

parameters were adjusted so that each dataset had between 1,500 and 4,000 HVGs; iii) HVG-data was then centered and scaled (i.e., standardization or z-score transformation); iv) dimensionality reduced to 50 or 300 dimensions with PCA; v) neighborhoods were computed using UMAP fuzzy simplicial sets using the cosine metric distance; vi) the neighborhood graph was projected with UMAP and t-SNE; and vii) the neighborhood graph was clustered with the Leiden community detection algorithm, with the *resolution* parameter set to 2. These steps correspond to the workflow applied in most manuscripts in which single-cell data is analyzed, being the norm in the field and regarded as the current gold-standard[65]. Thus, we refer to these as the standard workflow throughout the manuscript. Unless otherwise stated, this was performed with default parameters within the SCANPY toolkit for single-cell analysis.

When stated that stand-alone UMAP and clustering was performed on HVGs, we refer to the scaled data matrix of *cells vs. HVGs*. In these cases, all analyses were still performed within scanpy, but setting the parameters *use_rep='X'* and *n_pcs=0* in the *scanpy.pp.neighbors()* function, which causes the neighborhood graph to be computed on the scaled data instead of top principal components. UMAP projections and Leiden clustering results were then obtained normally using scanpy built-in functions.

Cluster marker genes were found with logistic regression[66] by using *scanpy's* in all demonstrated examples, which calls *scikit-learn*[67] *LogisticRegression* estimator with default parameters (L2 regularization). The number of maximum iterations was set to 1000 for all examples and clustering results.

Topologically Optimized geoMetry

Overview

Fundamentally, TopOMetry learns representations using the discrete approximation of the Laplace-Beltrami Operator (LBO; $\triangle f$), formally defined as the divergence of the gradient:

$$\triangle f \ = \nabla \cdot \nabla f$$

The LBO is a powerful descriptor of data geometry under the manifold assumption, as on a Riemannian manifold (a smooth manifold M with a Riemannian metric g defined at every point

$p \in M$), the LBO entirely determines the Riemannian metric g - and therefore, all topological properties of the manifold. This remarkable feature was previously explored by methods like Laplacian Eigenmaps (LE) and Diffusion Maps (DM)[9,10,68], in which Laplacian-type operators obtained from affinity matrices approximate the LBO. The eigendecomposition of these operators is then used to find latent subspaces that completely preserve the geometry of the underlying manifold. This approach has been shown to be particularly robust when using affinity matrices learned with variable bandwidth kernels[68], continuous-k-nearest-neighbors (CkNN)[13] or fuzzy simplicial sets[15,69].

TopOMetry combines and expands these previous approaches into a comprehensive framework that approximates the LBO iteratively to learn faithful low-dimensional representations of high-dimensional data. First, a k-nearest-neighbors (kNN) graph is obtained from the standardized data (the *base kNN graph*), and then used to estimate affinity matrices using different kernels (the *base kernel*). Laplacian-type operators are obtained from these matrices (the 'geometric' Laplacian with DM, also referred to as the diffusion operator, and the 'random-walk' Laplacian with LE), and their eigenvectors are used to form a new orthogonal basis in which the latent geometry of the original data is preserved. The learned orthogonal bases (the *eigenbases*) can be used for numerous downstream tasks, such as computing specialized neighborhood graphs for graph layout optimization (GLO) algorithms such as t-SNE and PaCMAP. It has been previously shown that multiple eigenvectors can be needed to encode data heterogeneity[43,70,71], each encoding a different aspect of data structure, i.e., distinct cellular trajectories, and dozens or hundreds of diffusion components could be needed to describe large-scale, whole-organism single-cell data. Thus, these orthogonal bases in which all latent geometry is encoded are still high-dimensional, and as such, can be described through a second round of LBO approximation. In this step, TopOMetry learns a new kNN graph from these bases, from which new affinity matrices and Laplacian-type operators can be computed - these operators encode the latent geometry on a higher level, and can be used for other downstream tasks such as clustering and GLO. The diffusion potential is used by default for clustering (e.g., with the Leiden community-detection algorithm) and GLO (e.g., with MAP, a simplified version of UMAP), and its eigendecomposition is used as initialization for GLO algorithms. As a result, TopOMetry harnesses LBO approximations at each step of dimensional reduction, offering a pipeline that is completely unsupervised and dependent uniquely on the

intrinsic geometry of the underlying manifold, while also allowing the flexibility of testing multiple options of kernels and types of Laplacian operators.

Computational implementation

TopOMetry was designed to be easy to use, and as computationally efficient as possible. Popular approximate nearest-neighbor libraries (ANN, e.g., *NMSlib, hnswlib, and annoy*) are seamlessly integrated into the toolkit. Nearly all graph operations are performed on sparse matrices and are highly scalable with multithreading. Each step of the workflow involves the use of a modular class based on *scikit-learn*[67] transformers, thus allowing users to construct custom-made pipelines from them. The main classes are estimators of neighborhood graphs, affinity kernels (and associated graph operators), eigendecompositions and graph layout optimizations. The analysis is centered on the *TopOGraph* class, which orchestrates these classes into a streamlined workflow (Supplementary Figure 1). The modular nature of the software allows it to be easily integrated into diverse machine-learning and sc-data analysis workflows. It also includes additional classes and functions for evaluating lower-dimensional spaces, estimating intrinsic dimensionalities, and visualizing results. Some of these functions harness the *TopOGraph* class into high-level APIs that allow users to compute, evaluate and inspect the results of dozens of possible layouts from a single dataset using a single line of code, or to do so using the standard *AnnData* object[52]. Extensive documentation and tutorials are available at https://topometry.readthedocs.io/en/latest/.

Construction of k-nearest-neighbor graphs

One of the most fundamental aspects of machine-learning algorithms dealing with high dimensional data is the curse of dimensionality: in higher-dimensional spaces (HDS), the capacity of distance metrics to discriminate dissimilarity decays, i.e. as dimensionality increases, the Euclidean distance between the most similar samples tends to approach the distance between the most dissimilar one[42]. Thus, the task of accurately and efficiently computing distances between samples in a large HDS is a challenge by itself, and several algorithms and tools have been developed to address it[72]. The most commonly used metric in single-cell data

analysis is the Euclidean distance, followed by the cosine and the manhattan distances - yet, in principle, any metric distance could be applied. Recent benchmarks[73,74] have achieved initial insight on the suitability of different metrics to estimate distances between single-cells, pointing towards correlation metrics as a favorable choice, although these were limited to only some single-cell RNA sequencing datasets in PCA-based workflows, and further studies are needed. This guided the choice of the cosine distance metric as a default in TopOMetry and in all presented analyses, as it trivially corresponds to the Pearson correlation distance when using standardized data (which is the current standard in single-cell analysis).

TopOMetry can employ *scikit-learn* and various ANN libraries to construct kNN graphs. Guided by a recent benchmark between distance estimation algorithms[75], we defined the Hierarchical Navigable Small Worlds (HNSW)[76,77] as a default. HNSW has been shown to achieve fast and near-optimal results while being useful for computing a variety of distance metrics (e.g. euclidean, cosine, manhattan). Similarly to other non-linear algorithms, the first step of the diffusion basis is to find a k-nearest-neighbors graph (kNN) graph. As previous studies have shown that using only highly variable features (HVF) for computations increases the power of single-cell analysis[65,78], we estimated distances from the cell by HVF matrix for the shown experiments. As a reference, ~1,500 to ~4,000 genes are generally considered a reasonable range of HVFs in single-cell RNA sequencing data analysis.

Affinity estimation

Variable bandwidth kernels

The implemented kernels draw from recent advances in kernel operators and expand them to include two other possible kernels. Instead of the classic Gaussian kernel with a fixed ε bandwidth, several studies have described that variable bandwidth kernels[13,68,70,79] act as manifold-adaptive estimators, rendering more robust results. These variable bandwidth kernels were adapted into TopOMetry as a bandwidth-adaptive kernel[68,68,71,79] in which the distance between two samples (cells) is normalized by each sample's distance to its median *k*-nearest-neighbor (i.e. the k/2 - nearest-neighbor). As an extension of this concept, we introduce a novel kernel that includes a neighborhood expansion procedure in which the original *k*NN graph is expanded to additional neighbors using a data-derived estimate of neighborhood

sparseness. In addition, we introduce a novel adaptively decaying kernel, in which the rate of similarity learning is adaptively adjusted by each cell neighborhood density instead of by a fixed hyperparameter as otherwise proposed by PHATE's 'alpha-decaying kernel'[43].

Guided by previous approaches, we first construct a k-nearest-neighbors graph for a given dataset, with N samples (cells) and M measures (genes), using a given distance metric as a local similarity measure. Let $\mathcal{M} \subset \mathbb{R}^n$ be a Riemannian manifold with unknown sampling distribution q (not necessarily uniformly distributed) in ambient dimension *n*. For a given sample $x_i \in X_{n \times m} \subset \mathcal{M}$ (the manifold hypothesis - data is assumed to be contained in the proximities of the manifold), consider $nbrs\ (x_i)$ to be the ordered set of its k-nearest-neighbors under some distance metric $d(x_i, y_i)$:

$$nbrs\ (x_i) \ = \ \{y_{i\,(1)},\ y_{i\,(2)}, \ldots,\ y_{i\,(\frac{k}{2})},\ \ldots, y_{i\,(k)}\} \mid k < N,\ k \subset \mathbb{N}$$

Given that the vector of sample observations x_i, a scaling factor σ_i corresponding to the distance of sample i to its median nearest-neighboring sample(s) in the high-dimensional space is defined:

$$\sigma_i \ \equiv \ d(x_i,\ y_{i\,(\frac{k}{2})})$$

Drawing from previous work on adaptive bandwidth diffusion kernels, the following kernel function could be initially used to convert distances into affinities:

$$K\ (x_i,\ x_j) \ = \ exp(\frac{(x_i - x_j)(x_j - x_i)}{\sigma_i + \sigma_j})$$

It is of notice that the inverse of the scaling factor σ_i is employed in the manifold-adaptive FSA method of intrinsic dimensionality estimation[39].

To minimize the impact of the choice of the number of k neighbors hyperparameter on the learned graph structure, we harness the potential of the scaling factor σ_i to carry information regarding the neighborhood density of x_i. For this, a new indirect, continuous metric named *pseudomedian*(Ω) is defined for each sample x_i as the interpolation of all σ_i into the new interval:

$$\Omega_i = \frac{(k-1)(\sigma_i - argmin(\sigma))}{argmax(\sigma) - argmin(\sigma)} + 1$$

Ω_i represents a simple, indirect, and yet intrinsic reference from the distribution of $nbrs\ (x_i)$ at a given fixed k, thus providing information on the neighborhood density of each individual sample and about the discrete graph-building process. A sample with a sparser neighborhood will have a larger value of Ω than one with a dense neighborhood, indicating the relative need for search expansion in the sparser region. This estimate is implemented in the proposed diffusion process as a way to obtain enriched information about x_i neighborhood density, and also as a measure to alleviate the impact of a fixed k hyperparameter, as the graph search can be extended as a function of maximum Ω to help avoid insufficient sampling at sparser neighborhoods:

$$k' = k + (k - argmax(\Omega_i)) \mid k' > k,\ k' \subset \mathbb{N}$$

It is of note that PHATE[43] introduced a kernel with variable bandwidth ε equipped with a novel alpha-decay, in which α is a hyperparameter that controls the kernel decay rate. This was a first attempt to handle cases in which a sparsely sampled neighborhood can present with distant neighbors that are weighted with a high affinity. Note that $\alpha = 2$ leads to the classic Gaussian kernel, and ε is a fixed bandwidth

$$K_\alpha(x_i, x_j) = \frac{1}{2} exp(-\left(\frac{d(x_i, x_j)}{\varepsilon_i}\right)^\alpha) + \frac{1}{2} exp(-\left(\frac{d(x_i, x_j)}{\varepsilon_j}\right)^\alpha)$$

We harness the concept of the alpha-decaying kernel to implement an *adaptively-decaying kernel*, in which the rate of decay is related to the neighborhood sampling density distribution Ω:

$$K_{\alpha,k'}(x_i, x_j) = \frac{1}{2} exp(-\left(\frac{d(x_i, x_j)}{\sigma_i}\right)^{\alpha_i}) + \frac{1}{2} exp(-\left(\frac{d(x_i, x_j)}{\sigma_j}\right)^{\alpha_j}) \mid \alpha_i = 2^{\frac{k'-\Omega_i}{\Omega_i}}$$

In summary, TopOMetry includes the fixed-bandwidth kernel, adapts the *bandwidth-adaptive* kernel, and introduces the *bandwidth-adaptive* kernel *w*ith neighborhood expansion, the *bandwidth-adaptive adaptively-decaying* kernel, and the *bandwidth-adaptive adaptively-decaying* with neighborhood expansion. By default, the *bandwidth-adaptive* kernel is used, due to its simplicity and relation to the FSA i.d. estimator.

Continuous k-nearest neighbors

Continuous k-nearest neighbors (CkNN) is a recently introduced[80] method that constructs a single unweighted graph from which the underlying manifold's structure (i.e. topological information) can be retrieved, given any compact Riemannian manifold. In its introductory manuscript, it was also shown that CkNN is the only unweighted graph construction strategy for which the graph Laplacian converges to the Laplace-Beltrami Operator (LBO). Conceptually, CkNN points to the superiority of *consistent homology* over *persistent homology*. However, differently from variable-bandwidth kernels or fuzzy simplicial sets, this is achieved through an unweighted graph rather than through a weighted graph.

CkNN's goal is to create an unweighted undirected graph from a given set of samples with distances given by a metric d. This translates to deciding whether to connect samples with edges. The two traditional approaches for constructing such a graph are a) fixed ϵ-balls (i.e. connect samples i, j if $d(i, j) < \epsilon$); and b) k-Nearest Neighbors (kNN), in which a fixed integer k given as a user defined hyperparameter is used to define edges (i.e. connect samples i, j if $d(i, j) \leq d(i, i_k)$ or if $d(i, j) \leq d(j, j_k)$; i_k and j_k being i and j k-th nearest neighbor, respectively). As shown in CkNN's manuscript and in its excellent performance within TopOMetry, a less discrete and rather continuous version of kNN circumvents several limitations imposed by these traditional strategies.

CkNN connects samples i, j with edges if $d(i, j) < \delta(\sqrt{d(i, i_k)\, d(j, j_k)})$. Here, δ is a unitless scale parameter allowed to continuously vary. Given a finite set of data, there is only a finite set of possible values for δ, which is set within TopOMetry as a user-provided hyperparameter. There are two main advantages to this approach: first, it allows one to interpret the CkNN graph construction as a kernel method, enabling interpreting the graph Laplacian in terms of δ; second, it allows fixing the k hyperparameter, enabling the use of $d(i, i_k)$ as a local density estimate, similarly to the approximation of geodesics with fuzzy simplicial sets and diffusion harmonics. In practice, this captures the natural multiscale topology of data, wielding smooth geometries where data is sparsely sampled and fine-grained details where data is densely sampled.

Fuzzy simplicial sets

The application of 1-skeleton fuzzy simplicial sets as topological *metrics* on TopOMetry draws entirely from previous work, particularly from UMAP[15] theory and the work of David Spivak[69]. Our innovation is represented by the use of these metrics to construct new orthogonal bases that capture latent topology, and subsequently to construct topological graphs prior to layout optimization. When using this method to learn metrics from some data $X_{n \times m}$, it is assumed $X$is uniformly distributed on a locally connected $\mathcal{M}$ Riemannian manifold with a locally constant metric g. UMAP originally addresses the uniformity assumption by creating a custom distance for each sample by normalizing distances with respect to their k-th nearest-neighbor to effectively approximate geodesics. To merge each of the discrete metric spaces generated per sample into a global informational space, the metric spaces are converted into fuzzy simplicial sets. A consensus metric can then be obtained by taking a fuzzy union across all metric spaces encoded as fuzzy simplicial sets, rendering a single fuzzy simplicial set which is granted to capture relevant topological metrics on $\mathcal{M}$. The Laplace-Beltrami Operator acting on $\mathcal{M}$ can then be approximated from the normalized laplacian of this fuzzy graph representation.

The aforementioned approach is grounded on solid category theory and no major modifications to the fuzzy simplicial set algorithms defined and implemented in UMAP were made, aside from improving its computational performance with approximate-nearest-neighbors libraries. For brevity, we direct the reader to UMAP's[15] and David Spivak's[69] manuscripts for in-depth descriptions of the involved category theory. A summarized description follows.

Simplicial sets are generalizations of simplicial complexes that provide a combinatorial approach to the study of topological spaces. Let Δbe a category with the finite order sets $[n] = \{1, \ldots, n\}$ as objects and morphisms given by order-preserving maps. Δ^{op} denotes a category with the same objects as Δ and morphisms equivalent to those of Δ with reversed direction. Then, a simplicial set is defined as a functor from Δ^{op} to $Sets$(the category of sets). Given a simplicial set $X : \Delta^{op} \rightarrow Sets$, we refer to the set $X([n])$ as X_n , and to the elements of the set as the n-simplices of X. In the category of simplicial sets, there's a direct correspondence between the n-simplices of $X$and morphisms of the standard simplices Δ^n ($\Delta^n \rightarrow X$). For each

$x \in X_n$ there's a corresponding morphism x: $\Delta^n \to X$, and therefore $colim_{x \in X_n} \Delta^n \approx X$. The standard covariant functor $|\cdot| : \Delta \to Top$ (the category of topological spaces) that sends $[n]$ to the standard n-simplex $\left|\Delta^n\right| \subset \mathbb{R}^{n+1}$ with standard subspace topology is defined as:

$$\left|\Delta^n\right| \equiv \left\{(t_0, \dots, t_n) \in \mathbb{R}^{n+1} \mid \sum_{i=0}^{n} t_i = 1,\ t_i \geq 0\right\}$$

And because $X : \Delta^{op} \to Sets$ is a simplicial set, the realization of X ($|X|$) is $|X| = colim_{x \in X_n} |\Delta^n|$, which associates the topological space with the simplicial set. On the other hand, given a topological space Y, the associated simplicial set $S(Y)$ (the singular set of Y) is defined as $S(Y) : [n] \to hom_{Top}(|\Delta^n|, Y)$. The theoretical contributions of David Spivak and UMAP's authors involve exploiting the fact that the realization functor and singular set functors form an adjunction, translating between topological spaces and simplicial sets - for this, a categorical presentation of fuzzy sets was outlined.

A fuzzy set can be defined as a set for which membership is not binary - rather, it is a fuzzy property represented by continuous values in the unit interval I (with $I = (0, 1] \subseteq \mathbb{R}$, and with topology given by intervals of the form $[0, a)\ for\ a \in (0, 1]$). This allows one to define presheafs: a presheaf $\mathcal{P}$ on I is a functor from Δ^{op} to $Sets$. A fuzzy set is a presheaf on I such that all maps $\mathcal{P}(a \leq b)$ are injections. $\mathcal{P}([0, a))$ is the set of all elements with at least a membership strength. With this concept in mind, the category $Fuzz$ of fuzzy sets is defined as the full subcategory of sheaves on I spanned by fuzzy sets. Consequently, defining fuzzy simplicial sets (FSS) corresponds to considering presheaves of Δ to be valued in $Fuzz$ instead of $Sets$. The objects of the category of FSS $sFuzz$ are thus functors from Δ^{op} to $Fuzz$, and its morphisms are given by natural transformations. Conversely, FSS can be viewed as sheafs over the product topology $\Delta \times I$, allowing one to apply them to a broad category of extended-pseudo-metric spaces. This will be useful as the initial geodesic normalization step involves creating a unique metric space for each sample.

Extended-pseudo-metric-spaces (X, d) are defined as sets X and maps $d : X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$, being objects of the category $EPMet$, which has non-expansive maps as morphisms. A further subcategory of finite extended-pseudo-metric-spaces is defined as $FinEPMet$. Previously[69], the realization and singular set functors were extended to FSS by the construction of natural

extension functors, $Real$and $Sing$, between the categories $sFuzz$and $EPMet$. $Real$is defined in terms of standard fuzzy simplices ($\Delta^n_{<a}$), similarly to the the classic realization functor $|\cdot|$:

$$Real(\Delta^n_{<a}) \equiv \left\{(t_0, \dots, t_n) \in \mathbb{R}^{n+1} \mid \sum_{i=0}^{n} t_i =- log(a),\ t_i \geq 0\right\}$$

A morphism $\Delta^n_{<a} \to \Delta^m_{<b}$ exists only if $a \leq b$, and is determined by a Δ morphism $\sigma: [n] \to [m]$. The action of Real on such a morphism is then given by the non-expansive map

$$(x_0, x_1, \dots, x_n) \to \frac{log(b)}{log(a)}\left(\sum_{i_0 \in \sigma^{-1}(0)} x_{i_0}, \sum_{i_0 \in \sigma^{-1}(1)} x_{i_0}, \dots, \sum_{i_0 \in \sigma^{-1}(m)} x_{i_0}\right)$$

This map is then extended to a more general simplicial set X by defining $Real(X) \equiv colim_{\Delta^n_{<a} \to X} Real(\Delta^n_{<a})$. Once again, similarly to the classical case, the right adjoint functor $Sing$is then defined for a given extended-pseudo-metric space Yin terms of its action on the category of product topology $\Delta \times I$:

$$Sing(Y): ([n], [0, a)) \to hom_{EPMet}(Real(\Delta^n_{<a}),\ Y).$$

As only finite metric spaces are of interest in this case, the associated FSS $Fin - sFuzz$is considered and the correspondent $FinReal$and $FinSing$are used. Formally, the functor $FinReal: Fin - sFuzz \to FinEPMet$ is defined by:

$$FinReal(\Delta^n_{<a}) \equiv (\{(x_0, x_1, \dots, x_n\},\ d_a)\ ,$$

where $d_a(x_i, x_j)$ is equivalent to$- log(a)$ if $i \neq j$ and to 0 otherwise, allowing

$$FinReal(X) \equiv colim_{\Delta^n_{<a} \to X} FinReal(\Delta^n_{<a})$$

to be defined. Analogously, $FinSing: FinEPMet \to Fin - sFuzz$ can be defined by:

$$FinSing(Y): ([n], [0, a)) \to hom_{FinEPMet}(FinReal(\Delta^n_{<a}), Y)$$

By construction, we have that the functors $FinReal: Fin - sFuzz \to FinEPMet$ and $FinSing: FinEPMet \to Fin - sFuzz$form an adjunction in which $FinReal$is the left adjoint and

$FinSing$is the right adjoint. After defining fuzzy simplicial sets and their functors, one can merge the rather incompatible metric spaces created during geodesic approximation to preserve topological information from each metric by first translating each to a fuzzy simplicial set with the $FinSing$functor, and then taking a union across all FSS, resulting in a single FSS that is ought to capture the underlying structure of the manifold $\mathcal{M}$. Given a dataset $X = \{X_1, \ldots, X_n\} \in \mathbb{R}^n$and its associated family of extended-pseudo-metric spaces $\{(X, d_i)\}_{i=1\ldots N}$ with common carrier set X, it follows that $d_i(X_i, X_k) = d_{\mathcal{M}}(X_i, X_k) - \rho$if $i = j$or $i = k$, and that $d_i(X_i, X_k) = \infty$if otherwise. ρis the distance between $X_i$and its nearest neighbor, and $d_{\mathcal{M}}$is the approximate geodesic distance on the manifold $\mathcal{M}$. Then, the fuzzy topological representation of X corresponds to the union: $\bigcup_{i=1}^{n} = FinSing((X, d_i))$.

Learning a latent basis from Laplacian-type operators

TopOMetry employs eigendecomposition to learn latent bases from Laplacian-type operators that approximate the LBO. Diffusion Maps (and its multiscale version) result from the eigendecomposition of the diffusion operator. The construction of a diffusion process on a graph from a given kernel function is a classical topic in spectral graph theory. Given a symmetric positive semi-definite affinity matrix W, we define the degree matrix D as a diagonal matrix composed of the sum of columns of W: $d_{ii} = \sum_j w_{ij}$. The unnormalized (or 'combinatorial') graph Laplacian L is given by: $L = D - W$. Then the normalized (or 'symmetrized normalized') graph Laplacian L_{sym} is given by: $L_{sym} = D^{-1/2} L D^{-1/2}$. The random-walk Laplacian L_{rw} is given by: $L_{rw} = D^{-1} L = I - D^{-1} W$. The diffusion operator P can be simply defined as $P = D^{-1} W$, but can also be parametrized as $P_{(\alpha)}$ by the α *anisotropy* value, which controls how much the data distribution is allowed to bias the obtained results so that only the underlying geometry is considered and the LBO is approximated when $\alpha = 1$. For that, we first obtain an α-normalized

$W_{(\alpha)}$: $W_{(\alpha)} = D^{-\alpha}\ W\ D^{-\alpha}$ and form a new degree matrix $D_{(\alpha)}$: $d_{(\alpha),ii} = \sum_j w_{(\alpha),ij}$, which is used to form the anisotropic diffusion operator $P_{(\alpha)} = D_{(\alpha)}^{-1}\ W_{(\alpha)}$.

These operators can then be decomposition using the generalized eigenvector problem. In classic Laplacian Eigenmaps (LE), the system $Lf = \lambda Df$ is solved to obtain a number of m eigenvectors, and these are used to map the data into a m-dimensional embedding ($x_i \rightarrow (f_1(i), \ldots, f_m(i))$). By default, TopOMetry uses the random-walk Laplacian L_{rw}, and eigendecomposes it by $L_{rw}f = \lambda f$. In LE, the eigenvectors with smallest eigenvalues are used, the first one being dropped as it is trivial. In Diffusion Maps (DM), the eigenvectors with the highest eigenvalues are obtained by solving $P_{(\alpha)}f = \lambda f$.

Most approaches using DM rely on powering the diffusion operator $P_{(\alpha)}$ by a given number of t 'diffusion steps'[9,10,43], which consist of an additional hyperparameter. Instead, we harnessed an adaptation consisting of multiscaling, which avoids setting a fixed number of *t* diffusion steps and renders weighted diffusion distances across all possible timescales[71,81].

Given a diffusion operator $P_{(\alpha)}$ with a defined number of t steps and a given estimate M of the intrinsic dimensionality (number of non-trivial eigenvalues) of $\mathcal{M}$, one can achieve the spectral decomposition:

$$P_{(\alpha)}^{t} = \sum_{i \geq 0} \lambda_i^{2t} \phi_i \phi_j$$

In which $P_{(\alpha)}^{t}$ encodes the diffusion distance DD between samples x_i and x_j, $\phi_i^{(l)} = \phi(x_j)$ along the diffusion component ϕ:

$$DD^t(\phi_i, \phi_j)^2 = \sum_{t=1}^{L} \lambda_l^{2t} (\phi_i^{(l)} - \phi_j^{(l)})^2$$

We next multiscale distances to account for all possible t scales, rendering multiscale diffusion distances $msDD$:

$$msDD^t(\phi_i, \phi_j)^2 = \sum_{t=1}^{\infty} \sum_{t=1}^{L} \lambda_l^{2t} (\phi_i^{(l)} - \phi_j^{(l)})^2$$

Which can be reformulated as:

$$msDD^t(\phi_i, \phi_j)^2 = \sum_{t=1}^{L} \left(\frac{\lambda_l}{1-\lambda_l}\right)^2 (\phi_i^{(l)} - \phi_j^{(l)})^2 \mid 1 > \lambda_1 > \lambda_2 > \ldots > \lambda_L > 0$$

The eigenvalues of the resulting decompositions are useful to perform an empiric estimation of the data intrinsic dimensionality during computations, as the numerical values hit the limits of floating-point precision, which is encoded either by a sign-flip or by the decomposition algorithm outputting null outputs for the eigenvalues that pass this limit. We found that this is a useful way to set the key hyperparameter *n_eigs*, which defines a fixed number of latent dimensions to be computed, and recommend putative users to increase this parameter up to a point where such a spectral gap (or an eigengap) can be observed in scree plots. As shown in Figure 3, the identified spectral gap often corresponds to external estimates of intrinsic dimensionality.

Graph-layout optimization

Graph layout optimization (GLO) are techniques that aim to optimize a lower-dimensional representation (usually in two or three dimensions, for visualization) that preserves most of some properties from the original data by minimizing a given cost function. Differences between these methods are mostly ough to the different methods to learn similarities from distances and to the number and design of cost functions[82]. The seminal t-SNE method, for instance, employs t-Student scores to estimate similarities between samples, and minimizes the Kullback–Leibler divergence between the high-dimensional and lower-dimensional representations. UMAP, which followed t-SNE as the *de facto* gold standard for visualization, learns similarities with fuzzy simplicial sets (FSS), and then minimizes the cross-entropy between the high and lower-dimensional representations. PaCMAP has shed additional light on how these methods work by analyzing the asymptotics of cost functions and empirically testing and defining the characteristics of efficiently designed cost functions; it also introduced the use of more than one cost function during the layout optimization process (i.e. initial iterations follow one cost function, intermediate steps another cost function, final steps a third one). MDE generalizes these

concepts by situating layout optimization methods as special cases of convex optimization. With the exception of MAP, TopOMetry harnesses pre-existing implementations of layout algorithms in the python programming language, such as t-SNE, TriMAP, PaCMAP and MDE. A detailed description of the main techniques currently included in the framework and used in the benchmark follows:

t-Distributed Stochastic Neighborhood Embedding (t-SNE)

t-SNE[14] is an improved version of SNE[83] (Stochastic Neighborhood Embedding) that offers significantly superior performance when applied to real-world data. SNE first converts high-dimensional distances into conditional probabilities that represent similarities: the similarity between samples x_i and x_j corresponds to the probability $p_{i|j}$ that x_j would have x_i as neighbor if neighbors were chosen according to the probability density under a Gaussian centered at x_j.

$$p_{i|j} = \frac{exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2 / 2\sigma_i^2)}$$

A similar conditional probability $q_{i|j}$ can be computed for the points in the low-dimensional embedding. Traditionally, the gaussian employed in $q_{i|j}$ is set to $\frac{1}{\sqrt{2}}$, and these similarities are defined by:

$$q_{i|j} = \frac{exp(-||x_i - x_j||^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2)}$$

SNE aims to find a low-dimensional representation that minimizes the difference between these two probability distributions by considering the sum of all Kullback-Leibler (KL) divergences between the high- and low-dimensional representation of each data point. Considering P_i as the conditional probability distribution of samples in their ambient high-dimensional space and Q_i its low-dimensional counterpart, the sum of KL divergences for all samples is then minimized through gradient descent, with a cost function C defined by:

$$C = \sum_i KL(P_i || Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}} .$$

The gradient of of this cost function has the form:

$$\frac{\delta C}{\delta \gamma_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(\gamma_i - \gamma_j).$$

And its update with a momentum term (to avoid local minima) is given by:

$$\gamma^{(t)} = \gamma^{(t-1)} + \eta\frac{\delta C}{\delta \gamma} + \alpha^{(t)}(\gamma^{(t-1)} - \gamma^{(t-2)})$$

t-SNE's superiority to SNE is due to two differences: first, the implementation of a symmetrized version of the SNE cost function; second, the use of heavy-tailed distributions to address the crowding problem. Significant advances have also been made regarding the optimization of the t-SNE cost function to avoid local minima and increase performance. To symmetrize the cost function, instead of minimizing the sum of all KL divergences between high-dimensional and low-dimensional probability distributions, a single KL divergence between the joint probabilities distributions P (high-dimensional) and Q (low-dimensional) is used:

$$C = KL(P||Q) = \sum_i \sum_j p_{j|i} log \frac{p_{ji}}{q_{ji}}$$

In t-SNE, a Student t-distribution with one degree of freedom (the same as a Cauchy distribution) is used as a heavy-tailed distribution for the low-dimensional mapping. This defines the joint probabilities q_{ij}:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l} (1 + ||y_k - y_l||^2)^{-1}}$$

The gradient of the KL divergence between P and Q (computed using the above equation) is given by:

$$\frac{\delta C}{\delta \gamma_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(\gamma_i - \gamma_j)$$

With an update with momentum identical to that of SNE.

There are numerous modifications and adaptations of the t-SNE algorithm (e.g. heavy tailing, density preservation, perplexity combination), and these exceed the scope of this manuscript. The version of t-SNE included in TopOMetry relies on the MulticoreTSNE[84,85] package.

Uniform Manifold Approximation and Projection (UMAP) and Manifold Approximation and Projection (MAP)

The similarity metric when using the 'fuzzy' algorithm in TopOMetry corresponds to this 1-skeleton of the fuzzy simplicial sets, that is, a fuzzy graph (i.e. a fuzzy set of edges). In essence, UMAP's algorithm consists of only a couple more steps to learn a low-dimensional representation from X by constructing fuzzy topological representations of X and its putative low-dimensional representation Y. First, the fuzzy set of edges is represented by a reference set A and a membership strength function $\mu : A \rightarrow [0, 1]$. Using a presheaf representation P, it's possible to define $A = \bigcup_{a \in (0,1]} \mathrm{P}([0, a))$, and $\mu(x) = sup\{a \in (0, 1] \mid x \in \mathrm{P}([0, a))\}$. Then, to generate visualizations, the graph layout is optimized by minimizing the cross-entropy C between the high-dimensional and low-dimensional fuzzy sets (A, μ) and (A, v)in terms of the reference set A:

$$C((A, \mu), (A, v)) \equiv \sum_{a \in A} \left(\mu(a) \, log\left(\frac{\mu(a)}{\nu(a)}\right) + (1 - \mu(a)) \, log\left(\frac{1 - \mu(a)}{1 - v(a)}\right)\right)$$

The MAP approach included in TopOMetry corresponds to a near-identical implementation of UMAP's optimization procedure, but using arbitrary graphs that are directly used for layout optimization (i.e. the manifold has already been approximated), thus only assuming that the cross-entropy optimization finds global near-optima solutions.

Pairwise Controlled Manifold Approximation and Projection (PaCMAP)

PaCMAP holds significant similarities with UMAP and TriMAP, but introduces its own unique insights. The core idea of PaCMAP is establishing robust loss functions that take into account not only a sample's local neighborhood, but how that neighborhood relates to closely located

neighborhoods and with far away points. First, the neighborhood affinity graph from data is built in a manner identical to TriMAP triplet weighting procedure, in which distances between samples are scaled by the distance between the target sample and its 4th to 6th nearest neighbor:

$$d_{ij}^2 = \frac{||x_i - x_j||^2}{\sigma_{ij}}$$

Next, pairs of samples are selected as follows:

- Near pairs - a sample's *k*-nearest-neighbors, based on the scaled distance.
- Mid-near pairs - for each sample *i*, 6 other samples are randomly selected, and the second-closest to *i* is paired to it in a mid-near pair. The number of mid-near pairs to be sampled is *k* times a hyperparameter (*MN_ratio, default 0.5)*.
- Further pairs - uniformly sample non-neighbors of *i*. The number of further pairs to be sampled is *k* times a hyperparameter (*FP_ratio, default 2)*.

Given an initialization (PCA in default PaCMAP, LE in TopOMetry), the embedding is then optimized in three different stages, each with different weights for near-pairs, mid-near pairs and further pairs. Consider a sample *i,* its neighbor *j*, its mid-near pair *k* and its further pair *l.* Given the low-dimensional distance $d'_{ab} = || y_a - y_b ||^2 + 1$between samples *a* and *b*, PaCMAP minimizes the loss:

$$L_{PacMAP} = (w_{nb} \times \sum \frac{d'_{ij}}{10 + d'_{ij}}) + (w_{mn} \times \sum \frac{d'_{ik}}{10^4 + d'_{ik}}) + (w_{fp} \times \sum \frac{1}{10 + d'_{il}})$$

Where $w_{nb}$is a weight given to *i* neighbors, w_{mn}a weight given to its mid-near pairs and w_{fp}a weight given to its further pairs. The optimization is performed in three stages, each with a number of iterations *t* that can be controlled by the user as hyperparameters. In the first stage (i.e. first 100 *t* iterations), $w_{nb} = 2$,

$w_{mn} = 1000 \times (1 - \frac{t-1}{100}) + 3 \times (\frac{t-1}{100})$, and $w_{fp} = 1$. In the second stage, $w_{nb} = 3$,

$w_{mn} = 3$, and $w_{fp} = 1$. In the third and last stage, $w_{nb} = 1$, $w_{mn} = 0$, and $w_{fp} = 1$.

The uniqueness and novelty of PaCMAP lies in the fact that local and global structure are balanced throughout the optimization process by the choice of sampling neighbors, mid-near pairs and further pairs simultaneously. When employed in TopOMetry, PaCMAP uses a denoised topological orthogonal basis instead of the raw data; similarly to TriMAP, feeding it the downstream topological graph may be counterproductive due to its peculiar neighborhood sampling process.

Estimation of intrinsic dimensionalities

Intrinsic dimensionalities were estimated using the Maximum-Likelihood Estimator (MLE)[47] and the modified FSA[39] algorithms. A summary of these methods follow:

Maximum Likelihood Estimator (MLE)

Consider X i.i.d. observations representing an embedding of a lower-dimensional sample $X_i = g(Y_i)$, where Y_i are sampled from an equally lower-dimensional and unknown density function f, g being a continuous and smooth mapping. Given a point x, it is assumed that $f(x)$is constant in a small sphere $S_x(R)$ of radius R centered on x, and the observations are treated as a homogeneous Poisson process in $S_x(R)$. Substituting a number of *k*-nearest-neighbors instead of a radius R, one can find that the estimate for the intrinsic lower-dimensionality $m_k(x)$ around x is:

$$m_k(x) = \left[\frac{1}{k-1}\sum_{j=1}^{k-1} log\frac{T_k(x)}{T_j(x)}\right]^{-1},$$

in which $T_k(x)$ represents the distance between x and its k-nearest-neighbor. The global estimate is obtained by averaging all local i.d. Estimates.

Manifold adaptive dimensionality estimation (FSA)

The FSA (Farahmand, Szepesvári & Audibert) method[39,86] is extremely simple: it uses two neighborhoods around a data point to estimate its local intrinsic dimensionality $\delta(x)$. In particular, it uses the ratio between the distance R from x to its *k* and *k/2* nearest neighbors:

$$\delta_k(x) = \frac{log(2)}{log(\frac{R_k}{R_{k/2}})}$$

Note that the ratio $\frac{R_k}{R_{k/2}}$ corresponds to the scaling factor σ used in the bandwidth-adaptive kernel estimator employed in TopOMetry. The median of local estimates[39] is then used to represent the global intrinsic dimensionality of the dataset.

Estimating and representing distortions with the Riemannian metric

We draw from the seminal work of Perrault-Joncas and Meila[36], which first used the Riemannian metric to visualize distortions in the data and learn locally isometric embeddings by correcting projections, using fixed bandwidth kernels. In that regard, our contribution is restricted to extending their work using *kNN* graphs and variable bandwidth kernels and to applying these principles to single-cell data.

The Riemannian metric g is a symmetric positive definite tensor field that defines an inner product $<,>_g$ on the tangent space T_pM for every $p \in M$, M being a Riemannian manifold. The Laplace-Beltrami operator (LBO) $\triangle f = \nabla \cdot \nabla f$ can be expressed by means of g:

$$\triangle f = \frac{1}{\sqrt{det(g)}} \frac{\partial}{\partial x^l} (\sqrt{det(g)} g^{lk} \frac{\partial}{\partial x^k} f)$$

Because $\triangle f$ is approximated by the random-walk graph Laplacian L_{rw} and the anisotropic diffusion operator P_α, one can then use it to estimate the inverse of the Riemannian metric $g(p)^{-1}$ using an arbitrary coordinate chart or embedding:

$g^{ij} = \frac{1}{2}\Delta f(x^i - x^i(p))(x^j - x^j(p)) \mid x^i = x^i(p),\ x^j = x^j(p)$, with $i, j = 1,\ldots, d$. Computing the inverse of g is then straightforward.

The method can then be extended to work with any embedding f of the manifold M. Consider d the manifold intrinsic dimension and s the dimension of the embedding space. For for any embedded point f_p, there will be a corresponding $s \times s$ matrix h_p with rank d defining a scalar product and with null space orthogonal to the tangent space $T_{f_{(p)}} f(M)$. h can be defined so that $(f(M), h)$ is isometric with (M, g), corresponding to the embedding Riemannian metric. The embedding metric h_p is given by the pseudoinverse of its dual, $\hat{h}$, where:

$$\hat{h}^{ij} = \frac{1}{2}\Delta f(x^i - x^i(p))(x^j - x^j(p)) \mid x^i = x^i(p),\ x^j = x^j(p)$$

In practice, using the graph Laplacian L to approximate Δ, we have:

$$\hat{h}^{ij} = \frac{1}{2}\left[L(f^i \cdot f^j) - f^i \cdot (Lf^j) - f^j \cdot (Lf^i)\right]$$

$\hat{h}$ can then be eigendecomposed, with eigenvalues ordered from largest to smallest, and the embedding metric h is given by the pseudoinverse of $\hat{h}$. For each point, the ellipsoid representation can then be obtained using its eigendecomposition. When using two dimensions, the ellipse orientation is given by the arctangent of the ratio between the eigenvectors, and the size of the major and minor axes are given by $2 \times \sqrt{\lambda} \times \kappa$ for the largest and smallest λ eigenvalues, respectively, κ being a scaling factor.

Quantitative evaluations

Global score

To evaluate the preservation of global structure by embeddings and layouts, we elected the PCA resemblance score. It was first introduced in the TriMAP manuscript[41] and roughly captures how much a given embedding ***Y*** performs compared to PCA when it comes to preserving the

global structure of the original data ***X***. By design, PCA has the lowest possible *Minimum Reconstruction Error* (MRE), defined as:

$$\varepsilon(Y|X) \equiv min_{A \in \Re^{m \times n}} ||X - AY||_F^2$$

Where $|| \cdot ||_F$ denotes the Frobenius norm. A global score (GS) is then defined by normalizing a given MRE by PCA's MRE (which by construction is 1):

$$GS(Y|X) \equiv exp(-\frac{\varepsilon(Y|X) - \varepsilon_{PCA}}{\varepsilon_{PCA}}) \in [0, 1].$$

Interestingly, as exposed in the results, the global score remains rather stable and close to 1.0 across all embedding and layout algorithms that were tested, given a PCA or spectral initialization. We empirically show that this score is approximately the same using either PCA or a spectral layout (LE) and implement functions to perform such tests within TopOMetry.

Trustworthiness

For evaluating the preservation of local structure from an isomorphic perspective, we harnessed the well-known *trustworthiness*[40] metric available on the *scikit-learn* machine-learning toolkit. Given a small number of k-nearest-neighbors, this metric is defined as:

$$T(k) = 1 - \frac{2}{nk(2n-3k-1)} \sum_{i=1}^{n} \sum_{j \in N_i^k} max(0, (r(i,j) - k)),$$

In which for each sample i, N_i^k are its k-nearest-neighbors in the low-dimensional space, and j is its $r(i,j)$-th nearest-neighbor in the original high-dimensional space. Essentially, this penalizes for samples (cells) that are nearest neighbors in the low-dimensional space, but not in the original space, in proportion to their original rank. A score of 0 implies complete loss of local geometry, and a score of 1 implies complete preservation of local geometry, at the rank of *k*.

Geodesic correlation

For evaluating the preservation of local structure from an isometric perspective, we built upon previous approaches that explored geodesic distances and the Spearman correlation between original high-dimensional and downstream low-dimensional distances[43,87]. This task is computationally very intensive and thus was not performed by default in the benchmark. kNN graphs are built from both the original high-dimensional data matrix and the representations to be scored. Next, the geodesic distances (i.e., the weighted shortest-path pairwise distances) are obtained for each of these graphs, and their distributions are then scored accordingly to the Spearman correlation (R). Cells can then be ranked accordingly to their neighborhood indices - let n be the number of cells and d_i be the difference between ranks for cell i in the original high-dimensional space and in the latent low-dimensional space. Then, we can calculate the Spearman correlation as:

$$R = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

The Spearman rank-order correlation coefficient (R) is a nonparametric measure of association between two sets of data, and superior to the Pearson correlation in this case because it assumes only monotonicity, not linearity. Intuitively, this score tells whether increasing distances in the low-dimensional embedding is related to increasing distances in the high-dimensional data: a score of 0 implies no association at all, and a score of 1.0 implies perfect association.

Benchmark

For the benchmark presented in Figure 2 and Supplementary Figure 3, we used the global score and the trustworthiness score. 200 principal components or eigencomponents were used for each non-biological dataset, and 600 principal components or eigencomponents were used for each single-cell dataset. For all datasets, the cosine metric distance was used with a value of 30 nearest-neighbors. All other hyperparameters used in t-SNE, UMAP, PaCMAP, and TopOMetry were set to defaults. We restricted the tested GLO techniques to t-SNE, UMAP, and PaCMAP due to practical limitations in calculating results for additional methods (e.g., including

TriMAP and MDE or the plethora of available options would lead to an unreasonable increase in runtime).

RNA velocity analysis and lineage inference

For RNA velocity analysis of the murine pancreas development dataset[48] , we used the python toolkit *scvelo*[26]. Analysis was performed with standard parameters to reproduce *scvelo* documentation. TopOMetry analysis was performed using the msDM eigenbasis (obtained with the bandwidth_adaptive kernel) and the MAP projection method was used on the learned diffusion potential from this eigenbasis with standard parameters. RNA velocity was then recomputed using the msDM representation to compute neighborhood graphs using the *scvelo.pp.moments* function, with all other hyperparameters being identical to the standard analysis.

For the MOCA dataset, TopOMetry msDM eigenbasis was used to decompose the data into 800 eigenvectors (796 positive). Clustering with the Leiden community detection algorithm and projection with MAP were then carried out using the diffusion potential learned from this eigenbasis, using default parameters.

Analysis of T cell diversity and clonality

The datasets used to explore T cell diversity were cell-type annotated using CellTypist, using the graph connectivities learned from the scaled HVG data matrix for overclustering and majority voting[51]. The *Immune_All_High* model was used to annotate main cell types, and the *Immune_All_Low* to annotate subtypes. For quantifying the amount of detected T cell clusters, a threshold of 0.8 prediction probability was used.

All datasets were analyzed following the standard workflow within *scanpy*, using 50 or 300 principal components and with the Leiden clustering algorithm set to *resolution=2.* All other hyperparameters were set to defaults. Marker genes were found using logistic regression[66] with *max_iters=1000* and all other hyperparameters set to defaults. TopOMetry analysis was performed using the msDM eigenbasis with the bandwidth-adaptive kernel to calculate a total of

500 eigencomponents for all datasets, from which the diffusion potential was used to compute MAP projections. All other TopOMetry hyperparameters were set to defaults.

For the analysis of the TICA dataset[51], an *AnnData* object containing scRNA-seq and VDJ information for the T cell compartment was obtained from tissueimmunecellatlas.org (*GEX+TCRab data*). We then reanalyzed this data with *scirpy*[60] following the steps exactly as detailed in its documentation without any change to parameters other than to use TopOMetry latent representations and clustering results. TopOMetry analysis was performed using the same parameters for all other PBMC datasets.

Computational Environment

The benchmark presented in Figure 2 and Supplementary Figure 3 was computed on a virtual machine with Intel(R) Xeon(R) Gold 6252 CPU @ 2.10GHz x 128 threads, with a total of 320GB of ERCC DDR4 SDRAM with the 64-bit Ubuntu 18.04.6 LTS distribution of the Linux operating system, provided by the Institute of Biology from the University of Campinas. All other analyses were performed on a six-core Lenovo ThinkPad P52 Intel® Xeon(R) E-2176M CPU @ 2.70GHz × 12 threads, with a total of 128GB of ERCC DDR4 SDRAM with 64-bit 21.10 Pop!OS distribution of the Linux operating system. Package versions are listed in Methods Table 1.

Funding

DS-O was supported by grant #2020/04074-2, São Paulo Research Foundation (FAPESP), and LAV was supported by grant #2013/07607-8, São Paulo Research Foundation (FAPESP).

Acknowledgments

We thank Gustavo B. Bellini, Valcir V. Vargas, and Marcos Akira at the Institute of Biology from the University of Campinas for their most kind support with computational infrastructure. We thank Leland McInness, Dmitry Kobak, and Akshay Agrawal for their valuable comments on

TopOMetry. We thank Bruno Loyola Barbosa for his kind assistance in testing the early implementations of TopOMetry. We thank Ebru Erbay, Helder Nakaya, and Ana Domingos for their generous feedback during the development of TopOMetry.

Author Contributions

DS-O and LAV jointly conceived the idea of harnessing phenotypic topology to visually represent biological hallmarks from single-cell data. DS-O designed and built TopOMetry and conceived and performed all bioinformatic analyses. Both authors wrote the manuscript and agreed to the final submitted version.

Competing interests

The authors have no competing interests.

References

1. Shapiro, E., Biezuner, T. & Linnarsson, S. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nat. Rev. Genet.* **14**, 618–630 (2013).
2. Potter, S. S. Single-cell RNA sequencing for the study of development, physiology and disease. *Nat. Rev. Nephrol.* **14**, 479–492 (2018).
3. Ma, Y. & Zhu, L. A review on dimension reduction. *Int. Stat. Rev.* **81**, 134–150 (2013).
4. Anowar, F., Sadaoui, S. & Selim, B. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE).

Comput. Sci. Rev. **40**, 100378 (2021).

5. Xiang, R. *et al.* A Comparison for Dimensionality Reduction Methods of Single-Cell RNA-seq Data. *Front. Genet.* **12**, (2021).
6. F.R.S, K. P. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **2**, 559–572 (1901).
7. Jolliffe, I. T. Principal Component Analysis and Factor Analysis. in *Principal Component Analysis* (ed. Jolliffe, I. T.) 115–128 (Springer, 1986). doi:10.1007/978-1-4757-1904-8_7.
8. Belkin, M. & Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.* **15**, 1373–1396 (2003).
9. Coifman, R. R. *et al.* Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proc. Natl. Acad. Sci.* **102**, 7426–7431 (2005).
10. Coifman, R. R. & Lafon, S. Diffusion maps. *Appl. Comput. Harmon. Anal.* **21**, 5–30 (2006).
11. Reuter, M., Wolter, F.-E., Shenton, M. & Niethammer, M. Laplace–Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. *Comput.-Aided Des.* **41**, 739–755 (2009).
12. Brand, M. Continuous nonlinear dimensionality reduction by kernel eigenmaps. in *Proceedings of the 18th international joint conference on Artificial intelligence* 547–552 (Morgan Kaufmann Publishers Inc., 2003).
13. Berry, T. & Sauer, T. Consistent manifold representation for topological data analysis. *Found. Data Sci.* **1**, 1 (2019).
14. Maaten, L. van der & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
15. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv* (2018).

16. Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.* **37**, 38–44 (2019).
17. Wang, Y., Huang, H., Rudin, C. & Shaposhnik, Y. Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, and PaCMAP for Data Visualization. *ArXiv201204456 Cs Stat* (2021).
18. Agrawal, A., Ali, A. & Boyd, S. Minimum-Distortion Embedding. *ArXiv210302559 Cs Math Stat* (2021).
19. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
20. Huang, S. The molecular and mathematical basis of Waddington's epigenetic landscape: A framework for post-Darwinian biology? *BioEssays* **34**, 149–157 (2012).
21. 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).
22. A comparison of single-cell trajectory inference methods | Nature Biotechnology. https://www.nature.com/articles/s41587-019-0071-9.
23. Wagner, A., Regev, A. & Yosef, N. Revealing the vectors of cellular identity with single-cell genomics. *Nat. Biotechnol.* **34**, 1145–1160 (2016).
24. Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol. Syst. Biol.* **15**, e8746 (2019).
25. La Manno, G. *et al.* RNA velocity of single cells. *Nature* **560**, 494–498 (2018).
26. Bergen, V., Lange, M., Peidli, S., Wolf, F. A. & Theis, F. J. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nat. Biotechnol.* **38**, 1408–1414 (2020).
27. Stuart, T. *et al.* Comprehensive Integration of Single-Cell Data. *Cell* **177**, 1888-1902.e21 (2019).

28. Polański, K. *et al.* BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics* btz625 (2019) doi:10.1093/bioinformatics/btz625.

29. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* **37**, 547–554 (2019).

30. Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nat. Methods* **13**, 845–848 (2016).

31. Gayoso, A. *et al.* A Python library for probabilistic analysis of single-cell omics data. *Nat. Biotechnol.* **40**, 163–166 (2022).

32. Sun, S., Zhu, J., Ma, Y. & Zhou, X. Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. *Genome Biol.* **20**, 269 (2019).

33. Heiser, C. N. & Lau, K. S. A Quantitative Framework for Evaluating Single-Cell Data Structure Preservation by Dimensionality Reduction Techniques. *Cell Rep.* **31**, 107576 (2020).

34. Hadi, A. S. & Ling, R. F. Some Cautionary Notes on the Use of Principal Components Regression. *Am. Stat.* **52**, 15–19 (1998).

35. Jolliffe, I. T. & Cadima, J. Principal component analysis: a review and recent developments. *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.* **374**, 20150202 (2016).

36. Perraul-Joncas, D. & Meila, M. Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery. Preprint at http://arxiv.org/abs/1305.7255 (2013).

37. Tenenbaum, J. B., Silva, V. de & Langford, J. C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **290**, 2319–2323 (2000).

38. Lee, J. M. *Riemannian manifolds: an introduction to curvature*. (Springer, 1997).

39. Benkő, Z. *et al.* Manifold-adaptive dimension estimation revisited. *PeerJ Comput. Sci.* **8**,

e790 (2022).

40. Venna, J. & Kaski, S. Neighborhood Preservation in Nonlinear Projection Methods: An Experimental Study. in *Artificial Neural Networks — ICANN 2001* (eds. Dorffner, G., Bischof, H. & Hornik, K.) 485–491 (Springer, 2001). doi:10.1007/3-540-44668-0_68.
41. Amid, E. & Warmuth, M. K. TriMap: large-scale dimensionality reduction using triplets. *arXiv* (2019).
42. Kobak, D. & Linderman, G. C. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nat. Biotechnol.* **39**, 156–157 (2021).
43. Moon, K. R. *et al.* Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.* **37**, 1482–1492 (2019).
44. Fukunaga, K. *Introduction to Statistical Pattern Recognition*. (Academic Press, 1990).
45. Camastra, F. & Staiano, A. Intrinsic dimension estimation: Advances and open problems. *Inf. Sci.* **328**, 26–41 (2016).
46. Campadelli, P., Casiraghi, E., Ceruti, C. & Rozza, A. Intrinsic Dimension Estimation: Relevant Techniques and a Benchmark Framework. *Math. Probl. Eng.* **2015**, e759567 (2015).
47. Levina, E. & Bickel, P. Maximum Likelihood Estimation of Intrinsic Dimension. in *Advances in Neural Information Processing Systems* (eds. Saul, L., Weiss, Y. & Bottou, L.) vol. 17 (MIT Press, 2004).
48. Bastidas-Ponce, A. *et al.* Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development* **146**, dev173849 (2019).
49. Arnes, L., Hill, J. T., Gross, S., Magnuson, M. A. & Sussel, L. Ghrelin Expression in the Mouse Pancreas Defines a Unique Multipotent Progenitor Population. *PLOS ONE* **7**, e52026 (2012).

50. Cao, J. *et al.* The single-cell transcriptional landscape of mammalian organogenesis. *Nature* **566**, 496–502 (2019).

51. Domínguez Conde, C. *et al.* Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science* **376**, eabl5197 (2022).

52. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15–15 (2018).

53. Ding, J. *et al.* Characterisation of CD4+ T-cell subtypes using single cell RNA sequencing and the impact of cell number and sequencing depth. *Sci. Rep.* **10**, 19825 (2020).

54. Greene, E. *et al.* New interpretable machine-learning method for single-cell data reveals correlates of clinical response to cancer immunotherapy. *Patterns* **2**, (2021).

55. Waickman, A. T. *et al.* Temporally integrated single cell RNA sequencing analysis of PBMC from experimental and natural primary human DENV-1 infections. *PLOS Pathog.* **17**, e1009240 (2021).

56. Nehar-Belaid, D. *et al.* Mapping systemic lupus erythematosus heterogeneity at the single-cell level. *Nat. Immunol.* **21**, 1094–1106 (2020).

57. Wauters, E. *et al.* Discriminating mild from critical COVID-19 by innate and adaptive immune single-cell profiling of bronchoalveolar lavages. *Cell Res.* **31**, 272–290 (2021).

58. Huang, Z. *et al.* Effects of sex and aging on the immune cell landscape as assessed by single-cell transcriptomic analysis. *Proc. Natl. Acad. Sci.* **118**, (2021).

59. Vinh, N. X., Epps, J. & Bailey, J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance.

60. Sturm, G. *et al.* Scirpy: a Scanpy extension for analyzing single-cell T-cell receptor-sequencing data. *Bioinformatics* **36**, 4817–4818 (2020).

61. Bagaev, D. V. *et al.* VDJdb in 2019: database extension, new analysis infrastructure and a

T-cell receptor motif compendium. *Nucleic Acids Res.* **48**, D1057–D1062 (2020).

62. von Rohrscheidt, J. & Rieck, B. Topological Singularity Detection at Multiple Scales. Preprint at http://arxiv.org/abs/2210.00069 (2023).
63. Chari, T. & Pachter, L. The Specious Art of Single-Cell Genomics. 2021.08.25.457696 Preprint at https://doi.org/10.1101/2021.08.25.457696 (2022).
64. A. F. Duque, S. Morin, G. Wolf, & K. R. Moon. Geometry Regularized Autoencoders. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 7381–7394 (2023).
65. Luecken, M. D. & Theis, F. J. Current best practices in single‐cell RNA‐seq analysis: a tutorial. *Mol. Syst. Biol.* **15**, e8746 (2019).
66. Ntranos, V., Yi, L., Melsted, P. & Pachter, L. A discriminative learning approach to differential expression analysis for single-cell RNA-seq. *Nat. Methods* **16**, 163–166 (2019).
67. Buitinck, L. *et al.* API design for machine learning software: experiences from the scikit-learn project. in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* 108–122 (2013).
68. Berry, T. & Harlim, J. Variable bandwidth diffusion kernels. *Appl. Comput. Harmon. Anal.* **40**, 68–96 (2016).
69. Spivak, D. I. METRIC REALIZATION OF FUZZY SIMPLICIAL SETS. (2012).
70. Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nat. Methods* **13**, 845–848 (2016).
71. Setty, M. *et al.* Characterization of cell fate probabilities in single-cell data with Palantir. *Nat. Biotechnol.* **37**, 451–460 (2019).
72. Yang, L. & Jin, R. Distance Metric Learning: A Comprehensive Survey. 51.
73. Skinnider, M. A., Squair, J. W. & Foster, L. J. Evaluating measures of association for single-cell transcriptomics. *Nat. Methods* **16**, 381–386 (2019).

74. Kim, T. *et al.* Impact of similarity metrics on single-cell RNA-seq data clustering. *Brief. Bioinform.* **20**, 2316–2326 (2019).

75. Aumüller, M., Bernhardsson, E. & Faithfull, A. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Inf. Syst.* **87**, 101374 (2020).

76. Malkov, Y. A. & Yashunin, D. A. *Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs*. (2018).

77. Boytsov, L. & Naidan, B. Engineering Efficient and Effective Non-metric Space Library. in *Similarity Search and Applications - 6th International Conference, SISAP 2013, A Coruña, Spain, October 2-4, 2013, Proceedings* (eds. Brisaboa, N. R., Pedreira, O. & Zezula, P.) vol. 8199 280–293 (Springer, 2013).

78. Yip, S. H., Sham, P. C. & Wang, J. Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data. *Brief. Bioinform.* **20**, 1583–1589 (2018).

79. Berry, T. & Sauer, T. Local kernels and the geometric structure of data. *Appl. Comput. Harmon. Anal.* **40**, 439–469 (2016).

80. Berry, T. & Sauer, T. Consistent Manifold Representation for Topological Data Analysis. *ArXiv160602353 Math* (2019).

81. Nadler, B., Lafon, S., Coifman, R. R. & Kevrekidis, I. G. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harmon. Anal.* **21**, 113–127 (2006).

82. Wang, Y., Huang, H., Rudin, C. & Shaposhnik, Y. *Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, and PaCMAP for Data Visualization*. https://github.com/YingfanWang/PaCMAP.

83. Cook, J., Sutskever, I., Mnih, A. & Hinton, G. Visualizing Similarity Data with a Mixture of Maps. in *Proceedings of the Eleventh International Conference on Artificial Intelligence and*

Statistics 67–74 (PMLR, 2007).

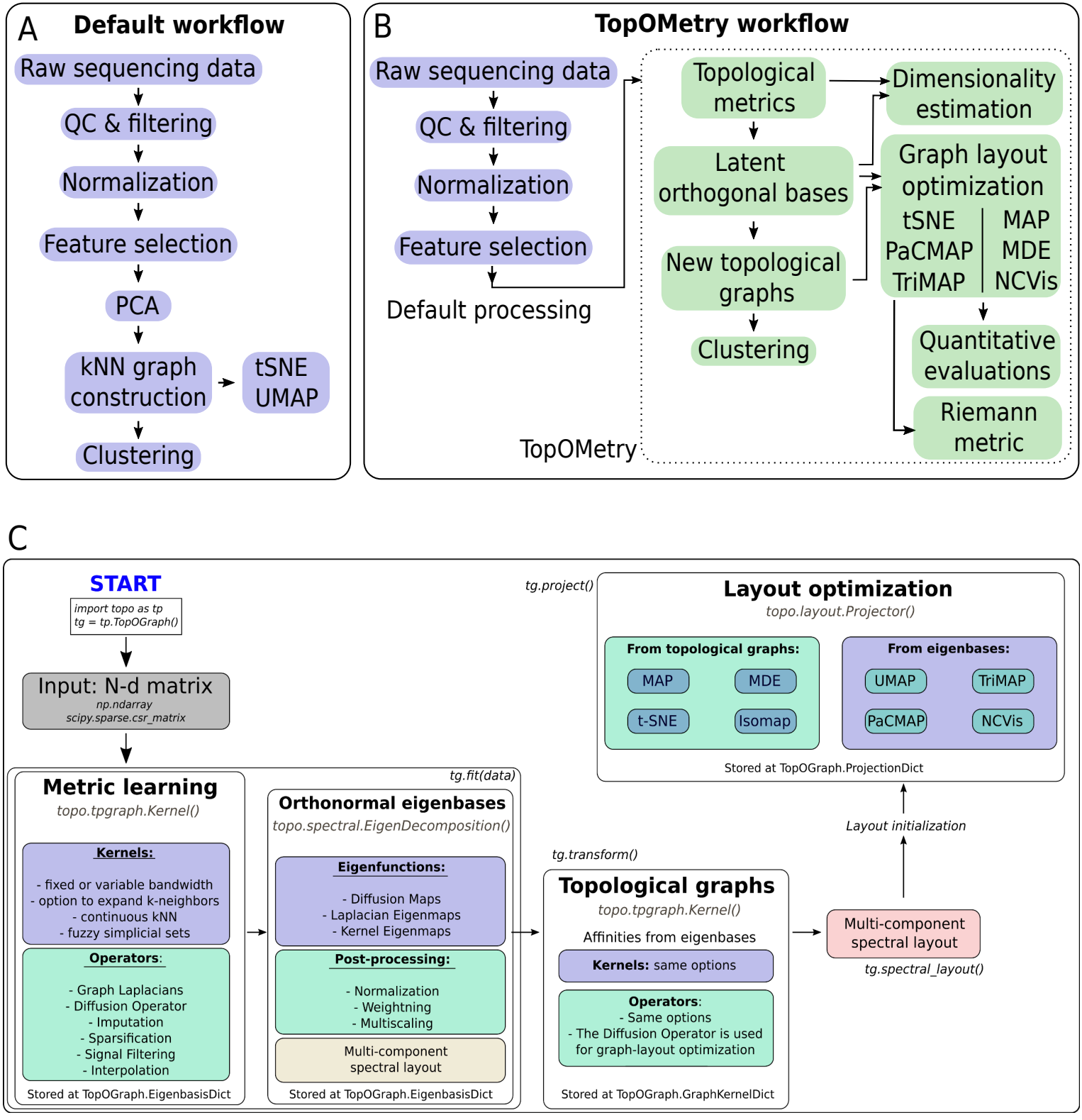
84. Ulyanov, D. Multicore t-SNE. (2021).
85. Maaten, L. van der. Accelerating t-SNE using Tree-Based Algorithms. *J. Mach. Learn. Res.* **15**, 3221–3245 (2014).
86. Farahmand, A. M., Szepesvári, C. & Audibert, J.-Y. Manifold-adaptive dimension estimation. in *Proceedings of the 24th international conference on Machine learning* 265–272 (ACM, 2007). doi:10.1145/1273496.1273530.
87. Heiser, C. N. & Lau, K. S. A Quantitative Framework for Evaluating Single-Cell Data Structure Preservation by Dimensionality Reduction Techniques. *Cell Rep.* **31**, 107576 (2020).

Supplementary figures and legends

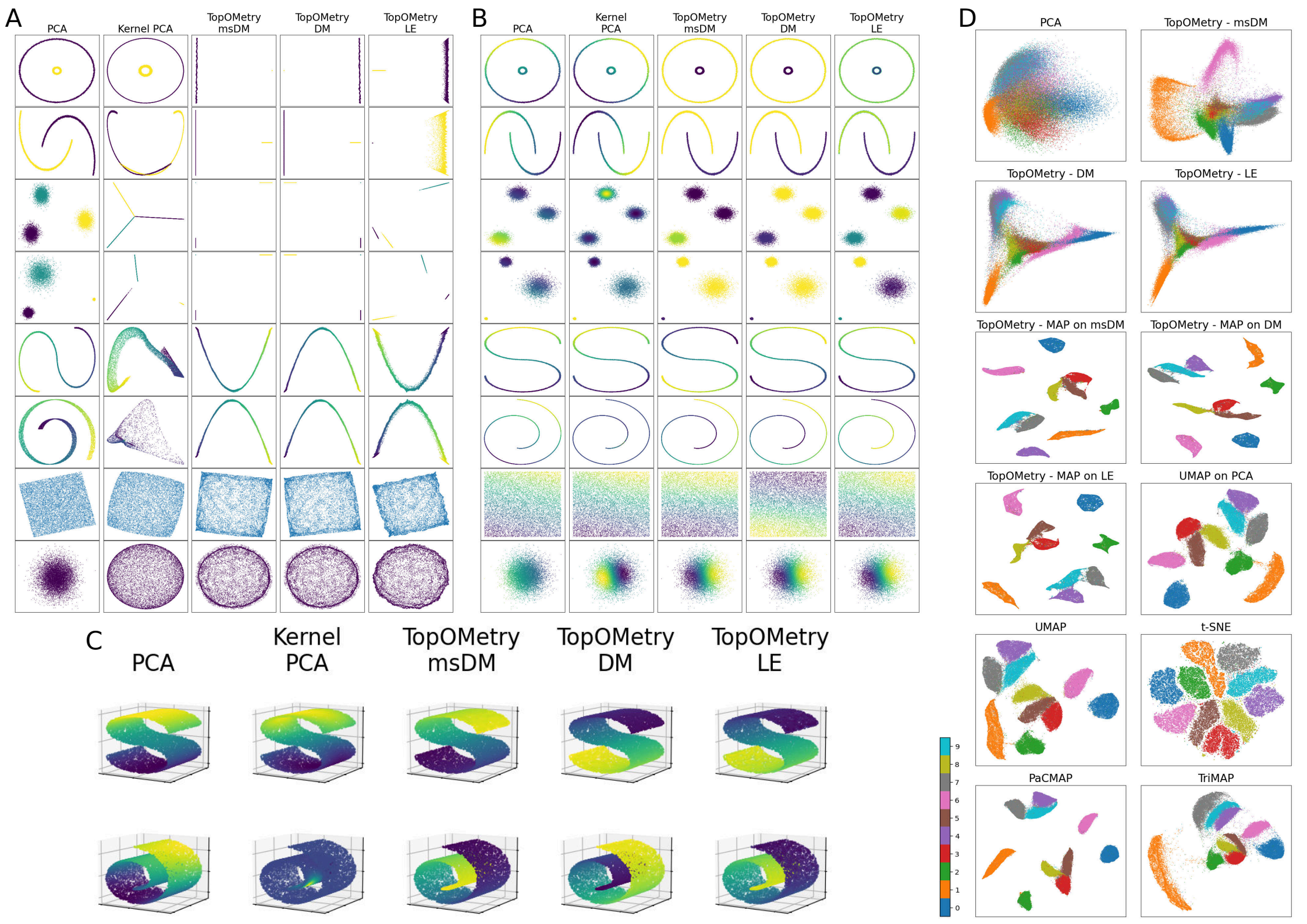
TopOMetry systematically learns and evaluates the latent dimensions of single-cell atlases

Davi Sidarta-Oliveira[1,2] and Licio A. Velloso[2]

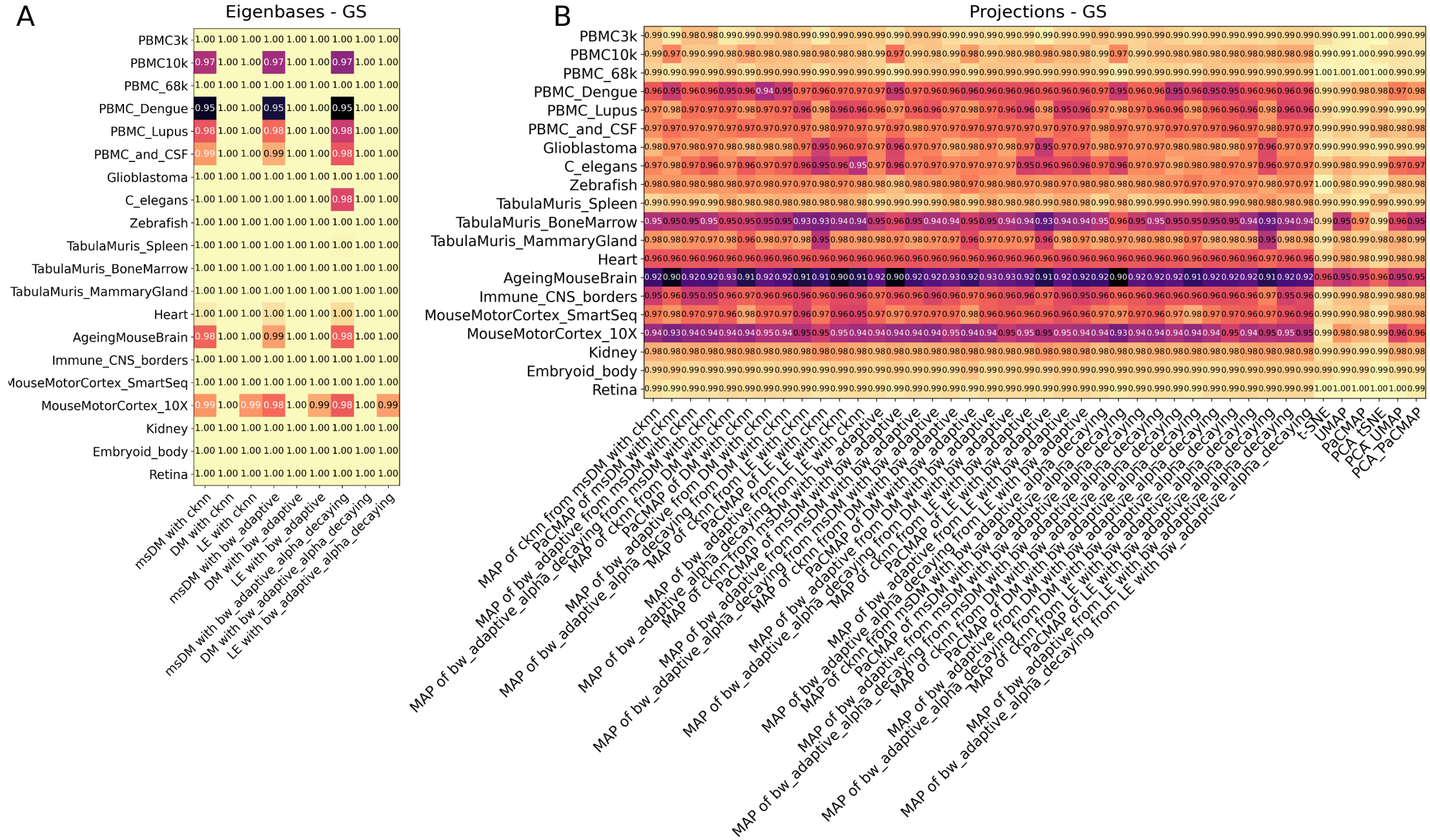
3. Department of Physiology, Anatomy and Genetics, University of Oxford, United Kingdom
4. Laboratory of Cell Signaling, Obesity and Comorbidities Research Center, School of Medical Sciences, University of Campinas, Brazil


A
Default workflow
Raw sequencing data
QC & filtering
Normalization
Feature selection
PCA
kNN graph construction
tSNE
UMAP
Clustering
B
TopOMetry workflow
Raw sequencing data
QC & filtering
Normalization
Feature selection
Default processing
Topological metrics
Latent orthogonal bases
New topological graphs
Clustering
Dimensionality estimation
Graph layout optimization
tSNE
PaCMAP
TriMAP
MAP
MDE
NCVis
Quantitative evaluations
Riemann metric
TopOMetry
C
START
import topo as tp
tg = tp.TopOGraph()
Input: N-d matrix
np.ndarray
scipy.sparse.csr_matrix
tg.fit(data)
Metric learning
topo.tpgraph.Kernel()
Kernels:
- fixed or variable bandwidth
- option to expand k-neighbors
- continuous kNN
- fuzzy simplicial sets
Operators:
- Graph Laplacians
- Diffusion Operator
- Imputation
- Sparsification
- Signal Filtering
- Interpolation
Stored at TopOGraph.EigenbasisDict
Orthonormal eigenbases
topo.spectral.EigenDecomposition()
Eigenfunctions:
- Diffusion Maps
- Laplacian Eigenmaps
- Kernel Eigenmaps
Post-processing:
- Normalization
- Weightning
- Multiscaling
Multi-component spectral layout
Stored at TopOGraph.EigenbasisDict
tg.transform()
Topological graphs
topo.tpgraph.Kernel()
Affinities from eigenbases
Kernels: same options
Operators:
- Same options
- The Diffusion Operator is used for graph-layout optimization
Stored at TopOGraph.GraphKernelDict
Multi-component spectral layout
tg.spectral_layout()
Layout initialization
tg.project()
Layout optimization
topo.layout.Projector()
From topological graphs:
MAP
MDE
t-SNE
Isomap
From eigenbases:
UMAP
TriMAP
PaCMAP
NCVis
Stored at TopOGraph.ProjectionDict

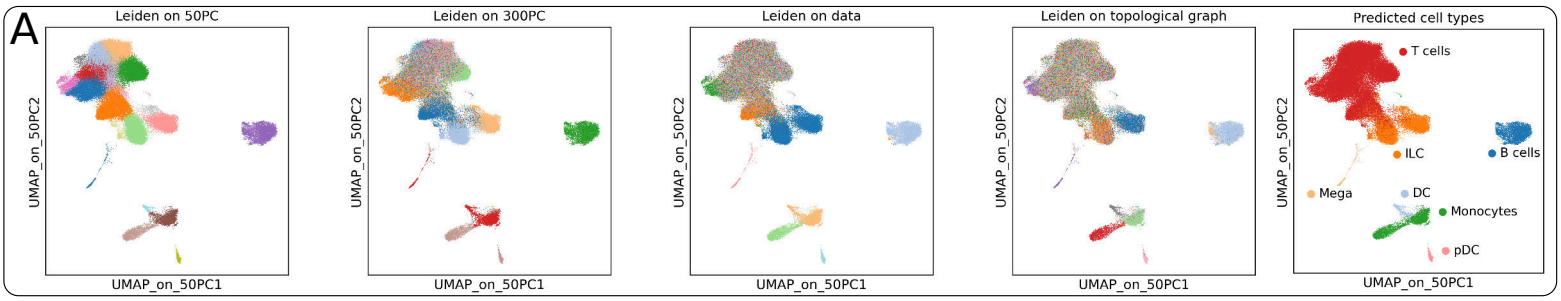
Supplementary Figure 1. Schematic overview of standard and TopOMetry workflows for single-cell analysis. (A) Schematic overview of the current standard workflow for single-cell analysis, starting from a high-dimensional matrix generated from raw sequencing data, that undergoes quality control (QC) and cell filtering and library-size normalization. High-variable genes are selected and used to compute PCA, from which the top principal components are used to find k-nearest-neighbors (kNN) graphs and affinity matrices that are used for clustering and projection into two-dimensional visualizations. (B) Schematic overview of the TopOMetry workflow, which consists of the same default processing steps, after which kNN graphs and specialized kernels are used to learn Laplacian-type operators that consist of topological affinity matrices. The eigendecomposition of such an operator yields eigenbases that preserve the latent topological information from the original high-dimensional manifold. The eigenbasis can then be used to learn new topological graphs using the same kernels and operators, which are used for clustering or learning two-dimensional projections for visualization through graph-layout optimization techniques. The learned projections are then evaluated for the preservation of global and local structure and the amount of introduced distortion. (C) Schematic overview of the computational implementation of TopOMetry, which is centered around the TopOGraph class.

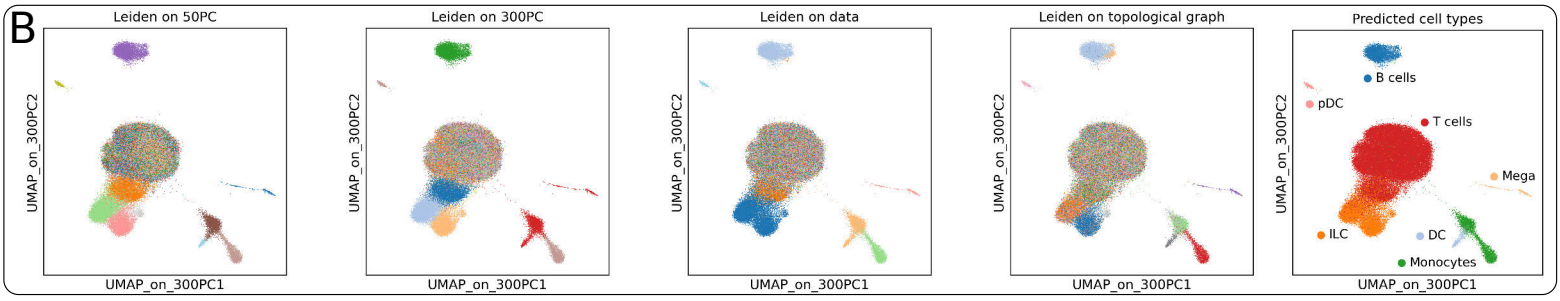

A
PCA
Kernel PCA
TopOMetry msDM
TopOMetry DM
TopOMetry LE
B
PCA
Kernel PCA
TopOMetry msDM
TopOMetry DM
TopOMetry LE
C
PCA
Kernel PCA
TopOMetry msDM
TopOMetry DM
TopOMetry LE
D
PCA
TopOMetry - msDM
TopOMetry - DM
TopOMetry - LE
TopOMetry - MAP on msDM
TopOMetry - MAP on DM
TopOMetry - MAP on LE
UMAP on PCA
UMAP
t-SNE
PaCMAP
TriMAP
9
8
7
6
5
4
3
2
1
0

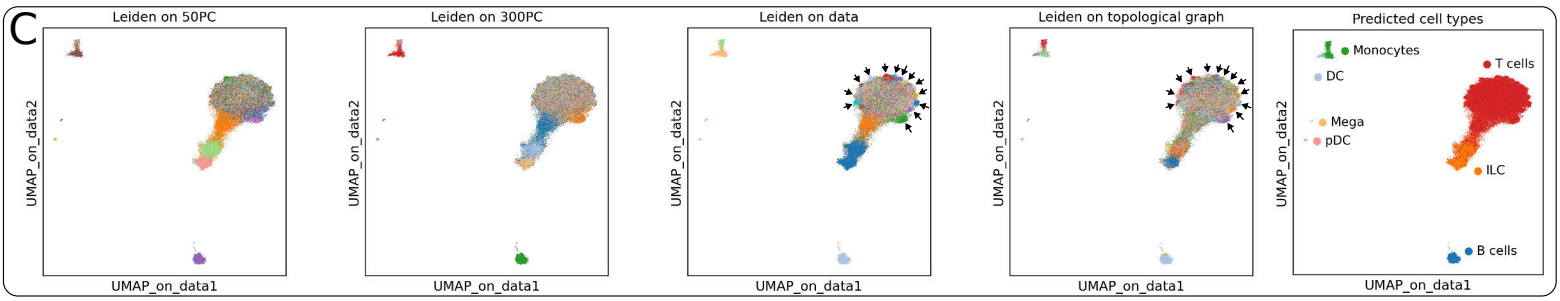
Supplementary Figure 2. Qualitative evaluation of TopOMetry and existing methods on synthetic and toy data. (A) Two-dimensional projections obtained with PCA, kernel PCA, and TopOMetry eigenbases for eight synthetic datasets: a circle inside a larger circle, two half moons, gaussian distributions with equal variances, gaussian distributions with different variances, the 'S' shape, the 'swiss-roll' shape, a uniformly sampled square and Gaussian noise. Sample points are colored by their original classes for the first four datasets and by their distribution on the fifth and sixth. Note how the eigenbases used in TopOMetry do not induce artificial clustering structure in the uniform square and the Gaussian noise, and successfully unfold the (B) The same projections, with samples colored by the value of the first component of the eigenbasis. Note how the first component of PCA and Kernel PCA fails to discriminate discrete submanifolds in the first two datasets and to unfold the 'swiss-roll' shape. (C) Three-dimensional visualization of the 'S' and 'swiss-roll' shapes, colored by the first component of each method. Note how the first component of PCA and Kernel PCA fails to discover the notoriously non-linear geometry. (D) Two-dimensional projections of the MNIST handwritten-digits dataset obtained with PCA, the eigenbases used in TopOMetry, a MAP projection of the diffusion potential of these eigenbases, UMAP on the first 100 principal components, UMAP on the data, t-SNE, PaCMAP and TriMAP. Samples (images) are colored by the digit they correspond to.

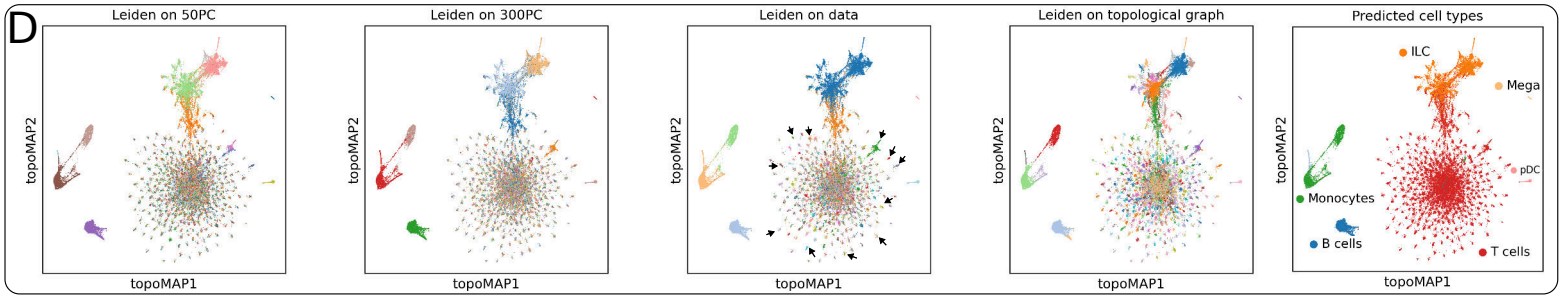

A
Eigenbases - GS
PBMC3k
PBMC10k
PBMC_68k
PBMC_Dengue
PBMC_Lupus
PBMC_and_CSF
Glioblastoma
C_elegans
Zebrafish
TabulaMuris_Spleen
TabulaMuris_BoneMarrow
TabulaMuris_MammaryGland
Heart
AgeingMouseBrain
Immune_CNS_borders
MouseMotorCortex_SmartSeq
MouseMotorCortex_10X
Kidney
Embryoid_body
Retina
msDM with cknn
DM with cknn
LE with cknn
msDM with bw_adaptive
DM with bw_adaptive
LE with bw_adaptive
msDM with bw_adaptive_alpha_decaying
DM with bw_adaptive_alpha_decaying
LE with bw_adaptive_alpha_decaying
B
Projections - GS
MAP of cknn from msDM with cknn
PaCMAP of msDM with cknn
MAP of bw_adaptive from msDM with cknn
MAP of bw_adaptive_alpha_decaying from msDM with cknn
MAP of cknn from DM with cknn
PaCMAP of DM with cknn
MAP of bw_adaptive from DM with cknn
MAP of bw_adaptive_alpha_decaying from DM with cknn
MAP of cknn from LE with cknn
PaCMAP of LE with cknn
MAP of bw_adaptive from LE with cknn
MAP of bw_adaptive_alpha_decaying from LE with cknn
MAP of cknn from msDM with bw_adaptive
PaCMAP of msDM with bw_adaptive
MAP of bw_adaptive from msDM with bw_adaptive
MAP of bw_adaptive_alpha_decaying from msDM with bw_adaptive
MAP of cknn from DM with bw_adaptive
PaCMAP of DM with bw_adaptive
MAP of bw_adaptive from DM with bw_adaptive
MAP of bw_adaptive_alpha_decaying from DM with bw_adaptive
MAP of cknn from LE with bw_adaptive
PaCMAP of LE with bw_adaptive
MAP of bw_adaptive from LE with bw_adaptive
MAP of bw_adaptive_alpha_decaying from LE with bw_adaptive
MAP of cknn from msDM with bw_adaptive_alpha_decaying
PaCMAP of msDM with bw_adaptive_alpha_decaying
MAP of bw_adaptive from msDM with bw_adaptive_alpha_decaying
MAP of bw_adaptive_alpha_decaying from msDM with bw_adaptive_alpha_decaying
MAP of cknn from DM with bw_adaptive_alpha_decaying
PaCMAP of DM with bw_adaptive_alpha_decaying
MAP of bw_adaptive from DM with bw_adaptive_alpha_decaying
MAP of bw_adaptive_alpha_decaying from DM with bw_adaptive_alpha_decaying
MAP of cknn from LE with bw_adaptive_alpha_decaying
PaCMAP of LE with bw_adaptive_alpha_decaying
MAP of bw_adaptive from LE with bw_adaptive_alpha_decaying
MAP of bw_adaptive_alpha_decaying from LE with bw_adaptive_alpha_decaying
t-SNE
UMAP
PaCMAP
PCA_tSNE
PCA_UMAP
PCA_PaCMAP

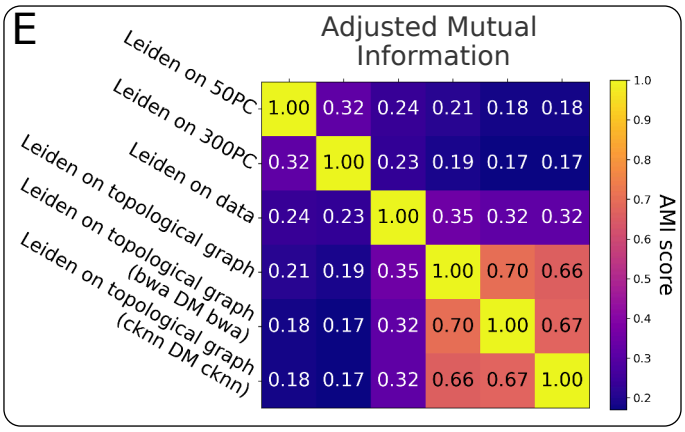
Supplementary Figure 3. Preservation of global structure across eigenbases and projections from single-cell datasets. (A) Global score for TopOMetry eigenbases for each of the 20 datasets used in the benchmark, with varying kernels. (B) Global score for projections obtained with t-SNE, UMAP, and PaCMAP with or without PCA preprocessing and with TopOMetry.

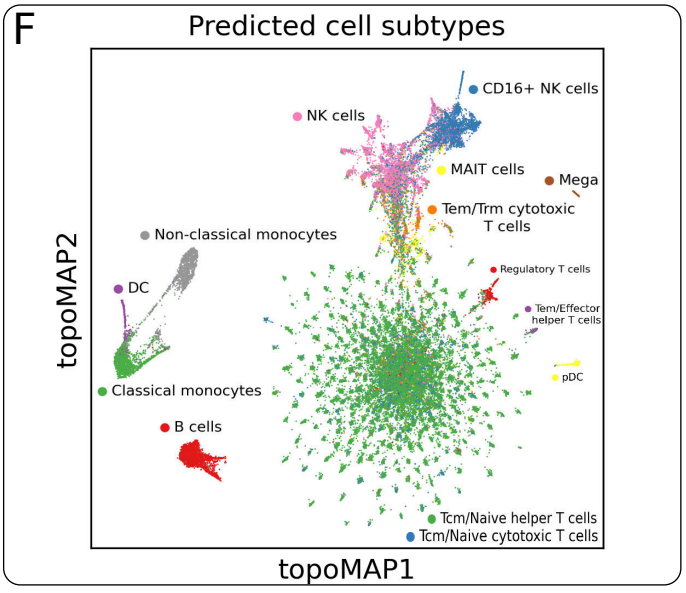

A
Leiden on 50PC
Leiden on 300PC
Leiden on data
Leiden on topological graph
Predicted cell types
UMAP_on_50PC2
UMAP_on_50PC1
T cells
ILC
B cells
Mega
DC
Monocytes
pDC

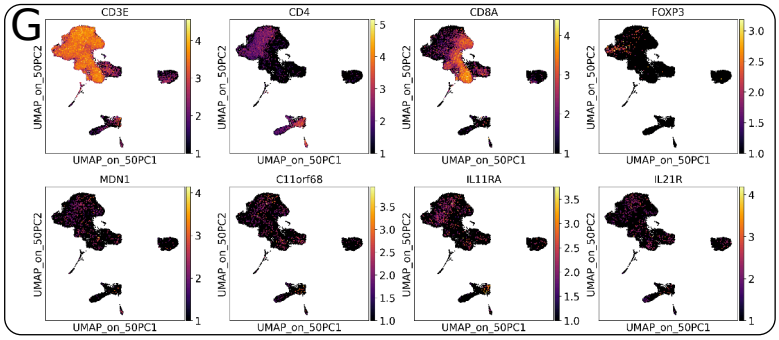

B
Leiden on 50PC
Leiden on 300PC
Leiden on data
Leiden on topological graph
Predicted cell types
UMAP_on_300PC2
UMAP_on_300PC1
B cells
pDC
T cells
Mega
ILC
DC
Monocytes

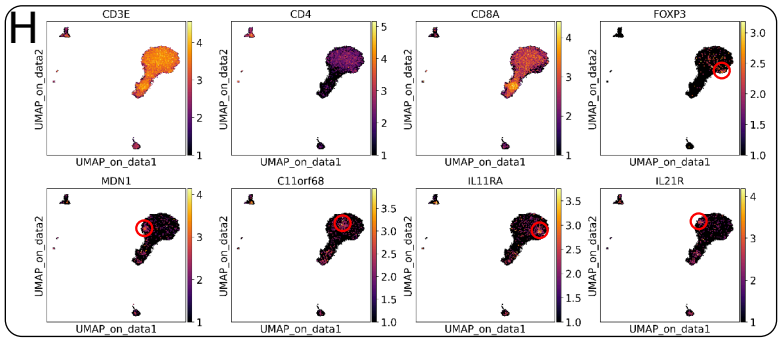

C
Leiden on 50PC
Leiden on 300PC
Leiden on data
Leiden on topological graph
Predicted cell types
UMAP_on_data2
UMAP_on_data1
Monocytes
DC
T cells
Mega
pDC
ILC
B cells

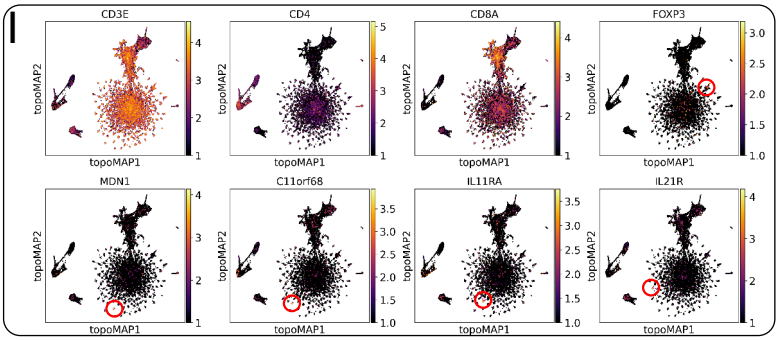

D
Leiden on 50PC
Leiden on 300PC
Leiden on data
Leiden on topological graph
Predicted cell types
topoMAP2
topoMAP1
ILC
Mega
pDC
Monocytes
B cells
T cells


E
Adjusted Mutual Information
Leiden on 50PC
Leiden on 300PC
Leiden on data
Leiden on topological graph
Leiden on topological graph (bwa DM bwa)
Leiden on topological graph (cknn DM cknn)
1.00 0.32 0.24 0.21 0.18 0.18
0.32 1.00 0.23 0.19 0.17 0.17
0.24 0.23 1.00 0.35 0.32 0.32
0.21 0.19 0.35 1.00 0.70 0.66
0.18 0.17 0.32 0.70 1.00 0.67
0.18 0.17 0.32 0.66 0.67 1.00
AMI score
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

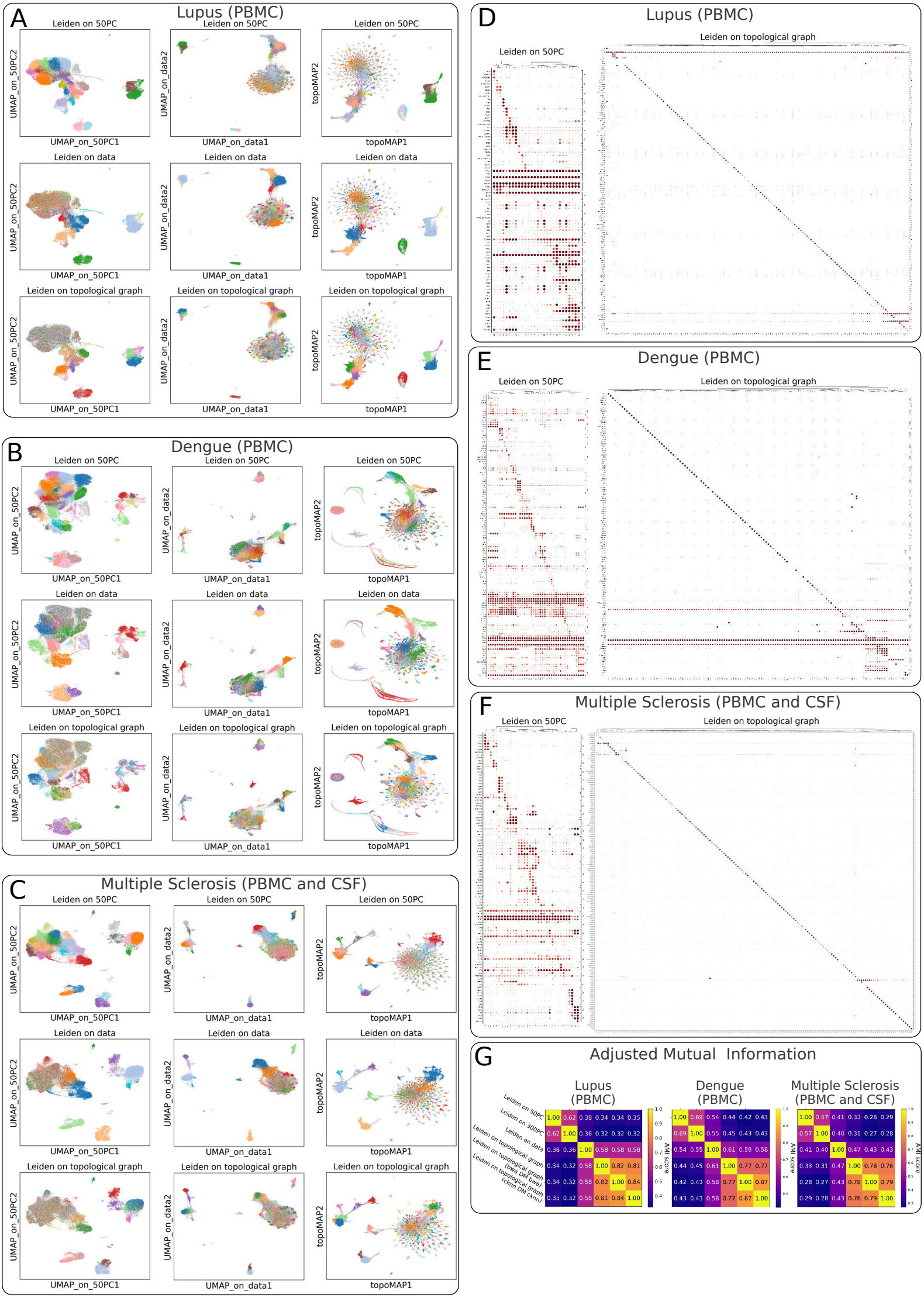

F
Predicted cell subtypes
CD16+ NK cells
NK cells
MAIT cells
Mega
Tem/Trm cytotoxic T cells
Non-classical monocytes
Regulatory T cells
DC
Tem/Effector helper T cells
pDC
Classical monocytes
B cells
Tcm/Naive helper T cells
Tcm/Naive cytotoxic T cells
topoMAP2
topoMAP1


G
CD3E
CD4
CD8A
FOXP3
MDN1
C11orf68
IL11RA
IL21R
UMAP_on_50PC2
UMAP_on_50PC1

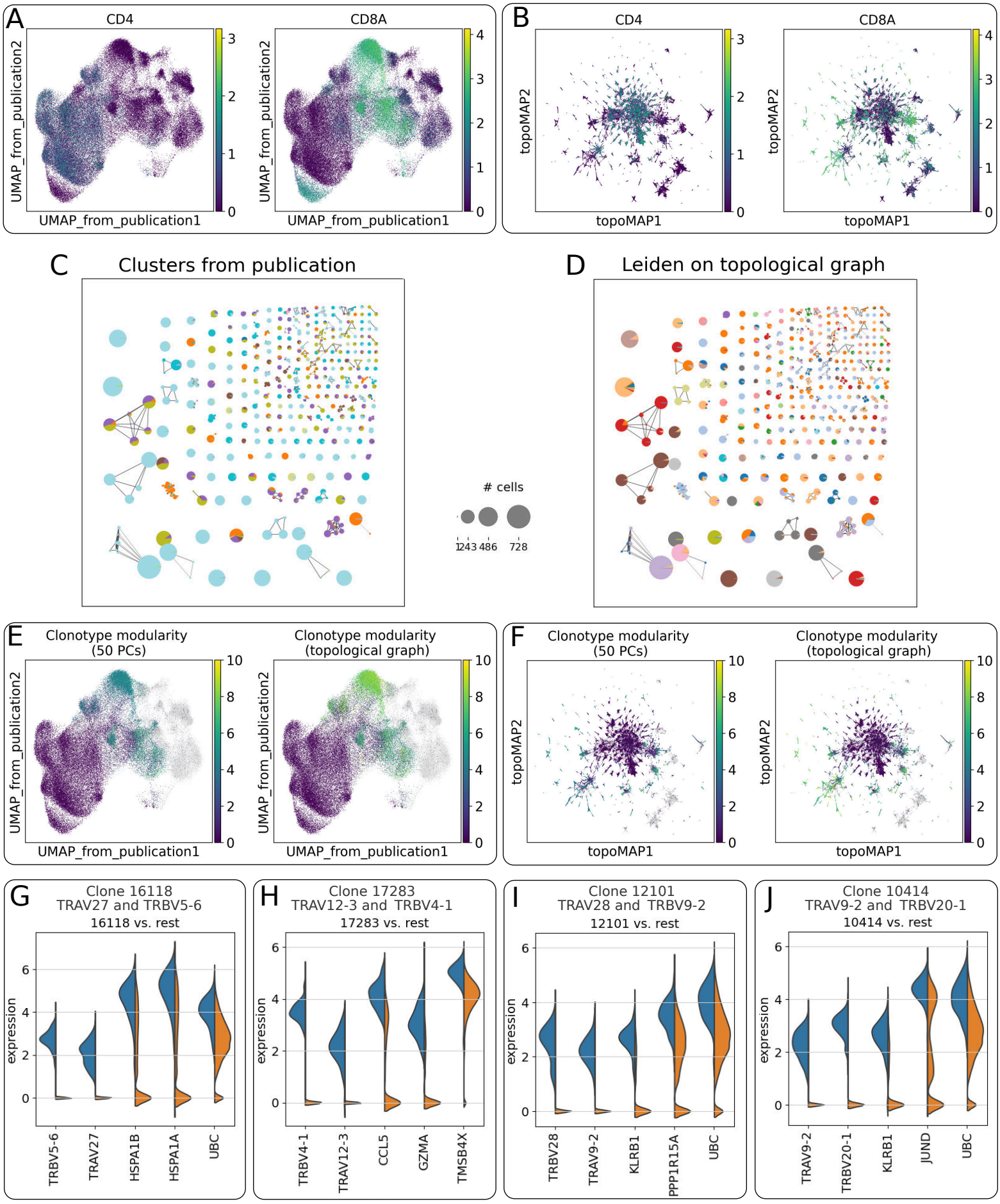

H
CD3E
CD4
CD8A
FOXP3
MDN1
C11orf68
IL11RA
IL21R
UMAP_on_data2
UMAP_on_data1


I
CD3E
CD4
CD8A
FOXP3
MDN1
C11orf68
IL11RA
IL21R
topoMAP2
topoMAP1

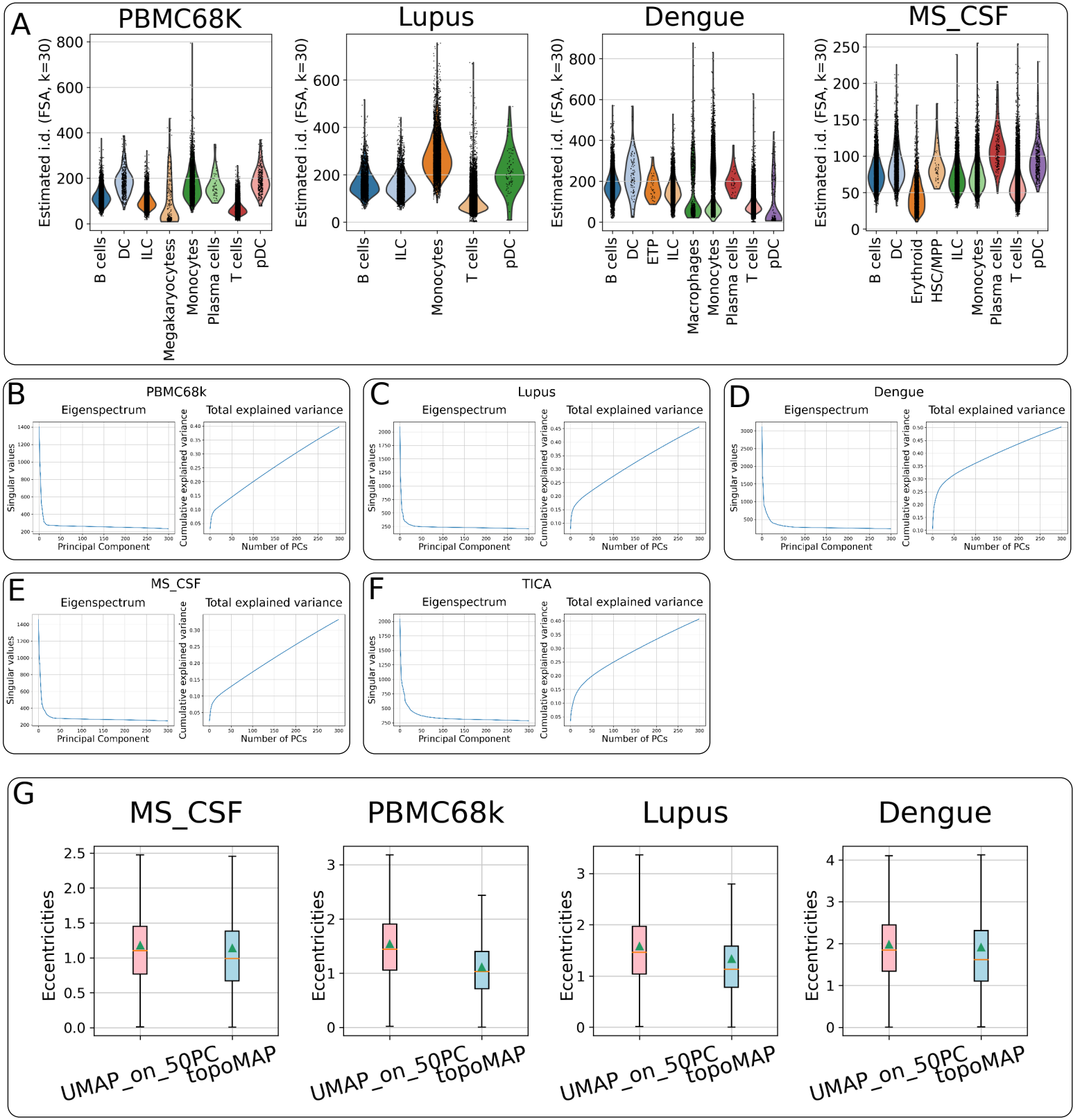
Supplementary Figure 4. Transcriptional diversity of T cells in a healthy human donor. (A) UMAP projections of the *PBMC68k* dataset obtained using the top 50 principal components (PCs), colored by clustering results obtained from the top 50 PCs, top 300 PCs, gene expression matrix, TopOMetry diffusion potential of the msDM eigenbasis and cell types predicted with CellTypist. (B) UMAP projections of the same data using the top 300 PCs. Note how clusters obtained with 50 PCs do not correspond to the structure uncovered using 300 PCs. (C) UMAP projections of the same data using the gene expression matrix of highly-variable genes. (D) topoMAP (MAP of the diffusion potential of the msDM eigenbasis) projection of the same data. Arrows in (C) and (D) highlight agreeing clustering results when using the gene expression matrix and TopOMetry, which can be detected on the periphery of the UMAP embedding of T cells. (E) Heatmap of adjusted mutual information (AMI) score between clustering results obtained with either approach, including additional TopOMetry eigenbases. The AMI score indicates the agreement between two clustering results when no ground-truth is known. Note how the results obtained with the high-dimensional expression data agree more with those obtained with TopOMetry than with those obtained with PCA. (F) topoMAP embedding of the same data, colored by cell subtypes predicted with CellTypist. Note how the majority of novel cell types uncovered correspond to CD4+ T and NK cells. (G) UMAP projection using the top 50 PCs, colored by gene expression of the canonical markers *CD3E*, *CD4, CD8A,* and *FOXP3,* and of marker genes for T cell subpopulations found with TopOMetry: *MDN1*, *C11*orf68, *IL11RA,* and *IL21R.* (H) UMAP projection using the expression data, colored as in (G), and with red circles highlighting the densely localized expression of marker genes of novel T cell subpopulations. (I) topoMAP projection of the same data, likewise colored, with red circles highlighting the densely localized expression of marker genes of novel T cell subpopulations.


A
Lupus (PBMC)
Leiden on 50PC
Leiden on data
Leiden on topological graph
UMAP_on_50PC1
UMAP_on_50PC2
UMAP_on_data1
UMAP_on_data2
topoMAP1
topoMAP2
B
Dengue (PBMC)
C
Multiple Sclerosis (PBMC and CSF)
D
Lupus (PBMC)
Leiden on topological graph
E
Dengue (PBMC)
F
Multiple Sclerosis (PBMC and CSF)
G
Adjusted Mutual Information
Lupus (PBMC)
Dengue (PBMC)
Multiple Sclerosis (PBMC and CSF)
Leiden on 50PC
Leiden on 300PC
Leiden on data
Leiden on topological graph (bwa DM bwa)
Leiden on topological graph (cknn DM cknn)
AMI score

Supplementary Figure 5. Transcriptional diversity of T cells across diseases. (A) Panel of projections of the *Lupus* dataset of peripheral mononuclear blood cells (PBMC) from healthy donors and Systemic Lupus Erythematosus patients, each obtained with UMAP on the first 50 principal components (PCs), UMAP on the expression data and TopOMetry, and colored by clustering results obtained with the first 50 PCs, expression data and TopOMetry. Each row corresponds to a clustering method and each column corresponds to a projection method. (B) Panel of projections of the *Dengue* dataset of PBMC from a dengue fever patient, obtained and colored as in (A). (C) Panel of projections of the *MS_CSF* dataset of mononuclear cells from the peripheral blood and cerebrospinal fluid (CSF) from healthy donors and multiple sclerosis patients, obtained and colored as in (A) and (B). Note how the findings observed in the *PBMC68k* dataset are replicated in a nearly identical fashion for all three datasets: clustering and projecting from expression data yield results that disagree with those obtained with the standard PCA-based approach and agree with those obtained with TopOMetry. The diffusion potential of the msDM eigenbasis was used for topoMAP projection and Leiden clustering in all datasets. (D) Dotplot of the top marker genes for clusters found with 50 PC and with TopOMetry on the *Lupus* dataset, with three genes shown per cluster for the former and one for the latter. (E) Same as in (D), but for the *Dengue* dataset. (F) Same as in (E) and (D), but for the *MS_CSF* dataset. Note how the finding that clusters of T cells found with the standard PCA-based approach present poor marker genes is replicated for all datasets, in sharp contrast to the highly specific marker genes presented by the clusters found with TopOMetry. (G) Heatmaps of adjusted mutual information (AMI) scores indicating agreement between different clustering strategies for the *Lupus*, *Dengue* and *MS_CSF* datasets, including additional TopOMetry eigenbases. Clustering results obtained with expression data disagree with those obtained using PCA and agree with those obtained using TopOMetry.


A
CD4
CD8A
UMAP_from_publication1
UMAP_from_publication2
B
topoMAP1
topoMAP2
C
Clusters from publication
D
Leiden on topological graph
cells
1243 486 728
E
Clonotype modularity (50 PCs)
Clonotype modularity (topological graph)
F
G
Clone 16118
TRAV27 and TRBV5-6
16118 vs. rest
expression
TRBV5-6
TRAV27
HSPA1B
HSPA1A
UBC
H
Clone 17283
TRAV12-3 and TRBV4-1
17283 vs. rest
TRBV4-1
TRAV12-3
CCL5
GZMA
TMSB4X
I
Clone 12101
TRAV28 and TRBV9-2
12101 vs. rest
TRBV28
TRAV9-2
KLRB1
PPP1R15A
UBC
J
Clone 10414
TRAV9-2 and TRBV20-1
10414 vs. rest
TRAV9-2
TRBV20-1
KLRB1
JUND
UBC

Supplementary Figure 6. Transcriptional diversity of T cell clonotypes. (A) PCA-based UMAP projection of T cells from the TICA dataset, as in the original study, colored by expression of *CD4* and *CD8A*. (B) topoMAP (MAP of the diffusion potential of the msDM eigenbasis) of the TICA dataset, colored by expression of *CD4* and *CD8A*. (C) Clonotype network for T cells with a detectable TCR in this dataset, based on amino acid sequence similarity. Each dot represents an individual clonotype, and its size represents the number of cells belonging to that clonotype. Dots are colored by the clusters from the original study. Note how different clones are grouped together under the same cluster assignment. (D) Same as in (C), but with dots colored by the clusters found with TopOMetry. Note how most clonotypes belong to a single cluster assignment, without overlap. (E) PCA-based UMAP and (F) topoMAP projections of the same data, colored by clonotype modularity. Clonotype modularity was calculated using a neighborhood graph learned from the top 50 principal components (PCs) or the diffusion potential graph. Note how modularities are higher with the latter, indicating a better connection between the manifold learned from RNA-seq and the clonotypes learned from VDJ-seq. (G-J) Violin plots of comparative gene expression between cells belonging to a clone and the rest, detected akin to marker genes using the Wilcoxon test, showing gene expression of the receptor chains that define clonotypes identified by VDJ-seq.


A
PBMC68K
Lupus
Dengue
MS_CSF
Estimated i.d. (FSA, k=30)
B cells
DC
ILC
Megakaryocytes
Monocytes
Plasma cells
T cells
pDC
ETP
Macrophages
Erythroid
HSC/MPP
B
PBMC68k
C
Lupus
D
Dengue
E
MS_CSF
F
TICA
Eigenspectrum
Total explained variance
Singular values
Principal Component
Cumulative explained variance
Number of PCs
G
Eccentricities
UMAP_on_50PC
topoMAP

Supplementary Figure 7. PCA introduces linear distortion into the manifold region inhabited by T cells. (A) Violin plots of intrinsic dimensionality (i.d.) estimates for the *PBMC68k, Lupus, Dengue,* and *MS_CSF* datasets, grouped by cell types predicted with CellTypist. Note how T cells do not present particularly high i.d. Estimates. (B-F) Eigenspectrum (left) and cumulative explained variance (right) of PCA for the *PBMC68k, Lupus, Dengue, MS_CSF,* and TICA datasets. An *ad hoc* 'elbow point' is found for all datasets around 30 principal components (PCs), at which PCA explains less than 15% of the data covariance. Such feature is a known hallmark of non-linear data. (G) Boxplot of eccentricities of the ellipses representing the Riemannian metric to evaluate distortions, obtained from T cells from the *PBMC68k, Lupus, Dengue,* and *MS_CSF* datasets. For all datasets, the eccentricities (an indirect measure of induced distortion) were significantly lower in the topoMAP projection when compared to UMAP on the first 50 PCs ($p < 10^{-90}$, two-sided Wilcoxon test).

Dataset name	Number of cells	Number of HVG	Species	Assay	Tissue	Source
Embryoid_body	16,825	2,122	Homo sapiens	10x 3' v3	Embryoid body	https://doi.org/10.1038/s41587-019-0336-3
Heart	60,668	7,987	Homo sapiens	10x 3' v2 and v3	Heart	https://doi.org/10.1038/s44161-022-00183-w
Kidney	39,176	5,156	*Homo sapiens*	10x 3' v3 10x 5' v1	Kidney	https://doi.org/10.1038/s41467-022-32972-z
Immune_CNS_borders	130,725	1,746	*Homo sapiens*	10x 3' v2 and v3 10x 5' v1	Several	https://doi.org/10.1016/j.coi.2022.102188
MouseMotorCortex_10X	71,183	2,149	*Mus musculus*	10x 3' v3	Brain	https://doi.org/10.1038/s41586-021-03500-8
MouseMotorCortex_SmartSeq	6,288	8,710	*Mus musculus*	Smart-seq		
Retina	20,091	6,397	*Homo sapiens*	10x 3' v3	Macula lutea and peripheral region of retina	https://doi.org/10.1038/s41467-019-12780-8
PBMC3K	2,700	1,838	*Homo sapiens*	10x 3' v1	Peripheral mononuclear cells	10X Genomics https://www.10xgenomics.com/resources/datasets/
PBMC10K	10,119	5,175	*Homo sapiens*	10x 3' v3.1		
PBMC_68k	66,834	2,318	*Homo sapiens*	10x 3' v1		
PBMC_Lupus	53,170	1,641	*Homo sapiens*	10x 3' v2		
PBMC_Dengue	72,412	1,924	*Homo sapiens*	10x 3' v2		https://doi.org/10.1371%2Fjournal.ppat.1009240
PBMC_and_CSF	65,326	1,361	*Homo sapiens*	10x 3' v1	Peripheral mononuclear cells and cerebrospinal fluid	https://doi.org/10.1038/s41467-019-14118-w
Glioblastoma	4,043	7,613	*Homo sapiens*	10x 5' v1	Glioblastoma multiforme	10X Genomics https://www.10xgenomics.com/resources/datasets/
C_elegans	2,800	943	*Caenorhabditis elegans*	sci-RNA-seq	C. elegans embryos	https://doi.org/10.1126%2Fscience.aam8940
Zebrafish	63,530	1,416	*Danio rerio*	inDrops	Zebrafish development	https://doi.org/10.1126/science.aar4362
TabulaMuris_BoneMarrow	3,652	2,698	*Mus musculus*	DropSeq	Mouse bone marrow	https://doi.org/10.1038/s41586-018-0590-4
TabulaMuris_Spleen	9,552	1,275	*Mus musculus*	DropSeq	Mouse spleen	
TabulaMuris_Mammary_gland	4,481	3,224	*Mus musculus*	DropSeq	Mouse mammary gland	

Table 1. Public single-cell RNA-seq datasets that were used in this study benchmark.