Interpretable models for scRNA-seq data embedding with multi-scale structure preservation

David Novak[1,2*], Cyril de Bodt[3,4], Pierre Lambert[5], John A. Lee[6], Sofie Van Gassen[1,2] and Yvan Saeys[1,2]

[1]Department of Mathematics, Computer Science and Statistics, Ghent University, Krijgslaan 281-S9, Ghent, 9000, Oost-Vlaanderen, Belgium.
[2]Data Mining and Modeling for Biomedicine, Center for Inflammation Research, VIB-UGent, Technologiepark-Zwijnaarde 71, Ghent, 9052, Oost-Vlaanderen, Belgium.
[3]Mathematics Department, Université de Namur, Rue Joseph Grafé 2, Namur, 5000, Namur, Belgium.
[4]Namur Institute for Complex Systems (naXys), Université de Namur, Rue Joseph Grafé 2, Namur, 5000, Namur, Belgium.
[5]ICTEAM/ELEN, Université catholique de Louvain, Place du Levant 3 bte L5.03.02, Louvain-la-Neuve, 1348, Brabant wallon, Belgium.
[6]IREC/MIRO, Université catholique de Louvain, Avenue Hippocrate 55 B1.54.07, Brussels, 1200, Brussels, Belgium.

*Corresponding author. E-mail: davidno@irc.vib-ugent.be;
Contributing authors: cyril.debodt@unamur.be; pierre.h.lambert@uclouvain.be; john.lee@uclouvain.be; sofievangassen@ugent.be; yvan.saeys@ugent.be;

Abstract

The ability to explore high-dimensional single-cell transcriptomics data efficiently is crucial in many biological studies. Dimensionality reduction techniques have therefore emerged as a basic building block of analytical workflows. They generate low-dimensional embeddings that capture important structures in the data, and are often used in discovery, quality control, and downstream analysis. However, the trustworthiness of current methods and the rigour of popular evaluation criteria are limited. We tackle this in an empirical study of structure-preserving data embeddings, delivering two new tools. First, we introduce *ViScore*: a robust scoring framework that improves both unsupervised and supervised quality metrics, with emphasis on scalability and fairness. Second, we introduce *ViVAE*: a deep learning model that achieves better multi-scale structure preservation and is equipped with new tools for interpretability. We demonstrate the potential of our framework to advance the trustworthiness of single-cell dimensionality reduction in a quantitative comparison and focused case studies.

Keywords: scRNA-seq, interpretability, dimensionality reduction, deep learning

Context

- This paper advances **empirical evaluation of dimensionality reduction (DR)** in single-cell transcriptomics and proposes a new DR model that achieves a **favourable balance between local and global structure preservation**. Both the evaluation methodology and the DR model are **readily applicable** to a broad range of single-cell data.
- Evaluation of single-cell DR methods has relied on very informal definitions of 'local' and 'global' structures. There has been **an emphasis on evaluation metrics that are straightforward to use in benchmarks** involving representative datasets (57; 65; 24). This has been in lieu of extensive descriptions of the underlying data manifolds, their intrinsic dimensionality, and dataset-specific topologies, which is infeasible with much of single-cell omics data.
- Mathematical descriptions of multi-scale structures in high-dimensional point clouds are possible within the topology domain (1). However, applying them to real biological data is difficult and can require considerable supervision and tuning (21). For this reason, **the present study focuses on broadly applicable evaluation strategies that make few assumptions about the datasets at hand**.
- The proposed DR model is shown to **separate distinct cell compartments and represent continuities and discontinuities well**, based on case studies with both single-snapshot and developmental single-cell transcripmtomics data.
- Although some DR methods are equipped with quality-control and explainability measures (14; 35), their general adoption has been limited. ***Encoder indicatrices*, a tool for visualising unwanted embedding distortions**, are presented here to advance the explainability of single-cell data embeddings.
- The **new scoring framework (*ViScore*) and DR model (*ViVAE*) are published as Python packages**, along with reproducible experiments, guided tutorials, and an automated benchmarking framework. Individual elements of both can be integrated into alternative models, contributing to advancement in the single-cell DR field.

Introduction

Data embedding using dimensionality reduction (DR) is common in the computational analysis of single-cell RNA-sequencing (scRNA-seq) data (4; 27). DR converts high-dimensional (HD) gene expression data to low-dimensional (LD) embeddings (Figure 1**A**). These embeddings are suitable for describing cell type diversity (27), quality control (QC) after data integration (54), or representating tentative trajectory structures (18). They generally improve the interpretability of single-cell datasets, which are sparse and noise-prone (16). DR is therefore crucial for effective exploratory data analysis in the single-cell bioinformatics domain.

Current methods differ considerably in the criteria they optimise. *Principal component analysis* (PCA) is a popular deterministic method that obtains ordered orthogonal vectors explaining the most variance in the data: *principal components* (PCs). To reduce dimensionality, a set number of top PCs is taken as the basis for a LD vector space, used for projection. PCA is widely applied to preprocess HD sparse data including scRNA-seq (23). It is typically used to generate 50- to 100-dimensional representations amenable to downstream processing, including further DR. Alternatively, the top 2 or 3 PCs can be taken directly as the embedding. However, this inherently limits the ability to capture complex non-linear relationships between features, as PCA only uses linear transformations.

Non-linear DR methods overcome this, with t-SNE (38) and *UMAP* (39) gaining special prominence in single-cell biology (as explored in (27; 68)). They both fall into the family of *neighbour embedding* (NE) algorithms, and can be formulated as optimising point positions via attractive and/or repulsive forces (8). NE algorithms aim to preserve local point neighbourhoods, and are suitable for visualising clusters and comparing alternative labellings. However, their inherent *'locality bias'* can distort the relative positions of cohesive cell clusters, introducing artificial gaps and arbitrary global layouts (36). By *locality bias*, we mean that local structures in the data (bounded by some *'scale'* parameter that governs size of the point neighbourhoods of interest), are optimised (see Figure 1**B**). While *scale* can be increased (*e.g.*, the *perplexity* parameter in t-SNE or neighbourhood count in *UMAP*), the limitation to local structures remains. Some initialisation strategies partially compensate for this, yielding better preservation of large-scale structures (27).

Current single-cell bioinformatics literature lacks formal definitions of the terms 'local' and 'global' that would lend themselves clearly to extensive comparative evaluation across many datasets. Our work here, while focused on application more than on theory, offers a conceptual framework for that dichotomy. By 'local structures' we refer to structures in some close neighbourhood of any given vantage point in the input data, with neighbourhood size governed by the scale parameter described above. Conversely, global structures are not limited by a scale parameter (Figure 1**C**), and can extend further across the point cloud under this conceptual framework. The term 'structure preservation' (SP) is then used to indicate a measure of how well the original HD structures are preserved in the LD embeddings (despite inherent constraints introduced by the reduced dimensionality).

A number of recent methods have set out to increase the preservation of global structures, or to find a favourable local-global balance. Some, including *TriMap* (3) and *PaCMAP* (66), are variants of NE. While NE methods are typically susceptible to partitioning data points into artificial 'islands' (Figure 1**D**), both *TriMap* and *PaCMAP* sample some points outside of local neighbourhoods in the optimisation process to improve the global layout of embeddings.

SQuad-MDS (30) is a recent adaptation of *multidimensional scaling* (MDS), which preserves relative HD pairwise distances between some points regardless of neighbourhood

or structure scales. In practice, this optimises the global layout of embeddings, similarly to PCA despite algorithmic differences (Figure 1**E**).

Models such as *scvis* (14) and *ivis* (57) are examples of variational autoencoders (VAEs; (26)), a flexible neural network architecture. VAEs embed HD data in a smooth latent space, favourable for representing continuous transitions between cell states (Figure 1**F**).

Overall, the trend of attempting to better balance local and global SP can reduce misinterpretations of t-SNE and *UMAP*, which are often applied to data with important global or hierarchical structures, despite not being optimised to represent them faithfully (60; 61). However, a major open problem remains. This is the issue of systematically evaluating the accuracy of embeddings. Recent comparisons apply both unsupervised and supervised scores (24; 65; 28), often via measuring performance in downstream inference taks, such as clustering or classification (41; 57). These approaches exhibit some shortcomings. First, when scoring local structure preservation (SP), setting a hard threshold for a 'local' neighbourhood (as in (65)) is potentially heavy-handed. Second, to assess global SP some metrics effectively downsample data, computing distances between representative points determined via clustering (65), thus sacrificing rigour. Third, evaluation of embeddings via downstream tasks (as in (24)) yields a proxy score, fundamentally dependent on the classification or clustering algorithm used.

More broadly, a recent critique in (10) questioned the usefulness of single-cell DR due to failures to preserve distances between points, to which (31) respond by demonstrating that t-SNE and *UMAP* preserve point neighbourhoods and separate biologically relevant clusters. This further underscores a need for clear and fair evaluation criteria and indicators of embedding trustworthiness.

Our *ViScore* framework addresses those concerns regarding evaluation and builds on the conceptual 'local-versus-global' distinction we drew above. *ViScore* introduces highly scalable unsupervised SP scoring with few assumptions on the data and embeddings (novel R_{NX} curve approximations), an improved qualitative supervised evaluation method (*Extended Neighbourhood-Proportion-Error*; xNPE), and a fully-fledged benchmarking framework, all of which we describe in this paper.

With an empirical evaluation framework in place, we set out to develop a DR model for representing both local and global structures faithfully. *ViVAE*, our flexible deep learning-based method, exhibits a favourable balance between the two criteria. Furthermore, it incorporates *encoder indicatrices* (EIs): a visual aid for describing unwanted distortions in existing embeddings. Grounded in previous work within differential geometry (44), EIs serve as a straightforward QC check that is model- and dataset-specific and allows the investigator to pinpoint specific parts of the embedding that are artificially expanded, shrunk, or stretched. We propose EIs as a flexible approach that represents as much information about embedding distortions in a 2-dimensional projections as possible, while remaining easy to interpret.

The following *Results* section describes our methodological contributions and validates them in an empirical study. First, we present the *ViScore* approach to SP scoring. Second, we employ it in a quantitative comparison of *ViVAE* to other DR methods. Third, we perform a case study with a longitudinal developmental dataset (18) to validate the capacity for multi-scale SP by *ViVAE* in an applied context with known trajectory structures. Fourth, we demonstrate the use of EIs as a tool for on-demand detection of distortions, using the same dataset. Finally, we compare meaningful differences between t-SNE, *UMAP*, and *ViVAE* on a non-developmental dataset (50), evaluating localised distortions with the use of more focused, supervised evaluation measures implemented in *ViScore*.

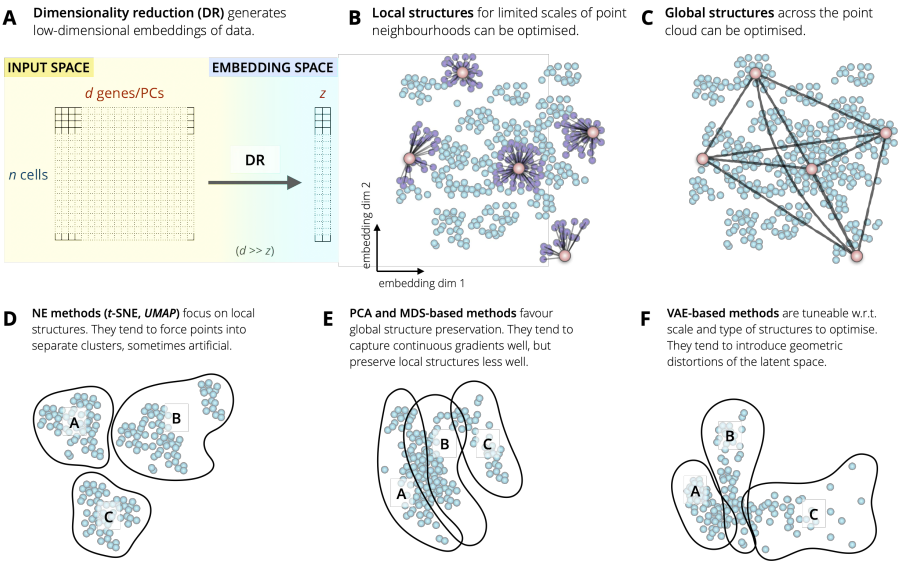

Fig. 1: **Schematic overview of dimensionality reduction (DR) and structure preservation. A.** Tabular d-dimensional input data is transformed into a z-dimensional embedding by DR. While the original d features correspond to specific genes or principal components (PCs), the z features of the embedding do not. The number of data points n does not change between the inputs and the embedding. **B, C.** DR can focus on preserving neighbourhood relationships between points, respectively, at a localised scale or globally. **D.** Neighbour embedding (NE) methods are typically 'cluster-forming'. **E.** Principal component analysis (PCA) and methods based on multidimensional scaling (MDS) do not have a locality bias and favour global structure preservation. **F.** Methods based on variational autoencoders (VAEs) are highly flexible in the objectives they optimise. The models' embedding spaces tend to introduce geometric distortions with respect to the input data.

Results

First, we present a framework for evaluating DR in both supervised and unsupervised settings (with and without the use of cell labels, respectively). This is implemented in the *ViScore* Python package, which includes an automated large-scale benchmarking solution.

Figure 2**A** introduces R_{NX} curves (34) as the basis for assessing *structure preservation* (SP). This is a mathematical framework for quantifying preservation of structure at any given scale. Crucially, we solve the outstanding issue of poor scalability of R_{NX} with a custom approximation algorithm that extends their use to large scRNA-seq data. This has been unfeasible previously, thus dramatically restricting the evaluation of single-cell DR. We use R_{NX} to derive two metrics which score, respectively, local (locally biased) and global (scale-agnostic) SP, and use them to compare *ViVAE* to other methods.

Figure 2**B** (left side) shows the *Extended Neighbourhood-Proportion-Error* (xNPE): a new score for supervised (cell label-based) evaluation. xNPE, based on the NPE (28), measures embedding quality per cell population. It captures distortions that change the relative positions between labelled cell populations. Providing cell population-level scores instead of embedding-level ones, xNPE serves as a QC check especially for comparing multiple candidate embeddings of the same dataset.

Another QC tool shown in Figure 2**B** (right side) are *neighbourhood composition plots* (NCPs), which diagnose sources of mis-embedding. They provide a qualitative description of errors, in contrast to the quantitative approach of the xNPE. NCPs visualise, for a given population of interest, which cells surround it, respectively, in the HD input space and in the evaluated embedding. This is done using a range of local neighbourhood scales.

Second, with a suitable evaluation framework in place we developed *ViVAE*: a new DR model for balanced local and global structure preservation. This is a regularised variational autoencoder (VAE) that learns non-linear transformations of scRNA-seq data (Figure 2**C**). The input data is first denoised to prevent the model fitting to noise. Next, the model is trained on the denoised data with the use of a novel stochastic multidimensional scaling (*stochastic-MDS*) regulariser. Stochastic-MDS is a key extension of the 'stochastic quartet' approach from (29), making crucial adaptations to yield a differentiable loss function that is usable in a deep learning context. In practice, stochastic-MDS improves the preservation of global structures and embedding distortions (as shown further).

The trained model can generate data embeddings (Figure 2**D**) and map unseen data points from the same domain as training data, facilitating the assessment of technical artifacts by comparing data batches.

Figure 2**E** introduces *encoder indicatrices* (EIs): a novel QC tool that allows the investigator to visually inspect unwanted distortions which the model introduces into the embedding. Appearing as ellipses in the figure, EIs denote both the direction and the magnitude of unwanted expansion, shrinkage, or stretching of parts of the embedding space, to the extent that these can be visualised in 2-*d*.

Additionally, *ViVAE* integrates with *FlowSOM* (19), which couples clustering and graph embedding via minimum spanning trees (MSTs), as shown in Figure 2**F** (left side). This helps in validating the MST topology against the embedding, and vice versa, and examining whether expected ontologies, hierarchies, or developmental lineages are reflected. Projecting values from the input feature space onto representative clusters also quickly captures major sources of variability, as a dataset-specific explanatory mechanism (right side).

Both *ViScore* and *ViVAE* are available in dedicated GitHub repositories (*saeyslab/ViScore* and *saeyslab/ViVAE*), with several guided tutorials, experiments, and benchmarking guidelines provided alongside the software itself.

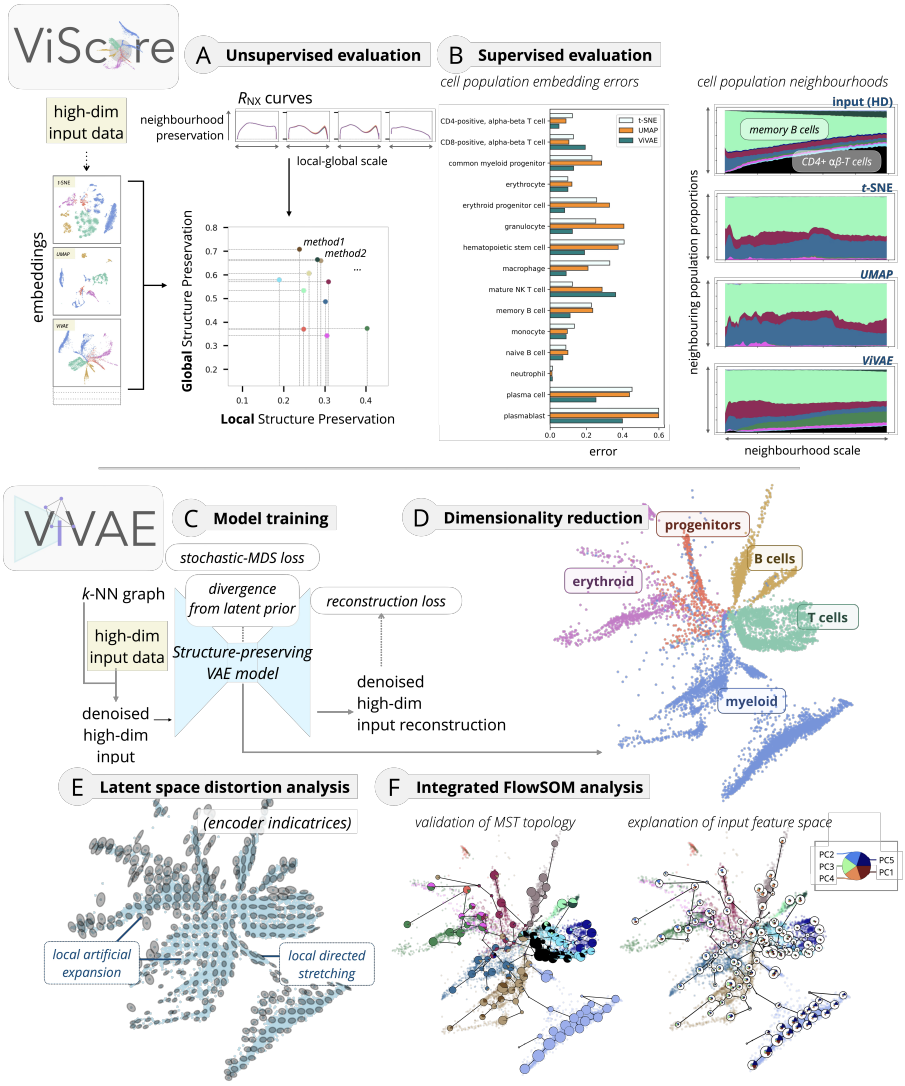

Fig. 2: Schematic overview of proposed workflows. ViScore–A. Unsupervised evaluation based on R_{NX} curve analysis quantifies neighbourhood preservation for local and global structures. **B.** *(Left side.)* Supervised evaluation via *Extended Neighbourhood-Proportion-Error* (xNPE) quantifies embedding error per cell population. *(Right side.) Neighbourhood composition plots* (NCPs) reveal inaccuracies in relative positions of cell populations in evaluated embeddings. **ViVAE–C.** Input data is denoised and used to train a regularised VAE model. **D.** The trained model is used to produce a low-dimensional embedding for visualisation. **E.** Encoder indicatrices (EIs) evaluate distortions of the latent space learned by the model's encoder. **F.** The model can be used to validate a *FlowSOM* tree topology *(left side)* and project input feature values onto representative clusters *(right side)*.

ViScore enables the evaluation of structure preservation across multiple scales

The proposed *ViScore* methodology assesses the preservation of high-dimensional (HD) structures in their low-dimensional (LD) embeddings across different scales. The term *'Structure Preservation'* (SP) is taken to denote the degree to which an embedding retains the neighbourhood structures between HD points in the original data. The choice to focus on (Euclidean distance-induced) neighbourhoods rather than distances themselves is motivated by the analysis in (31), which demonstrates their conceptual relevance in single-cell DR.

Figure 3**A** shows an example of 4 alternative embeddings of a developmental dataset from (18). A single 'vantage-point' cell is chosen (orange cross); a colour gradient shows the true neighbourhood-ranks of all other cells with respect to the vantage point. These neighbourhood-ranks are computed from the HD input data, not the embeddings.

t-SNE and *UMAP* separate the data into islands, achieving a high level of detail (note that these separations always need to be validated as non-artificial). On the other hand, the gradient of neighbour ranks is inconsistent: this demonstrates failure to preserve some structures. *SQuad-MDS*, at the other extreme, shows minimal separation into islands, but the gradient is preserved better. Finally, *ViVAE* balances these two criteria (as examined further in the following section).

Despite under-defined notions of 'local' and 'global' in relevant literature, we devise an exact definition for a *Global SP* score and *Local SP* score, based on R_{NX} curves (34) (see *Methods*). This is shown in Figure 3**B**. It is crucial that we avoid setting a hard neighbourhood-size cut-off for what constitutes a 'local' structure, and we compute Global SP as a truly scale-agnostic score (*i.e.*, small-scale and large-scale structures are treated as equally important), thereby reducing arbitrary designs.

In overview, both SP scores measure the preservation across all possible scales; for any point in a dataset of size N, these are scales from 1 to $N-1$. While *Global SP* treats all scales as equally important for the final score (*scale-agnostic*), *Local SP* applies a log_{10} rescaling of neighbourhood scales, which upweights the importance of smaller-scale structures (as proposed in (34); *locally biased*).

The SP scores are upper-bounded by 1, with 0 corresponding to a random embedding (baseline), and negative values to a worse-than-random representation of neighbourhood structures. A realistic upper bound, however, is dataset-specific, since full preservation of neighbourhoods would only be possible for some cases of noiseless data lying on a 2-manifold that can be flattened. Determining this upper bound would translate to describing the true manifold of each single-cell dataset; this is not currently feasible. However, based on comparative analyses presented here and elsewhere (24; 65), we know some DR methods to effectively prioritise Global SP (PCA, *SQuad-MDS*) while others (mainly *t*-SNE) focus on Local SP. This serves as an empirically determined anchor for good performance with respect to each criterion.

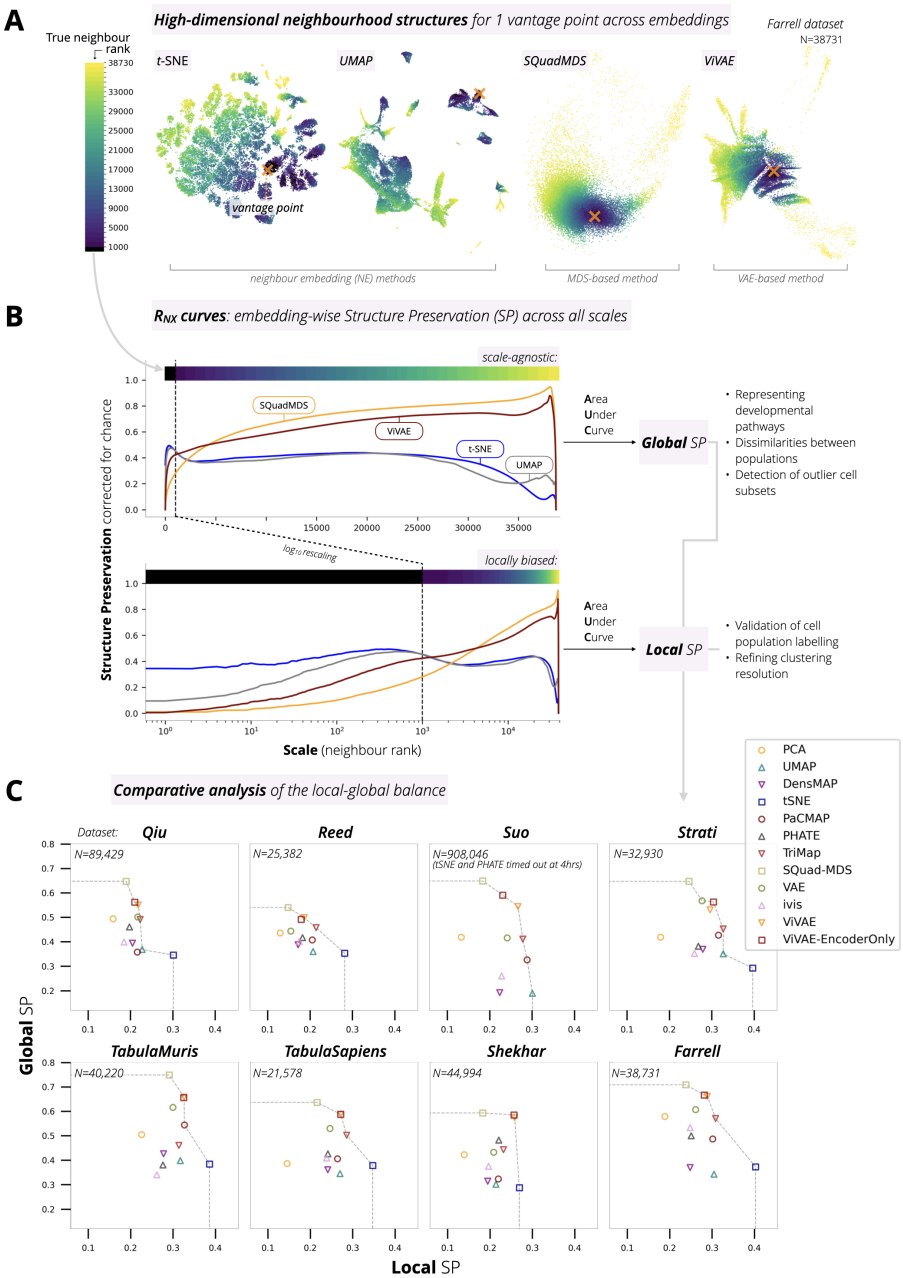

Fig. 3: Structure Preservation (SP) scoring methodology and comparative analysis results. A. The ranking of neighbours to one fixed 'vantage point' in the *Farrel* data (18), computed in the 100-PC space, is projected onto a *t*-SNE, *UMAP*, *SQuad-MDS* and *ViVAE* embedding of that data, with a corresponding binned colour scale. **B.** R_{NX} curves integrate neighbourhood preservation across all vantage points, yielding an SP score corrected against a random embedding baseline for each scale from 1 to $N-1$ (where N is the cell count). The curves can be compared between the different embeddings of the data, with a scale-agnostic curve (top) used to compute *Global SP* and a locally biased curve with $\log_{10}$-rescaled neighbourhood scale (bottom) used to compute *Local SP*. Both SP scores are computed as the corresponding area-under-curve (AUC). **C.** Mean Local and Global SP values are shown for each dataset and DR method, with a subplot per dataset. The horizontal and vertical axes, respectively, quantify Local and Global SP, with common axis ranges across all subplots. Distinctive symbol-colour combinations show the mean SP values across 5 independent runs. The Pareto front with respect to Local and Global SP is indicated using a dotted line in each subplot.

ViVAE is a parametric model that balances local and global structure

We compare 12 DR models run on 8 publicly available scRNA-seq datasets with respect to Local and Global SP. These models are PCA, *UMAP* (39), *DensMAP* (a *UMAP* variant), *t*-SNE (38), *PHATE* (41), PaCMAP (66), TriMAP (3), SQuad-MDS (29), ivis (57), VAE (a standard variational autoencoder) and our own: *ViVAE* (default) and *ViVAE-EncoderOnly*. While default *ViVAE* is trained using reconstruction loss, a variational regulariser, and stochastic-MDS, *ViVAE-EncoderOnly* omits reconstruction loss (and does not have a decoder). This yields a loss function for the *ViVAE* and *ViVAE-EncoderOnly*, respectively:

$$l_{\text{ViVAE}} = l_{\text{recon}} + l_{\text{kldiv}} + 100 l_{\text{MDS}}$$
$$l_{\text{ViVAE-EncoderOnly}} = l_{\text{kldiv}} + 100 l_{\text{MDS}}$$

where l_{recon} is the autoencoder reconstruction loss (mean squared error), l_{kldiv} is the Kullback-Leibler divergence from an isotropic latent prior standard to VAEs and l_{MDS} is the novel stochastic-MDS loss. We elaborate on the used models, their hyperparameters, and the comparative analysis set-up in the *Methods* subsection *Model and hyperparameter specification*.

We test the DR methods on the following datasets:

- *Qiu*: 89*k* T cells from mouse embryos (49). Chosen as immunologically relevant data from a common model organism.
- *Reed*: 25*k* immune cells from human breast (50). Chosen due to containing widely studied immune cell compartments, and because DR methods struggle to separate some of the labelled cell populations clearly (*e.g.*, NK cells within the T+NK compartment, which we evaluate in a corresponding case study). Also as a representative of non-developmental data, for which cluster-forming embeddings (*t*-SNE, *UMAP*) are traditionally used.
- *Suo*: 908*k* human pre-natal T and NK cells across organs (56). Chosen as a recent representative of a large multi-organ dataset with developmental gradients.
- *Strati*: 33*k* human bone marrow cells from healthy donors (53). Chosen due to the prominence of bone marrow lymphopoiesis as use case in trajectory inference, as well as its relevance in immunology and haematology.
- *TabulaMuris* (subset): 40*k* mouse bone marrow cells (2). Chosen as a popular mouse dataset complementary to *Strati*.
- *TabulaSapiens* (subset): 22*k* human thymic cells (25). Similarly to *Strati*, chosen due to the prominence of T-lymphopoiesis as a use case in TI analysis and its relevance in immunology and hematology.
- *Shekhar*: 45*k* cells from mouse retina (52). Chosen as a popular transcriptomic dataset where a documented batch effect can be revealed by non-linear DR.
- *Farrell*: 39*k* cells from zebrafish embryos at different developmental stages (18). Chosen to evaluate the preservation of smooth gradients and discontinuities, both of which are present, due to a longitudinal component. The provided annotation of developmental lineages allowed us to validate embeddings of developmental pathways in a dedicated case study.

Following a standard preprocessing protocol (27; 57), we normalise and scale expression matrices from each dataset and apply PCA to extract orthogonal components as

input features, retaining 100 PCs (as a more conservative approach). The full preprocessing workflow is given in Methods section *Resource availability* and the online *ViScore* repository.

Figure 3**C** compares Local and Global SP by dataset, averaged across 5 repeated runs. For all datasets, at least one of the *ViVAE* models lies on the Pareto front, meaning no other model outperforms it at both tasks. Notably, *t*-SNE gives the highest Local SP wherever applied (training timed out at 4 hours for the largest *Suo* dataset) and *SQuad-MDS* gives the highest Global SP across all datasets. *TriMap*, on the Pareto front in 6 out of 8 experiments, achieves similar Local SP to *UMAP*, while improving Global SP, which aligns generally with past results in (24).

Supplementary Table 1 contains results with and without the use ofdenoising of input data for each method, which amounts to an ablation study that demonstrates a boost to Local SP with denoising with VAE-based models. Additionally, the *ViVAE* and VAE models in our comparison use the same architecture, therefore comparing them yields an ablation study that demonstrates a boost to Global SP with stochastic-MDS. *Supplementary Table 1* further includes *p*-values from one-sided Wilcoxon tests comparing the default and encoder-only variant of *ViVAE* with each other method per dataset for Local SP, Global SP, and a tentative 'Balanced SP' metric (computed as the harmonic mean between the two, as described in Methods section *Multi-scale evalution of structure preservation*). The Local, Global, and Balanced SP means and standard deviations are plotted separately plotted in *Supplementary Figure 1*. The change to R_{NX} curves induced by input data denoising is shown in *Supplementary Figure 2*.

While these SP results give an overview of different methods' capabilities to represent data faithfully, there are other aspects of *ViVAE* that we see as valuable contributions. Based on a VAE architecture, *ViVAE* is a parametric model with a latent space that is amenable to QC and explainability measures. This includes encoder indicatrices (as demonstrated below) and integration with *FlowSOM* (demonstrated in online materials). Furthermore, *ViVAE* does not have an effective 'scale' hyperparameter determining the size of relevant point neighbourhoods for which structures are optimised during training, unlike *t*-SNE, *UMAP*, *DensMAP*, *PaCMAP*, *TriMap*, *PHATE*, or *ivis*. This approach, using the truly multi-scale stochastic-MDS loss, simplifies potential hyperparameter tuning.

In the following sections, we demonstrate ways in which *ViVAE* meaningfully outperforms its alternatives in two case studies: one with the developmental *Farrell* data and one with the non-developmental *Reed* data. The *Farrell* case study shows clearly why the local-global SP balance achieved by *ViVAE* is desirable.

Stochastic-MDS regularisation in *ViVAE* achieves more faithful embeddings of developing cell lineages in zebrafish embryos

The *Farrell* Drop-Seq dataset from (18) aggregates 38,731 cells from 12 annotated developmental stages across 694 zebrafish embryos, from 3.3 to 12 hours post-fertilisation (*hpf*). In addition to *hpf*-sorted embryonic stage labels, the authors provide annotations of developmental lineages and inferred pseudotime values inducing the ordering of cells within each lineage. As shown in our comparison results, '*ViVAE*' and '*ViVAE-EncoderOnly*' models achieve high Global SP, lower only than *SQuad-MDS*, which they exceed at Local SP. Both *ViVAE* models improve Local and Global SP over a standard VAE.

Figure 4**A** compares embeddings from *t*-SNE and *UMAP* (examples of neighbour-embedding; NE), *SQuad-MDS* (example of multidimensional scaling; MDS), and VAE, default *ViVAE*, and *ViVAE-EncoderOnly* (examples of variational autoencoders; VAEs). Embryonic stages are marked by colour and schematic annotation bubbles with arrows in the direction of development (increasing cell differentiation). As a generally known phenomenon, the NE methods separate data into clusters, with the rest forming smoother embeddings. Of the smooth embeddings, *SQuad-MDS* shows the least separation of 11- and 12-*hpf* cells into distinct branches, with VAE or default *ViVAE* separating them the most; panel **B** maps these separations onto trajectory endpoints. The smooth embeddings place most 6-*hpf* ('shield phase') cells clearly along the major trajectory, in contrast to NE embedding separating them out. The correctness of the smoother embeddings is supported by lower Extended Neighbourhood-Proportion-Error (xNPE) embedding errors for shield phase cells: ranging from 0.781 in *t*-SNE, 0.858 in *UMAP* and 0.620 in *SQuad-MDS* to 0.243 in VAE, 0.242 in *ViVAE*, and 0.128 in *ViVAE-EncoderOnly*. Lower xNPE indicates a more accurate layout of this population with respect to others (see Methods section *Extended Neighbourhood-Proportion-Error*).

Figure 4**B** shows 6 distinctive developmental lineages, with cells coloured by normalised ranks along the progression. Terminal cells in each lineage are highlighted in red. Of particular interest is the challenge of embedding the *enveloping layer cell* (ELC) and *primordial germ cell* (PGC) lineages, terminating at 10.0-*hpf*. They contain few cells and are distinct from other lineages, as described in the original study. All 6 embeddings reflect this, with both lineages partially separating from others. Additionally, both lineages show some degree of bifurcation. The pathways are highly fragmented in *t*-SNE and *UMAP*, while in the smooth embeddings they retain more continuity, despite the low cell counts.

This raises the question of whether smooth embeddings represent transitional states correctly. Consider that this data is longitudinal, still allowing for gaps in trajectories between *hpf* stages. To test this, we quantify how continuous or fragmented each lineage is in the input (HD) space. In Figure 4**C**, we show Euclidean distances between cells ordered by pseudotime ('transitions') within the *prechordal plate* lineage. This is computed for input space and for each of the embeddings, and transformed to account for scale differences (via *min-max* and *log1p* scaling). Large distance values at the 3.3- and 3.8-*hpf* stages in input space validate that the less dense representation of these cell subsets by smooth embeddings is correct; *t*-SNE and *UMAP* collapse them artificially.

To measure the distortion of these transitions by each embedding, we report L1 distances between the input and embedded distance distributions, normalised by each lineage's total cell count. These values (Figure 4**D**) show that the smooth DR models indeed achieve a more faithful representation of the lineages. Furthermore, Figure 4**E** shows transition distances in the ELC lineage to validate the bifurcation observed in embeddings. The

real separation is shown to be gradual, more similar to the representation by the smooth embeddings.

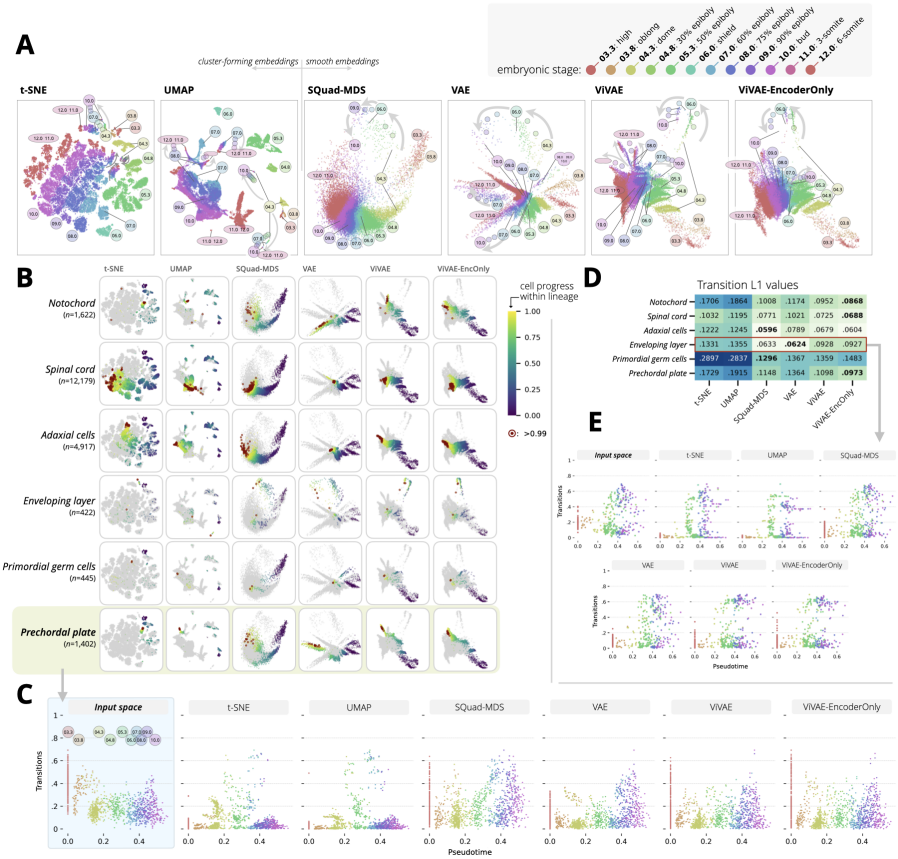

Fig. 4: ***Farrell* data embeddings with developmental lineages in zebrafish embryos.** **A.** 2-d embeddings by *t*-SNE, *UMAP*, *SQuad-MDS*, VAE, *ViVAE* and *ViVAE-EncoderOnly* are shown, with colour-coding by embryonic stage. The layout of cell subsets by stage is indicated in each embedding, showing major developmental gradients using *hpf* labels and arrows indicating progression. **B.** Six annotated developmental lineages are projected onto each embedding, using colouring by pseudotime-derived rank, with terminal cells in the highest one percent highlighted in red. **C.** Scaled distances between cells along the *prechordal plate* lineage are shown for input data and the embeddings. **D.** Average L1 distances between input-space and embedding distances for lineages shown in *B.* are plotted in a heatmap. **E.** Transition distances for the enveloping layer lineage validate that trajectories start diverging around the 4.3- or 4.8-*hpf* stage.

Encoder indicatrices reveal distortions in VAE model space

Figure 5**A** gives an overview of the algorithm. We propose *encoder indicatrices* (EIs): a new tool to visualise distortions of (V)AE-based latent spaces, and use them to analyse VAE, default *ViVAE*, and *ViVAE-EncoderOnly* embeddings of the *Farrell* data. In each case, an embedding is generated by a trained encoder model. Representative points are sampled from the embedding using a uniform grid. They are then mapped to their coordinates in the input (HD) space. For each sampled point, a small hypersphere around it forms its HD *indicatrix*. Here, 'indicatrix' denotes a shape that helps us describe how the particular part of the point cloud gets projected by the trained model. Using a local linearisation of the encoding, we then flatten the indicatrix and transform it using the encoder again.

This transformation introduces size and shape distortions, which are mapped onto the embedding coordinates. In this way, we objectively describe local embedding artifacts throughout the latent space, to the extent that this is achievable in a 2-dimensional visualisation. In line with previous work in (44), we adopt the use of ellipses superimposed on the

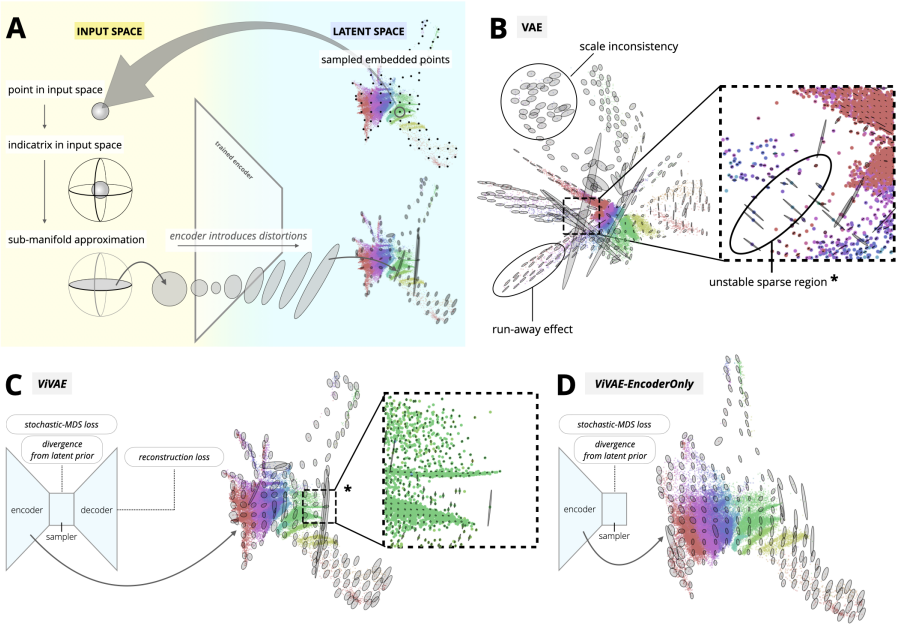

Fig. 5: **Analysis of latent space distortion in VAE-based embeddings of *Farrell* data. A.** Overview of *encoder indicatrix* (EI) computation. Points in input space are obtained via grid-sampling from the embedding. Using local linearisations of the trained encoder, HD indicatrices for the points are flattened along corresponding horizontal tangent spaces, yielding a piecewise approximation of the embedding 2-manifold. Flattened indicatrices are transformed by the encoder and are plotted over the embedding. **B.** EIs reveal commonly observed distortions of VAE latent space: scale inconsistencies, run-away effect inducing directed stretching and instabilities in sparse regions. **C.** The default *ViVAE* model gives less varied indicatrices but some instabilities in sparse regions remain. **D.** The additionally evaluated encoder-only *ViVAE* model learns a latent space that exhibits gradual distortion of scale and shape and less drastic instabilities due to sparsity (less stretching and fewer changes in direction of the stretching across different parts of the embedding space).

embedding for a readily interpretable visual description of artifacts. The *Methods* section describes the EI algorithm in detail.

Figure 5**B** uses EIs to highlight commonly observed embedding distortions in VAEs. First, scale inconsistences make some parts of the latent space artificially sparser. Second, a 'run-away' effect causes directed stretching of some point subsets. Third, some sparse regions are ill-defined, showing drastic extension where the embedding should be smoother.

Figure 5**C** shows that the default *ViVAE* model, with denoising and stochastic-MDS regularisation, gives more uniform indicatrices, reducing scale inconsistencies and run-away effects. Some instabilities in sparse regions as seen with VAEs remain. Figure 5**D** shows that the encoder-only *ViVAE* model creates a latent space with some gradual scale and shape distortion, but further reduces instabilities due to sparsity. Overall, this shows a notable reduction of latent space distortion in *ViVAE* over a standard VAE.

ViVAE achieves structure-preserving dimensionality reduction of non-developmental immune cell data with low population embedding errors

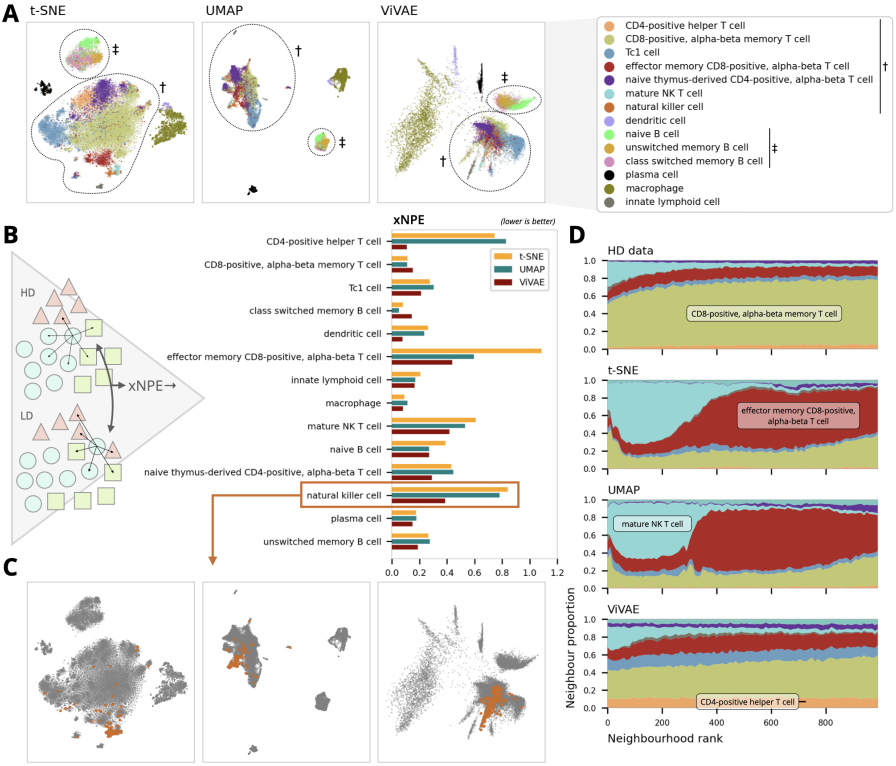

Fig. 6: ***Reed* dataset embedding and analysis of population-wise error. A.** 2-d embeddings *t*-SNE, *UMAP* and *ViVAE* are shown, with populations indicated by colour, according to legend on the right. Symbols mark the the T- and NK-cell (†) and B-cell (‡) compartments. **B.** xNPE (population-wise embedding error) is computed using 1000-nearest-neighbour relations in input space and embeddings. Distributions of neighbour identities in terms of self and non-self are compared. **C.** NK cells are highlighted in orange as population-of-interest in each embedding. **D.** Neighbourhood composition plots ($k = 1000$) show distribution of other populations in the neighbourhoods of the population-of-interest. Colour-coding from panel *A*. is used, along with text annotations.

Here we demonstrate the usefulness of *ViVAE* as a general-purpose DR algorithm for other than developmental data, where Global Structure Preservation (SP) by smoother embedding algorithms is less of a focus. To that end, we demonstrate novel supervised (label-based) evaluation tools from *ViScore*: the Extended Neighbourhood-Proportion-Error (xNPE) and neighbourhood composition plots (NCPs).

Specifically, we use the *Reed* immune cell dataset, originating from a single-cell transcriptomic atlas of the adult human breast (50). As opposed to the *Farrell* embryological

data (in the previous section), we expect more discontinuity between cell types, as the breast tissue is seeded with distinct populations of mature immune cells, likely presenting a challenge to our smoother VAE-based embedding method.

Figure 6**A** compares *ViVAE* to the prevalent NE methods (*t*-SNE and *UMAP*), showing a degree of major cell-compartment separation in each. Since DR invariably introduces artifacts, we seek to quantify and describe them here, using a higher-resolution approach than embedding-level SP scoring. Figure 6**B** quantifies population-level embedding errors using xNPE, which leverages the provided cell labelling. For each *population of interest* (POI), the xNPE aggregates the distribution of 'self versus non-self' neighbours (how many neighbouring cells belong to the POI itself, versus a different population) over all the cells in the POI. It does so separately in the HD input space and each LD embedding, ending up with two distributions. These two distributions are then compared using the earth mover's distance (EMD). The lower the EMD value, the lower the distribution of the POI is with respect to the ground truth.

By this metric, *t*-SNE fares less well than *UMAP* for 8 of 14 populations and less well than *ViVAE* for 12 of 14 populations, despite a high Local SP score based in an unsupervised setting. Importantly, in contrast to Local SP, xNPE only focuses on how each population interfaces with the others around it; small positional inaccuracies or swapping positions within the population are not penalised. (The online *ViScore* GitHub repository includes a proof-of-concept experiment with synthetic data.)

We investigate how the natural killer (NK) cells population is distributed in different embeddings, picking them as the POI because all three embeddings place them within the same compartment as T cells (marked '†' in panel **A**), but assign them somewhat scattered positions within it (orange points in panel **C**). Both *UMAP* and *ViVAE* outperform *t*-SNE in terms of xNPE. To derive a more meaningful interpretation, we generate neighbourhood composition plots (NCPs) for the NK-cell population (Figure 6**D**). These are stacked area plots showing smoothed proportions of neighbouring populations across different neighbourhood sizes (up to 1000), using a sliding-window approach. These proportions can be compared between input (high-dimensional, non-reduced) data and the different embdeddings, and they exclude neighbouring cells from the POI itself.

According to the top NCP, the NK cell population is close to a $CD8^+$ α-β T-cell population (olive green) in input space. All three embeddings underestimate this (the population is under-represented), but *ViVAE* does so the least. Notably, *t*-SNE and *UMAP* show an intrusion of mature NK T cells (light blue) into a close neighbourhood of the NK cells, meaning their proximity is exaggerated; *ViVAE* avoids this artifact. The proximity to $CD4^+$ helper T cells (orange), relative to the HD expression data, is artificially reduced by *t*-SNE and *UMAP* and increased in *ViVAE*. *t*-SNE and *UMAP* also mis-embed NK cells with respect to effector memory $CD8^+$ α-β T cells, in contrast to *ViVAE*.

This case study shows separation of major immune cell compartments by *ViVAE* and low population embedding errors, validating its competitiveness at general-purpose single-cell data embedding, and thus extending its functionality beyond visualising trajectory data. The supervised evaluation metrics presented here can leverage known cell population labelling, making it possible for the domain expert to examine embeddings of single-cell data more meaningfully, in an ad-hoc setting.

Discussion

We have shown *ViVAE* to achieve favourable multi-scale structure preservation and faithful embedding of both trajectory and non-trajectory data, according to comparisons with other methods on an array of real biological datasets as well as focused case studies. Integrating denoising of inputs and a novel stochastic-MDS component with a VAE model, *ViVAE* balances the continuous nature of VAE latent spaces with an ability to separate major cell compartments. To improve interpretability, we equip *ViVAE* with new QC functionality in the form of *encoder indicatrices*; crucially, these can be implemented for any differentiable DR model. In addition to a regularised VAE, we propose and evaluate a decoder-less model (‘*ViVAE-EncoderOnly*’). We take the full model as default due to mostly slightly better preservation of local structures, but evaluate both.

We have demonstrated that balancing local and global structure preservation can lead to more faithful embeddings of developmental, longitudinal, but also non-trajectory (single-snapshot) data. In essence, the proposed methodology is designed for a broad range of exploratory data analysis workflows.

Throughout the work, we use Euclidean distances in PCA-reduced input space, which yield evaluations that are generally congruent with the known trade-offs between local and global structure preservation by established methods. However, for adaptability to sparser and higher-dimensional data (*e.g.*, expression of a filtered gene set, without initial PCA reduction), we implement the use of cosine distances as an alternative for use with the stochastic-MDS loss, as well as in the unsupervised structure-preservation scoring.

The development of *ViVAE* followed our initial implementation of model-agnostic structure preservation scores in *ViScore*, which were necessary for a quantitative evaluation. Overcoming previous conceptual and computational limitations, we coin unsupervised embedding-level scores that quantify structure preservation without using hard local/global cut-offs, landmark points based on clustering, or other previously proposed proxy measures. In addition, we introduce supervised population-level evaluation metrics for comparing embeddings and explaining artifacts.

Our contributions to evaluation and interpretability are motivated by increasing sizes of single-cell datasets, the availability of large public databases and atlases (59; 40; 17), and ongoing efforts to systematically evaluate single-cell data analysis algorithms (36). DR currently aids in exploration and hypothesis formulation, and its usage will likely rise with more public data. This will require efforts to prevent erroneous conclusions from spurious patterns or artifacts in embeddings.

Having adapted notions from differential geometry described in (44) to visualise distortions in the model space using *encoder indicatrices*, we estimate that an increased use of such explainability tools will help in describing method-specific artifacts. Future endeavours in interpretable non-linear single-cell DR may include hierarchical embeddings, as pioneered in (63), and interactive tools for evaluating embedding quality at local levels (46; 22).

Another major avenue of research involves topological data analysis (TDA) to describe point clouds. We acknowledge that concepts such as local and global structure have undergone thorough and formal examination within this field (as surveyed in (1)). Further, some studies have sought to tackle the problem of interesting structures existing at multiple scales in single-cell data (64; 55). Nonetheless, this work within TDA has not been fully incorporated within the single-cell DR field. While interesting work on topology-preserving autoencoders exists (42), our familiarity with a recent study on topological regularisation

of autoencoders for single-cell DR (21) emphasises to us that dataset-specific descriptions of topologies can prove cumbersome and non-trivial.

More broadly, describing the true manifolds of real biological data is currently intractable. Therefore, the focus of our study here, and a handful of others that compare DR methodologies (65; 24; 57; 31), is instead on empirical evaluation across different datasets that are representative of a range of real-world scenarios.

In line with this approach, our work seeks to provide ways to measure and optimise multi-scale structure preservation without requiring too much tuning or too many assumptions on underlying structures. This is why our current study does not use TDA and remains restricted to the use of nearest-neighbour graphs in lieu of more elaborate topologies.

Nonetheless, our ongoing research is focusing on alternative graph structures for both denoising and DR evaluation, such as networks from the Louvain algorithm (5), witness complexes (used in (55)), or branched spanning trees robust to noise (51). Our integration of *ViVAE* with *FlowSOM* (13) (shown in Figure 1) is a tentative step toward the use of graphs together with embedded point clouds to capture relevant structures.

To support the adoption of *ViVAE* and *ViScore*, we provide detailed tutorials and guidelines in the *saeyslab*/*ViVAE* and *saeyslab*/*ViScore* GitHub code repositories. The code needed to reproduce the case studies in this paper is also included.

Methods

Variational autoencoders for dimensionality reduction

An autoencoder (AE) is a type of neural network trained to reconstruct an input $\mathbf{X_{n \times d}} \in \mathcal{X}$ as an approximation $\mathbf{\hat{X}_{n \times d}}$ while compressing $\mathbf{X}$ into a lower-dimensional representation $\mathbf{Z_{n \times z}} \in \mathcal{Z}$ (where $\mathcal{X} = \mathbb{R}^d, \mathcal{Z} = \mathbb{R}^z, z < d$). It consists of two stacked feed-forward networks: the encoder $E_\Phi : \mathcal{X} \rightarrow \mathcal{Z}$, which transforms $\mathbf{X}$ to $\mathbf{Z}$, and the decoder $D_\theta : \mathcal{Z} \rightarrow \mathcal{X}$, which transforms $\mathbf{Z}$ to $\mathbf{\hat{X}}$. The network parameters (encoder weights Φ and decoder weights Θ) are learned so as to reduce a reconstruction loss: in *ViVAE*, input data is scaled (see *Data availability and preprocessing*) and reconstruction loss is computed as the mean square error $l_{\text{recon}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{h} \sum_{i=1}^{h} (x_i - \hat{x}_i)^2$ for $\mathbf{x}$ and $\hat{\mathbf{x}}$ being points or batches of points drawn, respectively, from $\mathbf{X}$ and $\mathbf{\hat{X}}$. Each node of the network transforms inputs using a non-linear activation function, allowing for learning complex, non-linear patterns through back-propagation of error at output.

The variational autoencoder (VAE) is a generative model based on the AE. Instead of encoding and decoding a latent representation $\mathbf{Z}$ directly, a distribution $\mathcal{D}$ in latent space is learned, and a sampler generates a non-deterministic representation $\mathbf{Z}$ and decodes it. The dissimilarity (Kullback-Leibler divergence) between $\mathcal{D}$ and a latent prior distribution (typically isotropic Gaussian) is used as an additional loss term l_{kldiv} during training. The model is trained in mini-batches drawn from $\mathbf{X}$ to minimise l_{recon} and l_{kldiv} jointly. We refer to (26) for a complete explanation.

Since the VAE learns a salient representation $\mathbf{Z}$ of the original data $\mathbf{X}$, we can use $\mathbf{Z}$ for visualisation of important high-dimensional structures in $\mathbf{X}$ that are initially obscured. This gives us a parametric DR model. By default, our *ViVAE* model uses the Gaussian error linear unit (GELU) activation function and a symmetric architecture of 32,64,128 32, and 2 (z) nodes, respectively, in consecutive encoder layers, and a reverse of this for the decoder. The model generalises to higher latent space dimensionalities. The default optimiser used in training is Adam, with a learning rate of 0.001 and weight decay of 0.0001.

PCA-initialisation using imitation loss

In our online *ViVAE* repository, we implement an optional pre-training procedure for PCA initialisation of (V)AEs. We devise the *imitation loss*, which computes L2 distances between embedded points and the first two PCs of the input dataset. In practice, pre-training the model to imitate PCA prior to the full training stabilises results, reducing variability between repeated runs of the training.

Input denoising

The denoising algorithm used here is based on an approach described in (9) and applied previously in (55); it is equivalent to the mean shift algorithm (11). The procedure is written out using pseudocode in Algorithm 1. Denoising requires a k-nearest-neighbour graph (k-NNG) over preprocessed input data; to this end, we use a fast approximate k-NNG construction algoritm (15). By default, we take $k = 100$. Each point is then moved closer toward the average coordinates of its k nearest neighbours, where a hyperparameter λ determines size of the shift. This can be done in multiple iterations, each shift potentially exposing the underlying manifold and making the point cloud less scattered. In our workflow, we use

$\lambda = 1$ and apply 1 iteration of denoising to input data. One may experiment with these parameters, or apply denoising to LD embeddings (using the k-NNG in input space) to achieve a sharper defined shape of embedded cell clusters or pathways.

According to ablations conducted in our full comparative analysis (results in *Supplementary Table 1*), the regularised VAE model used in *ViVAE* is particularly amenable to input denoising, achieving better Local SP in the resulting embedding. Crucially, we still use the original (preprocessed, but non-denoised) input data as ground truth in evaluating the embeddings, so as to ensure a fair comparison between set-ups. We test all of the evaluated methods with and without denoising, further concluding that the structure-preserving properties of *ViVAE* are not exclusively attributable to the denoising step.

Algorithm 1 Denoising

function Denoising(X, K, λ, n_i)
  ▷ X: coordinate matrix, K: row-wise k-nearest-neighbour indices matrix, λ: shift coefficient, n_i: number of iterations
  $n \leftarrow \text{rowCount}(X)$
  for iter $\in 1..x_i$ **do**
    for $i \in 1..n$ **do**
      $\text{idcs} \leftarrow K_{i*}$ ▷ neighbour indices
      $\text{target} \leftarrow \text{columnMeans}(X_{\text{idcs},*})$ ▷ target coordinates
      $\text{shift} \leftarrow \text{target} - X_{\text{i},*}$
      $X_{i*} \leftarrow X_{i*} + \lambda\text{target}$
    end for
  end for
  return X
end function

Stochastic-MDS loss and encoder-only model

Similarly to some other VAE-based DR algorithms (*scvis* (14), *ivis* (57), *VAE-SNE* (20)), *ViVAE* formulates constraints on the LD embedding with respect to HD inputs. We achieve this using a *stochastic multidimensional scaling* (*stochastic-MDS*) loss function. This is an adaptation of the stochastic quartet loss introduced in *SQuad-MDS* (29; 30), which we previously generalised and reimplemented as *group loss* for use in differentiable models in a preliminary study (45).

For stochastic-MDS, samples in each training mini-batch drawn from $\mathbf{X}$ are divided randomly into quartets (group of 4 points) and a cost value is computed based on the difference between normalised distances within the quartet in $\mathbf{X}$ and within the corresponding embedded quartet in $\mathbf{Z}$. An update to each sample's position in LD is computed by differentiating a cost function with respect to all of the 6 intra-quartet distances. Therefore, each sample's position in LD moves to jointly minimise the error of its quartet as a whole. For data points indexed by i and j, we use pairwise Euclidean distance functions, δ_{ij} in HD and d_{ij} in LD (an option of using cosine distances is also given). We compute relative distances between pairs of points among quartets as

$$\delta_{ij}^r = \frac{\delta_{ij}}{\sum_{a=1}^{3}\sum_{b=a+1}^{4}\delta_{ab}}, d_{ij}^r = \frac{d_{ij}}{\sum_{a=1}^{3}\sum_{b=a+1}^{4}d_{ab}} \tag{1}$$

The loss function is then defined as

$$l_{\mathrm{MDS}} = \sum_{a=1}^{3} \sum_{b=a+1}^{4} (\delta_{ab}^{r} - d_{ab}^{r})^2 \tag{2}$$

To increase coverage of different parts of the point cloud, we re-sample quartets multiple times (4 times by default) and average out the computed gradients at each pass. Regarding our fixed choice of 4 points as group size, our past experiments in (45) showed no improvement in terms of structure preservation when group size was increased.

For our default '*ViVAE*' model (labelled that way in our comparative analysis results), we use a weight factor of 1 for l_{recon}, 1 for l_{kldiv} and 100 for l_{MDS}: this amounts to a combination of the structure-preserving capabilities of a VAE and the MDS model. We also evaluate a '*ViVAE-EncoderOnly*' model, which assigns a weight of 0 to l_{recon}, 1 to l_{kldiv} and 100 to l_{MDS}. In this case the decoder is not used, and we obtain a probabilistic MDS model consisting only of an encoder and a sampler. Technically, this is then not an autoencoder, but we use the name *ViVAE* because other parameters of the model are kept fixed between this model and the default one, yielding an ablation experiment for the use of reconstruction error.

Regarding our choice of loss term weights, we observe and report good performance with these default weights across the datasets included in our comparison, at least as long as a consistent data preprocessing workflow (described in *Data availability and preprocessing*) is used. We ensure that the scale of the stochastic-MDS loss term is invariant with respect to chosen batch size and number of quartet re-samplings. Our software implementation enables the usage of cosine distances in HD for computing the stochastic-MDS loss. This is explored in the online *ViVAE* repository.

Encoder indicatrices for detection of local distortion

During training, AEs and VAEs both learn a smooth latent space, defined by the encoder. The parametric nature of these models, along with the fact that the transformations they learn are differentiable, allow us to explore properties of this latent space. Specifically, we can detect local contraction, expansion and directed stretching of its different parts.

Our methodology extends the work on geometric regularisation for (non-variational) AEs (44). To visualise undesirable artifacts, the use of *indicatrices* (7) is proposed to quantify latent space distortion. An indicatrix can be understood, in a simplified way, as a circle on the surface of a small hypersphere in the original (HD) ambient space, embedded into the (LD) latent space. In (44), multiple indicatrices from different parts of the HD point cloud are sampled and then pulled back into the latent space through the decoder, to reveal distortions introduced by the model.

Algorithm 2 Encoder Indicatrices

function CircleOnTangentPlane(x, h_1, h_2, r, n_p)
  ▷ x: coordinates of point of origin, h_1, h_2: tangent vectors, r: indicatrix radius, n_p: number of polygon vertices to estimate indicatrix
  $h_1^n \leftarrow h_1 / L_2\text{norm}(h_1)$
  $h_2^n \leftarrow h_2 / L_2\text{norm}(h_2)$
  $s \leftarrow n_p^{-1}$ ▷ stepsize
  $P \leftarrow []$ ▷ points on circle
  for $i \in 0..(n_p - 1)$ **do**
    $a \leftarrow s \cdot 2\pi i$
    $P_i \leftarrow x + r\cos(a)h_1^n + r\sin(a)h_2^n$
  end for
  return P
end function

function EncoderIndicatrices(E, X, S, r, n_p)
  ▷ E: trained encoder, X: row-wise input coordinates, D: indices of points for which to compute indicatrices
  $L_D \leftarrow E(X_D)$ ▷ 2-d embedding of selected points
  $Ind \leftarrow []$ ▷ list of embedded indicatrix points $\forall x \in X_D$
  for $i \in 1..\text{length}(D)$ **do**
    $d \leftarrow D_i$
    $J \leftarrow \text{jacobianMatrix}(\text{function} = E,\ \text{point} = L_d)$
    $U, S, V \leftarrow \text{singularValueDecomposition}(J)$ ▷ such that $J = U\text{diag}(S)V^T$
    $h_1 \leftarrow V_{*1}$ ▷ horizontal tangent vectors
    $h_2 \leftarrow V_{*2}$
    $P \leftarrow \text{CircleOnTangentPlane}(X_d, h_1, h_2, r, n_p)$
    $Ind_i \leftarrow E(P)$
  end for
  return Ind
end function

A perfect, non-distorted embedding results in all indicatrices remaining circular and identical (circular because shape is preserved and identical because there are no scale inconsistencies across different parts of the latent space). The distortion that each indicatrix reveals is specific with respect to the ambient space, transformation function and the coordinates of the sampled point. In (44), indicatrices are computed for the transformation learned by the decoder: we can refer to them therefore as 'decoder indicatrices'. Crucially, decoder indicatrices quantify distortion of latent space specifically with respect to the decoded reconstruction $\hat{\mathbf{X}}$.

In contrast, we derive a solution to the 'encoder indicatrix' problem, *i.e.*, quantifying local embedding (and latent space) distortion with respect to the original high-dimensional inputs. Since a general analytical solution is not feasible (essentially due to the encoder not being injective), we provide a numerical solution, described in Algorithm 2. A schematic overview of the algorithm is given in Figure 5**A**.

To compute encoder indicatrices, we begin by sampling points from the existing embedding closest to uniform grid over it and mapping them onto the respective high-dimensional inputs X_s. For each point $x \in X_s$, we compute the Jacobian matrix $J(x, \Phi) = \left[\frac{\partial E_\Phi(x)}{\partial x_1}, ..., \frac{\partial E_\Phi(x)}{\partial x_n}\right]$ which, when developed, has dimensions $z \times d$ (z being latent dimension and d being input dimension), and gives a local linearisation of the encoder with respect to this point. Under the assumption that the E_Φ function is approximately linear for very small neighbourhoods of any given $x \in X_s$, only z vectors from a small hypersphere

around it are needed. This is because all other orthogonal vectors will be sent to zero by E_Φ.

We obtain the z useful vectors as the bases of the *horizontal tangent space* specific to the given point and the encoder. The horizontal tangent space is approximated using singular value decomposition of the Jacobian matrix. We then sample points from a small circle in this approximation of the horizontal tangent space for the queried point. The sampled points are taken as the flattened indicatrix, which is then transformed by the encoder to reveal the local size and shape distortions with respect to input data. Same as in the decoder indicatrix case, non-distortion yields identical, circular indicatrices. In practice, the indicatrices differ in size and are stretched into ovals. Note that the use of indicatrices with a linear AE would result in a degenerate case, which is explored in (44).

Integration with *FlowSOM*

To boost interpretability of *ViVAE* embeddings for practitioners familiar with *FlowSOM*, we provide the option to superimpose the topology of a *FlowSOM* MST along with median marker expression, proportion of manually labelled population, or metacluster assignment of clusters over the *ViVAE* embedding. This is demonstrated in our online *ViVAE* repository. In practice, we adapt code from the recent Python clone of *FlowSOM* (13) to enable the projection of clusters into the latent space of a trained model.

Rank-based neighbourhood preservation scoring

We base our unsupervised quality-assessment measures on widely adopted rank-based DR evaluation criteria (32). The basis for these is the concept of a *co-ranking matrix*, which quantifies the changes of ranks of neighbours to each point in terms of pairwise distances in LD, as compared to HD.

For any pair of HD points, we denote distance from ϵ_i to ϵ_j as δ_{ij}. Conversely, we denote the distance from an LD point $\mathbf{x}_i$ to $\mathbf{x}_j$ as d_{ij}. We define the rank in HD of ϵ_j with respect to ϵ_i as the set cardinality $\rho_{ij} = |\{k : \delta_{ik} < \delta_{ij} \vee (\delta_{ik} = \delta_{ij} \wedge k < j)\}|$. (This can be re-stated more simply as 'ϵ_j is the ρ_{ij}-th neighbour to ϵ_i'.) The corresponding rank in LD is similarly defined as $r_{ij} = |\{k : d_{ik} < d_{ij} \vee (d_{ik} = d_{ij} \wedge k < j)\}|$. The non-reflexive (non-self) neighbourhoods of ϵ_i and $\mathbf{x}_i$ are then written as $\nu_i^K = \{j : 1 \leq \rho_{ij} \leq K\}$ and $v_i^K = \{j : 1 \leq r_{ij} \leq K\}$ respectively, for a neighbourhood size of K. We then define the co-ranking matrix for a dataset of size N as $\mathbf{Q} = [q_{kl}]_{1 \leq k,l \leq N-1}$, where $q_{kl} = |\{(i,j) : \rho_{ij} = k \wedge r_{ij} = l\}|$. Any off-diagonal positive values in this matrix indicate a change of rank in the LD embedding versus the HD input dataset, and are undesirable. The co-ranking matrix records changes in the order of relative distances between points as we go from the original, high-dimensional data, toward a lower-dimensional embedding. A reduction in rank is labelled an *intrusion* and is recorded left of the diagonal, whereas an increase is termed an *extrusion*, and is recorded right of the diagonal. A change in rank that crosses a given close-neighbourhood upper bound of K (a sample enters or leaves the K-ary neighbourhood of its reference) is termed a *hard-K intrusion* or *hard-K extrusion*, respectively.

Based on (34), we measure structure preservation at different scales by recording the changes in rank within a limited neighbourhood size K as we expand size from 1 to N-1.

We start with defining the average structure preservation in a K-ary neighbourhood as

$$Q_{\mathrm{NX}}(K) = \frac{1}{KN} \sum_{i=1}^{N} |\nu_i^K \cap v_i^K| \tag{3}$$

and record these values for the different values of K. We denote this as the Q_{NX} curve. A rescaling of the Q_{NX} curve is the R_{NX} curve; it shows improvement of neighbourhood preservation over a random embedding, which would result in an average $Q_{\mathrm{NX}}(K) \approx \frac{K}{N-1}$. This is given by

$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K} \tag{4}$$

for $1 \leq K \leq N-2$. The R_{NX} curve helps evaluate how the structure of the HD dataset is preserved across different scales in the LD embedding.

Multi-scale evalution of structure preservation

In reference to concepts introduced in the previous section, we denote the area under the R_{NX} curve with a linear (non-transformed) scaling of K as the global structure-preservation score (s_g, or 'Global SP'). We also report the area under the R_{NX} curve with a logarithmic ($\log_2$) scale for K, up-weighting local point neighbourhoods, as the local structure-preservation score (s_l, or 'Local SP'). The Global SP score is computed as the area-under-curve

$$\mathrm{AUC} = \left(\sum_{K=1}^{N-2} K^{-1} \right)^{-1} \cdot \sum_{K=1}^{N-2} \frac{R_{\mathrm{NX}}(K)}{K} \tag{5}$$

while the Local SP uses $log_2 K$ instead of K (otherwise the same equation is used). Additionally, we compute a balanced structure-preservation score ('Balanced SP') s_b, calculated as the harmonic mean (F-measure) between the s_l and s_g:

$$s_b = \frac{2 \cdot s_l \cdot s_g}{s_l + s_g} \tag{6}$$

However, we recommend looking at the values of s_l and s_g separately when possible, since overwhelming performance in only one of them can still result in high s_b in cases where this is unintuitive.

It is noteworthy that evaluating Q_{NX} and R_{NX} curves for K ranging from 1 to $N-2$ entails a $\mathcal{O}\left(N^2 \log N\right)$ time complexity. Their computation requires sorting, for each data point, its HD and LD distances with respect to all $N-1$ remaining data points (33). These criteria, in their original implementation, are hence exhaustive and slow to assess the low-dimensional embeddings of very large datasets. We address this below by presenting a novel algorithm that circumvents scaling issues.

Q_{NX} and R_{NX} curve approximation

The $\mathcal{O}\left(N^2 \log N\right)$ time complexity required to evaluate the unsupervised DR quality criteria for all neighbourhood sizes, reviewed above, is unaffordable in big data contexts. One could trivially display the quality curves for very small neighbourhood sizes only, smaller than $\overline{K} \ll N$ (6). The ensued time complexity would scale as $\mathcal{O}\left(\overline{K} N \log N\right)$, thanks to dedicated algorithms for fast nearest neighbour searches (70; 43; 15). Nevertheless, such a

trick cannot enable fairly comparing several DR methods, which may preserve structures and trends at different scales. Methods with global properties such as PCA would be systematically disadvantaged with respect to local schemes such as t-SNE. For this reason, quality curves should still be computed for a much wider proportion of neighbourhood sizes, to grasp the behaviour of DR methods at different data scales. We hence introduce an algorithm to approximate Q_{NX} and R_{NX} curves for large datasets quickly and with high precision, based on repeated sub-sampling via vantage-point trees.

One can first note that Q_{NX} gives a score computed with respect to each data point index $i \in \mathcal{I}$, with the set $\mathcal{I} = \{1, \ldots, N\}$ containing all point indices, averaged over the entire dataset. This score evaluates, for each data point, the proportion of its K nearest HD neighbours preserved in the LD embedding. Faithfully estimating this score average, however, may only require relying on a fraction of the data samples, instead of the whole dataset. In this view, a set $\widetilde{\mathcal{I}} \subset \mathcal{I}$ indexing data representatives could be defined to summarise the data cloud. A candidate approximation of Q_{NX} hence develops as

$$\widetilde{Q}_{\text{NX}}(K) = \frac{1}{n} \sum\nolimits_{i \in \widetilde{\mathcal{I}}} \frac{\left|\nu_i^K \cap v_i^K\right|}{K} \in [0, 1], \tag{7}$$

where $n = \left|\widetilde{\mathcal{I}}\right|$. Compared to Q_{NX}, the time complexity of evaluating $\widetilde{Q}_{\text{NX}}(K)$ for all neighbourhood sizes $K \in \{1, \ldots, N-1\}$ drops to $\mathcal{O}(nN \log N)$; one must indeed only compute the $N-1$ HD and LD distances with respect to each data representative indexed in $\widetilde{\mathcal{I}}$, instead of all N data samples. From $\widetilde{Q}_{\text{NX}}(K)$, an approximation of $R_{\text{NX}}(K)$ is obtained as

$$\widetilde{R}_{\text{NX}}(K) = \frac{(N-1)\,\widetilde{Q}_{\text{NX}}(K) - K}{N-1-K}. \tag{8}$$

To define the set $\widetilde{\mathcal{I}}$, a first option consists in randomly sampling n representatives among the N data samples. Nevertheless, to ensure the consistency of the approximation of $Q_{\text{NX}}(K)$ by $\widetilde{Q}_{\text{NX}}(K)$, the samples indexed in $\widetilde{\mathcal{I}}$ should faithfully represent the HD data distribution. On the other hand, the number n of representatives should remain small to keep the lowest possible time complexity. In this view, defining $\widetilde{\mathcal{I}}$ through pure random sampling indeed well summarises the HD data distribution, but only as soon as n gets sufficiently large, which undesirably increases running times. To keep the n value as small as possible while enforcing the representatives to cover the entire HD distribution, a well-balanced partition of the HD data distribution may first be computed. Space-partitioning trees such as vantage-point (VP) trees are extremely fast and efficient for this purpose (70; 62), being widely used to speed up nearest neighbour searches in big datasets (69; 37).

At each non-leaf node of a VP tree, a vantage point is first selected and the distances between the vantage point and the remaining data points are computed. The median distance is then computed in linear time, thanks to a quick-select procedure (12). Two child nodes are finally created, respectively storing the remaining data points closer and further to the vantage point than the median. The median threshold ensures that the tree remains complete and balanced, with similar proportions of data points in all parts of the tree. Constructing the whole VP tree has $\mathcal{O}(N \log N)$ time complexity (70). However, as only n representatives need to be determined, we may only grow the tree up to depth $\lfloor \log_2 n \rfloor$, in $\mathcal{O}(N \log n)$ time. Indeed, the $2^{\lfloor \log_2 n \rfloor}$ leaves of the resulting tree define a balanced partition of the HD data, with each leaf containing a subset of the database. One can hence randomly sample a representative in each leaf, or possibly two if $2^{\lfloor \log_2 n \rfloor} < n$, yielding n

representatives which summarise the HD distribution well. Furthermore, only the leaves of the tree need to be stored, since only the latter are required to sample the representatives. The VP tree can thus be iteratively created in-place in a single array. In each node, the vantage point is selected as the furthest datum from the most central one.

Note that neighbour searches in VP trees do not scale well with the HD dimension due to the distance concentration (58), which annihilates their time complexity advantage. Nevertheless, this only concerns neighbour searches performed in these trees, not their construction since in this case, data points are evenly spread with respect to the median distance to the vantage point. As we only seek to construct VP trees to partition the HD distribution without performing neighbour searches, they can be safely employed in our context without suffering from increasing HD dimension. As another advantage, VP trees solely require to be able to evaluate distances between the HD data points, without constraining them to be expressed as HD vectors. These trees can hence be generally computed even with complex data formats, unlike k-d trees for instance (62), which involve Cartesian coordinates of the Euclidean space. For this reason, the size of k-d trees grows exponentially with the HD dimension, whereas the median threshold in VP trees prevents such increase, with depths scaling at most as $\mathcal{O}\left(\log_2 N\right)$.

Preliminary experiments revealed that n values between $\log N$ and $10 \log N$ dramatically accelerate the DR quality criteria computation, while yielding close to perfect approximations of Q_{NX} and R_{NX}; when considering big datasets with N growing to very large numbers, average scores estimated on small yet representative subsets of the database indeed intuitively quickly converge towards the actual average computed over the entire dataset. Values of $\lfloor 10 \log N \rceil$ are hence subsequently employed for n, with $\lfloor \cdot \rceil$ denoting rounding. Therefore, the time complexity required to evaluate $\widetilde{Q}_{\text{NX}}$ for all neighbourhood sizes scales as $\mathcal{O}\left(N \log^2 N\right)$, which is considerably more affordable than the $\mathcal{O}\left(N^2 \log N\right)$ time required to evaluate Q_{NX}.

Extended Neighbourhood-Proportion-Error

The *Neighbourhood Proportion Error* (NPE), introduced in (28) and applied in a recent DR benchmark in (65), quantifies preservation of the structure of manually defined populations of cells. We coin the *Extended Neighbourhood-Proportion-Error* (xNPE) as a modification of the NPE.

For each labelled population, NPE iterates over all the points it contains as vantage points. For each vantage point, the number of points in its k-ary neighbourhood belonging to the same population is counted. This is done for the HD input data, as well as the LD embeddings. The distribution of counts of same-population neighbours per vantange point is computed, and it is compared between HD and LD using total variation distance (TVD), resulting in a population-wise error score. The sum of these population-wise error scores is then returned as a measure of error for the entire embedding.

In xNPE, we use proportions of same-population neighbours rather than counts. Further, we use Earth mover's distance (EMD) (47; 48) to compute differences between distributions, since it uses the entire distribution instead of the (more likely spurious) single largest difference between distributions, and return population-wise scores to avoid an assumption of consistent resolution of the cell population labelling.

Algorithm 3 Extended Neighbourhood-Proportion-Error

```
function LikenessDistributions(K, A)
                    ▷ K: row-wise k-nearest-neighbour indices matrix, A: vector of labels
    n ← rowCount(K)
    P ← unique(A)                                                    ▷ populations
    p ← length(P)
    M ← [ ]_n                                   ▷ point-level same-neighbour proportions
    D ← [ ]_p          ▷ population-level distributions of same-neighbour proportions
    for i ∈ 1..n do
        r ← K_{i*}
        M_i ← |A_r = A_i|/length(A_r)   ▷ point-level proportion of neighbours from same
population
    end for
    for j ∈ 1..p do
        a ← |A = P_j|
        D_j ← histogram(M_a) ▷ population-level distribution of same-neighbour counts
    end for
    return D
end function

function BaselineLikenessDistributions(A)
    n ← rowCount(K)
    P ← unique(A)
    M ← [ ]_n                ▷ same-neighbour proportions assuming random positions
    p ← length(P)
    for i ∈ 1..n do
        M_i ← length(A_i)/n
    end for
    for j ∈ 1..p do
        a ← |A = P_j|
        D_j ← histogram(M_a)
    end for
end function

function xNPE(K_H, K_L, A)
                    ▷ K_H and K_L: k-NN matrices for HD and LD data
    P ← unique(A)
    p ← length(P)
    H ← LikenessDistributions(K_H, A)
    L ← LikenessDistributions(K_L, A)
    B ← BaselineLikenessDistributions(A)
    U ← columnCount(K_H)                                             ▷ upper bound for EMDs
    δ ← [ ]_p                                                        ▷ population-level errors
    for i ∈ 1..p do
        δ_i ← EarthMover'sDistance(H_i, L_i)
        δ_i ← δ_i/U                                                  ▷ rescaling
        δ_i ← δ_i/B_i                                                ▷ random baseline adjustment
    end for
    return δ
end function
```

Additionally, we scale each population-wise score. This is first done such that a score of 0 (best) is perfect and 1 (worst) is the upper bound which represents largest possible EMD value. By default, we then adjust the score again such that 1 corresponds to a random embedding baseline (as described in the corresponding pseudocode in Algorithm 3).

Rather than summing or averaging population-wise errors to obtain an embedding-wise score, we inspect and compare the population errors to detect which parts of the input point cloud are the most mis-embedded. A proof-of-concept with synthetic data that compares xNPE to the unsupervised Local and Global SP scores is presented in the online *ViScore* repository.

Neighbourhood composition plots

To identify and visualise sources of embedding error at a population level, we use ***neighbourhood composition plots*** (NCPs). These plots visualise the composition of different-size neighbourhoods of a chosen labelled population of interest (POI) in HD data and its embeddings. This enables us to compare the position of the POI relative to other populations and how it is preserved in a given embedding.

For this, we query k nearest neighbours ($k = 1000$ by default) of each point in the POI, overlooking points in the POI itself. This is distinct from the xNPE approach described earlier, which makes the binary distinction between 'self' and 'non-self'. While neighbourhood composition plots do not directly quantify introduced errors, they visualise the type of error (*e.g.*, intrusion or extrusion of a specific population into or out of the neighbourhood of the POI). Composition of different-size (K-ary) neighbourhoods (1 to 100, with step size 10 by default) in terms of population labels is plotted for the POI as a stacked area chart, where each distribution of labels across the K axis is pooled for a range of neighbourhood ranks and for all points in the POI.

Model and hyperparameter specification

Model specifications and hyperparameters used in our quantitative comparisons of structure preservation (SP) by 12 DR methods, presented in *Results*, are given in *Supplementary Table 2*.

The full configuration for our comparative analysis of structure preservation by different DR methods (as presented in *Results*) is available in the *ViScore* GitHub repository. It includes installation scripts for each method and instructions for deployment on a high-performance computing (HPC) cluster. We use default hyperparameters for running each method, except for larger batch size of 1024 for VAE-based models. For consistency, PCA-reduced data is used as input to each method, with reference to (27; 23) as resources on standard preprocessing procedures. While *PHATE* can be applied directly to the sparse count matrices from scRNA-seq experiments, it performs an initial PCA reduction anyway.

For '*VAE*; '*ViVAE*', and '*ViVAE-EncoderOnly*' we use 50 training epochs as default, because this training procedure is typically feasible on a consumer laptop within a few minutes even without the use of a dedicated GPU for acceleration. However, a higher number of training epochs typically leads to better convergence and structure preservation with many datasets. '*VAE*' uses the same architecture and hyperparameters as '*ViVAE*', so as to serve as a fair reference in an ablation experiment for the stochastic-MDS loss regularisation. Furthermore, for the comparison to simultaneously serve as ablation study for denoising, each method is evaluated both on non-denoised and denoised inputs.

CPU-based methods (PCA, t-SNE (*scikit-learn* implementation), *UMAP* and *DensMAP* (*umap-learn* implementation), *PaCMAP*, *PHATE*, *TriMap*, and *SQuad-MDS*) were trained on a CPU cluster with an *Intel Xeon Gold 6140* processor. 16 GB of RAM were allocated for the processing of all datasets except *Suo*, for which 32 GB of RAM were allocated. GPU-based methods (VAE, *ivis*, *ViVAE*, and *ViVAE-EncoderOnly*) were trained on a GPU cluster with *Intel Xeon Gold 6242* and an *NVIDIA Volta V100* graphics card. 2 cores with 8 GB of GPU memory each were allocated.

Resource availability

Code and data to reproduce our study are openly available. We include instructions for downloading and preprocessing scRNA-seq datasets we used in our qualitative comparison in the *ViScore* GitHub repository, and in a Python code snippet below. The *scanpy* Python package (version 1.9.8) (67) is used, and the preprocessing procedure is based on that in (27), with the scaling step parametrised the same way as in the *ivis* workflow in (57).

```
import numpy as np
import scanpy as sc

hd = sc.read_h5ad('./scrnaseq.h5ad')

## If we filter by some condition (eg. tissue=='blood'):
# hd = hd[hd.obs['tissue']=='blood']
## If we are given raw counts:
# sc.pp.normalize_total(hd)
# sc.pp.log1p(hd)
sc.pp.scale(hd, max_value=10.)
sc.tl.pca(hd, svd_solver='arpack', n_comps=100)
pc = hd.obsm['X_pca']
np.save(f'pc.npy'), pc, allow_pickle=True)
```

Changes to this procedure or use of other data modalities may require corresponding changes to method hyperparameters, including individual loss term weights in *ViVAE*, for good performance.

ViVAE and *ViScore* are both available as Python packages. *ViVAE* itself, associated tutorials, and code used to run analyses in this paper can be found at `github.com/saeyslab/ViVAE`. The models are implemented using PyTorch and can be run with GPU acceleration (via CUDA or MPS). To boost accessibility for R users, especially from the cytometry community, we provide a tutorial on using *ViVAE* within the R programming language. *ViScore* is provided along with a documented extensible framework for running large benchmarks or hyperparameter sweeps and code to run a synthetic proof-of-concept experiment demonstrating differences between SP scored and the xNPE.

Acknowledgements

D.N. was funded by the FWO (Fonds Wetenschappelijk Onderzoek) and this research was conducted as part of the FWO Strategic Basic research project 1S40421N. C.d.B. is a beneficiary of an FSR Incoming Post-doctoral Fellowship from UCLouvain. P.L. is a FRIA grantee with the Belgian FNRS (Fonds National de la Recherche Scientifique). J.A.L. is a Research Director with the Belgian F.R.S.-FNRS (Fonds National de la Recherche Scientifique). S.V.G. is supported by an FWO postdoctoral research grant (1272823N, Research Foundation – Flanders). This work was supported by the Flemish Government under the Flanders AI Research Program (Project 174K02325).

We thank Sebastian Damrich for his invaluable advice regarding the implementation of encoder indicatrices and Anna Konstorum for her advice and guidance regarding supervised evaluation that allowed us to build on top of the Neighborhood Proportion Error algorithm.

Author contributions

D.N. conceived of and implemented the *ViVAE* methodology, supervised evaluation metrics in *ViScore*, and the large-scale evaluation framework, designed and ran experiments, and drafted the manuscript. C.d.B. conceived of and implemented the R_{NX} curve approximation and wrote the corresponding manuscript section. C.d.B., P.L., J.A.L., S.V.G., and Y.S. provided extensive feedback on the manuscript. Y.S. edited the manuscript. S.V.G. and Y.S. supervised the study.

Declaration of interests

The authors declare no competing interests.

References

[1] Adams H, Moy M (2021) Topology applied to machine learning: From global to local. Frontiers in Artificial Intelligence 4:668,302

[2] Almanzar N, Antony J, Baghel AS, et al (2020) A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. Nature 583(7817):590–595. https://doi.org/10.1038/s41586-020-2496-1, URL https://doi.org/10.1038/s41586-020-2496-1

[3] Amid E, Warmuth MK (2019) TriMap: Large-scale Dimensionality Reduction Using Triplets URL http://arxiv.org/abs/1910.00204

[4] Becht E, McInnes L, Healy J, et al (2019) Dimensionality reduction for visualizing single-cell data using UMAP. Nature Biotechnology 37(1):38–44. https://doi.org/10.1038/nbt.4314, URL https://doi.org/10.1038/nbt.4314

[5] Blondel VD, Guillaume JL, Lambiotte R, et al (2008) Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10):P10,008. https://doi.org/10.1088/1742-5468/2008/10/p10008, URL http://dx.doi.org/10.1088/1742-5468/2008/10/P10008

[6] de Bodt C, Mulders D, Verleysen M, et al (2022) Fast Multiscale Neighbor Embedding. IEEE Transactions on Neural Networks and Learning Systems 33(4):1546–1560. https://doi.org/10.1109/TNNLS.2020.3042807, URL https://ieeexplore.ieee.org/document/9308987

[7] Brun A, Knutsson H (2009) Tensor Glyph Warping: Visualizing Metric Tensor Fields using Riemannian Exponential Maps, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 139–160. https://doi.org/10.1007/978-3-540-88378-4_7, URL https://doi.org/10.1007/978-3-540-88378-4_7

[8] Böhm JN, Berens P, Kobak D (2022) Attraction-repulsion spectrum in neighbor embeddings. Journal of Machine Learning Research 23(95):1–32. URL http://jmlr.org/papers/v23/21-0055.html

[9] Carlsson G, Ishkhanov T, Silva VD, et al (2008) On the local behavior of spaces of natural images. International Journal of Computer Vision 76:1–12. https://doi.org/10.1007/s11263-007-0056-x

[10] Chari T, Pachter L (2023) The specious art of single-cell genomics. PLOS Computational Biology 19(8):1–20. https://doi.org/10.1371/journal.pcbi.1011288, URL https://doi.org/10.1371/journal.pcbi.1011288

[11] Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5):603–619. https://doi.org/10.1109/34.1000236

[12] Cormen TH, Leiserson CE, Rivest RL, et al (2009) Introduction to algorithms. MIT press

[13] Couckuyt A, Rombaut B, Saeys Y, et al (2024) Efficient cytometry analysis with FlowSOM in Python boosts interoperability with other single-cell tools. Bioinformatics 40(4):btae179. https://doi.org/10.1093/bioinformatics/btae179, URL https://doi.org/10.1093/bioinformatics/btae179, https://arxiv.org/abs/https://academic.oup.com/bioinformatics/article-pdf/40/4/btae179/57341836/btae179.pdf

[14] Ding J, Condon A, Shah SP (2018) Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nature Communications 9(1):2002. https://doi.org/10.1038/s41467-018-04368-5, URL https://doi.org/10.1038/s41467-018-04368-5

[15] Dong W, Moses C, Li K (2011) Efficient k-nearest neighbor graph construction for generic similarity measures. In: Proceedings of the 20th international conference on World wide web, ACM, pp 577–586

[16] Eling N, Morgan MD, Marioni JC (2019) Challenges in measuring and understanding biological noise. https://doi.org/10.1038/s41576-019-0130-6

[17] Elmentaite R, Domínguez Conde C, Yang L, et al (2022) Single-cell atlases: shared and tissue-specific cell types across human organs. Nature Reviews Genetics 23(7):395–410. https://doi.org/10.1038/s41576-022-00449-w, URL https://doi.org/10.1038/s41576-022-00449-w

[18] Farrell JA, Wang Y, Riesenfeld SJ, et al (2018) Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. Science 360. https://doi.org/10.1126/science.aar3131

[19] Gassen SV, Callebaut B, Helden MJV, et al (2015) FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry Part A 87:636–645. https://doi.org/10.1002/cyto.a.22625

[20] Graving JM, Couzin ID (2020) VAE-SNE: a deep generative model for simultaneous dimensionality reduction and clustering. BioRxiv

[21] Heiter E, Vandaele R, De Bie T, et al (2023) Topologically regularized data embeddings. arXiv preprint arXiv:230103338

[22] Heiter E, Martens L, Seurinck R, et al (2024) Pattern or Artifact? Interactively Exploring Embedding Quality with TRACE. URL https://arxiv.org/abs/2406.12953, 2406.12953

[23] Hu J, Wang Y, Zhou X, et al (2022) Pre-processing, Dimension Reduction, and Clustering for Single-Cell RNA-seq Data, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 37–51. https://doi.org/10.1007/978-3-662-65902-1_2, URL https://doi.org/10.1007/978-3-662-65902-1_2

[24] Huang H, Wang Y, Rudin C, et al (2022) Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization. Communications Biology 5. https://doi.org/10.1038/s42003-022-03628-x

[25] Jones RC, Karkanias J, Krasnow MA, et al (2022) The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. Science 376(6594):eabl4896. https://doi.org/10.1126/science.abl4896, URL https://www.science.org/doi/abs/10.1126/science.abl4896, https://arxiv.org/abs/https://www.science.org/doi/pdf/10.1126/science.abl4896

[26] Kingma DP, Welling M (2022) Auto-encoding variational bayes. URL https://arxiv.org/abs/1312.6114, 1312.6114

[27] Kobak D, Berens P (2019) The art of using t-SNE for single-cell transcriptomics. Nature Communications 10. https://doi.org/10.1038/s41467-019-13056-x

[28] Konstorum A, Jekel N, Vidal E, et al (2018) Comparative Analysis of Linear and Non-linear Dimension Reduction Techniques on Mass Cytometry Data https://doi.org/10.1101/273862, URL https://doi.org/10.1101/273862

[29] Lambert P, de Bodt C, Verleysen M, et al (2021) Stochastic quartet approach for fast multidimensional scaling. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, pp 417–422, URL http://hdl.handle.net/2078.1/252550

[30] Lambert P, de Bodt C, Verleysen M, et al (2022) SQuadMDS: A lean Stochastic Quartet MDS improving global structure preservation in neighbor embedding like t-SNE and UMAP. Neurocomputing 503:17–27. https://doi.org/10.1016/j.neucom.2022.06.108

[31] Lause J, Kobak D, Berens P (2024) The art of seeing the elephant in the room: 2D embeddings of single-cell data do make sense

[32] Lee JA, Verleysen M (2009) Quality assessment of dimensionality reduction: Rank-based criteria. Neurocomputing 72:1431–1443. https://doi.org/10.1016/j.neucom.2008.12.017

[33] Lee JA, Verleysen M (2010) Scale-independent quality criteria for dimensionality reduction. Pattern Recognition Letters 31(14):2248–2257

[34] Lee JA, Peluffo-Ordóñez DH, Verleysen M (2015) Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. Neurocomputing 169:246–261. https://doi.org/10.1016/j.neucom.2014.12.095

[35] Lopez R, Regier J, Cole MB, et al (2018) Deep generative modeling for single-cell transcriptomics. Nature methods 15(12):1053–1058

[36] Luecken MD, Theis FJ (2019) Current best practices in single-cell rna-seq analysis: a tutorial. Molecular Systems Biology 15(6):e8746. https://doi.org/10.1371/journal.pcbi.1011288, URL https://www.embopress.org/doi/abs/10.15252/msb.20188746, https://arxiv.org/abs/https://www.embopress.org/doi/pdf/10.15252/msb.20188746

[37] Maaten Lvd (2014) Accelerating t-SNE using tree-based algorithms. Journal of machine learning research 15(1):3221–3245

[38] Maaten LVD, Hinton G (2008) Visualizing Data using t-SNE. Journal of Machine Learning Research 9:2579–2605

[39] McInnes L, Healy J, Melville J (2018) UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction URL http://arxiv.org/abs/1802.03426

[40] Megill C, Martin B, Weaver C, et al (2021) cellxgene: a performant, scalable exploration platform for high dimensional sparse matrices. bioRxiv https://doi.org/10.1101/2021.04.05.438318, URL https://www.biorxiv.org/content/early/2021/04/06/2021.04.05.438318, https://arxiv.org/abs/https://www.biorxiv.org/content/early/2021/04/06/2021.04.05.438318.full.pdf

[41] Moon KR, van Dijk D, Wang Z, et al (2019) Visualizing structure and transitions in high-dimensional biological data. Nature Biotechnology 37(12):1482–1492

[42] Moor M, Horn M, Rieck B, et al (2020) Topological Autoencoders. In: III HD, Singh A (eds) Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 119. PMLR, pp 7045–7054, URL https://proceedings.mlr.press/v119/moor20a.html

[43] Muja M, Lowe DG (2014) Scalable nearest neighbor algorithms for high dimensional data. IEEE transactions on pattern analysis and machine intelligence 36(11):2227–2240

[44] Nazari P, Damrich S, Hamprecht FA (2023) Geometric Autoencoders - What You See is What You Decode. In: Krause A, Brunskill E, Cho K, et al (eds) Proceedings of the 40th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 202. PMLR, pp 25,834–25,857, URL https://proceedings.mlr.press/v202/nazari23a.html

[45] Novak D, Gassen SV, Saeys Y (2023) GroupEnc: encoder with group loss for global structure preservation. URL https://arxiv.org/abs/2309.02917, 2309.02917

[46] Ovchinnikova S, Anders S (2020) Exploring dimension-reduced embeddings with Sleepwalk. Genome research 30(5):749–756

[47] Pele O, Werman M (2008) A linear time histogram metric for improved sift matching. In: Computer Vision–ECCV 2008. Springer, pp 495–508

[48] Pele O, Werman M (2009) Fast and robust earth mover's distances. In: 2009 IEEE 12th International Conference on Computer Vision, IEEE, pp 460–467

[49] Qiu C, Martin BK, Welsh IC, et al (2024) A single-cell time-lapse of mouse prenatal development from gastrula to birth. Nature 626(8001):1084–1093. https://doi.org/10.1038/s41586-024-07069-w, URL https://doi.org/10.1038/s41586-024-07069-w

[50] Reed AD, Pensa S, Steif A, et al (2024) A single-cell atlas enables mapping of homeostatic cellular shifts in the adult human breast. Nature Genetics 56(4):652–662. https://doi.org/10.1038/s41588-024-01688-9, URL https://doi.org/10.1038/s41588-024-01688-9

[51] Sanmartín EF, Schnörr C, Hamprecht FA (2024) The Central Spanning Tree Problem. URL https://arxiv.org/abs/2404.06447, 2404.06447

[52] Shekhar K, Lapan SW, Whitney IE, et al (2016) Comprehensive Classification of Retinal Bipolar Neurons by Single-Cell Transcriptomics. Cell 166:1308–1323.e30. https://doi.org/10.1016/j.cell.2016.07.054

[53] Strati P, Li X, Deng Q, et al (2023) Prolonged cytopenia following CD19 CAR T cell therapy is linked with bone marrow infiltration of clonally expanded IFN-expressing CD8 T cells. Cell Reports Medicine 4(8):101,158. https://doi.org/https://doi.org/10.1016/j.xcrm.2023.101158, URL https://www.sciencedirect.com/science/article/pii/S2666379123003117

[54] Stuart T, Butler A, Hoffman P, et al (2019) Comprehensive integration of Single-Cell data. Cell 177(7):1888–1902.e21

[55] Stuchly J, Novak D, Brdickova N, et al (2023) Deconstructing Complexity: A Computational Topology Approach to Trajectory Inference in the Human Thymus with tviblindi https://doi.org/10.1101/2023.07.13.547329, URL http://dx.doi.org/10.1101/2023.07.13.547329

[56] Suo C, Dann E, Goh I, et al (2022) Mapping the developing human immune system across organs. Science 376(6597):eabo0510. https://doi.org/10.1126/science.abo0510, URL https://www.science.org/doi/abs/10.1126/science.abo0510, https://arxiv.org/abs/https://www.science.org/doi/pdf/10.1126/science.abo0510

[57] Szubert B, Cole JE, Monaco C, et al (2019) Structure-preserving visualisation of high dimensional single-cell datasets. Scientific Reports 9. https://doi.org/10.1038/s41598-019-45301-0

[58] Tang J, Liu J, Zhang M, et al (2016) Visualizing large-scale and high-dimensional data. In: Proceedings of the 25th international conference on world wide web, International World Wide Web Conferences Steering Committee, pp 287–297

[59] Tarhan L, Bistline J, Chang J, et al (2023) Single Cell Portal: an interactive home for single-cell genomics data. BioRxiv

[60] Tasic B, Yao Z, Graybuck LT, et al (2018) Shared and distinct transcriptomic cell types across neocortical areas. Nature 563(7729):72–78. https://doi.org/10.1038/s41586-018-0654-5, URL https://doi.org/10.1038/s41586-018-0654-5

[61] Triana S, Vonficht D, Jopp-Saile L, et al (2021) Single-cell proteo-genomic reference maps of the hematopoietic system enable the purification and massive profiling of precisely defined cell states. Nature Immunology 22(12):1577–1589. https://doi.org/10.1038/s41590-021-01059-0, URL https://doi.org/10.1038/s41590-021-01059-0

[62] Uhlmann JK (1991) Satisfying general proximity/similarity queries with metric trees. Information processing letters 40(4):175–179

[63] van Unen V, Höllt T, Pezzotti N, et al (2017) Visual analysis of mass cytometry data by hierarchical stochastic neighbour embedding reveals rare cell types. Nature Communications 8(1):1740. https://doi.org/10.1038/s41467-017-01689-9, URL https://doi.org/10.1038/s41467-017-01689-9

[64] Venkat A, Bhaskar D, Krishnaswamy S (2023) Multiscale geometric and topological analyses for characterizing and predicting immune responses from single cell data. Trends in Immunology 44(7):551–563. https://doi.org/https://doi.org/10.1016/j.it.2023.05.003, URL https://www.sciencedirect.com/science/article/pii/S1471490623000844, special issue: Systems immunology II

[65] Wang K, Yang Y, Wu F, et al (2023) Comparative analysis of dimension reduction methods for cytometry by time-of-flight data. Nature Communications https://doi.org/10.1038/s41467-023-37478-w

[66] Wang Y, Huang H, Rudin C, et al (2021) Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization. URL http://jmlr.org/papers/v22/20-1061.html.

[67] Wolf FA, Angerer P, Theis FJ (2018) SCANPY: Large-scale single-cell gene expression data analysis. Genome Biology 19. https://doi.org/10.1186/s13059-017-1382-0

[68] Xiang R, Wang W, Yang L, et al (2021) A Comparison for Dimensionality Reduction Methods of Single-Cell RNA-seq Data. Frontiers in Genetics 12:646,936. https://doi.org/10.3389/fgene.2021.646936, URL https://www.frontiersin.org/articles/10.3389/fgene.2021.646936/full

[69] Yang Z, Peltonen J, Kaski S (2013) Scalable Optimization of Neighbor Embedding for Visualization. In: ICML (2), pp 127–135

[70] Yianilos PN (1993) Data structures and algorithms for nearest neighbor search in general metric spaces. In: SODA, pp 311–321