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

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

[1]Department of Applied 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]ICTEAM/ELEN, Université catholique de Louvain, Place du Levant 3 bte L5.03.02, Louvain-la-Neuve, 1348, Brabant wallon, Belgium.

[4]IREC/MIRO, Université catholique de Louvain, Avenue Hippocrate 55 B1.54.07, Brussels, 1200, Brussels, Belgium.

*Corresponding author(s). E-mail(s): david.novak@ugent.be;
Contributing authors: cyril.debodt@uclouvain.be; pierre.h.lambert@uclouvain.be; john.lee@uclouvain.be; sofievangassen@ugent.be; yvan.saeys@ugent.be;

Abstract

The power to efficiently explore high-dimensional single-cell omics data is crucial for many biological applications. Dimensionality reduction techniques have therefore emerged as a basic building block of analytical workflows. By producing salient low-dimensional embeddings, they capture important structures in the data and are often used in visualisation, quality control, discovery and downstream analysis. However, current methods lack trustworthiness measures, resulting in the danger of misinterpretation. We tackle this with *ViVAE*: a deep learning model that achieves multi-scale preservation of structure and enhances interpretability through a set of novel, visual quality control techniques. In addition, we provide *ViScore*: a robust scoring framework that advances quantitative evaluation of embeddings with both unsupervised and supervised metrics that are scalable and fair. By applying our novel methods to a variety of datasets, we demonstrate the potential of our framework to spearhead the development of more trustworthy dimensionality reduction.

Keywords: single-cell omics, interpretability, dimensionality reduction, deep learning

Main

Data embedding using dimensionality reduction (DR) is commonly used in biological single-cell analysis to convert high-dimensional input data to low-dimensional projections (3; 26). These projections are suitable for describing cell type diversity (26), checking data quality after integration (54) or embedding proposed trajectory structures (18). The aim of single-cell DR generally is to improve interpretability of datasets which, in addition to being high-dimensional, are affected by noise (16).

In creating suitable lower-dimensional (LD) embeddings of high-dimensional (HD) data, DR methods differ in the criteria for which they optimise. Principal component analysis (PCA) is a popular deterministic method that obtains ordered orthogonal vectors that explain the most variance in the data, referred to as principal components (PCs). To reduce dimensionality, a set number of the top PCs can be taken as the basis for a LD vector space into which the HD inputs are projected. PCA is widely used to pre-process high-dimensional sparse data including scRNA-seq, as described in (22), typically applied to generate 50- to 100-dimensional, less sparse representations. As such, it is commonly used to generate inputs to other DR algorithms. Alternatively, the top 2 or 3 PCs can be taken directly to represent the data; however, this discards much of the variability and does not capture any non-linear relationships between features.

Various non-linear DR methods have been devised to overcome this limitation (51; 4; 10), with *t*-SNE (36) and *UMAP* (37) gaining special prominence in single-cell biology, as explored in (26; 69). These algorithms aim at preserving local neighbourhood structures, and are thus suitable for visualising clusters and comparing alternative label assignments. However, the inherent locality bias in *t*-SNE and *UMAP* can produce embeddings which are misleading in the relative positioning of cohesive cell clusters, and the embeddings can introduce artificial gaps between cell types and states (26; 34).

To overcome the limitations of focusing mostly on local structures, newly proposed algorithms such as *TriMap* (2) and *SQuad-MDS* (28) set out to preserve global relationships in input data better. Others aim to strike a favourable balance between local and global structure preservation, including *scvis* (14), *ivis* (57), *PHATE* (39), *fmSNE* (6) or *PaCMAP* (66). This is an important trend in the field, as it might reduce misinterpretation of *t*-SNE and *UMAP*, which are often used to visualise data with important global or hierarchical structures (60; 61). We elaborate on previous efforts to improve multi-scale structure preservation and chart an ontology of different DR approaches in Supplementary Note 4.

An interesting open challenge concerns the evaluation of the quality of DR algorithms. Recent comparisons have applied both unsupervised and supervised metrics to assess embedding quality (23; 65), often via measuring performance in subsequent inference taks, such as clustering or classification (27; 39; 57). However, these existing approaches exhibit some shortcomings. When scoring local structure preservation, setting a hard threshold for defining a 'local' neighbourhood remains a challenge. To assess larger-scale structures, some metrics circumvent issues with scalability by only using representative points determined via clustering, sacrificing rigor. Evaluating DR methods using downstream inference tasks introduces a proxy score that is fundamentally dependent on the classification or clustering algorithm used. This introduces a bias towards concordance with a particular machine learning approach, and it assumes that embeddings of very low dimensionality should be universally amenable to this downstream task. We elaborate on previously proposed evaluation approaches in Supplementary Note 7.

Some recent work has strongly criticised the use of non-linear DR in single-cell biology (9), questioning its overall usefulness in exploratory biological analyses and pointing out

failures to preserve distances. (30) respond to this critique by demonstrating that t-SNE and *UMAP* achieve neighbourhood preservation and separation of biologically relevant clusters of cells, even if pairwise distances are not preserved well. These recent discussions highlight the need for better frameworks to quantify the trustworthiness of DR methods, calling for new ways to interpret and evaluate them.

Here we present a novel framework for advancing trustworthy exploration of low-dimensional single-cell embeddings (Figure 1). The first component of our framework is *ViVAE*, a new deep learning-based DR method with demonstrable structure-preserving properties, accompanied by new quality control checks that facilitate better interpretation of the resulting embeddings. *ViVAE* achieves multi-scale neighbourhood preservation by making key improvements to the structure-learning capabilities of variational autoencoders (VAEs; (25)). To improve interpretability, we introduce encoder indicatrices as a novel method to detect local distortions of the learned latent space. The second component of our framework is *ViScore*, a flexible evaluation framework that tackles existing robustness and fairness concerns, and is designed to be scalable and reproducible, allowing for automated, large-scale evaluation (see Supplementary Note 2). *ViScore* implements both unsupervised and supervised evaluation methods that are fair and scalable, thereby overcoming some of the current limitations. The unsupervised approach enables distinct quantification of local and global structure preservation. The supervised approaches evaluate quality at a cell-population level, enabling validation of embedding structures in specific biological contexts. We describe all evaluation methodologies fully in *Methods*.

Results

We present *ViVAE*, a deep learning model that learns non-linear transformations of single-cell data in two steps (Figure 1**A**). Firstly, input data is denoised (smoothed) to prevent overfitting to noise during training. Secondly, the model is trained with the use of a custom regulariser for stochastic multi-dimensional scaling (MDS). Stochastic-MDS is conceptually related to the 'stochastic quartet' approach to DR proposed in (28), making crucial adaptations to yield a differentiable loss function applicable in a deep learning context.

Being a parametric model, *ViVAE* allows for the embedding of new data points from the same domain as original inputs. The model can therefore be trained on a subset of biological samples, for the trained encoder to subsequently transform new data as more measurements are acquired (Figure 1**B**), enabling, *e.g.*, revelation of batch effect or other technical artifacts across subsets.

Figure 1**C** shows *encoder indicatrices*, which we introduce as a novel quality control measure to inspect deformations introduced in the encoding from input to latent space. Encoder indicatrices build on previously developed methods grounded in differential geometry (42), presenting a way to examine the encoder-defined embedding transformation with respect to input data. This effectively reveals artificial expansion, shrinking or stretching of parts of the latent space.

The *ViVAE* model can be combined with *FlowSOM* (19), which couples clustering and a graph embedding procedure based on minimum spanning trees (MSTs), as shown in Figure 1**D**. This provides a new quality control measure that helps to validate the MST topology against the embedding (and vice versa), to examine whether expected ontologies or lineages are reflected (subfigure *(i)*). Projecting values from the input feature space onto representative clusters also quickly captures major sources of variability in the data, and can be used as an additional explanatory mechanism (subfigure *(ii)*). This is described further in Supplementary Note 3.

To facilitate comparisons of DR methods using multiple, complementary metrics, we developed *ViScore*, a new evaluation framework. *ViScore* includes an automated implementation of large-scale comparisons across many datasets, methods and hyperparameter configurations, which can be deployed efficiently using high-performance computing (HPC) clusters (described in Supplementary Note 2).

Figure 1**E** introduces R_{NX} curves (33) as the method of choice for assessing neighbourhood preservation across scales. While R_{NX} curves have been described earlier, we solve the outstanding issue of poor scalability to large datasets, introducing a novel approximation algorithm that renders their calculation feasible for large single-cell data. Furthermore, we newly derive two metrics tailored to measure either local or global structure preservation, and use them to provide a comprehensive comparison of *ViVAE* to other DR methods. Figure 1**F** *(i)* shows the Extended Neighbourhood-Proportion-Error (xNPE), a newly proposed metric for supervised (cell label-based) evaluation. This metric overcomes limitations of the NPE score, introduced in (27), and measures embedding quality per cell population. It is only sensitive to distortions that change relative positions of populations with respect to each other within the embedding. It is thus invariant to minor changes within a group of cells with the same label. A proof-of-concept study for xNPE is presented in Supplementary Note 11. As an additional quality measure, *ViScore* introduces neighbourhood composition plots (NCPs) for diagnosing sources of error (Figure 1**F** *(ii)*). NCPs visualise, for a given population of interest, which cells surround it in input space and in the evaluated embedding, using a range of different neighbourhood scales.

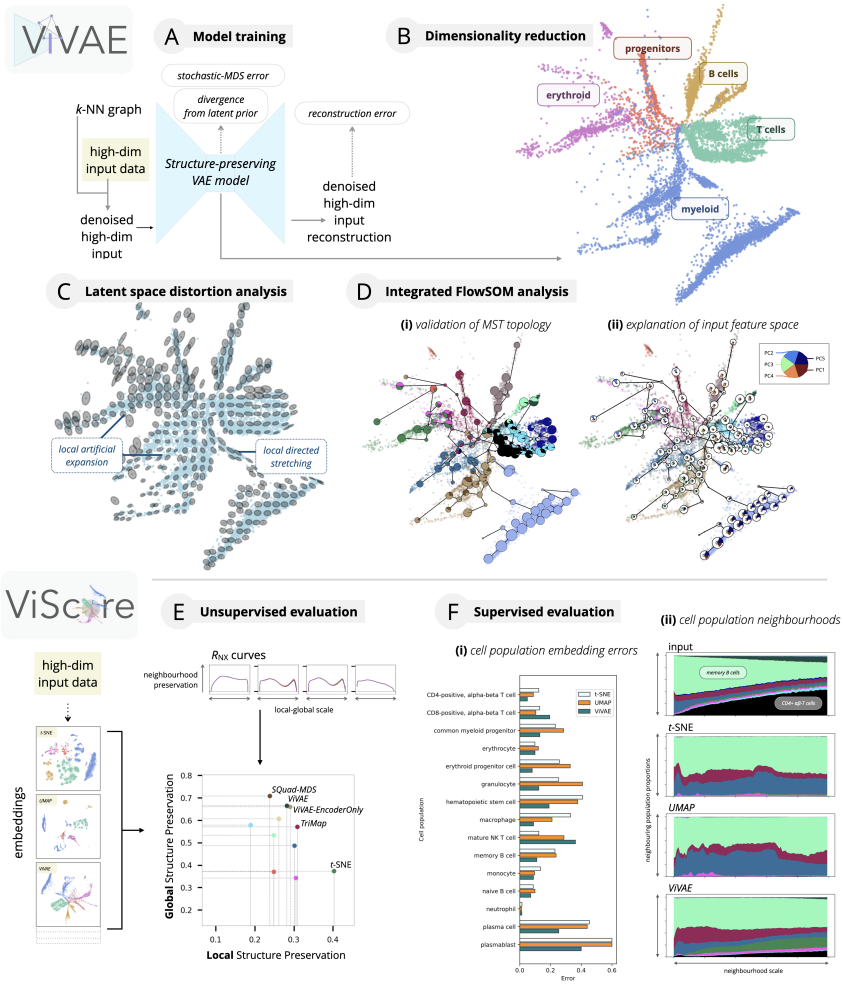

Fig. 1: **Schematic representation of proposed workflows. ViVAE–A.** Input data is denoised and used to train a regularised VAE model. **B.** The trained model is used to produce a low-dimensional embedding for visualisation. **C.** Encoder indicatrices are applied to evaluate distortions of the latent space learned by the model's encoder. **D.** The model can be used to *(i)* validate a *FlowSOM* tree topology and *(ii)* project input feature values onto representative clusters. **ViScore–E.** Unsupervised evaluation based on R_{NX} curve analysis quantifies neighbourhood preservation for local and global structures, separately. **F.** *(i)* Supervised evaluation via Extended Neighbourhood-Proportion-Error (xNPE) quantifies embedding error per cell population. *(ii)* Neighbourhood composition plots (NCPs) reveal inaccuracies in relative positions of cell populations in evaluated embeddings.

ViVAE strikes a favourable balance between local and global structure preservation

We present a comparison of 12 DR models with respect to their capacity to faithfully embed structures in 8 publicly available scRNA-seq datasets (47; 48; 56; 53; 1; 24; 52; 18) that we describe in Supplementary Note 1. We examine DR to 2 dimensions specifically, computing *Local* and *Global Structure Preservation* (SP) computed separately, so as to capture a trade-off between the two (see *Methods* for a detailed description).

Following a standard pre-processing protocol based on (26), expression matrices from each dataset were normalised and scaled and PCA was applied to extract orthogonal components as features, retaining 100 PCs as a more conservative approach. The following DR methods were compared: PCA, *UMAP* (37), *DensMAP* (a *UMAP* variant), *t*-SNE (36), *PHATE* (39), PaCMAP (66), TriMAP (2), SQad-MDS (28), ivis (57), VAE (a standard variational autoencoder) and our own contributions, *ViVAE* (default) and *ViVAE-EncoderOnly*. While '*ViVAE*' is trained using reconstruction loss, a variational regulariser and stochastic-MDS, '*ViVAE-EncoderOnly*' does not use reconstruction loss, omitting training of the decoder entirely.

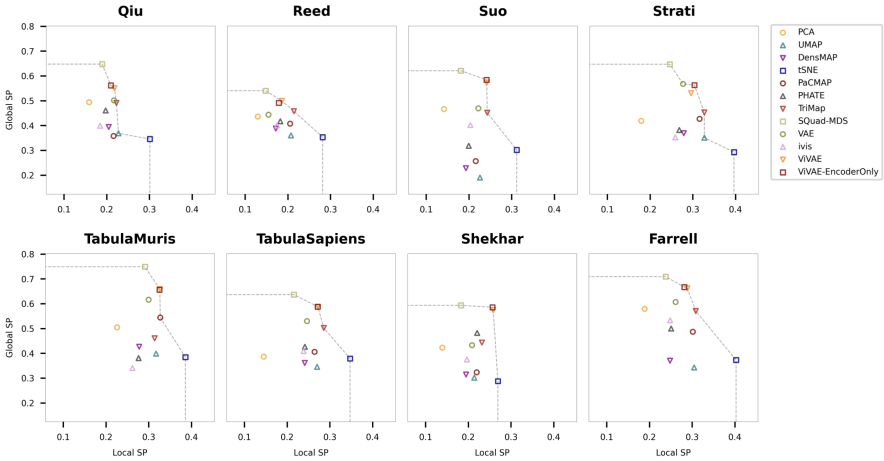

Fig. 2: **Local and Global Structure Preservation scores.** Mean Local and Global Structure Preservation (SP) values are shown for each benchmark dataset and DR method, with a subplot per dataset. The horizontal and vertical axes, respectively, quantify Local and Global SP, with common axis ranges across 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.

Figure 2 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. In cases where one of the two *ViVAE* models does not lie on the Pareto front, it is dominated only by the other one. Notably, *t*-SNE gives the highest Local SP and *SQuad-MDS* gives the highest Global SP across all tested datasets. *TriMap*, which lies on the Pareto front in 6 out of 8 experiments, tends to achieve similar Local SP to that of *UMAP*, while also improving Global SP, which aligns generally with past results in (23)). Supplementary Note 8 addresses separate influence of

denoising and stochastic-MDS on SP, stability of results. Supplementary Note 9 reports running times. Supplementary Note 13 shows embeddings from each experimental set-up (dataset-method combination).

Regularised VAEs achieve 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 which induce the ordering of cells within each lineage. As shown in our comparison results, '*ViVAE*' and '*ViVAE-EncoderOnly*' models achieve high global SP with this data, lower only than that of *SQuad-MDS*, which they exceed at local SP. Both *ViVAE* models improve local and global SP over a standard VAE.

Figure 3**A** shows *t*-SNE, *UMAP*, *SQuad-MDS*, VAE, default *ViVAE* and *ViVAE-EncoderOnly* embeddings, with embryonic stages marked by colour and schematic annotation of developmental gradients. As commonly observed, *t*-SNE and *UMAP* tend to separate data into clusters, whereas *SQuad-MDS* and the VAE-based models produce smoother embeddings. Of the smoother methods, *SQuad-MDS* shows the least separation of 11- and 12-*hpf* cells into distinct branches, with VAE or default *ViVAE* separating them the most visibly; a mapping of these separations onto trajectory endpoints is then shown in panel **B**. The smoother embeddings place a majority of 6-*hpf* shield phase cells clearly along the major temporal trajectory, in contrast to *t*-SNE and *UMAP* separating them out. The correctness of this embedding is supported by lower 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*. Panel **B** shows 6 distinctive developmental lineages mapped onto each embedding.

Figure 3**B** shows six distinctive developmental lineages, with each cell belonging to a given lineage coloured by its normalised rank within the trajectory. The terminal cells within 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, which terminate at 10.0-*hpf*. They contain few cells and are distinct from other lineages, as described in the original study. All six embeddings reflect this, with both lineages partially separating from others. Additionally, the ELC lineage shows a bifurcation. The pathway representations are highly fragmented in *t*-SNE and *UMAP*, while in the smooth embeddings they retain more continuity, despite the low numbers of cells in these lineages.

This leads to the question of whether the smooth embeddings represent transitional states correctly, given that this data is longitudinal, still allowing for gaps in trajectories. To test this, we quantify how continuous or fragmented each lineage is in the input space. In Figure 3**C**, we show L2 distances ('transitions') between cells ordered by pseudotime within the prechordal plate lineage. This is computed for input space and 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 the sparser representation of these cell subsets by smooth embeddings; *t*-SNE and *UMAP* collapse them artificially.

To compare the distortion level of transitions between embeddings, we report L1 distances between the input and embedded distance distributions, normalised by number of

cells within a lineage. These values (Figure 3**D**) show that the smooth DR models indeed achieve a more faithful representation of the cell lineages. Furthermore, Figure 3**E** shows transition distances in the ELC lineage to validate the diverging pathways we observe in embeddings. The separation is shown to be gradual, more similar to the representation by smooth DR models.

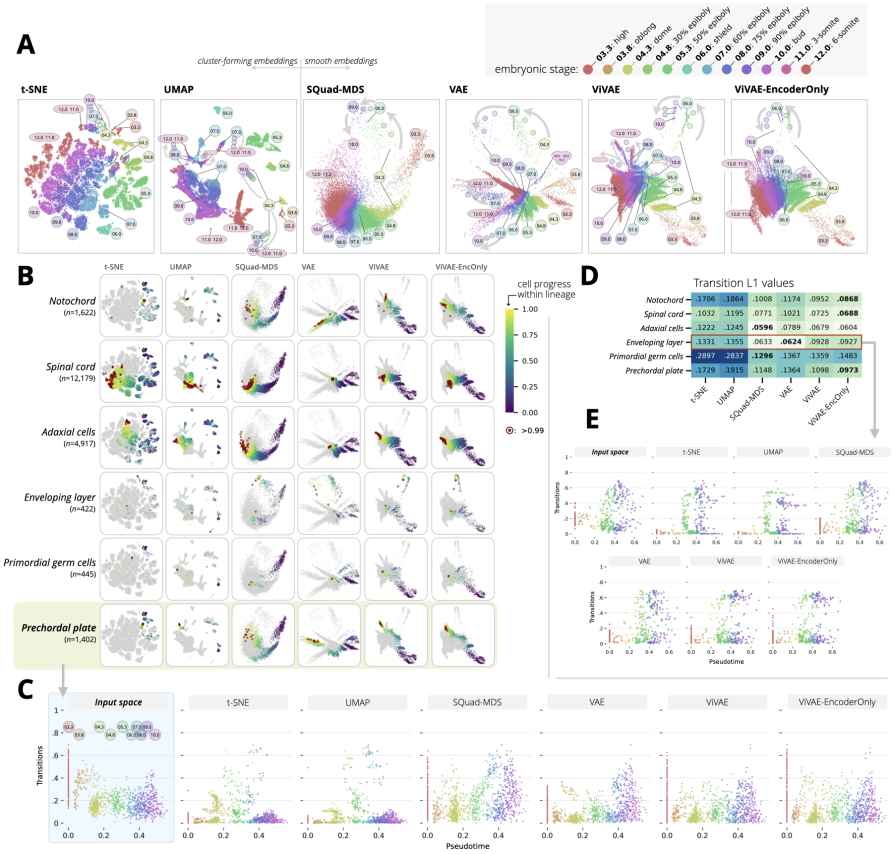

Fig. 3: ***Farrell* data embeddings with developmental lineages in zebrafish embryos.** **A.** 2-d embeddings by *t*-SNE, *UMAP*, *SQuad-MDS*, VAE, *ViVAE* and *ViVAE-EncoderOnly* are shown, with colour-coding by embryonic stage. The layout of cell subsets by stage is indicated in each embedding, showing 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 VAE-based latent space distortions

Using encoder indicatrices, we carry out an analysis of latent space distortion for VAE-based methods trained on the *Farrell* dataset. Figure 4**A** summarises the algorithmic novelty. Representative points are grid-sampled in an embedding from a trained encoder and mapped to their coordinates in input space. For each sampled point, a small hypersphere around it forms its HD indicatrix; using a local linearisation of the encoding, we then flatten the indicatrix and transform it using the encoder. This transformation introduces size and shape distortions, which are mapped onto the embedding coordinates to describe local embedding artifacts throughout the latent space.

Figure 4**B** applies indicatrices to highlight commonly observed embedding distortions in VAEs. First, inconsistencies in scaling make some parts of the latent space artificially sparser than others. Second, a 'run-away effect' causes directed stretching of some point subsets. Third, some sparse regions of the latent space are ill-defined, showing drastic extension where the the embedding should be smoother.

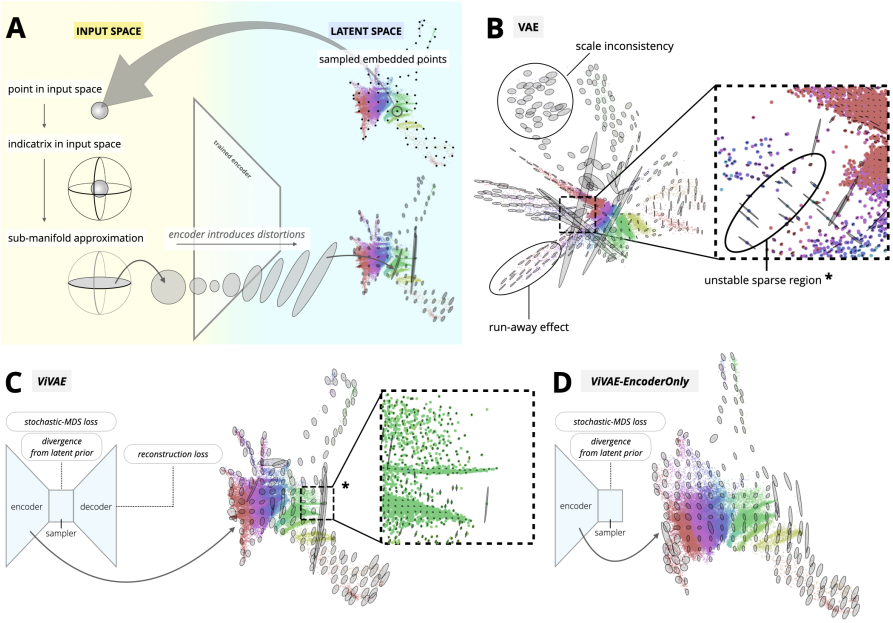

Fig. 4: **Analysis of latent space distortion in VAE-based embeddings of *Farrell* data. A.** Overview of *encoder indicatrix* computation. Points in input space are obtained via grid-sampling from the embedding. Using local linearisations of the trained encoder, HD indicatrices for the points are flattened along corresponding horizontal tangent spaces, yielding a piecewise approximation of the embedding 2-manifold. Flattened indicatrices are transformed by the encoder are plotted over the embedding. **B.** Encoder indicatrices 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.

Figure 4**C** shows that the default *ViVAE* model, with denoising and stochastic-MDS regularisation, gives more uniform indicatrices, reducing scale inconsistencies and runaway effects. Some instabilities in sparse regions, as seen with VAEs, remain. Figure 4**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, the analysis demonstrates 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

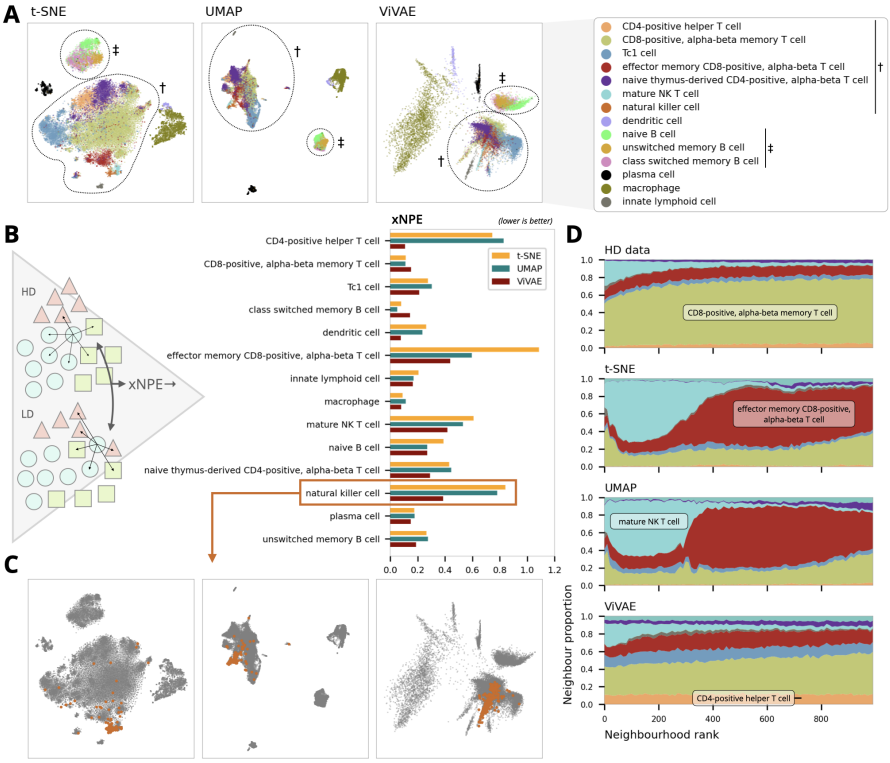

Fig. 5: ***Reed* dataset embedding and analysis of population-wise error. A.** 2-d embeddings *t*-SNE, *UMAP* and *ViVAE* are shown, with populations indicated by colour, according to legend on the right. Symbols mark the the T- and NK-cell (†) and B-cell (‡) compartments. **B.** xNPE (population-wise 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.

To demonstrate the usefulness of *ViVAE* as a general-purpose single-cell embedding method, we present a case study with data where cluster-forming embeddings can be reasonably expected to perform well. Specifically, we use the *Reed* immune cell dataset, originating from a single-cell transcriptomic atlas of the adult human breast (48). As opposed to the embryological zebrafish data, we expect more discontinuity between cell types as the breast tissue is seeded with distinct populations of mature immune cells, presenting a possible challenge to the smoother VAE-based embedding method.

Figure 5**A** displays the three embeddings, showing some degree of separation of major cell compartments in each. Recognising that DR invariably introduces artifacts, we seek to both quantify and describe them here, using a higher-resolution approach than embedding-level scoring. To this end, we compute the xNPE values for each cell population and display them in a barplot (Figure 5**B**). The xNPE aggregates the distribution of non-self neighbour counts (how many neighbouring cells belong to a differently-labelled population) over all cells in a population of interest. It does so separately for the points in input data and those in the evaluated embedding. These two distributions are then compared using Wasserstein distance (earth mover's distance). 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, even though it scores the highest in unsupervised Local SP. Crucially, in contrast to Local SP, xNPE focuses on points at the edges of each labelled population and, consequently, how the populations are positioned with respect to each other.

We further investigate the distribution of natural killer (NK) cells as a population of interest (POI), as all three embeddings place them in the same compartment as T cells, but assign them somewhat scattered positions within it, as shown in Figure 5**C**. Both *UMAP* and *ViVAE* outperform *t*-SNE here with respect to xNPE. In addition to quantifying distortion, we seek to explain sources of error in a more informative manner. To this end, we present neighbourhood composition plots (NCPs) for NK cells in Figure 5**D**. NCPs are stacked area plots showing smoothed proportions of neighbouring populations across different neighbourhood sizes (up to 1000), using a sliding window approach. These proportions are compared between input data and the different embdeddings.

The top NCP shows that, out of the labelled cell types, the NK cell population is close to a $CD8^+$ α-β T-cell population (olive green) in input space. All three embeddings underestimate this, as the population is graphically under-represented, but the *ViVAE* embedding does so the least. Notably, the *t*-SNE and *UMAP* embeddings show an intrusion of mature NK T cells (light blue) into a close neighbourhood of NK cells, meaning their proximity is exaggerated, while *ViVAE* avoids this artifact. The proximity of NK cells 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.

Discussion

We have shown *ViVAE* to achieve multi-scale structure preservation and faithful embedding of trajectory and non-trajectory data. Integrating a denoising and MDS component into a VAE model, *ViVAE* balances the continuous nature of autoencoder-based embeddings with an ability to separate major cell compartments. To improve interpretability, we equip *ViVAE* with novel quality-control functionality in the form of encoder indicatrices, which can be implemented for any differentiable DR model. In addition to a regularised VAE, we propose and evaluate a decoder-less model. We take the full VAE model as default due to slightly better preservation of local structures, but evaluate both, compare them and provide an open framework for hyperparameter tuning.

Beyond *ViVAE*, we implement novel model-agnostic evaluation measures in *ViScore*. 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; 38; 17) and ongoing efforts to systematically evaluate single-cell data analysis algorithms (34). 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 and extended a neighbourhood-based structure-preservation framework from (31) and notions from differential geometry described in (42), we estimate that an increased use of these 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), interactive tools for evaluating embedding quality at a local level (44; 21) and graph representations of data, possibly drawing from topological data analysis (in reference to (64) and (40)).

Our integration of *ViVAE* with *FlowSOM* (13) (demonstrated in Supplementary Note 3 is a tentative step toward the use of graphs together with embedded point clouds to capture relevant structures, and potential topologies in hierarchical data. Alternative graph structures, such as from the Louvain community detection algorithm (5) or witness complexes (used in (55)) could be used to substitute the *FlowSOM* MST. Recently proposed branched spanning trees robust to noise (50) may also prove useful in representing single-cell data.

Crucially, *ViVAE* is distinct from *UMAP* (37), *PHATE* (39) or *topological autoencoders* (40) in that it does not impose topological constraints or model input data as a graph structure. Instead, it optimises embeddings via repeated sampling and an MDS loss, which itself does not require a graph be built. This makes the stochastic-MDS loss applicable to any encoder-based model.

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 $\mathbf{Z_{n \times l}} \in \mathcal{Z}$ (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 pre-processing*) 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 from $\mathbf{X}$ and $\mathbf{\hat{X}}$, respectively. 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 (25) 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. We implement an optional pre-training procedure for PCA initialisation of (V)AEs, described in Supplementary Note 5.

Input denoising by smoothing

Our denoising algorithm, which we call *smoothing*, is based on an approach described in (8) and applied previously in (55); it is equivalent to the mean shift algorithm (11). The procedure is written out using pseudocode in Algorithm 1. Smoothing requires a k-nearest-neighbour graph (k-NNG) over pre-processed 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 smoothing to input data. One may experiment with these parameters, or apply smoothing to LD embeddings (using the k-NNG in input space) to achieve a sharper defined shape of embedded cell clusters or pathways.

According to our ablation study in Supplementary Note 10, 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 (pre-processed, 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 smoothing (see Supplementary Note 8), further demonstrating that the structure-preserving properties of *ViVAE* are not exclusively attributable to the denoising step.

Algorithm 1 Smoothing

```
function Smooth(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 ← rowCount(X)
    for iter ∈ 1..x_i do
        for i ∈ 1..n do
            idcs ← K_{i*}                                   ▷ neighbour indices
            target ← columnMeans(X_{idcs,*})                ▷ target coordinates
            shift ← target − X_{i,*}
            X_{i*} ← X_{i*} + λtarget
        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 multi-dimensional scaling (MDS) loss function. This is an adaptation of the stochastic quartet loss introduced in *SQuad-MDS* (28; 29), which we previously generalised and re-implemented as *group loss* for use in differentiable models in a preliminary study (43).

For stochastic-MDS, samples in each training mini-batch drawn from $\mathcal{X}$ 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 $\mathcal{X}$ and within the corresponding embedded quartet in $\mathcal{Z}$. 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 (43) 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 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 pre-processing workflow (described in *Data availability and pre-processing*) 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. We explore the usage of cosine distances in HD instead of the default Euclidean distances in Supplementary Note 6.

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 (42). 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 (42), 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 (42), 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 4**A**.

To compute encoder indicatrices, we begin by grid-sampling points from the latent space 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 ellipses. Note that the use of indicatrices with a linear AE would result in a degenerate case, which is explored in (42).

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, as demonstrated in Supplementary Note 3. 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 (31). 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 (33), 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 re-scaling 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 *Rank-based neighbourhood preservation scoring* 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 SP scores are both computed as the area-under-curve

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

with the two different scales for values of K. 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 advise 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 (32). 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 (71; 41; 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 (71; 62), being widely used to speed up nearest neighbour searches in big datasets (70; 35).

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 (71). 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 $\mathcal{O}(\log_2 N)$.

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 (27) 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) (45; 46) 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                               ▷ re-scaling
        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 Supplementary Note 11.

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 in 4, 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.

Data availability and pre-processing

We provide instructions needed for downloading and pre-processing scRNA-seq datasets we used in the benchmark in the *ViScore* GitHub repository. We use the *scanpy* Python package (version 1.9.8) (68) and apply the following pre-processing procedure, based on that in (26).

```
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 changes in method hyperparameters (including individual loss term weigths in *ViVAE*) for good performance. The *Samusik* mass cytometry dataset (49) used in Supplementary Note 3 was obtained using the *HDCytoData* R package (67). We applied the recommended pre-processing instructions (using an inverse hyperbolic sine transformation with cofactor 5: $X \leftarrow \operatorname{arcsinh}(X/5)$), kept columns corresponding to 'type' markers and exported the expression matrix and cell labels as a NumPy binary file. We used loss term weights $l_{\mathrm{recon}} = 1, l_{\mathrm{kldiv}} = 1, l_{\mathrm{MDS}} = 10$ to train the *ViVAE* model.

Code availability

ViVAE and *ViScore* are both available as Python packages. *ViVAE* itself, associated tutorials and code to run analyses in this paper can be found at `github.com/saeyslab/ViVAE`. The models are implemented using PyTorch and are best 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 reproduce the synthetic proof-of-concept experiment in Supplemetary Note 11. It is available at `github.com/saeyslab/ViScore`. Supplementary Note 12 elaborates on ensuring reproducibility.

Acknowledgements

D.N. is funded by the FWO (Fonds Wetenschappelijk Onderzoek) and this research was conducted as part of the FWO Strategic Basic research project 1S40421N. CdB is a beneficiary of an FSR Incoming Post-doctoral Fellowship from UCLouvain. J. A. Lee 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).

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

References

[1] 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

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

[3] 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

[4] Belkin M, Niyogi P (2001) Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In: Dietterich T, Becker S, Ghahramani Z (eds) Advances in Neural Information Processing Systems, vol 14. MIT Press, URL https://proceedings.neurips.cc/paper_files/paper/2001/file/f106b7f99d2cb30c3db1c3cc0fde9ccb-Paper.pdf

[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] 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

[9] 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

[10] Coifman RR, Lafon S, Lee AB, et al (2005) Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the National Academy of Sciences 102(21):7426–7431. https://doi.org/10.1073/pnas.0500334102, URL https://pnas.org/doi/full/10.1073/pnas.0500334102

[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, 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

[22] 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

[23] 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

[24] 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

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

[26] 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

[27] 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

[28] 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

[29] 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

[30] 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

[31] 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

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

[33] 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

[34] 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

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

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

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

[38] 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

[39] 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

[40] 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

[41] 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

[42] 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

[43] 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

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

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

[46] 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

[47] 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

[48] 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

[49] Samusik N, Good Z, Spitzer MH, et al (2016) Automated mapping of phenotype space with single-cell data. Nature Methods 13:493–496. https://doi.org/10.1038/nmeth.3863

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

[51] Schölkopf B, Smola A, Müller KR (1998) Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10(5):1299–1319. https://doi.org/10.1162/089976698300017467, URL https://doi.org/10.1162/089976698300017467, https://arxiv.org/abs/https://direct.mit.edu/neco/article-pdf/10/5/1299/813905/089976698300017467.pdf

[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] Weber LM, Soneson C (2019) HDCytoData: Collection of high-dimensional cytometry benchmark datasets in Bioconductor object formats. F1000Research 8:1459. https://doi.org/10.12688/f1000research.20210.1

[68] 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

[69] 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

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

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