A framework for quantifiable local and global structure preservation in single-cell dimensionality reduction

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

Dimensionality reduction techniques are essential in current single-cell 'omics approaches, offering biologists a first glimpse of the structure present in their data. These methods are most often used to visualise high-dimensional and noisy input datasets, but are also frequently applied for downstream structure learning. By design, every dimensionality reduction technique preserves some characteristics of the original, high-dimensional data, while discarding others. We introduce *ViScore*, a framework for validation of low-dimensional embeddings, consisting of novel quantitative measures and visualisations to assess their quality in both supervised and unsupervised settings. Next, we present *ViVAE*, a new dimensionality reduction method which uses graph-based transformations and deep learning models to visualise important structural relationships. We demonstrate that *ViVAE* strikes a better balance in preserving both local and

global structures compared to existing methods, achieving general-purpose visualisation but also facilitating analyses of developmental trajectories.

Keywords: single-cell data, dimensionality reduction, deep learning, denoising

1 Introduction

Dimensionality reduction (DR) is an essential component of single-cell analysis pipelines, applied in transcriptomics and cytometry. (4; 22) Enabling detection of RNA transcripts or labelled proteins, respectively, helps characterise each cell in a tissue sample of interest on a previously unattainable scale. Applications of transcriptomic analyses using single-cell RNA sequencing (scRNA-seq) include the study of immune system dynamics (49), developmental trajectories (16; 45; 11) or intercellular communication (7), with recent notable progress in spatial transcriptomics in particular (55; 60). Flow and mass cytometry analyses (38; 41) are deployed to describe cell populations and transient cell states based on expression levels of proteins of interest, with a wide array of scientific and medical applications (31; 46; 54). scRNA-seq and cytometry datasets are inherently noisy (14) and high-dimensional, and their computational analysis, including DR, must be adapted to these properties.

DR refers to a class of methods that isolate or construct a limited set of features that preserve important structures from the high-dimensional (HD) input data while transforming it into a lower-dimensional (LD) embedding (most commonly 2-dimensional for visualisation purposes). Since visualising HD data is non-trivial, learning useful representations helps to get an overview of cell populations and their relative positions within the feature space.

A well-established deterministic method of DR via feature engineering is principal component analysis (PCA). PCA computes principal components (PCs) as linear combinations of original features, computed iteratively to explain the most possible variance in the original data, while remaining orthogonal to each other. PCA, while used broadly to pre-process high-dimensional sparse data (including scRNA-seq) (21), is (by its definition) unable to capture non-linear relationships between features, which limits it considerably when used for visualisation.

Various non-linear methods have been devised since (47; 5; 9), with t-SNE (35) and UMAP (37) gaining special prominence in single-cell biology (22; 62). The high degree to which t-SNE, a stochastic kernel-based embedding algorithm, preserves local neighbourhood structures of cell embeddings (which we validate in this paper) is useful in retaining clusters of cell populations present in the analysed tissue. In a similar vein, *UMAP* reduces dimensionality while preserving local neighbourhoods using an approach drawing considerably from topological data analysis. Since its release, *UMAP* has positioned itself as another standard DR technique, able to model relationships between cells on a local level (4; 42).

Crucially, the local structure-preserving objective can lead to embeddings that are misleading in the relative positioning of cohesive clusters (global structure) and the obscuring of connections between subpopulations (22; 33). Past efforts have partially compensated for this via PCA-based initialisation, adaptive setting of perplexity (a hyperparameter controlling sizes of the relevant local neighbourhoods) and other optimisation parameter changes (23). PCA initialisation, however, suffers from large populations contributing overwhelmingly to the first few PCs, while the global relationships that involve smaller populations are not captured, and larger perplexity values can only expand the relevant local neighbourhood up to

a point (rather than forcing the embedding algorithm to look past it). UMAP, while claimed to preserve global structures better than t-SNE, is also limited to optimising layouts of the nearest neighbourhoods of any given embedded point, and does not optimise global structure preservation (4; 22).

More recently proposed algorithms (*TriMap* (1), *SQuadMDS* (26), *fmSNE* (6)) set out to preserve global relationships better or to strike a new balance between local and global structure preservation (*ivis* (50), *PaCMAP* (57)). We welcome this trend, due to the promise of reducing misinterpretation of t-SNE and *UMAP* as two popular, locally focused methods that are used heavily to visualise data with important hierarchical or global structures.

Given the large size of single-cell datasets, along with the non-trivial nature of evaluating correctness of embeddings, benchmarking DR algorithms is an open problem. (We discuss some previously proposed evaluation metrics in 6.1.) In this work, we motivate and introduce novel scalable methods for unsupervised evaluation of local and global structure preservation (distinctly), as well as supervised population-level evaluation criteria for validation in specific biological contexts. We implement these metrics in *ViScore*, a Python package, to provide an evaluation framework for our new method, as well as a general tool for DR pipelines.

Having a powerful evaluation framework allowed us to develop a new method that balances structure preservation at local and global scales better than the state-of-the-art. To this end, we developed *ViVAE*: a new DR algorithm and stand-alone Python package. *ViVAE* combines the ability of variational autoencoders (VAEs), a family of neural networks, to compress HD data into LD latent representations by learning a series of non-linear data transformations, with recently published and novel optimisation criteria that improve preservation of HD structures in the resulting embeddings. We compare *ViVAE* to other methods in a quantitative comparison, showing trade-offs between preserving local and global structures and how *ViVAE* strikes a balance between these criteria well.

2 Results

Extraction of salient features, invariably, affects downstream analysis, and the distortions inherent to DR increase the risk of faulty reasoning about data. The contribution we make is two-fold: *(1)* an easy-to-use framework with novel quantitative metrics and intuitive visualisations for comparing candidate embeddings of a selected dataset based on their truthfulness and *(2)* a new, generally applicable DR method that is shown to strike a balance between preserving local and global relationships within HD data.

Figure 1 depicts the *ViVAE* algorithm for generating informative lower-dimensional embeddings of single-cell data, as well as the *ViScore* evaluation: firstly in an unsupervised setting and secondly with the use of known cell labels, to diagnose sources of error.

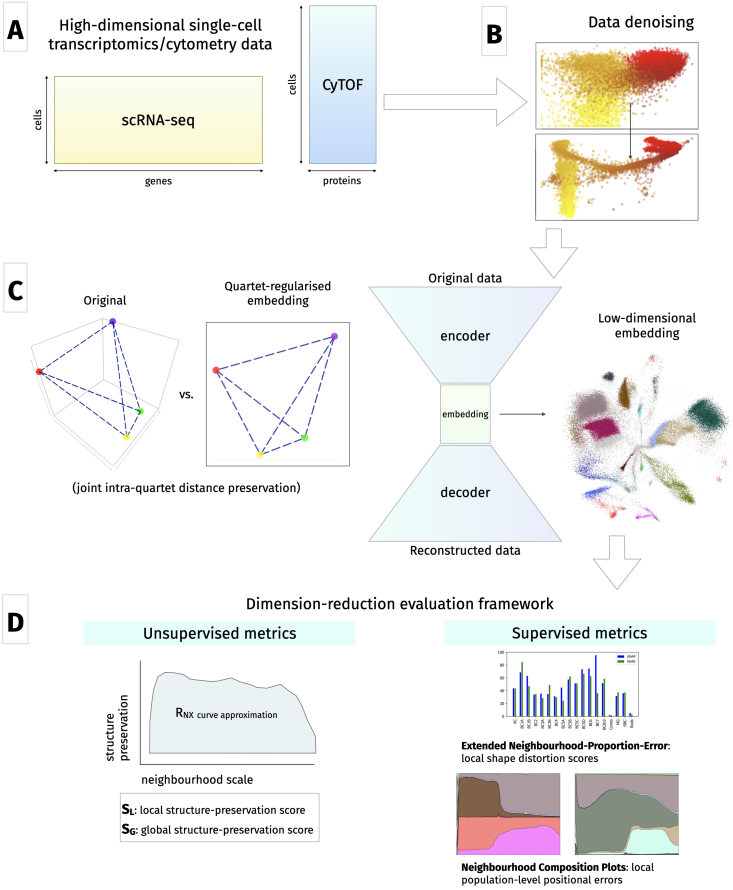


Fig. 1: **Schematic overview of the contributions of this paper. A.** High-dimensional single-cell data, typically from single-cell RNA-sequencing (scRNA-seq) or mass cytometry (CyTOF) are the intended inputs in our DR workflow. **B.** *ViVAE* first applies denoising based on nearest-neighbour graphs to improve embedding quality downstream. **C.** A dimension-reduction model based on a variational autoencoder (VAE) architecture and quartet loss regularisation, adapted from *SQuadMDS* (26), is trained using denoised data. Normalised distances within randomly drawn quartets of points are optimised jointly, so as to impose a multi-scale structure-preservation constraint on the latent space. **D.** A unified *ViScore* evaluation framework offers unsupervised scores for local and global structure preservation and supervised scores for quantifying shape distortion or position errors of annotated populations.

2.1 Unsupervised and supervised evaluation metrics help quantify performance of dimensionality reduction methods with particular datasets

To help users in comparing dimensionality reduction methods and choosing an appropriate one, our first contribution consists of a quantitative evaluation framework, applicable to both

high-dimensional (HD) and large datasets. This allows the user to evaluate candidate methods on their dataset of interest, based on objective criteria of structure preservation (SP), rather than subjective judgement.

We propose two types of scores in our evaluation framework: unsupervised scores that can be applied to any dataset, and supervised scores, applicable to datasets where additional prior knowledge (*e.g.*, knowledge of cell types or trajectory structures) is present. The former metrics allow to assess the trustworthiness of low-dimensional embeddings using only the data itself, while the latter metrics allow using prior structural knowledge that a biologist already has, allowing to quantify how well certain regions in the low-dimensional embeddings conform to the prior.

Our unsupervised scoring methodology quantifies preservation of both local and global structures, but does not use a hard cut-off for neighbourhood size to make a distinction between 'local' and 'global' SP. We argue that this difference is inherently fuzzy, therefore using a binary distinction between local and global is unnecessarily distorting. Our truly global score, S_{G}, assesses preservation of all relative positions across neighbourhood sizes. The local score, S_{L}, also takes into account relative positions across all scales, but the nearest neighbour positions are strongly up-weighted. Both of these scores are area-under-the-curve (AUC) measures. In addition, the harmonic mean F of S_{L} and S_{G} can be used to indicate a balance between the two quality indicators. (We use these scores in 2.2 to benchmark our method against previously published tools.)

Unsupervised metrics are most interesting to us due to their objectivity and stringency. They lack bias with regard to cell type labelling or any other formulation of 'ground truth', which can itself be inaccurate. On the other hand, we believe supervised evaluation measures are also important in explaining sources of embedding error in an interpretable way. Gaining information about structural distortion on a cell-population level conveys a level of trustworthiness associated with the embedding of a chosen population, before the user makes inferences about their data. Therefore, we propose unsupervised scoring be applied for overall benchmarking of performance, and supervised scoring be optionally used downstream to elucidate the source of error in a given embedding.

To this end, we provide two novel quality control metrics for evaluating how correctly specific cell populations are embedded (described in Methods and Supplementary Information). Firstly, we propose the Extended Neighbourhood-Proportion-Error (xNPE), based on the previously published Neighbourhood Proportion Error (24). The xNPE measures population-level distortions comparable between different embeddings of the same data. This allows the user to pinpoint which part of an embedding is most problematic, rather than having a single score for the entire embedding. Secondly, we introduce *neighbourhood composition plots* to evaluate positional embedding error, allowing to compare the relative positions of known populations between original data and an embedding of interest, or between different embeddings of the same dataset. (We show a use-case of these evaluation criteria in 2.4.)

2.2 *ViVAE* strikes a good balance between preserving global and local structures in low-dimensional embeddings

ViVAE denoises high-dimensional input data and learns a lower-dimensional representation using a variational autoencoder (VAE) architecture, while imposing a structure-preserving constraint to optimise both local and global distances between points. This constraint, the *quartet loss*, is an adaptation of the main idea in *SQuadMDS* (26). It seeks to preserve relative distances within quartets (groups of 4) of points that are randomly (repeatedly) drawn

out of the input dataset and the lower-dimensional latent embedding of it. Crucially, the network is trained so as to optimise all the intra-quartet distances jointly at each iteration. We provide a methodology for quantifying both local and global structure preservation (SP), as well the balance between these two criteria, to ensure a means to compare DR methods based on objective criteria.

In our comparative analysis, we apply multiple existing DR algorithms as well as our newly proposed one. These are principal component analysis (PCA), *t*-SNE (35), *UMAP* (37), *PaCMAP* (57), *TriMap* (1), *SQuadMDS* (26), *ivis* (50) and *ViVAE*. We test all methods on a collection of large scRNA-seq (48; 16; 63; 43; 67) and mass cytometry (CyTOF) (31; 46) datasets to evaluate SP in their respective low-dimensional embeddings. We chose these datasets due to their size and, in the case of mass cytometry (3), relatively high dimensionality within the cytometry domain. We describe the selected datasets in Table 1. (See Supplementary Information for more information about the data and pre-processing steps applied to each of the datasets.)

Dataset name	**Biological source**	**Technology**	**# of features**	**# of datapoints**	**# of populations**
Ziegler	human nasopharynx	scRNA-seq	32,871	32,588	39
Shekhar	mouse retina	scRNA-seq	24,904	44,994	19
Ximerakis	mouse brain	scRNA-seq	14,699	37,069	25
Farrell	zebrafish embryos	scRNA-seq	17,239	38,731	12
Park	human thymus	scRNA-seq	33,694	125,810	55
Samusik	mouse bone marrow	CyTOF	39	86,864	24
Levine	human bone marrow	CyTOF	32	265,627	14

Table 1: Datasets used for comparison of DR methods. We name them according to author names of publications with which the datasets are associated. '# of features' refers to the number of detectable proteins in the case of CyTOF data, and to the number of detectable gene transcripts in the case of scRNA-seq data. '# of populations' refers to manual labelling of individual cells provided by the authors (the set of labels used may be one of multiple sets provided).

Our results, visualised in Figure 2, show that while *t*-SNE excels at preservation of local neighbourhoods, alternative methods exist that preserve global structure better. *ViVAE* preserves local neighbourhoods mostly to a similar degree as *UMAP*, *PaCMAP* or *TriMap*, while achieving better global structure preservation than these methods. *SQuadMDS* is consistently the best at keeping global relationships least distorted, but local structures are not preserved as well. *ViVAE* consistently strikes a good balance between local and global structure preservation, obtaining the highest F-scores, thus serving as an appropriate general-purpose DR method for single-cell data. (See 6.2 in Supplementary Information for a full table of scores.)

2.3 *ViVAE* improves visualising developmental pathways in zebrafish development

We have shown that *ViVAE* preserves global and local structures to a high degree, using a collection of single-cell RNA-sequencing and mass cytometry datasets for testing. We have quantified this by measuring how much distortion occurs in the form of change in neighbourhood ranks (relative distances between measured cells) as we go from the original high-dimensional space to a 2-dimensional latent space created by the model.

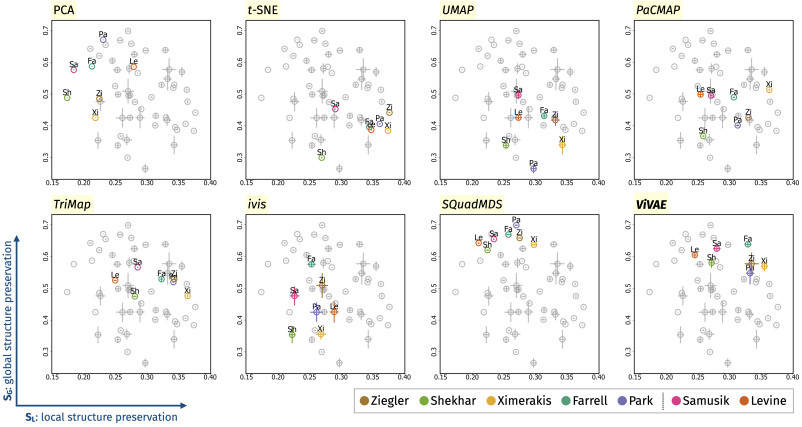


Fig. 2: **Balance between local and global structure preservation.** In each panel, the positions of circled points indicate mean values of S_L and S_G scores for a particular dataset-method combination (higher, therefore top and right, is better), as reported in Table 2. Lines drawn through the points indicate standard deviation. In each panel, only points corresponding to the method in the title are coloured in, with 2-letter initial abbreviations for dataset names (see Table 1) shown just above. The rest of the data points (those for other methods) are plotted in grey.

To illustrate the performance of *ViVAE* in a specific context, we use the case study of the Farrell zebrafish embryos data (16), collected using Drop-Seq (36) and analysed using URD, a trajectory analysis tool the authors propose. They conducted a semi-supervised analysis involving the computation of diffusion pseudotime (20), labelling of terminally differentiated cell populations and detection of branch-points via repeated simulation of random walks. This approach reconstructed trajectories described in the literature, as well as revealing branching events and gene expression patterns that were previously unknown. Terminally differentiated cell stages were specified manually in the course of the trajectory inference pipeline. We note that an initial *t*-SNE embedding of the normalised single-cell transcript counts did not immediately make the directedness of development apparent.

A major goal of ours is to maximise the usefulness of unsupervised data visualisation in similar analyses of development (*e.g.*, in trajectory inference pipelines). The aim is to reveal structures that turn out to be largely concordant with the developmental pathways of distinct cell lineages, *without* actively modelling tentative trajectories. This keeps the algorithm generally applicable.

Figure 3 compares 2-dimensional embeddings of the transcriptomic profiles of the embryonic cells, with distinct curated developmental lineages highlighted separately. Furthermore, we show the positions of embedded points separated by pseudotemporal stages, as annotated by the authors, using a colour scale, and we highlight terminally differentiated populations ('leaves' in the developmental tree), provided as prior information in the trajectory analysis, and a point with the lowest pseudotime value, corresponding to the cell of origin (progenitor).

In all of the embeddings, some pathway-like structures are apparent once cells are labelled by pseudotemporal stages. However, in the case of *t*-SNE, the overall layout, without additional annotation, does not show clear directedness from origin to terminal populations. The *UMAP* and *PaCMAP* embeddings, similar to each other overall, appear less generic at

first glance, and some directedness is apparent. However, both of them suffer from a high degree of fragmentation into distinct islands, which obscures the gradient from progenitors toward increasingly differentiated cells. *TriMap*, *ivis*, *SQuadMDS* and *ViVAE* embeddings, scoring higher in S_{G} (global structure preservation), achieve a layout where transitions between cells within the same lineage are more gradual, with less fragmentation.

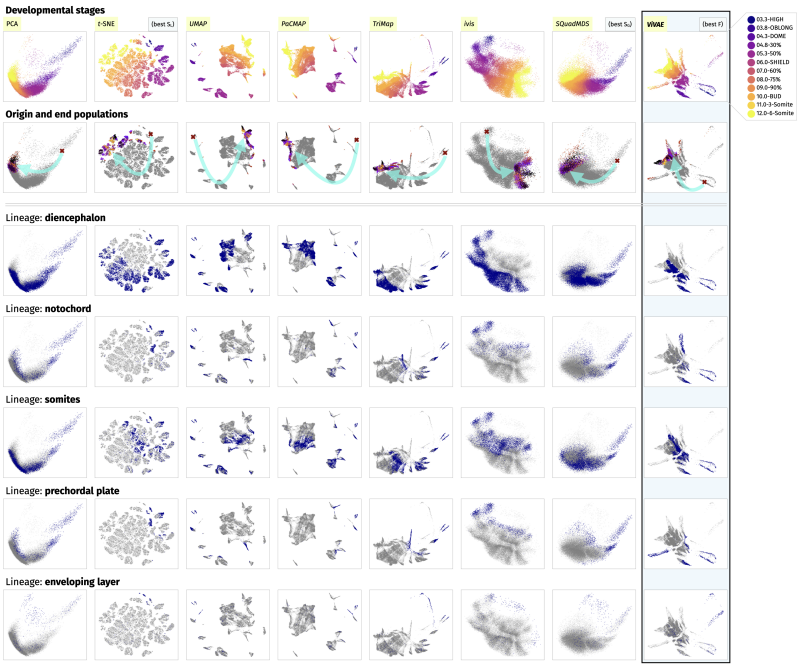


Fig. 3: **Pathways in zebrafish embryo transcriptomic data.** 2-dimensional embeddings of scRNA-seq data (16), generated using different methods, are shown in each row. The first row highlights developmental stages based on pseudotime simulations from the original study, independent of any dimensionality reduction, projected onto the embeddings. In the second row, a red 'X' denotes the point with lowest pseudotime, corresponding to a progenitor cell of origin. Other colours distinct from grey give an overview of populations which are identified, a priori, as terminally differentiated populations (*ie.* 'leaves' of the corresponding developmental tree). An arrow is drawn to denote roughly how the developmental gradient maps onto each projection (from progenitor toward more differentiated cells). The following rows use dark blue to highlight cells which fall into curated and labelled developmental lineages, which represent distinct pathways in development, while following the pseudotemporal gradient.

The *ViVAE* embedding, with the highest F score, quantifying the balance between S_{L} and S_{G} (see Table 2), achieves a more distinct separation of endpoints than *SQuadMDS*, and overall a high degree of contrast between the individual lineages, as projected onto the embedded points, compared to other embeddings.

2.4 Supervised evaluation metrics quantify and qualify embedding errors in mouse retina analysis

Our evaluation framework allows for interrogation of specific embeddings on a cell population level. As stated above, we provide the *Extended Neighbourhood-Proportion-Error* (xNPE) algorithm for pinpointing population embeddings with relatively high level of distortion. xNPE requires a 'ground truth', in the form of a mapping of single measured cells onto known populations. The scores computed using xNPE map onto individual populations, rather than the entire embedding. Contrasting xNPE scores of multiple embeddings of a single dataset can reveal the difference between how faithfully a particular population is represented. (See 4.2.1 for description of algorithm.)

Further, we present *neighbourhood composition plots* for describing errors in relative positioning of populations within an embedding. This is shown in terms of how points from other populations invade or leave a close neighbourhood of a population of interest, and can be compared between the HD input data and an embedding of interest, or between different candidate embeddings. (See 4.2.2 for description of algorithm.)

We show an example comparison of *UMAP* and *ViVAE*, with *ViVAE* comparatively more geared toward global structure preservation. We embed the Shekhar (48) mouse retina scRNA-seq dataset using both of these methods. Figure 4 shows a subsequent supervised analysis of the results. Panel *A.* shows differences in xNPE scores of *UMAP* and *ViVAE* for each labelled population (cells labelled '*nan*' and '*Doublets/Contaminants*' were aggregated as '*NA/Debris*' and omitted from this evaluation). We inspect the two populations of bipolar retinal cells with a notably higher embedding error in the UMAP embedding: *BC5A* and *BC7*. We show that UMAP separates these into distinct islands (a manifestation of experimental batch effect specific to the data that applies to more than these two populations), whereas *ViVAE*, while showing a separation into two sub-groups in each case, keeps the populations compact, avoiding the artificial de-coupling of these points and achieving a lower level of shape distortion.

Further analysis describes the change in relative positions between populations involved in embedding the dataset. We chose to inspect population *BC5A*, since its embeddings within the *UMAP* and *ViVAE* plot clearly differ in terms of which other populations surround them. Panel *B.* shows insets (zoomed-in areas) of the local neighbourhoods of the population, with text labels for surrounding cells. Note that the *ViVAE* plot appears smoother, which is due to the continuous nature of data embedding with variational autoencoders.

Panel *C.* shows neighbourhood composition plots for *BC5A* cells, which we label as the population of interest (POI). These plots visualise populations in different neighbourhoods (by rank) of the POI. The neighbourhood scale K is varied from 1-10 up to 991-1000, in increments of 10. This way, neighbours are shown for different neighbourhood rings (denoted by K) around the vantage point, aggregated across all cells in the POI (eg. for $K = 501$, the composition of neighbours ranked 501 to 510 is evaluated, aggregated for all cells in the POI). The composition is not cumulative across K. The neighbours are queried anew, with *BC5A* cells and debris excluded from the search (since most nearest neighbours tend to come from the same population as the POI). We can see that, in congruence with the *BC5A* neighbourhood

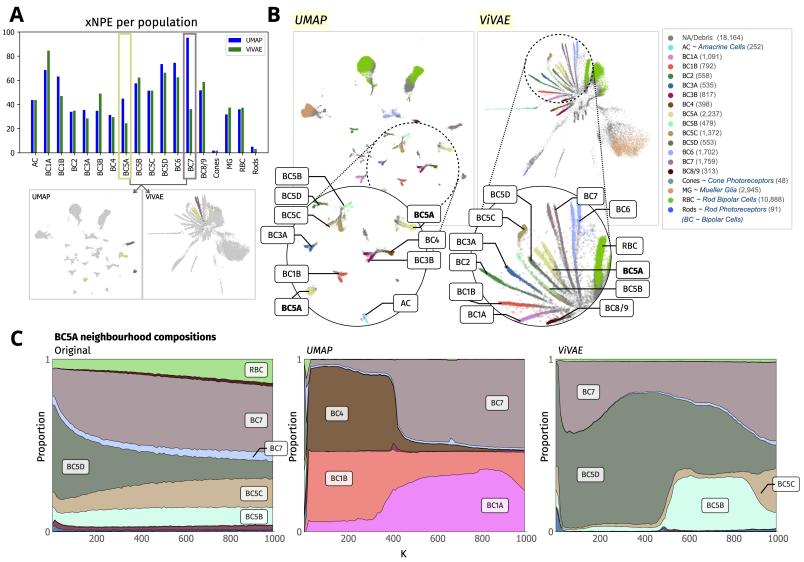


Fig. 4: **Population-level supervised evaluation of Shekhar retina data embeddings. A.** xNPE scores for a 2-dimensinoal *UMAP* and *ViVAE* embedding per labelled population are shown (in blue and green, respectively). Below, populations *BC5A* (red) and *BC7* (black) are highlighted in both embeddings, revealing increased fragmentation of both populations by *UMAP*. **B.** The evaluated *UMAP* and *ViVAE* embeddings are displayed, with a colour legend for each labelled population. Local neighbourhoods of the *BC5A* population are zoomed in, with surrounding populations tagged. **C.** Neighbourhood composition plots for original (HD) data, the *UMAP* embedding and the *ViVAE* embedding are shown, for neighbourhoods of 1000 (non-self, non-debris) points in *BC5A*.

composition in the HD input data, *BC5D* cells are featured prominently in the composition within the *ViVAE* projection, but not in *UMAP*. An intrusion of *BC4*, *BC1B* and *BC1A* is also revealed as an artifact of *UMAP*. Other artifacts, such as extrusion of *RBC*, which is underrepresented in both of the embeddings, are clearly visible in this plot.

While the xNPE makes a distinction between same-population and different-population neighbours (capturing shape distortion or change in density of neighbourhoods), neighbourhood composition plots discriminate between the surrounding populations (capturing error in how populations are positioned with respect to each other) and are qualitative as well as quantitative. As we have shown, they can be used together to describe embedding errors in a supervised setting.

3 Discussion

With increasing sizes of single-cell datasets, including organ-level cell atlases (15), the single-cell bioinformatics community should ensure good interpretability of data throughout the analytical pipeline. Dimensionality reduction (DR) is a well-established component in these pipelines that aids in exploration and hypothesis testing. As such, it is imperative that a quantitative framework be established for measuring overall distortion in lower-dimensional embeddings, as well as to identify local sources of error as they relate to cell populations of interest. This way, the potential for drawing erroneous conclusions from embeddings of high-dimensional data will be reduced.

As we set out to create a new DR method that offers a better balance between preserving local and global structures, we decided to bundle it with a framework of scoring functions that meet these requirements for evaluation. Thanks to this framework (*ViScore*), we demonstrate the performance of our method (*ViVAE*) on 7 publicly available datasets, leveraging intrinsic properties of the input data, as well as external labelling of cell populations. We encourage heavy use of these evaluation criteria to compare which tool to use for creating useful embeddings of any given transcriptomic or cytometry experimental data.

In an effort to obtain a parametric model that allows for projection of newly acquired data samples in larger experimental cohorts after training on a subset, we based *ViVAE* on a variational autoencoder (VAE), a generative model which learns an inherently continuous transformation for the high-dimensional input. The generative nature of *ViVAE* was chosen with future uses in mind, namely for structure learning and subsequent batch effect correction, community detection or imputation based on less sparse embeddings of the original data.

Based on results reported above, we recommend our DR method for all-purpose visualisation, and choose it in particular to visualise developmental trajectories and outlier populations. (The performance of *ViVAE* in comparison to other candidate tools can and should be quantified using *ViScore*.)

With regard to future development by us and others in this field, we estimate that the use of DR techniques for single-cell data outside of visualisation will grow in importance. Coupling an autoencoder model with clustering functionality (2; 32; 25) has previously shown good results, and this speaks to the value of lower-dimensional embeddings as inputs amenable to clustering or community detection. Furthermore, we believe constraining DR using topological priors (53) holds further promise for generating better embeddings with regard to developmental trajectories.

4 Methods

Our evaluation methodology includes unsupervised and supervised quality assessment metrics for lower-dimensional (LD) embeddings of high-dimensional (HD) data. The *ViVAE* pipeline itself consists of a denoising component and a dimension-reduction component, which are independent and we describe them separately.

We describe 3 alternative *ViVAE* models, each using a variational autoencoder (VAE) with an extra loss function in addition to the ones used commonly for VAEs. We mainly focus on the *ViVAE*-quartet model, applying it as the default ('*ViVAE*') model, unless stated otherwise (see results for other *ViVAE* models in Supplementary Information).

Our aim with *ViVAE* was to combine the strength of VAEs (parametric, generative models suitable for representation of continuous data, with GPU acceleration) with various modes of regularisation, adapted, more or less loosely, from other methods. These methods are *ivis* (for *ViVAE*-triplet), *TriMap* (for *ViVAE*-trimap) and *SQuadMDS* (for *ViVAE*-quartet), . We explain the specifics of all these models in 4.6, 4.5.1 and 4.5.2, respectively.

While we determine *ViVAE*-quartet as the most powerful model generally (and propose it as the default new, general-purpose DR method), we report on the performance of other variants of VAE regularisation.

4.1 Unsupervised structure-preservation measures

We base our unsupervised quality-assessment measures on widely adopted rank-based DR evaluation criteria (28). 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 $r_{ij} = |\{k : d_i k < d_i j \vee (d_i k = d_i j \wedge k < j)\}|$. The non-reflexive (non-self) neighbourhoods of ϵ_i and $\mathbf{x}_i$ are then written as $\nu_i^K = \{j : 1 \le \rho_{ij} \le K\}$ and $\upsilon_i^K = \{j : 1 \le r_{ij} \le K\}$ respectively, for a neighbourhood size of K. We then define the co-ranking matrix for a dataset of size N as

$$\mathbf{Q} = [q_{kl}]_{1 \le k,l \le N-1} \tag{1}$$

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. (We make reference to these terms in describing the triplet loss.)

Based on (30), 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

$$Q_{\mathrm{NX}}(K) = \frac{1}{KN} \sum_{i=1}^{N} |\nu_i^K \cap \upsilon_i^K| \tag{2}$$

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

$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K} \tag{3}$$

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. We denote the area under the R_{NX} curve with a linear scaling of K as the global structure preservation score S_{G}.

We also report the area under the R_{NX} curve with a *log* scale for K, up-weighting local point neighbourhoods as the local structure-preservation score S_{L}.

Keeping the different scales for values of K, the S_{G} and S_{L} scores can be computed as

$$\mathrm{AUC} = \left(\sum_{K=1}^{N-2} K^{-1} \right)^{-1} \cdot \sum_{K=1}^{N-2} \frac{R_{\mathrm{NX}}(K)}{K} \tag{4}$$

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 indeed requires sorting, for each data point, its HD and LD distances with respect to all $N-1$ remaining data points (29). These criteria, in their original version, are hence exhaustive and slow to assess the low-dimensional embeddings of very large datasets.

4.1.1 Q_{NX} and R_{NX} curve approximation

The $\mathcal{O}\left(N^2 \log N\right)$ time complexity required to evaluate the DR quality criteria reviewed in 4.1 for all neighbourhood sizes 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 (65; 39; 13). 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} consists in an average, over the entire dataset, of 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. 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 may however only require relying on a fraction of the data samples, instead of the whole dataset. In this view, a set $\tilde{\mathcal{I}} \subset \mathcal{I}$ indexing data representatives could be defined to summarise the

data cloud. A candidate approximation of Q_{NX} hence develops as

$$\widetilde{Q}_{\mathrm{NX}}(K) = \frac{1}{n}\sum_{i\in\widetilde{\mathscr{I}}} \frac{\left|\nu_i^K \cap v_i^K\right|}{K} \in [0,1], \tag{5}$$

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

$$\widetilde{R}_{\mathrm{NX}}(K) = \frac{(N-1)\widetilde{Q}_{\mathrm{NX}}(K) - K}{N-1-K}. \tag{6}$$

To define the set $\widetilde{\mathscr{I}}$, a first option consists in randomly sampling n representatives among the N data samples. Nevertheless, to ensure the consistency of the approximation of $Q_{\mathrm{NX}}(K)$ by $\widetilde{Q}_{\mathrm{NX}}(K)$, the samples indexed in $\widetilde{\mathscr{I}}$ 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 $\widetilde{\mathscr{I}}$ 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 (65; 52), being widely used to speed up nearest neighbour searches in big datasets (64; 34).

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 (10). 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 $\mathscr{O}(N\log N)$ time complexity (65). However, as only n representatives need to be determined, we may only grow the tree up to depth $\lfloor \log_2 n \rfloor$, in $\mathscr{O}(N\log n)$ 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.

Notice that neighbour searches in VP trees do not scale well with the HD dimension due to the distance concentration (51), 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 (52), 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 $10 \log N$ incomparably 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 $\widetilde{Q}_{\mathrm{NX}}$ 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}.

4.2 Supervised evaluation metrics

We describe a population-wise supervised evaluation metric, as well as a qualitative tool for comparing population neighbourhoods in input data and its embeddings.

4.2.1 *Extended Neighbourhood-Proportion-Error*

The *Neighbourhood Proportion Error* (NPE) (24) (applied in a recent DR benchmark (56)) quantifies preservation of the structure of manually defined sample populations. The per-population distribution of same- versus other-population samples across ranks in close neighbourhoods to each vantage-point sample is compared in HD and LD using total variation distance (TVD). The sum of these distances between each pair of the discrete distributions per population is then returned as a measure of error.

We re-implement the NPE as the *Extended Neighbourhood-Proportion-Error* (xNPE) to work faster and use the Earth mover's distance (EMD) instead of TVD, since it uses the entire distribution instead of the, more likely spurious, single largest difference between distributions.

Moreover, in addition to a summed or averaged error value, the error for each population can be returned. This indicates which populations are reconstructed more or less faithfully in the LD in any given embedding.

The strength of the xNPE, as showcased above in our Results, is the ability to incorporate prior information about the input dataset in the form of labels, and to obtain a score for each population of points in an embedding, rather than the embedding as a whole. We posit that the xNPE, as a local measure of embedding accuracy, effectively captures distortion in shape and relative position of populations.

4.2.2 Neighbourhood composition plots

For identifying sources of embedding error on a cell population level, we recommend inspecting *neighbourhood composition plots* (demonstrated in 2.4). These visualise the composition of different-size neighbourhoods of a chosen labelled population of interest (POI) in HD data or its embeddings, allowing for comparing the relative position of this population to others.

A new k-NNG is built for all input points except the POI (and any unlabelled or debris points if indicated). This allows to query nearest neighbours at different cut-offs for points in the POI, while excluding points in the POI itself (which often make up the majority of neighbours).

Composition of different-size (K-ary) neighbourhoods (1 to 100, with step size 10 by default) in terms of population labels is then 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 (the size of which is the step size) and for all points in the POI.

4.3 Data denoising using nearest-neighbour graphs

Our denoising algorithm, which we call smoothing, is based on an approach described in (8) It starts with identifying a fixed number (k) of nearest neighbours to each point in a k-nearest-neighbour graph (k-NNG). Each point is then moved closer toward the average coordinates of its k nearest neighbours (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.

We use the term *pre-smoothing* to mean denoising of a HD point cloud ahead of using it as input to a DR or clustering algorithm. The term *post-smoothing* then refers to denoising a LD embedding using nearest-neighbour relations from the k-NNG built on top of the HD data.

4.4 Variational autoencoder architecture for dimension reduction

An autoencoder (AE) is a type of neural network trained to reconstruct an input $\mathbf{X_{n \times d}} \in \mathscr{X}$ as an approximation $\hat{\mathbf{X}}_{\mathbf{n \times d}}$ while compressing $\mathbf{X}$ into a lower-dimensional representation $\mathbf{Z_{n \times l}} \in \mathscr{Z}$ (where $\mathscr{X} = \mathbb{R}^d, \mathscr{Z} = \mathbb{R}^l, l < d$). It consists of two stacked feed-forward networks: the encoder $E_\Phi : \mathscr{X} \rightarrow \mathscr{Z}$, which transforms $\mathbf{X}$ to $\mathbf{L}$, and the decoder $D_\theta : \mathscr{Z} \rightarrow \mathscr{X}$, which transforms $\mathbf{L}$ to $\hat{\mathbf{X}}$. 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 in Supplementary Information) and reconstruction loss is computed as the mean square error $l_{\text{recon}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{h} \sum_{i=1}^{h} (x_i - \hat{x}_i)^2$.

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. Furthermore, dropout layers can be used for regularisation to avoid overfitting.

The variational autoencoder (VAE) is a generative model based on a basic AE. Instead of encoding and decoding a latent representation $\mathbf{L}$ directly, a distribution $\mathscr{D}$ in latent space is learned, and a sampler generates a non-deterministic representation $\mathbf{L}$ and decodes it. The dissimilarity (Kullback-Leibler divergence) between $\mathscr{D}$ and a latent prior distribution (isotropic Gaussian or mixture of Gaussians (19)) is used as an additional loss term during training.

Since the model is trained to learn a salient representation $\mathbf{L}$ of the original data $\mathbf{X}$, we use $\mathbf{L}$ for visualisation of important high-dimensional structures in $\mathbf{X}$ that are initially obscure due to inability to visually represent HD data directly.

By default, the *ViVAE* model uses the SELU activation function and a Gaussian-mixture latent prior. (Other model hyperpamaters are specified in of Supplementary Information.)

ViVAE uses a VAE architecture with multiple optional loss function terms that can be added to the reconstruction error and the KL-divergence from prior.

4.5 Siamese neural network training

Instead of presenting a singular mini-batch of input samples at a time, *ViVAE* models can be trained on three related mini-batches in parallel, while enforcing a constraint on each triplet of corresponding samples. We refer to this training procedure as Siamese network training (as in (50)), and we call the samples within each triplet an *anchor* $\mathbf{x}_a$, a *positive reference* $\mathbf{x}_p$ and *negative reference* $\mathbf{x}_n$. An additional loss function term can then impose a constraint on the relative distances between these samples.

We provide three methods of sampling points to form the triplets, all using a k-nearest-neighbour graph (k-NNG) of the input data.

The first setting is adopted from the *ivis* algorithm (50). For an input dataset of n samples, we generate n triplets, iteratively taking each sample as $\mathbf{x}_a$ once and randomly picking $\mathbf{x}_p$ from its k-ary neighbourhood and a $\mathbf{x}_n$ from outside that neighbourhood, with uniform probability in both cases. This ensures the negative reference is not closer to the anchor than the positive reference.

In contrast, the second setting works with indices k' and l' where $k' < l' \leq k$ and $\mathbf{x}_p$ is sampled from neighbours with rank from $[1, k']$ and $\mathbf{x}_n$ is sampled from neighbours with rank from $(k', l']$. The motivation is capturing rank changes within small local neighbourhoods.

The third setting uses the sampling procedure from the *TriMap* method (1). In contrast to *ivis*, multiple triplets per anchor are sampled: we take m nearest neighbours of x_a and sample m' triplets per each. Additionally, r 'random' triplets are created where x_p and x_n are sampled uniformly, irrespective of the k-NNG (swapping x_p and x_n if needed to ensure $\rho_{ap} \leq \rho_{an}$).

Each triplet is also weighted. First a quantity σ_i is computed for each point $\mathbf{x}_i$, as the average HD Euclidean distance between the 4-th, 5-th and 6-th nearest neighbour of $\mathbf{x}_i$. Each triplet weight is then initialised as $\widetilde{\omega}_{apn} = \gamma_{an} - \gamma_{ap}$, using a density-aware re-scaling of squared HD Euclidean distance $\gamma_{ij} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma_i \sigma_j}$ (66). Finally, the weights are re-scaled again using tempered logarithm (40) to prevent extremely large weights:

$$\omega_{apn} = \frac{1}{1-t}\left(1 + \widetilde{\omega}_{apn} - \omega_{\min}\right)^{1-t} \tag{7}$$

with default $t = 0.5$ and $\omega_{\min}$ equal to the smallest weight value prior to re-scaling.

4.5.1 *ivis* triplet loss for penalising instrusions and extrusions

The *ivis*-based triplet loss (50) imposes a soft constraint on the latent representation, penalising intrusions and extrusions (change in neighbour ranks between points, as described in (28)) using a triangle inequality. An extra term is added to the loss function, suppressing hard-k intrusions and hard-k extrusions by penalising a smaller distance from x_{a} to x_{n} than to x_{p} in the embedding space:

$$l_{\text{triplet}} = \left[\sum_{apn} d_{ap} - min(d_{an}, d_{pn}) + m\right]_+ \tag{8}$$

where m (margin) is a small value and d is Euclidean distance $d_{ab} = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2}$ between d-dimensional points. The larger the value of m, the larger the enforced distance between the points.

4.5.2 Similarity-based *TriMap* triplet loss

Authors of *TriMap* (1) define a triplet-associated loss with the use of a similarity function

$$s_{ij} = \left(1 + \|\mathbf{x_i} - \mathbf{x_j}\|_2\right)^{-1}, \tag{9}$$

which is motivated by Student's t-distribution (used as a distribution of similarities between points in LD embeddings in (35), in that differences between short distances are up-weighted relative to those between faraway points). While the authors define their own triplet sampling procedure (see above), other sampling approaches can be used in combination with the *TriMap* loss function. It is computed as

$$l_{\text{TriMap}} = \sum_{apn} \omega_{apn} \frac{s_{an}}{s_{ap} s_{an}} \tag{10}$$

where ω_{apn} is the corresponding triplet weight, originating from the *TriMap* sampling procedure (otherwise defaulting to 1 for *ivis*-inspired triplet sampling).

4.6 Scale-agnostic quartet loss function

For the triplet loss, we pick reference samples based on membership in a close neighbourhood to each vantage point. However, there is utility to imposing constraints on relative distances of points picked irrespective of this close-versus-distant classification. This idea is explored by SQuadMDS (26; 27), and a variant thereof is implemented in *ViVAE* as the *quartet loss* (used by the *ViVAE*-quartet model).

Here, samples in each training mini-batch are divided randomly into quartets (groups of 4 points) and a cost value is computed based on the difference between normalised distances within the quartet.

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 minimise the error of its quartet as a whole. Using Euclidean distance functions for points in HD and LD δ and d, respectively, we compute relative distances between quartets of points as $\delta^r_{ij} = \frac{\delta_{ij}}{\sum_{a=1}^{3}\sum_{b=a+1}^{4}\delta_{ab}}$ and $d^r_{ij} = \frac{d_{ij}}{\sum_{a=1}^{3}\sum_{b=a+1}^{4} d_{ab}}$. The loss function can then easily be defined as

$$l_{\text{quartet}} = \sum_{a=1}^{3} \sum_{b=a+1}^{4} (\delta^r_{ab} - d^r_{ab})^2 \tag{11}$$

In *ViVAE*, we additionally regularise the model by re-shuffling points into random quartets within the mini-batch multiple times and averaging the loss. This ensures that the model iterates over many different quartet partitionings during training. The generic training procedure for a VAE that uses the quartet loss (*Stochastic Quartet VAE*, or *SQuadVAE*, in keeping with the naming convention from *SQuadMDS*) is shown in Figure 5.

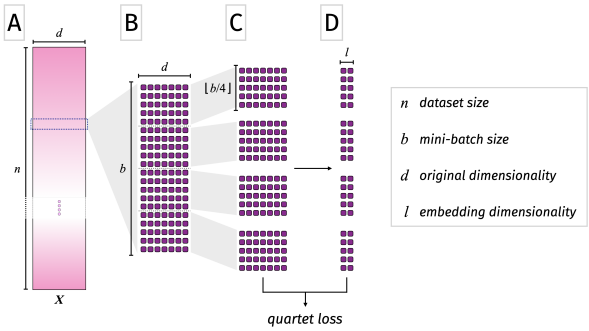


Fig. 5: **Training an encoder in *Stochastic Quartet VAE* (*SQuadVAE*). A.** A shuffled d-dimensional dataset X consisting of n points is presented as input to *SQuadVAE*. **B.** At each training iteration, a mini-batches of b samples is drawn from X. **C.** The mini-batch is divided randomly into equally large ($\lfloor b/4 \rfloor$) quarters, representing the 1st, 2nd, 3rd and 4th points in corresponding quartets. **D.** The mini-batch is encoded into an l-dimensional embedding and relative intra-quartet distances before and after embedding are used to calculate the quartet loss. The derivative of the quartet loss is used to update parameters of the encoder.

4.7 Code availability

ViVAE implements denoising and dimension reduction (`github.com/saeyslab/ViVAE`). This is a Python package, with `sklearn`-like API. The models are implemented using *TensorFlow* and enable GPU acceleration (eg. via *CUDA*, *Metal*).

ViScore implements unsupervised and supervised scoring functions for evaluation and benchmarking (`github.com/saeyslab/ViScore`); this is also a Python package. (See functions `score`, `xnpe`, `plot_xnpe_map`, `make_knn` and `neighbourhood_composition_plot`.) The `ViScore` repository also contains scripts for comparing performance of different methods, used in generating results shown in this paper.

5 Acknowledgements

This research was funded by the FWO (Fonds Wetenschappelijk Onderzoek) and conducted as part of the FWO Strategic Basic research project 1S40421N.

6 Supplementary Information

In this document, we elaborate more deeply on the methodology of *ViVAE*, approaches we take to evaluate lower-dimensional (LD) embeddings of high-dimensional (HD) data, more detailed results of the experiments outlined in the main text and all 2-dimensional embedding layouts for comparison.

6.1 Discussion of previous and newly proposed evaluation metrics

Various approaches to evaluating DR methods have been suggested in the past. Notably, change in relative distances between points (measured via neighbourhood rank) has been proposed as an indicator of embedding error, with measures such as continuity, trustworthiness and the local continuity meta-criterion (LCMC) (28). The notion of LCMC, which measures agreement of neighbourhood ranks in neighbourhoods of fixed size, was extended to a measure of structure preservation across different scales (local to global), using Q_{NX} and R_{NX} curves (30). The scalability of these approaches is hindered by the need to compute all pairwise distances for both the original and the reduced data (quadratic computational complexity).

Additionally, a 'global score' was proposed that bases a lower bound of error on principal component analysis (PCA) and considers the difference between a given embedding and an optimal linear mapping of the HD data (1). Computing correlation for distance matrices between centroids of manually defined populations of points in HD data and its LD embedding (or repeatedly generated subsets thereof) has been applied previously to assess global structure in a supervised setting (50). Alternatively, concordance of clustering in the original and reduced space has been proposed for evaluation of embedding quality of large mass cytometry datasets (56). A measure of distortion of pre-defined cell populations (the aforementioned Neighbourhood Proportion Error) has also been suggested as a measure focused on local distance relationships between cell embeddings (24).

While useful in various contexts, we suggest caution when applying techniques that assume a prior devised from linear DR, from an automated clustering step or from supposed knowledge of cell types at a consistent resolution across different compartments of the data. Furthermore, if supervised evaluation is required, we suggest computing population-level, rather than embedding-level, scores, so that the assumption of stable resolution of cell population labelling can be relaxed.

To provide an unsupervised solution, we build on the established methodology involving Q_{NX} and R_{NX} curves (30) and circumvent their scalability issues curves by devising an approximate solution (see 4.1.1) to calculate a local and global SP score (S_{L} and S_{G}, respectively).

For supervised scoring, we provide the xNPE, which mostly avoids the problem of equal resolution for cell population labelling, and neighbourhood composition plots, which provide qualitative as well as quantitative information to maximise their diagnostic potential.

6.2 Local and global structure preservation results for *ViVAE* and previously published methods

The unsupervised structure preservation results for 7 single-cell datasets consist of the local score (S_{L}), the global score (S_{G}) and the harmonic mean $F = \frac{2S_L S_G}{S_L + S_G}$ between the two criteria.

		Ziegler	Shekhar	Ximerakis	Farrell	Park	Samusik	Levine
S_{L}	PCA	0.22	0.17	0.22	0.21	0.23	0.18	0.28
	t-SNE	**0.46**±.00	**0.32**±.00	**0.43**±.00	**0.42**±.00	**0.41**±.00	**0.35**±.00	**0.40**±.00
	UMAP	0.34±.00	0.25±.00	0.43±.01	0.33±.00	0.31±.00	0.27±.00	0.26±.00
	PaCMAP	0.33±.00	0.26±.00	0.36±.00	0.31±.00	0.32±.00	0.27±.00	0.25±.00
	TriMap	0.34±.00	0.28±.00	0.36±.00	0.33±.00	0.34±.00	0.28±.00	0.24±.00
	ivis	0.25±.01	0.21±.01	0.25±.01	0.24±.01	0.25±.01	0.21±.01	0.28±.00
	SQuadMDS	0.28±.00	0.22±.00	0.30±.00	0.26±.00	0.27±.00	0.23±.00	0.21±.00
	ViVAE	0.33±.01	0.27±.00	0.35±.01	0.33±.01	0.33±.01	0.28±.01	0.24±.00
S_{G}	PCA	0.49	0.49	0.43	0.59	0.67	0.58	0.59
	t-SNE	0.43±.00	0.29±.00	0.38±.00	0.39±.00	0.40±.00	0.44±.00	0.38±.00
	UMAP	0.41±.02	0.33±.01	0.34±.03	0.43±.01	0.26±.01	0.49±.01	0.42±.01
	PaCMAP	0.42±.00	0.36±.00	0.51±.00	0.48±.00	0.40±.00	0.49±.00	0.50±.01
	TriMap	0.53±.00	0.47±.00	0.48±.00	0.53±.00	0.52±.00	0.56±.00	0.53±.00
	ivis	0.51±.04	0.35±.03	0.35±.02	0.57±.01	0.42±.03	0.47±.03	0.41±.03
	SQuadMDS	**0.66**±.00	**0.62**±.00	**0.64**±.00	**0.67**±.00	**0.70**±.00	**0.66**±.00	**0.64**±.00
	ViVAE	0.58±.03	0.58±.02	0.57±.01	0.64±.01	0.55±.03	0.63±.01	0.61±.01
F	PCA	0.31	0.26	0.29	0.31	0.34	0.28	**0.38**
	t-SNE	0.41±.00	0.28±.00	0.38±.00	0.37±.00	0.38±.00	0.35±.00	0.37±.00
	UMAP	0.37±.01	0.29±.00	0.34±.02	0.36±.00	0.28±.01	0.35±.00	0.33±.01
	PaCMAP	0.37±.00	0.30±.00	0.43±.02	0.38±.00	0.35±.00	0.35±.00	0.34±.00
	TriMap	0.42±.00	0.35±.00	0.41±.00	0.40±.00	**0.41**±.00	0.38±.00	0.34±.00
	ivis	0.35±.02	0.27±.01	0.31±.01	0.35±.01	0.32±.02	0.31±.01	0.34±.02
	SQuadMDS	0.39±.00	0.33±.00	0.41±.00	0.37±.00	0.39±.00	0.35±.00	0.32±.00
	ViVAE	**0.42**±.02	**0.37**±.00	**0.44**±.01	**0.44**±.01	**0.41**±.01	**0.39**±.01	0.35±.00

Table 2: Average structure-preservation scores for 5 runs of different algorithms on scRNA-seq and CyTOF datasets, with standard deviations.(Single values are shown for PCA, which is a deterministic method.) In each case, a 2-dimensional embedding is produced.In this default setting for *ViVAE*, we use the *ViVAE*-quartet (default) model. S_{L} is local structure preservation, S_{G} is global structure preservation and F is a harmonic mean between the S_{L} and S_{G}. For all three measures, higher score is better. Highest performer in each category (per dataset) is highlighted in bold and blue.

6.3 Local and global structure preservation by alternative *ViVAE* models

Structure-preservation scores for non-default *ViVAE* models, described in the Methods section of our paper, are shown below in Table 3. While we propose *ViVAE*-quartet (the default model) as the DR algorithm of choice, we included methodological description and results for these alternative models (*ViVAE*-triplet and *ViVAE*-trimap) for completeness.

		Ziegler	Shekhar	Ximerakis	Farrell	Park	Samusik	Levine
S_L	*ViVAE*-triplet	0.31±.01	0.24±.02	0.34±.00	0.30±.01	0.33±.00	0.23±.01	0.21±.00
	ViVAE-trimap	0.33±.01	0.27±.01	0.35±.01	0.32±.01	0.33±.01	0.27±.00	0.25±.01
S_G	*ViVAE*-triplet	0.53±.03	0.48±.06	0.53±.04	0.59±.02	0.49±.03	0.55±.04	0.49±.03
	ViVAE-trimap	0.54±.02	0.46±.05	0.50±.06	0.55±.03	0.48±.01	0.51±.02	0.46±.04
F	*ViVAE*-triplet	0.40±.02	0.32±.03	0.42±.01	0.40±.01	0.39±.01	0.33±.02	0.30±.01
	ViVAE-trimap	0.41±.01	0.34±.02	0.41±.02	0.41±.01	0.39±.01	0.35±.01	0.32±.01

Table 3: Average structure-preservation scores for 5 runs of alternative *ViVAE* models on scRNA-seq and CyTOF datasets, with standard deviations. In each case, a 2-dimensional embedding is produced. We use pre-smoothing and do not use post-smoothing.

6.4 Analysis of pre-smoothing and post-smoothing effects

ViVAE uses pre-smoothing and post-smoothing (see Methods): a denoising approach that uses near-neighbour relations to reduce the inherent noisiness of biological data. In pre-smoothing (default), each point in the HD data is shifted toward the mean of its nearest neighbours' coordinates. In post-smoothing (not default), each point in the LD embedding is shifted toward this mean, determined by coordinates in the embedding but the nearest-neighbour relations from the original HD data.

In Figure 6, we present the effects of pre-smoothing (denoising HD input data ahead of reducing dimensionality) and post-smoothing (denoising LD embeddings based on structure of the denoised or non-denoised HD data). To this end, we show the visual differences between corresponding embeddings of the Ximerakis (63) mouse brain transcriptomic dataset and structure-preservation scores of the respective embeddings.

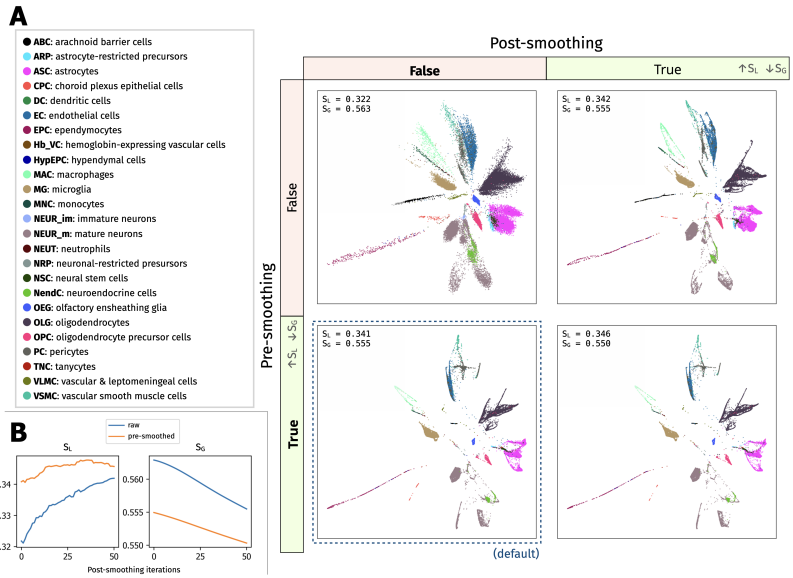


Fig. 6: **Effect of graph-based denoising on *ViVAE* embeddings. A.** A grid of 2-dimensional embeddings of the Ximerakis (63) mouse brain dataset using the *ViVAE*-quartet model is shown, with pre-smoothing (rows) and post-smoothing (columns) either disabled or enabled. For pre-smoothing, a shift coefficient $\lambda = 1$ and 1 iteration of the algorithm are applied. For post-smoothing, $\lambda = 0.05$ and 50 iterations are applied (to show how S_L and S_G change progressively in panel B.). A neighbourhood size of 50 is used in both cases. S_L and S_G scores are shown for each embedding. A corresponding colour-coded legend of cell populations, with abbreviations, is shown to the left of the grid. **B.** Values of the S_L and S_G scores with pre-smoothing disabled and enabled, and a varied number of post-smoothing iterations, are shown.

Both pre-smoothing and post-smoothing cause decrease in S_G and increase in S_L, indicating a trade-off between preserving the global structure faithfully on the one hand and preserving local neighbourhoods and relative positioning of populations on the other.

Furthermore, we demonstrate the usefulness of pre-smoothing in a more generic setting by evaluating its effect on downstream clustering of cells, separate from a DR analysis. In this case, we focus on the two CyTOF datasets, due to availability of reference labels of cells that originate from entirely manual analysis, via setting cut-offs on combined expression levels of known protein markers. We replicate the experimental setting in a popular clustering benchmark (58), applying the top-performing clustering algorithm, *FlowSOM* (18), to both datasets, reporting the F1 score of clusters bijectively (1-to-1) matched to labelled populations. In addition, we compute the adjusted Rand index (ARI) score, as an alternative supervised evaluation metric that does not require deciding a matching scheme (e.g., bijective) between clusters and annotated populations, and the Davies-Bouldin (DB) index, an unsupervised score evaluating the compactness and separation of clusters (Figure 7 A.).

We show that our denoising approach tends to improve performance of the clustering downstream, as evidenced by both supervised and unsupervised evaluation criteria. Additionally, Figure 7 (B.) shows the effects of applying smoothing to the (31) human bone marrow dataset. As an example, CD16$^-$ NK cells in particular end up less over-clustered when

pre-smoothing is used, as quantified by overlaps between the automatically generated partitioning by *FlowSOM* and the manually assigned cell labels shown in an ordered heatmap. Most bijectively matched cluster-label pairs (diagonal of the top-left submatrix) end up with larger overlaps, as desirable.

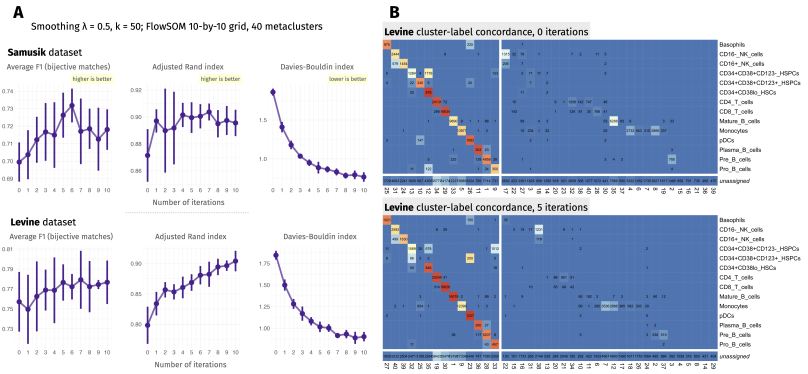


Fig. 7: **Evaluation of *FlowSOM* clustering on *CyTOF* data with smoothing (graph-based denoising) as a pre-processing step. A.** Average values of the F1 score (with F1-maximising bijective matching of clusters to labelled populations), adjusted Rand index and Davies-Bouldin index for 0 to 10 iterations of the denoising algorithm applied to protein expression data from (46) and (31) are shown. Ten runs of clustering with different random seeds were run on each input, with standard deviation indicated by error bars. **B.** Heatmaps of overlaps between clusters (columns) and populations (rows) for the Levine dataset, with 0 and 5 iterations of smoothing applied (top and bottom, respectively). Bijective cluster-to-population matches are on the diagonal of the top-left submatrix, while clusters matched to any best-fitting population are marked in green and populations matched to best-fitting clusters are marked in purple. Absolute counts of cells in each overlap are displayed (lack of number means 0).

6.5 xNPE algorithm description

The Extended Neighbourhood-Proportion-Error (xNPE) algorithm, based on the NPE (24)is written out using pseudocode below.

Algorithm 1 Extended Neighbourhood-Proportion-Error

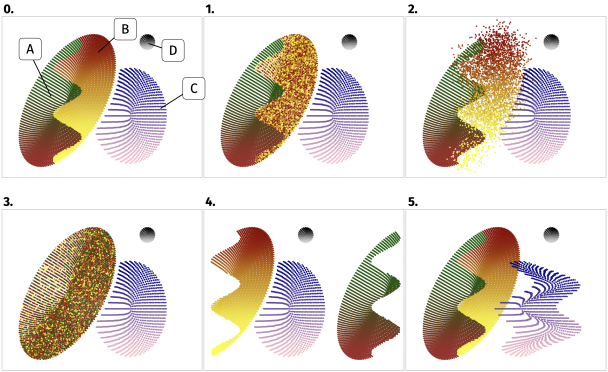
```
function SameNeighbourDistribution(K, A)
                    ▷ K: row-wise k-nearest-neighbour indices matrix, A: vector of labels
    n ← rowCount(K)
    P ← unique(A)                                                    ▷ populations
    p ← length(P)
    M ← [ ]_n                                                        ▷ same-neighbour counts
    D ← [ ]_p                                                        ▷ distributions of same-neighbour counts
    for i ∈ 1..n do
        r ← K_{i*}
        M_i ← |A_r = A_i|                                            ▷ count neighbours from same population
    end for
    for j ∈ 1..p do
        a ← |A = P_j|
        D_j ← M_a                                                    ▷ population-wise distribution of same-neighbour counts
    end for
    return D
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 ← SameNeighbourDistribution(X_H, A, k)
    L ← SameNeighbourDistribution(X_L, A, k)
    δ ← [ ]_p
    for i ∈ 1..p do
        δ_i ← EarthMover'sDistance(H_i, L_i)
    end for
    return δ
end function
```

6.6 xNPE algorithm demonstration on synthetic data

We present a simple experiment on a 2-dimensional synthetic dataset to showcase how the xNPE, a locally-focused supervised evaluation metric, contrasts with the unsupervised local structure-preservation score (S_{L}) in Figure 8. Since the scoring methodology generalises to any data transformation (and is not limited to dimension reduction), 2-dimensional mappings of the original 2-dimensional synthetic dataset are shown, demonstrating different kinds of distortions. The neighbourhood size for xNPE is set to 50. While both S_{L} and xNPE focus on local structure, they quantify a different type of error. While S_{L} will penalise any relative change in position of points that results in neighbourhood rank changes, the xNPE only penalises those changes that occur at the boundaries of close adjacent populations. It follows, for instance, that shuffling points within one population reduces the S_{L} score, but it does not cause the xNPE value of that population to increase. This is because the aggregated neighbourhood profile of the population remains unchanged (panel 1). However, adding noise to points that are on the frontiers of a population or shuffling points between populations (panels 2 and 3, respectively), increases xNPE, as well as decreasing S_{L}. A relative shift in position of non-isolated populations resulting in isolation (panel 4) or distortion of shape of non-isolated populations (panel 5) are also captured by the xNPE.

Essentially, while S_{L} is sensitive to shuffling or noise among points which are in the center of a labelled population (do not define the shape or relative positioning of that population within the embedding), the xNPE does not penalise this.



Score

Transformation	xNPE				S_L
	A	B	C	D	
1	0	0	0	0	0.55
2	2.8	5.0	2.1	0	0.66
3	22.2	22.5	0	0	0.21
4	2.1	2.2	0	0	0.85
5	0	0.9	1.6	0	0.94

Fig. 8: **Demonstration of xNPE and S_L scores on synthetic data.** A synthetic dataset (panel **0.**) is provided, with points partitioned into manually assigned populations A (4041 points), B (3939 points), C (2110 points) and D (276) and colour-coded using gradients to show the original distribution of single points. Neigbourhood size of 50 is used for the xNPE. Panel **1.** shows a transformation error which causes shuffling of points within population A. Panel **2.** has population B jittered using Gaussian noise. Panel **3.** shuffles points between populations A and B, causing these populations to be mixed together. Panel **4.** shows a shift in position (isolation) of population A. Panel **5.** distorts the shape of population C. The xNPE values for each transformed population, as well as the S_L score per transformation, are shown in a table on the right.

6.7 Pre-smoothing and post-smoothing algorithm description

We describe the pre-smoothing/post-smoothing algorithm for graph-based data denoising using pseudocode below.

Algorithm 2 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
  $n \leftarrow rowCount(X)$
  for $i \in 1..n$ **do**
    $r \leftarrow K_{i*}$
    $\mu \leftarrow X_r$
    $t \leftarrow$ columnMeans(μ) ▷ target coordinates
    $\Delta \leftarrow t - X_{i*}$
    $X_{i*} \leftarrow \lambda\Delta$
  end for
  return X

6.8 Comparison of running times

We compare running times of all tested methods and datasets. All of the evaluated methods are iterative, and have a default setting for iteration count (constant in the case of *UMAP*, *TriMap*, *PaCMAP* and *SQuadMDS* and adaptive in the case of PCA, t-SNE, *ivis* and *ViVAE*).

Intuitively, the number of iterations influences training duration for each model. If time requirements are prohibitive, one can find that embeddings obtained with a reduced number of iterations or resulting from transforming an entire input dataset after training only on a subset (possible with *UMAP*, *PaCMAP*, *ivis* and *ViVAE*) are of sufficient quality.

Furthermore, since *ivis* and *ViVAE* use the *TensorFlow* framework, they train notably faster if GPU acceleration is available. Early stopping criteria, which set the number of training epochs adaptively, also play a major role. For *ViVAE* models with triplet sampling, there is an addition to training time proportional to the number of additional samples in the training set.

Running times for each method are shown in Tables 4, 5 and 6. For *ViVAE*, we further divide running time by separate steps. Apart from training our network, smoothing and triplet-based loss functions (*ivis*- and *TriMap*-based losses) require k-nearest-neighbour graph construction.

The steps for which we measured running times are

1. K-nearest neighbour graph construction for original input data (for smoothing),
2. pre-smoothing,
3. K-nearest neighbour graph construction for pre-smoothed data (for post-smoothing if pre-smoothing is applied),
4. dimensionality reduction (model training),
5. post-smoothing.

By default (and in our summarised results), we do not apply post-smoothing (thus omitting step 5 for all *ViVAE* models and steps 3 and 5 for *ViVAE*-quartet). Our recommended general-purpose embdedding method is to apply pre-smoothing and train a *ViVAE*-quartet model on HD data, with the default early stopping conditions, or for an increased number of epochs without early stopping for improved quality and convergence.

PCA, t-SNE, *UMAP*, *PaCMAP* and *TriMap* model training, construction of k-nearest-neighbour graphs and smoothing were run on a single node of a CPU cluster (*Intel Xeon Gold 6140* processor) with 32 GB of usable RAM. *ivis* and *ViVAE* models were trained on a single node of a GPU cluster (*16-core Intel Xeon Gold 6242* processor with *NVIDIA Volta V100* GPU) with 32 GB of usable RAM.

	Ziegler	Shekhar	Ximerakis	Farrell	Park	Samusik	Levine
PCA	0.2±0.2	0.1±0.0	0.1±0.1	0.1±0.1	0.3±0.0	0.3±0.0	0.8±0.1
t-SNE	323.2±6.9	666.2±14.5	348.4±7.5	385.4±7.4	1490.9±19.7	1658.3±41.3	5187±136.9
UMAP	41.2±8.8	62.9±7.5	58.7±8.6	48.1±7.9	193.4±37.9	123.2±8.5	460.2±46.3
PaCMAP	26.4±0.4	44.1±1	33±0.5	35.1±0.4	168.8±2.5	121.1±1.1	581.5±54.8
TriMap	28.2±0.5	42.4±1	32.3±0.8	35.2±0.6	132.7±3.1	97.6±3.2	409.2±35.2
ivis	375.8±51.9	449.9±5.9	393.5±33.3	421.4±41.5	1216.1±2.2	865.8±67.9	3312.8±165.7
SQuadMDS	45.8±3.3	100.0±9.1	98.3±4.2	95.6±1.7	313.0±17.9	204.0±10.9	563.0±13.2

Table 4: Running times for training dimensionality-reduction models on single-cell datasets in seconds. Mean values with standard deviation across 5 runs are shown.

	Ziegler	Shekhar	Ximerakis	Farrell	Park	Samusik	Levine
k-NNG	15.4±0.1	24.0±0.3	16.9±0.1	17.7±0.2	57.1±0.5	41.5±0.3	150.8±5.1
pre-smoothing	0.8±0.0	1.3±0.0	0.9±0.0	1.0±0.0	3.4±0.0	3.5±0.3	12.1±0.0
k-NNG (pre-smoothed)	14.1±0.1	21.8±0.2	15.8±0.1	16.3±0.2	53.0±0.4	36.0±0.4	128.0±0.4
post-smoothing	34.7±0.2	46.3±0.4	38.8±0.2	40.7±0.2	135.4±0.2	92.1±1.1	278.5±0.7

Table 5: Running times for constructing approximate *k*-nearest-neighbour graphs and applying denoising by smoothing, for single-cell datasets. Mean values with standard deviation across 5 runs are shown.

	Ziegler	Shekhar	Ximerakis	Farrell	Park	Samusik	Levine
ViVAE-triplet training	101.4±11.2	69.1±23.3	97.6±8.9	69.1±13.7	265.1±28.5	90.2±23.7	252.8±49.8
ViVAE-trimap training	102.9±11.1	107.1±1.6	98.4±8	92.6±1.4	259.2±41.2	197.9±3.6	749.5±41
ViVAE-quartet training	110.9±4.7	98.2±3.3	101.8±3.1	85.9±2.7	254.1±4.4	153.6±28.2	637.7±64.5

Table 6: Running times for training *ViVAE* dimensionality-reduction models on single-cell datasets in seconds. Mean values with standard deviation across 5 runs are shown.

6.9 Data description

The Shekhar (48) dataset comes from mouse retinal bipolar cells, a diverse subtype of neurons. The Farrell (16) dataset aggregates transcriptomes of cells from zebrafish embryos at different stages of embryogenesis. The Ximerakis (63) data aggregates transcriptomes of cells sampled from mouse brains, coming from two cohorts of mice (young and old). The Ziegler (67) dataset comprises transcriptomic data from nasopharyngeal swabs of 35 Covid-19-infected and 23 healthy individuals. The data from Park (43) aggregates measurements from 15 human thymi at embryonic or fetal stages. We only take a subset of the data (3' gene expression libraries) to avoid a documented technical batch effect.

The Levine (31) dataset contains profiles of human bone marrow cells from 2 healthy donors. The Samusik (46) dataset contains data from a mouse bone marrow sample.

6.10 Data availability

The (48) dataset was downloaded via the R package *scRNAseq* (44) using the function `ShekharRetinaData` and converted to an `AnnData` object using the *scDIOR* packages (17). Other scRNA-seq datasets were obtained from the *Single Cell Portal* (https://singlecell.broadinstitute.org/single_cell), using the following links.

- (16): https://singlecell.broadinstitute.org/single_cell/study/SCP162/single-cell-reconstruction-of-developmental-trajectories-during-zebrafish-embryogenesis
- (63): https://singlecell.broadinstitute.org/single_cell/study/SCP263/aging-mouse-brain
- (67): https://singlecell.broadinstitute.org/single_cell/study/SCP1289/impaired-local-intrinsic-immunity-to-sars-cov-2-infection-in-severe-covid-19

The dataset from (43) was obtained from *The Human Developmental Cell Atlas* online repository: https://developmental.cellatlas.io/thymus-development (10x 3' v2 assay).

Mass cytometry datasets were obtained through the R package *HDCytoData* (59), using functions `Levine_32dim` and `Samusik_01`.

6.11 Data pre-processing

We used the *scanpy* package version 1.9.1 (61) for scRNA-seq data pre-processing. We applied the following snippet for scaling, normalisation and principal component analysis (PCA) prior to running the DR algorithms:

```
import scanpy as sc
hd = sc.pp.normalize_per_cell(X, copy=True) # assume X is count matrix (numpy.ndarray)
hd = sc.pp.log1p(hd, copy=True)
hd = sc.pp.scale(hd, max_value=10, copy=True)
data = sc.tl.pca(hd, svd_solver='arpack', n_comps=50, copy=True)
```

The (16) and (43) datasets were exceptions, where already scaled data was used, and only the PCA step remained.

For mass cytometry data, we applied the asinh transformation with a cofactor of 5, per recommendation in the *HDCytoData* package.

6.12 Model hyperparameters

Default parameters were kept for generating LD embeddings shown in results.

Both user-level and internal functions, classes and methods of the *ViVAE* Python package are documented. The default model uses an encoder of shape `[32,64,128,32]` and decoder of shape `[32,128,64,32]`, with a latent projection dimensionality of 2. The *SELU* activation function and a Gaussian-mixture latent prior are used by default.

6.12.1 Training defaults

Default arguments for the `fit` and `fit_transform` methods set the number of training epochs to 100, with an early-stopping callback halting training if there is no reduction in the loss function value for 5 successive epochs. The *Adam* optimiser is used for training, due to computational efficiency. Each loss function term is monitored throughout training and logging functionality (using a CSV logger or *TensorBoard*) can be enabled.

To replicate results in this paper run pre-smoothing first, using the `smooth` function, then create a `ViVAE` object and run `fit_transform`.

6.12.2 Loss function specification

A DR model is instantiated using the `ViVAE.ViVAE` constructor. The loss function to use in training is then specified in the `fit` or `fit_transform` method. In either of them, the argument `loss` can be specified using a string (`'VAE-triplet'`, `'VAE-trimap'`, `'VAE-quartet'`) or a dictionary object, specifying weights of loss function terms. The VAE reconstruction error and divergence from latent prior, keys `'reconstruction'` and `'kldiv'` are used. *ivis*-style triplet loss is specified as `'triplet'`, *TriMap*-style triplet loss as `'trimap'` and quartet loss as `'quartet'`. For experimental purposes, a `'quintet'` and `'sextet'` loss, analogous to quartet, can also be used. A t-SNE loss function (`'tsne'`), as described in (12), can also be enabled; however, due to the use of CPU for fitting distributions to local neighbourhoods in HD and LD, adding this loss term results in increased running time of the algorithm.

6.12.3 Triplet sampling defaults

If the `triplet` or `trimap` weight values are non-zero, Siamese network training is enabled. To use *ivis*-style triplet sampling, set `triplet_sampling` to `'simple'`, for *TriMap*-style sampling, set it to `'trimap'` (default). By default, the value is `trimap`. The *TriMap* parameters m, m' and r are specified by arguments `trimap_n_inliers` (default: 3), `trimap_n_outliers` (default: 3) and `trimap_n_random` (default: 1), respectively. The argument `trimap_maxpoints` allows to cap the number of samples in the training set after sampling. By default, this upper bound is set to 3 times the size of the original input training set, whereby this number of triplets is drawn from all the generated triplets randomly (with uniform probability distribution).

6.12.4 Smoothing defaults

We use 50-neighbour smoothing by default. For pre-smoothing, we set the λ parameter to 1.0 and run 1 iteration of smoothing. For our benchmark, we did not apply post-smoothing, as we suggest applying this step adaptively, and using the S_{L}, S_{G} and xNPE (if labels are provided) evaluation criteria to determine the optimal degree to which the LD embedding should

be denoised. The function `smooth` is called, with arguments `k`, `coef` and `n_iter` specifying neighbourhood size, the λ coefficient and number of iterations, respectively.

6.13 Embeddings of all datasets

Embeddings of each dataset by each of the methods are displayed in Figures 9, 10, 11, 12, 13, 14 and 15.

Fig. 9: **Comparison of 2-dimensional embeddings of the Ziegler (67) scRNA-seq dataset.** Scatter plots of 2-dimensional embeddings per method are displayed, with a colour scheme for labelled populations and legend.

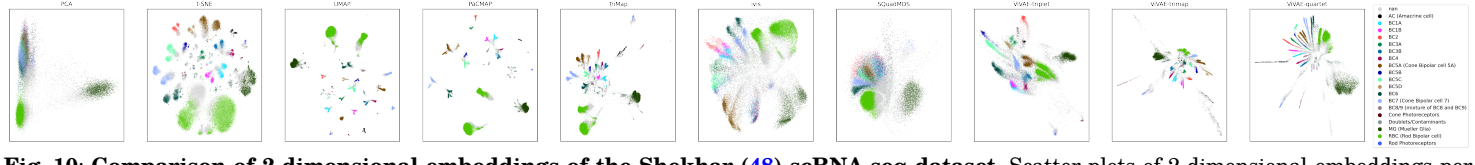


Fig. 10: **Comparison of 2-dimensional embeddings of the Shekhar (48) scRNA-seq dataset.** Scatter plots of 2-dimensional embeddings per method are displayed, with a colour scheme for labelled populations and legend.

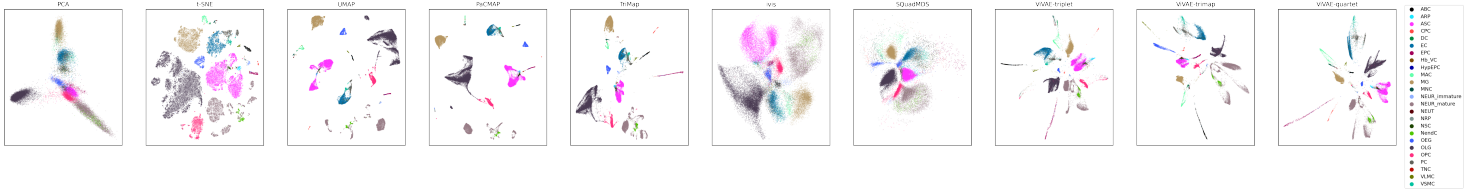


Fig. 11: **Comparison of 2-dimensional embeddings of the Ximerakis (63) scRNA-seq dataset.** Scatter plots of 2-dimensional embeddings per method are displayed, with a colour scheme for labelled populations and legend.

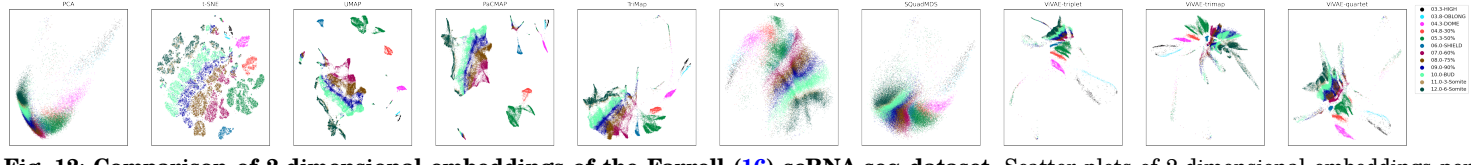


Fig. 12: **Comparison of 2-dimensional embeddings of the Farrell (16) scRNA-seq dataset.** Scatter plots of 2-dimensional embeddings per method are displayed, with a colour scheme for labelled populations and legend.

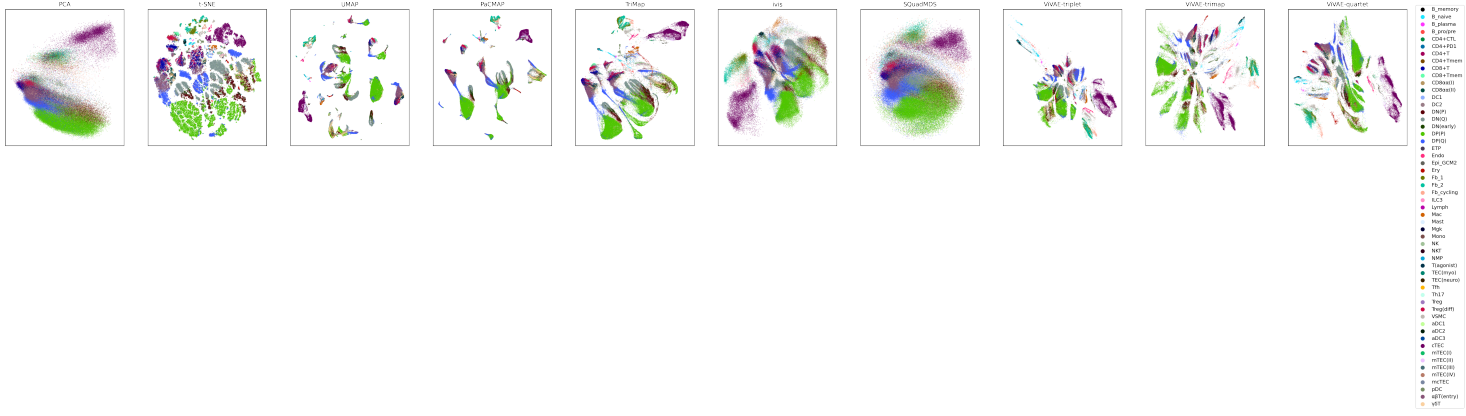


Fig. 13: **Comparison of 2-dimensional embeddings of the Park (43) scRNA-seq dataset.** Scatter plots of 2-dimensional embeddings per method are displayed, with a colour scheme for labelled populations and legend.

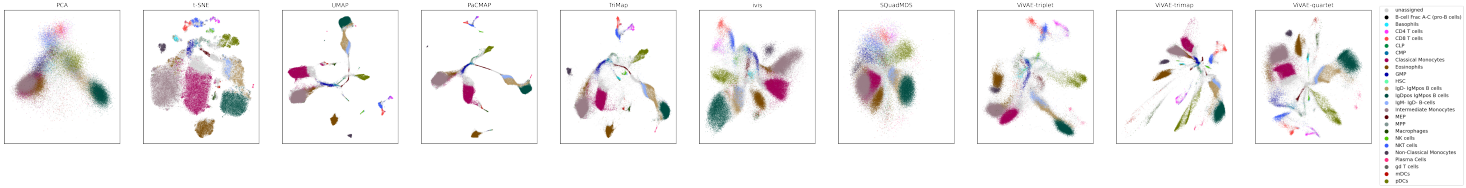


Fig. 14: **Comparison of 2-dimensional embeddings of the Samusik (46) CyTOF dataset.** Scatter plots of 2-dimensional embeddings per method are displayed, with a colour scheme for labelled populations and legend.

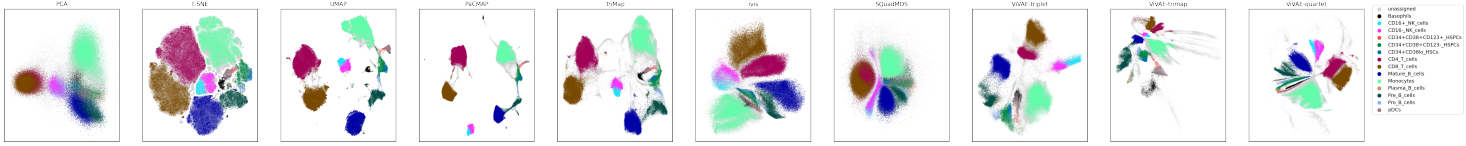


Fig. 15: **Comparison of 2-dimensional embeddings of the Levine (31) CyTOF dataset.** Scatter plots of 2-dimensional embeddings per method are displayed, with a colour scheme for labelled populations and legend.

References

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

[2] Amodio M, van Dijk D, Srinivasan K, et al (2019) Exploring single-cell data with deep multitasking neural networks. Nature Methods 16:1139–1145. https://doi.org/10.1038/s41592-019-0576-7

[3] Bandura DR, Baranov VI, Ornatsky OI, et al (2009) Mass cytometry: Technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry. Analytical Chemistry 81:6813–6822. https://doi.org/10.1021/ac901049w

[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] 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

[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] Browaeys R, Saelens W, Saeys Y (2020) NicheNet: modeling intercellular communication by linking ligands to target genes. Nature Methods 17:159–162. https://doi.org/10.1038/s41592-019-0667-5

[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] 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

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

[11] Denyer T, Ma X, Klesen S, et al (2019) Spatiotemporal Developmental Trajectories in the Arabidopsis Root Revealed Using High-Throughput Single-Cell RNA Sequencing. Developmental Cell 48:840–852.e5. https://doi.org/10.1016/j.devcel.2019.02.022

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

[13] 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

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

[15] 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

[16] 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

[17] Feng H, Lin L, Chen J (2022) scDIOR: single cell RNA-seq data IO software. BMC Bioinformatics 23. https://doi.org/10.1186/s12859-021-04528-3

[18] 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

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

[20] Haghverdi L, Büttner M, Wolf FA, et al (2016) Diffusion pseudotime robustly reconstructs lineage branching. Nature Methods 13(10):845–848. https://doi.org/10.1038/nmeth.3971, URL http://www.nature.com/articles/nmeth.3971

[21] 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

[22] 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

[23] Kobak D, Linderman GC (2021) Initialization is critical for preserving global data structure in both t-SNE and UMAP. Nature Biotechnology 39(2):156–157. https://doi.org/10.1038/s41587-020-00809-z, URL https://doi.org/10.1038/s41587-020-00809-z

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

[25] Kopf A, Fortuin V, Somnath VR, et al (2021) Mixture-of-Experts Variational Autoencoder for clustering and generating from similarity-based representations on single cell data. PLoS Computational Biology 17. https://doi.org/10.1371/journal.pcbi.1009086

[26] Lambert P, de Bodt C, Verleysen M, et al (2021) Stochastic quartet approach for fast multidimensional scaling. pp 417–422, URL http://hdl.handle.net/2078.1/252550

[27] 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

[28] 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

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

[30] 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

[31] Levine JH, Simonds EF, Bendall SC, et al (2015) Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. Cell 162:184–197. https://doi.org/10.1016/j.cell.2015.05.047

[32] Li X, Wang K, Lyu Y, et al (2020) Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. Nature Communications 11. https://doi.org/10.1038/s41467-020-15851-3

[33] 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/https://doi.org/10.15252/msb.20188746, 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

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

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

[36] Macosko E, Basu A, Satija R, et al (2015) Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. Cell 161(5):1202–1214. https://doi.org/10.1016/j.cell.2015.05.002, URL https://linkinghub.elsevier.com/retrieve/pii/S0092867415005498

[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] McKinnon KM (2018) Flow cytometry: An overview. Current Protocols in Immunology 2018:5.1.1–5.1.11. https://doi.org/10.1002/cpim.40

[39] 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

[40] Naudts J (2002) Deformed exponentials and logarithms in generalized thermostatistics

[41] Olsen LR, Leipold MD, Pedersen CB, et al (2019) The anatomy of single cell mass cytometry data. https://doi.org/10.1002/cyto.a.23621

[42] Palit S, Heuser C, De Almeida GP, et al (2019) Meeting the Challenges of High-Dimensional Single-Cell Data Analysis in Immunology. Frontiers in Immunology 10:1515. https://doi.org/10.3389/fimmu.2019.01515, URL https://www.frontiersin.org/article/10.3389/fimmu.2019.01515/full

[43] Park JE, Botting RA, Conde CD, et al (2020) A cell atlas of human thymic development defines T cell repertoire formation. Science 367. https://doi.org/10.1126/science.aay3224

[44] Risso D, Cole M (2022) scRNAseq: Collection of Public Single-Cell RNA-Seq Datasets

[45] Saelens W, Cannoodt R, Todorov H, et al (2019) A comparison of single-cell trajectory inference methods. Nature Biotechnology 37:547–554. https://doi.org/10.1038/s41587-019-0071-9

[46] 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

[47] 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

[48] 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

[49] Stubbington MJT, Rozenblatt-Rosen O, Regev A, et al (2017) Single-cell transcriptomics to explore the immune system in health and disease. Science 358:58–63. https://doi.org/10.1126/science.aan6828

[50] 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

[51] 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

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

[53] Vandaele R, Kang B, Lijffijt J, et al (2021) Topologically Regularized Data Embeddings URL http://arxiv.org/abs/2110.09193

[54] Vembadi A, Menachery A, Qasaimeh MA (2019) Cell Cytometry: Review and Perspective on Biotechnological Advances. https://doi.org/10.3389/fbioe.2019.00147

[55] Walker BL, Cang Z, Ren H, et al (2022) Deciphering tissue structure and function using spatial transcriptomics. https://doi.org/10.1038/s42003-022-03175-5

[56] 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

[57] 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.

[58] Weber LM, Robinson MD (2016) Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. Cytometry Part A 89:1084–1096. https://doi.org/10.1002/cyto.a.23030

[59] 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

[60] Williams CG, Lee HJ, Asatsuma T, et al (2022) An introduction to spatial transcriptomics for biomedical research. https://doi.org/10.1186/s13073-022-01075-1

[61] 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

[62] 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

[63] Ximerakis M, Lipnick SL, Innes BT, et al (2019) Single-cell transcriptomic profiling of the aging mouse brain. Nature Neuroscience 22:1696–1708. https://doi.org/10.1038/s41593-019-0491-3

[64] Yang Z, Peltonen J, Kaski S (2013) Scalable optimization of neighbor embedding for visualization. In: ICML (2), pp 127–135

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

[66] Zelnik-Manor L, Perona P (2004) Self-Tuning Spectral Clustering. In: Saul L, Weiss Y, Bottou L (eds) Advances in Neural Information Processing Systems, vol 17. MIT Press, URL https://proceedings.neurips.cc/paper_files/paper/2004/file/40173ea48d9567f1f393b20c855bb40b-Paper.pdf

[67] Ziegler CGK, Miao VN, Owings AH, et al (2021) Impaired local intrinsic immunity to SARS-CoV-2 infection in severe COVID-19 https://doi.org/10.1101/2021.02.20.431155, URL https://doi.org/10.1101/2021.02.20.431155