

Embedding to Reference t-SNE Space Addresses Batch Effects in Single-Cell Classification

Pavlin G. Poličar¹, Martin Stražar¹, and Blaž Zupan^{1,2}

¹ University of Ljubljana, SI-1000 Ljubljana, Slovenia

{pavlin.policar,martin.strazar,blaz.zupan}@fri.uni-lj.si

² Baylor College of Medicine, Houston, TX 77030, USA

Abstract. Dimensionality reduction techniques, such as t-SNE, can construct informative visualizations of high-dimensional data. When working with multiple data sets, a straightforward application of these methods often fails; instead of revealing underlying classes, the resulting visualizations expose data set-specific clusters. To circumvent these batch effects, we propose an embedding procedure that uses a t-SNE visualization constructed on a reference data set as a scaffold for embedding new data points. The new, secondary data is embedded one data point at the time. This prevents any interactions between instances in the secondary data and implicitly mitigates batch effects. We demonstrate the utility of this approach by analyzing six recently published single-cell gene expression data sets with up to tens of thousands of cells and thousands of genes. The batch effects are particularly strong as the used data comes from different institutions and was obtained using different experimental protocols. The visualizations constructed by our proposed approach are cleared of batch effects, and the cells from secondary data sets correctly co-cluster with cells of the same type from the primary data.

Keywords: Batch effects · Embedding · t-SNE · Visualization · Single-Cell Transcriptomics · Data Integration · Domain Adaptation.

1 Introduction

Two-dimensional embeddings and their visualizations may assist in the analysis and interpretation of high-dimensional data. Intuitively, two data instances should be co-located in the resulting visualization if their multi-dimensional profiles are similar. For this task, non-linear embedding techniques such as t-distributed stochastic neighbor embedding (t-SNE) [1] or uniform manifold approximation and projection [2] have recently complemented traditional data transformation and embedding approaches such as principal component analysis (PCA) and multi-dimensional scaling [3, 4]. While useful for visualizing data from a single coherent source, these methods may encounter problems with multiple data sources. Here, when performing dimensionality reduction on a merged data set, the resulting visualizations would typically reveal source-specific clusters instead of grouping data instances of the same class, regardless of data sources.

This source-specific confounding is often referred to as *domain shift* [5], *covariate shift* [6] or *data set shift* [7]. In bioinformatics, the domain-specific differences are more commonly referred to as *batch effects* [8–10].

Massive, multi-variate biological data sets often suffer from these source-specific biases. The focus of this work is single-cell genomics, a domain that was selected due high biomedical relevance and abundance of recently published data. Single-cell RNA sequencing (scRNA-seq) data sets are the result of isolating RNA molecules from individual cells, which serve as an estimate of the expression of cell's genes. The studies can exceed thousands of cells and tens of thousands of genes, and typically start with cell type analysis. Here, it is expected that cells of the same type would cluster together in two-dimensional data visualization [10]. For instance, Fig. 1.a shows t-SNE embedded data from mouse brain cells originating from the visual cortex [11] and the hypothalamus [12]. The figure reveals distinct clusters but also separates the data from the two brain regions. These two regions share the same cell types and — contrary to the depiction in Fig. 1.a — we would expect the data points from the two studies to overlap. Batch effects similarly prohibit the utility of t-SNE in the exploration of pancreatic cells in Fig. 1.b, which renders the data from a pancreatic cell atlas [13] and similarly-typed cells from diabetic patients [14]. Just like with data from brain cells, pancreatic cells cluster primarily by data source, again resulting in a visualization driven by batch effects.

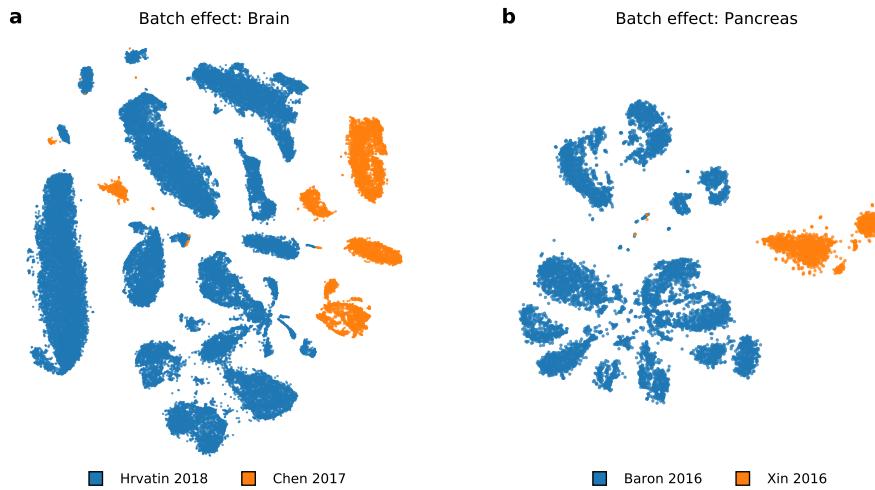


Fig. 1. Batch effects are a driving factor of variation between the data sets. We depict a t-SNE visualization of two pairs of data sets. In each pair, the data sets share cell types, so we would expect cells from the reference data (blue) to mix with the cells in a secondary data sets (orange). Instead, t-SNE clusters data according to the data source.

Current solutions to embedding the data from various data sources address the batch effect problems up-front. The data is typically preprocessed and transformed such that the batch effects are explicitly removed. Recently proposed procedures for batch effect removal include canonical correlation analysis [8] and mutual nearest-neighbors [9, 10]. In these works, batch effects are deemed removed when cells from different sources exhibit good mixing in a t-SNE visualization. The elimination of batch effects may require aggressive data preprocessing which may blur the boundaries between cell types. Another problem is also the inclusion of any new data, for which the entire analysis pipeline must be rerun, usually resulting in a different layout and clusters that have little resemblance to original visualization and thus require reinterpretation.

We propose a direct solution of rendering t-SNE visualizations to address batch effects. Our approach treats one of the data sets as a *reference* and embeds the cells from another, *secondary data set* to a joint low-dimensional space. We construct a t-SNE embedding using the reference data set, which is then used as a scaffold to embed the secondary data. The key idea underpinning our approach is that secondary data points are embedded independently of one another.

Independent embedding of each secondary datum causes the clustering landscape to depend only on the reference scaffold, thus removing data source-driven variation. In other words, when including new data, the scaffold inferred from the reference data set is kept unchanged and defines a “gravitational field”, independently driving the embedding of each new instance. For example, in Fig. 2, the cells from the visual cortex define the scaffold (Fig. 2.a) into which we embed the cells from the hypothalamus (Fig. 2.b). Unlike in their joint t-SNE visualization (Fig. 1.a), the hypothalamic cells are dispersed across the entire embedding space and their cell type correctly matches the prevailing type in reference clusters.

The proposed solution implements a mapping of new data into an existing t-SNE visualization. While the utility of such an algorithm was already hinted at in recent publication [15], here we provide its practical and theoretically-grounded implementation. Considering the abundance of recent publications on batch effect removal, we present surprising evidence that a computationally more direct and principled embedding procedure solves the batch effects problem when constructing interpretable visualizations from different data sources.

2 Methods

We describe an end-to-end pipeline that uses fixed t-SNE coordinates as a scaffold for embedding new (secondary) data, enabling joint visualization of multiple data sources while mitigating batch effects. Our proposed approach starts by using t-SNE to embed a reference data set, with the aim of constructing a two-dimensional visualization to facilitate interpretation and cluster classification. Then, the placement of each new sample is optimized independently via the t-SNE loss function. Independent treatment of each data instance from a secondary data set disregards any interactions present in that data set, and prevents the formation of clusters that would be specific to the secondary data.

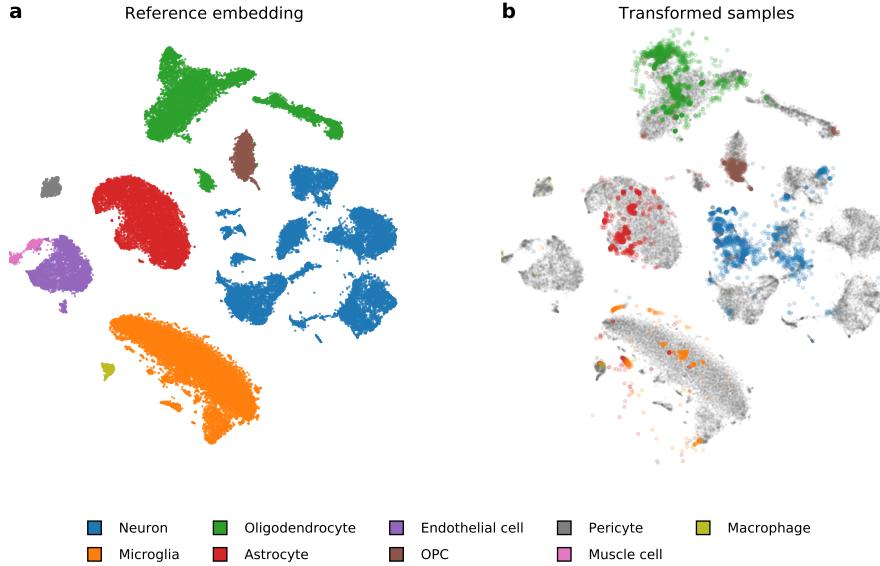


Fig. 2. A two-dimensional embedding of a reference containing brain cells (a) and the corresponding mapping of secondary data containing hypothalamic cells (b). The majority of hypothalamic cells were mapped to their corresponding reference cluster. For instance, astrocyte cells marked with red on the right were mapped to an oval cluster of same-typed cells denoted with the same color in the visualization on the left.

Below, we start with a summary of t-SNE and its extensions (Sec. 2.1, introducing the relevant notation, upon which we base our secondary data embedding approach (Sec. 2.2).

2.1 Data embedding by t-SNE and its extensions

Local, non-linear dimensionality reduction by t-SNE is performed as follows. Given a multi-dimensional data set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^D$ where N is the number of data points in the reference data set, t-SNE aims to find a low dimensional embedding $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\} \in \mathbb{R}^d$ where $d \ll D$, such that if points \mathbf{x}_i and \mathbf{x}_j are close in the multi-dimensional space, their corresponding embeddings \mathbf{y}_i and \mathbf{y}_j are also close. Since t-SNE is primarily used as a visualization tool, d is typically set to two. The similarity between two data points in t-SNE is defined as:

$$p_{j|i} = \frac{\exp(-\frac{1}{2}\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)/\sigma_i^2)}{\sum_{k \neq i} \exp(-\frac{1}{2}\mathcal{D}(\mathbf{x}_i, \mathbf{x}_k)/\sigma_i^2)}, \quad p_{i|i} = 0 \quad (1)$$

where \mathcal{D} is a distance measure. This is then symmetrized to

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}. \quad (2)$$

The bandwidth of each Gaussian kernel σ_i is selected such that the perplexity of the distribution matches a user-specified parameter value

$$\text{Perplexity} = 2^{H(P_i)} \quad (3)$$

where $H(P_i)$ is the Shannon entropy of P_i ,

$$H(P_i) = - \sum_i p_{j|i} \log_2(p_{j|i}). \quad (4)$$

Different bandwidths σ_i enable t-SNE to adapt to the varying density of the data in the multi-dimensional space.

The similarity between points \mathbf{y}_i and \mathbf{y}_j in the embedding space is defined using the t -distribution with one degree of freedom

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}, \quad q_{ii} = 0. \quad (5)$$

The t-SNE method finds an embedding \mathbf{Y} that minimizes the Kullback-Leibler (KL) divergence between \mathbf{P} and \mathbf{Q} ,

$$C = \text{KL}(\mathbf{P} \parallel \mathbf{Q}) = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (6)$$

The time complexity needed to evaluate the similarities in Eq. 5 is $\mathcal{O}(N^2)$, making its application impractical for large data sets. We adopt a recent approach for low-rank approximation of gradients based on polynomial interpolation which reduces its time complexity to $\mathcal{O}(N)$. This approximation enables the visualization of massive data sets, possibly containing millions of data points [16].

The resulting embeddings substantially depend on the value of the perplexity parameter. Perplexity can be interpreted as the number of neighbors for which the distances in the embedding space are preserved. Small values of perplexity result in tightly-packed clusters of points and effectively ignore the long-range interactions between clusters. Larger values may result in a more globally consistent visualizations — preserving distances on a large scale and organizing clusters in a more meaningful way — but can lead to merging small clusters and thus obscuring local aspects of the data [15].

The trade-off between the local organization and global consistency may be achieved by replacing the Gaussian kernels in Eq. 1 with a mixture of Gaussians of varying bandwidths [17]. Multi-scale kernels are defined as

$$p_{j|i} \propto \frac{1}{L} \sum_{l=1}^L \exp \left(-\frac{1}{2} \mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) / \sigma_{i,l}^2 \right), \quad p_{i|i} = 0 \quad (7)$$

where L is the number of mixture components. The bandwidths $\sigma_{i,l}$ are selected in the same manner as in Eq. 1, but with a different value of perplexity for each l . In our experiments, we used a mixture of two Gaussian kernels with perplexity values of 50 and 500. A similar formulation of multi-scale kernels was proposed in [15], and we found the resulting embeddings are visually very similar to those obtained with the approach described above (not shown for brevity).

2.2 Adding new data points to reference embedding

Our algorithm, which embeds new data points to a reference embedding, consists of estimating similarities between each new point and the reference data and optimizing the position of each new data point in the embedding space. Unlike parametric models such as principal component analysis or autoencoders, t-SNE does not define an explicit mapping to the embedding space, and embeddings need to be found through loss function optimization.

The position of a new data point in embedding space is initialized to the median reference embedding position of its k nearest neighbors. While we found the algorithm to be robust to choices of k , we use $k = 10$ in our experiments.

We adapt the standard t-SNE formulation from Eqs. 1 and 5 with

$$p_{j|i} = \frac{\exp\left(-\frac{1}{2}\mathcal{D}(\mathbf{x}_i, \mathbf{v}_j)/\sigma_i^2\right)}{\sum_i \exp\left(-\frac{1}{2}\mathcal{D}(\mathbf{x}_i, \mathbf{v}_j)/\sigma_i^2\right)}, \quad (8)$$

$$q_{j|i} = \frac{(1 + \|\mathbf{y}_i - \mathbf{w}_j\|^2)^{-1}}{\sum_i (1 + \|\mathbf{y}_i - \mathbf{w}_j\|^2)^{-1}}, \quad (9)$$

where $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M\} \in \mathbb{R}^D$ where M is the number of samples in the secondary data set and $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M\} \in \mathbb{R}^d$. Additionally, we omit the symmetrization step in Eq. 2. This enables new points to be inserted into the embedding independently of one another. The gradients of \mathbf{w}_j with respect to the loss (Eq. 6) are:

$$\frac{\partial C}{\partial \mathbf{w}_j} = 2 \sum_i (p_{j|i} - q_{j|i}) (\mathbf{y}_i - \mathbf{w}_j) (1 + \|\mathbf{y}_i - \mathbf{w}_j\|^2)^{-1} \quad (10)$$

In the optimization step, we refine point positions using batch gradient descent. We use an adaptive learning rate scheme with momentum to speed up the convergence, as proposed by Jacobs [18, 19]. We run gradient descent with momentum α of 0.8 for 250 iterations, where the optimization converged in all our experiments. The time complexity needed to evaluate the gradients in Eq. 10 is $\mathcal{O}(N \cdot M)$, however, by adapting the same polynomial interpolation based approximation, this is reduced to $\mathcal{O}(\max\{N, M\})$.

Special care must be taken to reduce the learning rate η as the default value in most implementations ($\eta = 200$) may cause points to “shoot off” from the reference embedding. This phenomenon is caused due to the embedding to a previously defined t-SNE space, where the distances between data points and

corresponding gradients of the optimization function may be quite large. When running standard t-SNE, points are initialized and scaled to have variance 0.0001. The resulting gradients tend to be very small during the initial phase, resulting in stable convergence. When embedding new samples, the span of the embedding is much larger, resulting in substantially larger gradients, and the default learning rate causes points to move very far from the reference embedding. In our experiments, we found that decreasing the learning rate to $\eta \sim 0.1$ produces stable solutions. This is especially important when using the interpolation-based approximation, which places a grid of interpolation points over the embedding space, where the number of grid points is determined by the span of the embedding. Clearly, if even one point “shoots off” far from the embedding, the number of required grid points may grow dramatically, increasing the runtime substantially. The reduced learning rate suppresses this issue, and does not slow the convergence because of the adaptive learning rate scheme, provided the optimization is run for a sufficient number of steps.

3 Experiments and Discussion

We apply the proposed approach to t-SNE visualizations of single-cell data. Data in this realm include a variety of cells from specific tissues and are characterized through gene expression. In our experiments, we considered several recently published data sets where cells were annotated with the cell type. Our aim was to construct t-SNE visualizations where similarly-typed cells would cluster together, despite systematic differences between data sources. To that end, we focus on comparing different ways of using t-SNE rather than differences to embeddings like PCA or MDS, which have been substantially covered before [1, 4]. Below, we list the data sets, describe single-cell specific data preprocessing procedures, and display the resulting data visualizations. Finally, we discuss the success of the proposed approach in alleviating the batch effects.

3.1 Data

We use three pairs of reference and secondary single-cell data sets originating from different organisms and tissues. The data in each pair were chosen so that the majority of cell types from the secondary data set were included in the reference set (Table 1). The cells in the data sets originate from the following three tissues:

Mouse brain. The data set from Hrvatin *et al.* [11] contains cells from the visual cortex exploring transcriptional changes after exposure to light. This was used as a reference for the data from Chen *et al.* [12], containing cells from the mouse hypothalamus and their reaction to food deprivation. From the secondary data, we removed cells with no corresponding types in the reference: tanyocytes, ependymal, epithelial, and unlabelled cells.

Human pancreas. Baron *et al.* [13] created an atlas of pancreatic cell types.

We used this set as a reference for data from Xin *et al.* [14], who examined transcriptional differences between healthy and type 2 diabetic patients.

Mouse retina. Macosko *et al.* [20] created an atlas of mouse retinal cell types.

We used this as a reference for the data from Shekhar *et al.* [21], who built an atlas for retinal bipolar cells.

Study	Organism/Tissue	Protocol	Cells	Cell Types	Sparsity (%)
Hrvatin <i>et al.</i>	mouse brain	inDrop	48,266	9	94
Chen <i>et al.</i>		Drop-seq	14,437	6	93
Baron <i>et al.</i>	human pancreas	inDrop	8,569	9	91
Xin <i>et al.</i>		SMARTer	1,492	4	86
Macosko <i>et al.</i>	mouse retina	Drop-seq	44,808	12	97
Shekhar <i>et al.</i>		Drop-seq	27,499	5	96

Table 1. Data sets used in our experiments. The first data set in a pair (Hrvatin *et al.*, Baron *et al.*, and Macosko *et al.*) was used as a reference. We relied on the quality control and annotations from the original publication and report the number of cell types after preprocessing. The cell annotations were made consistent to annotations from the Cell Ontology [22]. Notice that different RNA sequencing protocols were used to estimate gene expressions.

3.2 Single-cell data preprocessing pipeline

Due to the specific nature of single-cell data, additional steps must be taken to properly apply t-SNE. We use a standard single-cell preprocessing pipeline, consisting of the selection of 3,000 representative genes (see Sec. 3.3), library size normalization, log-transformation, standardization, and PCA-based representation that retains 50 principal components [10, 23]. To obtain the reference embedding, we apply multi-scale t-SNE using PCA initialization [15]. Due to high-dimensionality of the preprocessed input data we use cosine distance to estimate similarities between reference data points [24]. When adding new data points from the secondary data set to the reference embedding, we select 1,000 genes present in both data sets and use the cosine similarity to estimate the similarities between the secondary data item and reference data points. We note that similarities are computed using the raw count matrices. The preprocessing stages are detailed in accompanying Python notebooks (Sec. 3.5).

3.3 Gene selection

Single-cell data sets suffer from high levels of technical noise and low capture efficiency, resulting in sparse expression matrices [25]. To address this problem,

we use a specialized feature-selection method, which exploits the mean-dropout relationship of expression counts as recently proposed by Kobak and Berens [15]. Here, genes with higher than expected dropout rate are regarded as potential markers for cell subpopulations and are retained in the data.

Given an expression matrix $\mathbf{X} \in \mathbb{R}^{N \times G}$ where N is the number of samples and G is the number of genes in the data set, we compute the fraction of cells where a gene g was not expressed

$$d_g = \frac{1}{N} \sum_i I(X_{ig} = 0) \quad (11)$$

The mean \log_2 expression of gene g considers only cells expressing g :

$$m_g = \langle \log_2 X_{ig} \mid X_{ig} > 0 \rangle. \quad (12)$$

All genes expressed in less than ten cells are discarded. In order to select a specific number of \hat{G} genes, we use a binary search to find a value b such that

$$\sum_g I(d_g > \exp[-(m_g - b)] + 0.02) = \hat{G}. \quad (13)$$

3.4 Results and Discussion

Figs. 2, 3, and 4 show the embeddings of the reference data sets and their corresponding embeddings of the secondary data sets. In all the figures, the cells from the secondary data sets were positioned in the cluster of same-typed reference cells, providing strong evidence of the success of our approach. There are some deviations to these observations; for instance, in Fig. 2 several oligodendrocyte precursor cells (OPCs) were mapped to oligodendrocytes. This may be due to differences in annotation criteria by different authors, or due to inherent similarities of these types of cells. Examples of such erroneous placements can be found in other figures as well, but are uncommon and constitute less than 5% of the cells (less than 5% in brain, 1% in pancreas and 2% in retina secondary data).

Additional experiments with embedding secondary data sets from independent studies are provided in the accompanying Python notebooks. We embedded brain cells from Zeisel *et al.* [26] to the reference embedding shown in Fig. 2.a, and pancreas cells from Muraro *et al.* [27] to reference in 3.a. The conclusions are largely consistent, with the error rates of X% and Y%. To do ???

Notice that we could simulate the split between reference and secondary data sets using one data set only and perform cross-validation, however this type of experiment would not incorporate batch effects. We want to remind the reader that handling batch effects were central to our endeavour and that the disregard of this effect could lead to overly-optimistic results and data visualizations strikingly different from ours. For example, compare the visualizations from Fig. 1.a and Fig. 2.b, or Figs. 1.b and 3.b.

We use a number of additional, recently proposed modifications to enhance the t-SNE visualization of the reference data set. One important extension is the

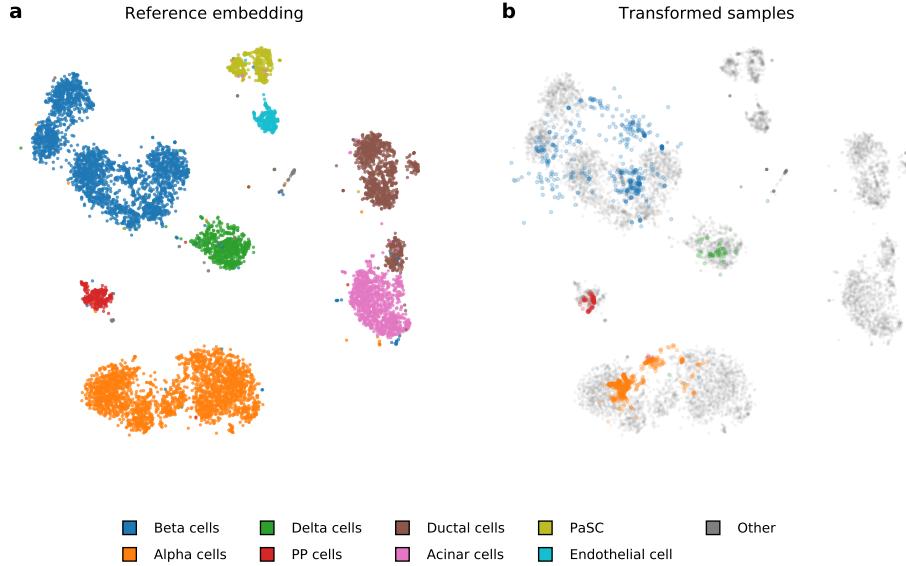


Fig. 3. Embedding of pancreatic cells from Baron *et al.* [13] and cells from the same tissue from Xin *et al.* [14]. Just like in Fig. 2 the vast majority of the cells from the secondary data set were correctly mapped to the same-typed cluster of reference cells.

use of multi-scale similarities, which, in addition to finding local structure, also ensure consistent organization of the resulting clusters. For illustration, consider visualizations with standard and multi-scale t-SNE in Fig. 5. Notice, for instance, that in multi-scale t-SNE (Fig. 5.b) the clusters with neuronal cells are clumped together, while their placement in standard t-SNE is arbitrary (Fig. 5.a).

We also observed the important role of gene selection in crafting the reference embedding spaces. We found that when selecting an insufficient number of genes, the resulting visualizations display overly-fragmented clusters. When the selection is too broad and includes lowly expressed genes, the subclusters tend to overlap. These effects can all be attributed to sparseness of the data sets and may be intrinsic to single-cell data. In our studies, we found that selection of 3,000 genes yields most informative visualizations (Fig. 6).

In principle, our theoretically-grounded embedding of secondary data into the scaffold defined by the reference embedding could be simplified with the application of the nearest neighbors-based procedure. For example, while describing a set of tricks for t-SNE, Kobak and Berens [15] proposed positioning new points into a known embedding by placing them in the median position of their 10 nearest neighbors, where the neighborhood was estimated in the original data space. Notice that we use this approach as well, but only for the initialization of positions of new data instances that are subject to further optimization. Despite both nearest-neighbors search and t-SNE optimization can be computed in lin-

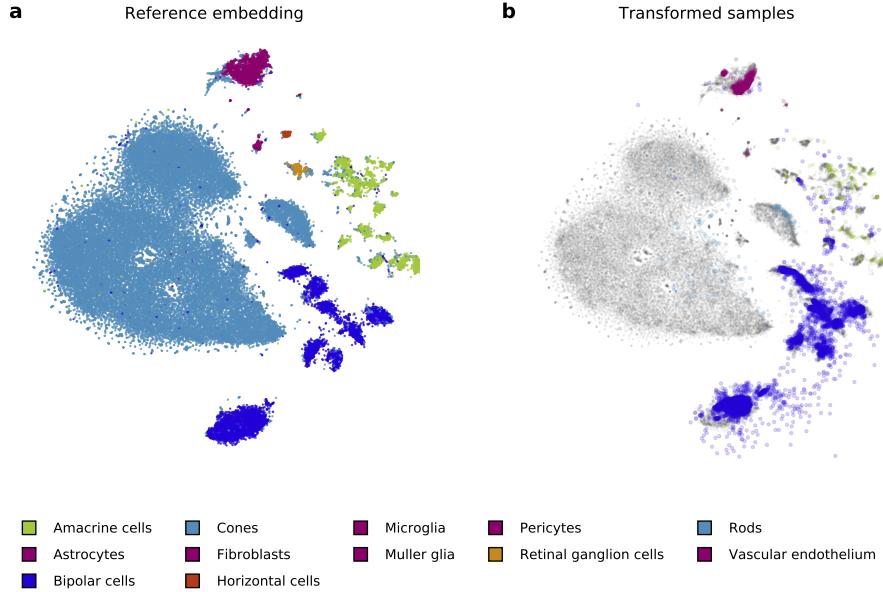


Fig. 4. An embedding of a large reference of retinal cells from Macosko *et al.* [20] (a) and mapping of cells from a smaller study that focuses on bipolar cells from Shekhar *et al.* [21] (b). We use colors consistent with the study by Macosko *et al.*.

ear time, the former dominates the runtime (mouse retina example; M reference, N secondary cells, X min N-search, Y min optimization). **To do (??)**

We demonstrate a case where nearest neighbor-based positioning is insufficient in Fig. 7. It may yield clumped visualizations where the optimal positioning using the t-SNE loss function is much more dispersed and rightfully shows a more considerable variation in the secondary data. Some points may also fall into the empty regions, while after optimization they typically move closer to same-typed groups. While the nearest neighbor-based approach always places new data points to their most similar cluster, our proposed t-SNE approach can identify extreme outliers and place them far from the reference cluster.

Our approach assumes that all cell types from the secondary data set are present in the reference. The method would fail to reveal novel cell types in the secondary data set, possibly positioning them arbitrarily close to unrelated clusters. Procedures such as scmap were recently proposed to cope with such cases and identify the cells whose type is new and not included in the reference [28].

Our procedure is, therefore, asymmetrical in the choice of reference and secondary data set. In practice, however, newly produced *secondary* data would be embedded into previously-prepared reference landscapes. Large collections of data *e.g.* the Human Cell Atlas initiative [29] make it possible to scale up our approach to wider sets of cell types.

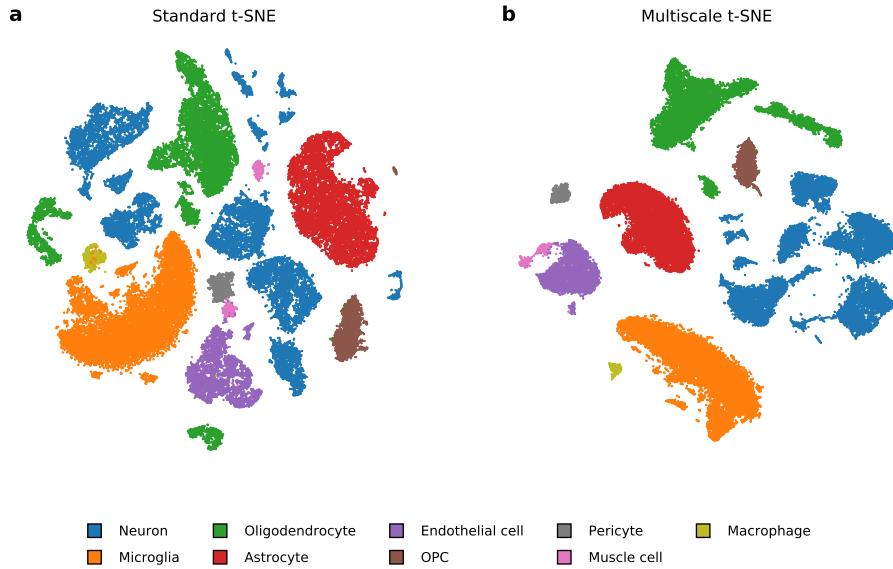


Fig. 5. A comparison of standard and multi-scale t-SNE on data from the mouse visual cortex [11]. (a) Standard t-SNE places clusters arbitrarily. (b) Augmenting t-SNE with multi-scale similarities and using proper initialization provides a more meaningful layout of the clusters. Neuronal types occupy one region of the space. Oligodendrocyte precursor cells (OPCs) are mainly progenitors to oligodendrocytes, but may also differentiate into neurons or astrocytes.

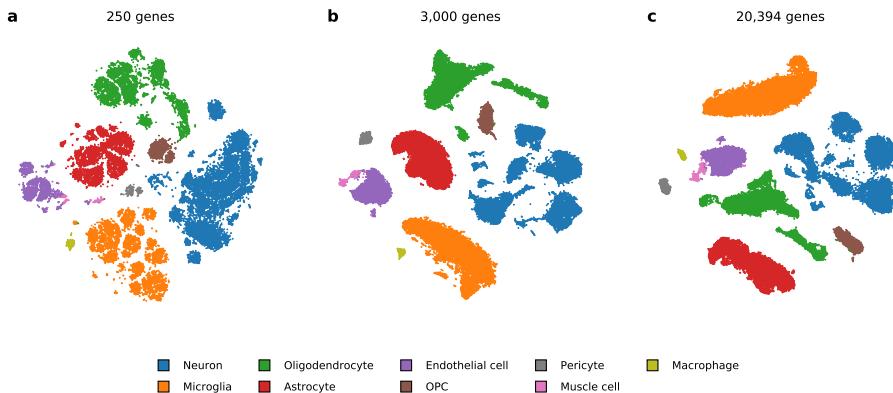


Fig. 6. Gene selection plays an important role when constructing the reference embedding. (a) Using too few genes results in fragmented clusters. (b) Using an intermediate number of genes reveals clustering mostly consistent with cell annotations. (c) Including all the genes may lead to under-clustering of the more specialized cell types.

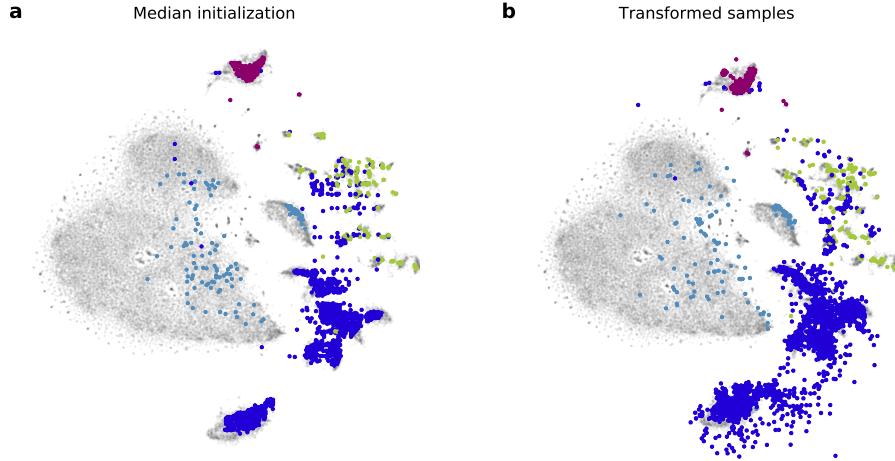


Fig. 7. Comparison of data placement using the nearest neighbors approach from Kobak and Berens [15] and the optimized placement using our algorithm. **(a)** Data points are placed to the median position of their 10 nearest neighbors in the reference set. **(b)** Point positions are optimized, revealing a different, more dispersed placement that better reflects the variety of cells in the secondary data set.

3.5 Implementation

The procedures described in this paper are provided as Python notebooks that are, together with the data, available in an open repository³. All experiments were run using `openTSNE`, our open and extensible t-SNE library for Python [30].

4 Conclusion

Almost all recent publications of single-cell studies begin with a two-dimensional visualization of the data that reveals cellular diversity. While many dimensionality reduction techniques are available, different variants of t-SNE are most often used to produce such visualizations. Single-cell studies enable the exploration of biological mechanisms at a cellular level, and their publications in the past couple of years are abundant. One of the central tasks in single-cell studies is the classification of new cells based on findings from previous studies. Such transfer of knowledge is often difficult due to batch effects present in data from different sources. Addressing batch effects by adapting and extending t-SNE, the prevailing method used to present single-cell data in two-dimensional visualization, motivated the research presented in this paper.

Our proposed approach uses a t-SNE embedding as a scaffold for the positioning of new cells within the visualization, and possibly for aiding in their

³ <https://github.com/biolab/tsne-embedding>

classification. The three case studies incorporating pairs of data sets from different domains but with similar classifications demonstrate that our proposed procedure can effectively deal with batch effects to construct visualizations that correctly map secondary data sets onto a reference data set from an independent study that possibly uses different experimental protocol. While we focused here on reference visualizations constructed using t-SNE, this approach can be applied using any existing two-dimensional visualization.

Acknowledgements This work was supported by the Slovenian Research Agency Program Grant P2-0209, and by the BioPharm.SI project supported from European Regional Development Fund and the Slovenian Ministry of Education, Science and Sport. We would also like to thank Dmitry Kobak for many helpful discussions on t-SNE.

References

1. Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
2. L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, February 2018.
3. Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-SNE effectively. *Distill*, 1(10):e2, 2016.
4. Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38, 2019.
5. Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 International Conference on Computer Vision*, pages 999–1006. IEEE, 2011.
6. Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(Sep):2137–2155, 2009.
7. Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.
8. Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411, 2018.
9. Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36(5):421, 2018.
10. Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive Integration of Single-Cell Data. *Cell*, 2019.
11. Sinisa Hrvatin, Daniel R Hochbaum, M Aurel Nagy, Marcelo Cicconet, Keiramarie Robertson, Lucas Cheadle, Rapolas Zilionis, Alex Ratner, Rebeca Borges-Monroy, Allon M Klein, et al. Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex. *Nature Neuroscience*, 21(1):120, 2018.

12. Renchao Chen, Xiaoji Wu, Lan Jiang, and Yi Zhang. Single-cell RNA-seq reveals hypothalamic cell diversity. *Cell Reports*, 18(13):3227–3241, 2017.
13. Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell Systems*, 3(4):346–360, 2016.
14. Yurong Xin, Jinrang Kim, Haruka Okamoto, Min Ni, Yi Wei, Christina Adler, Andrew J Murphy, George D Yancopoulos, Calvin Lin, and Jesper Gromada. RNA sequencing of single human islet cells reveals type 2 diabetes genes. *Cell Metabolism*, 24(4):608–615, 2016.
15. Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *bioRxiv*, page 453449, 2018.
16. George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nature Methods*, 16(3):243, 2019.
17. John A Lee, Diego H Peluffo-Ordóñez, and Michel Verleysen. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, 169:246–261, 2015.
18. Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1(4):295–307, 1988.
19. Laurens van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014.
20. Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.
21. Karthik Shekhar, Sylvain W Lapan, Irene E Whitney, Nicholas M Tran, Evan Z Macosko, Monika Kowalczyk, Xian Adiconis, Joshua Z Levin, James Nemesh, Melissa Goldman, et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166(5):1308–1323, 2016.
22. Jonathan Bard, Seung Y Rhee, and Michael Ashburner. An ontology for cell types. *Genome Biology*, 6(2):R21, 2005.
23. F Alexander Wolf, Philipp Angerer, and Fabian J Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018.
24. Pedro M Domingos. A few useful things to know about machine learning. *Communications fo the ACM*, 55(10):78–87, 2012.
25. Saiful Islam, Amit Zeisel, Simon Joost, Gioele La Manno, Pawel Zajac, Maria Kasper, Peter Lönnerberg, and Sten Linnarsson. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2):163, 2014.
26. Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015.
27. Mauro J Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon van Gurp, Marten A Engelse, Francoise Carlotti, Eelco JP de Koning, et al. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4):385–394, 2016.
28. Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell RNA-seq data across data sets. *Nature Methods*, 15(5):359, 2018.

29. Orit Rozenblatt-Rosen, Michael JT Stubbington, Aviv Regev, and Sarah A Teichmann. The Human Cell Atlas: from vision to reality. *Nature News*, 550(7677):451, 2017.
30. Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. *bioRxiv*, 2019.