Rapid single-cell cytometry data visualization with EmbedSOM

Miroslav Kratochvíl[a,b], Abhishek Koladiya[c], Jana Balounova[d], Vendula Novosadova[d], Karel Fišer[e], Radislav Sedlacek[d], Jiří Vondrášek[a], Karel Drbal[c]

[a]*Institute of Organic Chemistry and Biochemistry of the CAS, Prague*
[b]*Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague*
[c]*Department of Cell Biology, Faculty of Science, Charles University, Prague*
[d]*Czech Centre for Phenogenomics, Institute of Molecular Genetics of the CAS, Prague*
[e]*Childhood Leukaemia Investigation Prague (CLIP), 2nd Faculty of Medicine, Charles University and University Hospital Motol, Prague, Czech Republic*

Abstract

Efficient unbiased data analysis is a major challenge for laboratories handling large cytometry datasets. We present EmbedSOM, a non-linear embedding algorithm based on FlowSOM that improves the analyses by providing high-performance visualization of complex single cell distributions within cellular populations and their transition states. The algorithm is designed for linear scaling and speed suitable for interactive analyses of millions of cells without downsampling. At the same time, the visualization quality is competitive with current state-of-art algorithms. We demonstrate the properties of EmbedSOM on workflows that improve two essential types of analyses: The native ability of EmbedSOM to align population positions in embedding is used for comparative analysis of multi-sample data, and the connection to FlowSOM is exploited for simplifying the supervised hierarchical dissection of cell populations. Additionally, we discuss the visualization of the trajectories between cellular states facilitated by the local linearity of the embedding.

Keywords: flow cytometry, mass cytometry, automated analysis, high-dimensional data, embedding, clustering

1. Introduction

The ever-increasing size and dimensionality of data generated by flow and mass cytometry experiments drive interest in simplifying data analysis. Employing the usual repetitive manual gating and exploration techniques is tedious if the sample count is high and becomes imprecise on high complexity data. During the past decade, a multitude of automated analysis methods have been introduced, including various unsupervised clustering and phenotyping algorithms, and embedding methods. Comprehensive reviews of the algorithms are available (Saeys et al., 2016; Weber and Robinson, 2016; Kimball et al., 2018; Konstorum et al., 2018).

The preferred method to *display* cytometry datasets is embedding, in which cells are arranged into a 2-dimensional picture showing populations of agglomerated cells with similar properties. This provides a straightforward way to inspect the relative population sizes, their contents, and the presence of various features including subpopulations and trajectories of cell development. The performance of available embedding algorithms is constantly being improved. For example, tSNE Van Der Maaten (2014) has formed the basis for faster ASNE (Pezzotti et al., 2017) and HSNE (Pezzotti et al., 2016), and was further accelerated using GPU by Chan et al. (2018) and optimized in OptSNE and FItSNE (Belkina et al., 2018; Linderman et al., 2017). Similarly, the relatively new UMAP specifically aims to provide better, faster embedding than tSNE (McInnes and Healy, 2018). Despite these developments, two key objectives have not been met:

- Time and memory consumption of the embedding algorithm should scale linearly with the number of cells to be able to keep up with the increasing sizes of datasets;
- The algorithm should be able to process the data of volumes common in cytometry quickly, ideally within seconds, to allow interactive data inspection.

We introduce EmbedSOM, a new embedding algorithm that is designed to satisfy these two requirements. The algorithm uses a self-organizing map (SOM) that describes the multidimensional cell space. Existing algorithms successfully use SOMs for clustering this space — for example, FlowSOM (Van Gassen et al., 2015) uses SOM grid vertices as cluster centers to classify the cells into *pre-clusters* that are used as a basis for further analysis, such as for aggregating them into metaclusters. EmbedSOM differs by using the 'residual' spatial information

about SOM geometry — the SOM grid approximates a section of a *smooth* 2-manifold embedded in the multidimensional space in a manner such that the cells are uniformly distributed in its neighborhood. EmbedSOM computes the embedding by fitting a projection of each cell onto this manifold and transforming the projection coordinates to the 2-dimensional grid-relative coordinates, which are used as the result. The geometric interpretation of the method is similar to elastic maps (Gorban et al., 2001) or simplified generative topographic map-like manifold projections (Tino and Nabney, 2002).

The performance-oriented design of EmbedSOM differs substantially from other commonly used embedding algorithms. Most importantly, the usual time-consuming iterative optimization of single cell positions is replaced by relatively fast SOM training. Additionally, the separation of the SOM-training stage and single-cell projection stage introduces flexibility in manipulation of the intermediate SOM. As we demonstrate, the same SOM can be re-used for embedding cells different than those it was originally built for. This allows production of multiple embeddings with aligned cell population positions without any computational overhead, introduction of new cells into the existing embedding, and embedding of non-cellular features present in the multidimensional space (e.g., gates and other informative geometry).

Here, we describe the EmbedSOM algorithm and results of benchmarking it against other embedding algorithms on several datasets. In addition to evaluating performance, our benchmark also assessed the relevance of the embedding by relating the result to ground-truth present in manual gating and unsupervised clustering. Additionally, we present two use cases that exploit the properties of EmbedSOM to simplify common analyses of large, high-dimensional, multi-sample datasets. In the first, the embedding alignment is used to visualize development of samples over time and rapidly provide a clear view of high-dimensional sample differences. In the second, the information from SOM training is used to augment this view with information from FlowSOM-based automated clustering and statistical analysis, which we use to construct a hierarchical dissection workflow that is similar to manual gating but provides better precision and speed by offloading the most time-consuming and error-prone parts of the process onto the computer.

dataset	#cells×dim.	tSNE	UMAP	EmbedSOM	EmbedSOM (GPU)
`Levine-13`	82k×13	20m 04s	52s	10s	**2s**
`Levine-32`	104k×32	42m 44s	1m 27s	31s	**17s**
`Samusik-01`	53k×40	20m 25s	44s	20s	**9s**
`Samusik-all`	514k×40	3h 33m 15s	17m 05s	3m 10s	**1m 33s**
`iPSC`	125k×5	34m 20s	1m 57s	13s	**2s**
`Pregnancy`	16M×40	–	–	50m 40s	**22m 45s**

Table 1: Performance of the software on sample analyses as time required to compute the embedding. Displayed cell count and dimension corresponds only to that used for analysis (i.e., with debris removed, or downsampled in the `Pregnancy` dataset). Times for EmbedSOM are for combined SOM-building and projection stages; the first stage took roughly 70% of the total time in most cases.

2. Results

2.1. Implementation

Algorithm implementation consists of the SOM-building stage, which is shared with FlowSOM, and manifold projection stage, which is specific for EmbedSOM. The CPU-based implementation of EmbedSOM is made available in the `EmbedSOM` R package, to aid interoperability with FlowSOM and other R-based software packages. The embedding algorithm was implemented in C++ for efficiency. As both the SOM-building and projection stages are easily parallelizable, we also implemented GPU-accelerated versions using Vulkan® API for portability to most GPUs and platforms; the resulting package is called `vkEmbedSOM`. This accelerated implementation is based on the work of Xiao et al. (2015), with modifications to make it suitable for the relatively lower dimensionality (≤ 100) and higher data point ($\geq 10^6$) count of cytometry data.

Both R packages are available as free software at `http://bioinfo.uochb.cas.cz/embedsom/`, along with documentation and examples of common use-cases.

2.2. EmbedSOM provides superior embedding speed

The main advantage of EmbedSOM is its computational efficiency. A dataset of common size (approximately 300k cells and 20 markers) can be mapped by the SOM and embedded in less than a minute; the GPU-accelerated versions of the algorithms deliver the same result in seconds. Generally, embedding datasets with more than 10^7 cells and several dozen markers is possible in minutes using common office hardware. Moreover, since the major part of the required computation is shared with FlowSOM, EmbedSOM visualization adds only minor computational complexity to workflows that already use FlowSOM.

Quantitative measurements of the speed advantage on the benchmark computations are displayed in Table 1. The amount of memory used by EmbedSOM for the computation (excluding the raw loaded cell data) did not increase with the cell count because the algorithm does not need to retain any temporary information for single cells. EmbedSOM itself used less than 50MB of main memory and 30MB of GPU memory in all tests we conducted. UMAP and tSNE used less than 2GB of CPU memory in all cases.

Speed measurements confirmed the expected scaling difference between UMAP and EmbedSOM (see Figure 1b). This can be explained by differences in algorithm design. UMAP and tSNE are very efficient for high-dimensional data with a low datapoint count, because they remove the dimensionality overhead early in the process but require a quadratic amount of computation to optimize the final data point positions. The performance of both EmbedSOM stages is linear in the number of data points, but the dimensionality overhead is present throughout almost the entire computation. EmbedSOM benefits from this tradeoff when applied to high-volume flow and mass cytometry data that are physically limited to dozens of dimensions. Conversely, UMAP remains much faster on high-dimensional, low-volume datasets, such as raw single-cell RNA sequencing data, which must be pre-processed e.g. by linear dimensionality reduction for efficient use with SOM-based algorithms (Duò et al., 2018).

2.3. EmbedSOM visualization quality is competitive with other algorithms

The quality of embedding visualization was measured by relating the contents of k-neighborhoods in the embedding to the ground-truth available from manual gating and FlowSOM-based clustering. k-NN entropies and k-NN purities of the embeddings of benchmark data are shown for the `Levine-13` dataset in Figure 1a (see Section 4.2.3 for definitions of the measures used). All algorithms provided visualizations of comparable quality, and we consider the slight disadvantage of EmbedSOM to be a reasonable tradeoff for the performance gain. Measurements for other datasets and clustering methods yielded similar results. (Data from all benchmarks are available in section S2).

The measured quality difference between EmbedSOM, UMAP and tSNE, most visible in the more interspersed part of the data, arises from the design of the algorithms. Specifically, neither UMAP nor tSNE aims to preserve local linearity of the transformation, which allows them to take apart the clusters with noisy data and attach the residual noise to nearest clusters. This makes the embedding arguably more visually appealing by creating well-defined, undistorted borders, and at the same time improves the used embedding quality measures by reducing

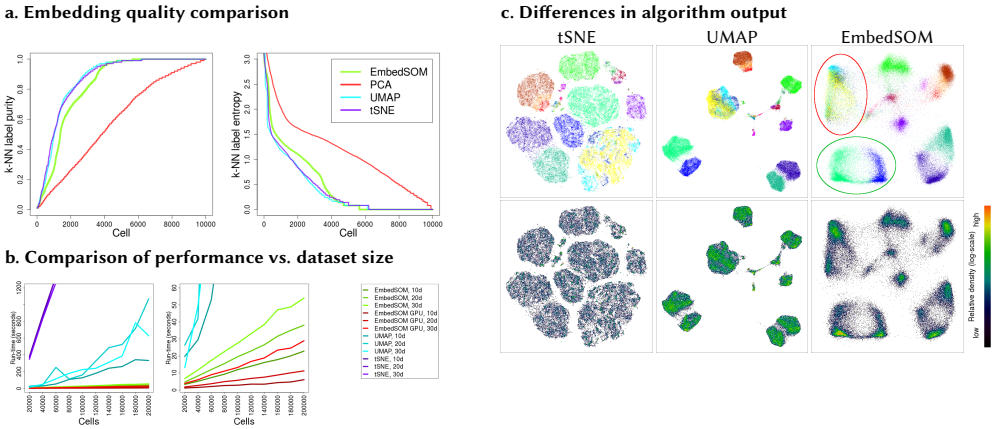

Figure 1: Benchmark results. **a.** Left: k-NN label purity of 10,000 randomly selected embedded cells in the `Levine-13` dataset for $k = 100$ plotted as a reliability distribution function for better comparison; higher value is better. Right: k-NN entropy in the same dataset, measured in bits; lower is better. **b.** Left: Benchmark of performance scaling shows time required by different algorithms for processing datasets of different sizes and dimensionalities. Right: A magnified plot is provided to show linear scaling of EmbedSOM. **c.** Comparison of population positioning (upper row) and preservation of internal population density (lower row) in tSNE, EmbedSOM and UMAP on the `Levine-13` dataset; color in the upper row represents the manual population classification. Cluster annotation is available in Table S1.

the chance of a cell from a different population occurring in a k-neighborhood. While this separation may be desirable if the embedding is expected to approximate the population boundaries, it may be inappropriate if the population environment is relevant for analysis. For example, tight packing of cells impairs the possibility to observe the natural population density distribution or to filter out noise manually. Additionally, potentially misplaced borders are highly undesirable in many situations, such as analysis of development trajectories. These differences can be observed in Figure 1c and in figures in section S2.

Despite the apparent noise in Figure 1c, the populations partially distorted by EmbedSOM can be precisely reconstructed from the underlying FlowSOM information, as described in Section 2.5. Conversely, the relevant visual information about the high variance in the population of erythroblasts highly interspersed with myelocytes and megakaryocytes (highlighted in red in Figure 1c, see Table S1 for complete annotation) is observable in neither UMAP nor tSNE embeddings. Additionally, only EmbedSOM shows differentiation of both naïve and mature CD4 cells to sub-populations based on presence of CD90 marker (highlighted in green).

2.4. Population alignment aids visualization of sample differences

The original aim of EmbedSOM was to simplify visualization of differences in time-series and other multi-sample data. Two examples from recent studies illustrate this functionality (Figure 2): To visualize differences in time series, we used data by Takeda et al. (2018), who examined the time course (from day 0 to day 15) expression of cell surface markers CD82, PDGFRα and CD13 in human-induced pluripotent stem cells (hiPSCs). In another example, we applied EmbedSOM to mass cytometry dataset by Aghaeepour et al. (2017) that was obtained from 18 women where the whole-blood samples were collected in 4 time points during and after pregnancy (early, mid- and late-pregnancy, and 6 weeks postpartum). All samples were measured in unstimulated state as well as stimulated with LPS, IFN-α and IL-2+IL-6, total 33 markers were used to study immune system function and regulation.

The EmbedSOM workflow used to generate the images enables a massive performance gain when processing a high volume of samples. This gain stems from the fact that the training of the SOM, which is the most computationally intensive part of the workflow, is run only once. After the SOM is built, any number of additional samples and cells can be embedded in time linear with the cell count.

Because the population positions are retained in embeddings of all samples, visual analysis of the data is simplified to identifying easily observable presence, density or position changes in the cell populations, followed by finding their biological meaning in the plots of marker expression (or vice versa). In this manner, the visualization of `iPSC` dataset clearly illustrates the movement of *in vitro* differentiating PDGFRα^+ and CD13$^+$ population (Figure 2a). The CD82$^+$ population starts to appear from day 4, peaks at day 7 (circled in black in the figure) and decreases gradually, which corresponds with the original observation of the study (Takeda et al., 2018, Fig. 1D). The unique CD82$^+$ PDGFRα^+ CD13$^+$, which was sorted and used to confirm that CD82 is a specific cell-surface marker for cardiomyocyte-fated progenitors in the original paper, appears at day 5 (circled in red). UMAP embedding of the `iPSC` dataset is provided for comparison of the differentiation trajectory shapes.

Visual comparison in the `Pregnancy` dataset is complicated by the high sample count and high variability in population distribution. To alleviate this problem, we ran statistical hypothesis testing on the contents of pre-clusters and metaclusters created by FlowSOM, and highlighted the statistically significant changes in the embedding. Similar bulk visualization of the cluster differences has been applied in other algorithms, such as diffcyt (Weber et al., 2018, Figure

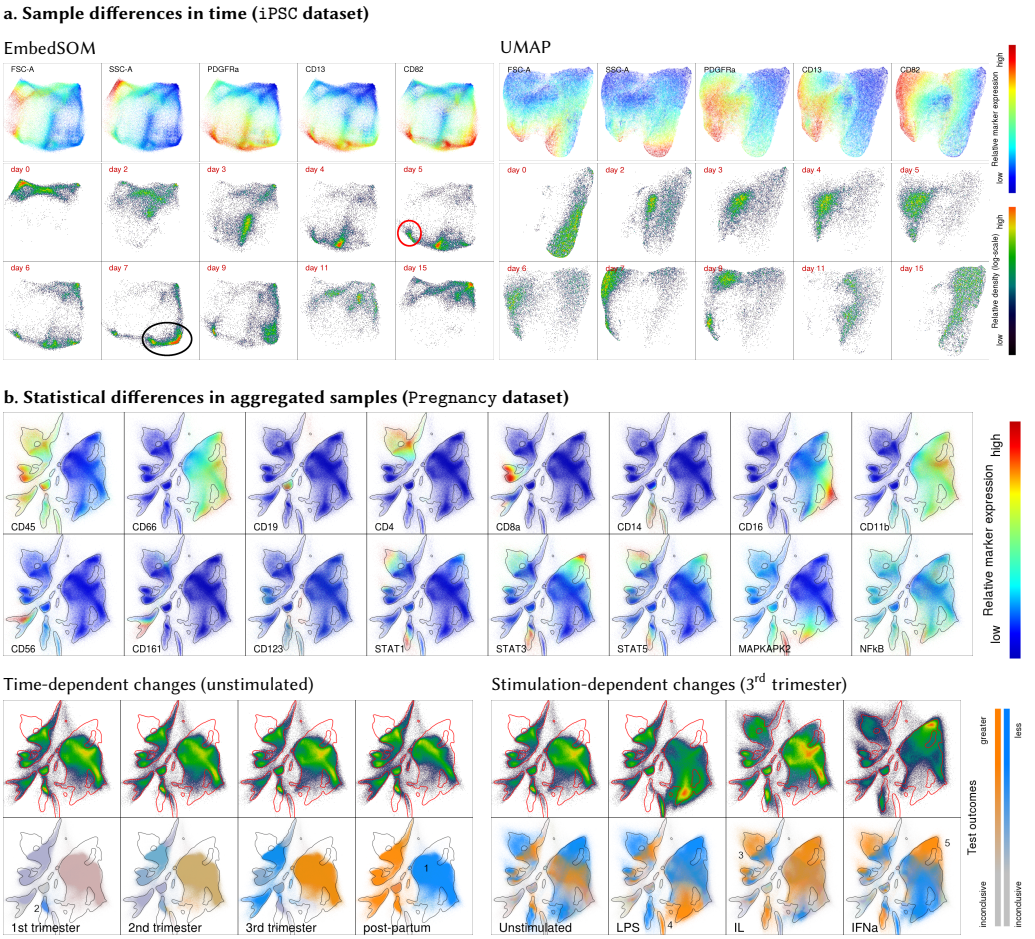

Figure 2: Examples of population alignment aiding the observation of differences in data. **a.** Reconstruction of the analysis of hiPSC differentiation to cardiomyocytes as seen on expression levels of CD13, CD82 and PDGFRα by Takeda et al. (2018, Figure 1) with EmbedSOM, compared with UMAP. Top: marker expressions; below: cell populations at different time points. The development trajectory and the CD82 peak in day 7 are clearly observable in EmbedSOM plots. **b.** Top: Selection of marker expressions from the `Pregnancy` dataset by Aghaeepour et al. (2017); contour is provided to aid population identification. Bottom left: Changes between the unstimulated population at different time points, displayed as cell densities (top) and population change significance in metaclusters (bottom). Bottom right: Same plots showing changes caused by sample stimulation in FlowSOM pre-clusters. Numeric labels are added for the declined neutrophils (1) and subpopulation of monocytes (2), activated CD4 and CD8 cells (3), MARKAPK2$^+$ monocytes and neutrophils (4) and STAT3$^+$ neutrophils (5).

2c,g) and Cytofast (Beyrend et al., 2018, Figure 2). The EmbedSOM view improves the presentation of results by directly connecting the information about differences with the population viewed in the embedding, and produces an easily comprehensible output even when the cluster count is too high to be examined as tabular data.

The resulting plots of the `Pregnancy` dataset (Figure 2b) allow straightforward observation of relative sample changes. The granularity of change highlighting is dependent on the granularity of the used clustering — for demonstration, we show the changes in large cell populations (defined by FlowSOM metaclusters) along with small populations and sub-populations (defined by FlowSOM pre-clusters). These are respectively used in the plots of time development and stimulation-induced changes to highlight various findings from the corresponding article (Aghaeepour et al., 2017). The plot of time development allows easy observation of the rapid post-partum decline of subpopulations of neutrophils, additionally it identifies a significantly lowered amount of monocytes in the samples from the first trimester. Stimulation-induced changes are clearly observable in cell populations of CD4 and CD8 T cells, which move to corresponding STAT5$^+$ regions, and in populations of neutrophils and monocytes, which, depending on the stimulation, move accordingly to MARKAPK2$^+$ or STAT3$^+$ regions.

2.5. Fast embedding improves supervised gating analysis

Embedding can be integrated with the FlowSOM workflow. EmbedSOM provides a more natural and fine-grained way to view the result of FlowSOM clustering: positions of cell populations in the embedding visually correspond to the positions of the pre-clusters in the grid view of FlowSOM (Van Gassen et al., 2015, Fig. 1 (ii) and 2). At the same time, the cells are distributed in a way that retains the topology and variance in the sample, which makes the EmbedSOM output locally similar to the view of populations in the usual dot plots.

We exploited the correspondence between FlowSOM and embedding output to connect the supervised gating process with the FlowSOM capabilities. Instead of separating cell populations with a manually drawn line, we let the user select cells in groups defined by FlowSOM metaclusters or pre-clusters. The result is both more natural to scientists who select the cells from the usual dot-plot view, and less prone to errors since human choice is restricted to a *discrete and reproducible* selection of clusters that have previously been proven to capture the respective cell populations very precisely (Weber and Robinson, 2016). After the user makes the selection, the dissection process continues hierarchically by

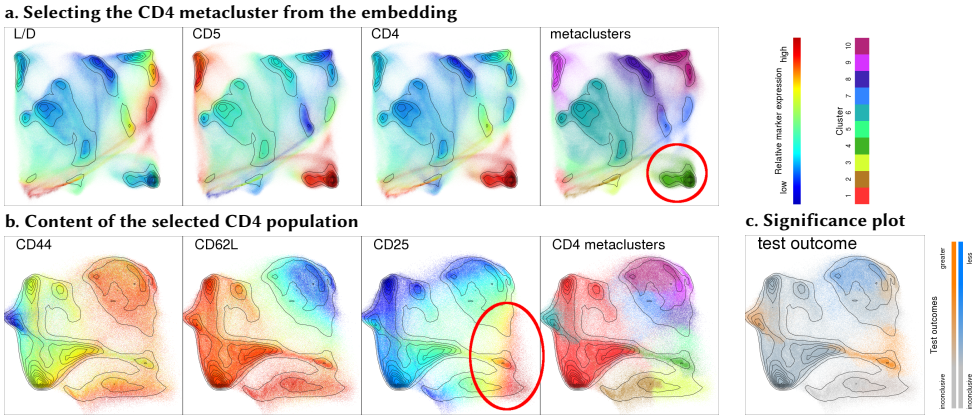

Figure 3: One level of hierarchical sample dissection augmented by EmbedSOM view, as demonstrated in Section 2.5. All marker expression plots are available in section S1. **a.** Embedding of all cells with highlighted important markers and colorized FlowSOM metaclustering. The user can choose the green metacluster of $CD5^+CD4^+$ live cells (circled in red) to investigate it more closely. **b.** Inner content of the selected CD4 metacluster; the population of Tregs is circled in red. **c.** The plot of significant differences in the population sizes between control and experimental groups clearly shows the increase in Treg cells in the experimental group (in orange).

re-embedding the selected subset of cells (i.e., 'zooming' as seen in HSNE (Unen et al., 2017)).

We demonstrated the application of this workflow on experimental screening for differences between a newly generated transgenic mouse model and the respective wild-type. All samples ($n = 14$) from a 10-dimensional dataset were aggregated and subjected to the semi-automatic workflow described in Section 4.2.4. The total runtime for all 5.5×10^6 cells was less than 3 minutes.

From the first stage of the workflow, we received an intelligible embedding of all aggregated cells in less than 2 minutes. As the demonstration experiment was primarily aimed at screening of possible alterations in the $CD4^+$ T cell compartment, we used the resulting marker expression plots to choose the corresponding FlowSOM metacluster (the choice is highlighted in Figure 3a) and re-embedded the cell sub-population. The updated embedding (seen in Figure 3b, computed in less than 15 seconds) showed the assorted CD4 T cells, mainly the effector and resting subpopulations of both Thelpers and Tregs. To simplify the screening process, the workflow then ran the bulk statistical testing of the differences in metacluster contents in wild-type vs. transgenic mice, and used the resulting p-values for coloring each metacluster in the embedding (Figure 3c) to provide the significance plot.

The final significance plot provided clear information regarding the relative differences in population size, which we used to guide a more exact analysis. In this specific case, after observing the p-values in the Treg population, we selected the appropriate metaclusters with all Tregs and repeated the hypothesis testing for the contained cells, which confirmed the hypothesis that Tregs abundance in the transgenic group is greater with $p = 0.04$. This reproduced the result obtained by manual analysis (Figure S1) in a fraction of the time.

3. Discussion

EmbedSOM alleviates the long-standing unavailability of fast, scalable non-linear embedding algorithm for single-cell cytometry data. In its current version, it extends the usefulness of commonly available hardware for running analyses and producing visualizations of high-volume datasets. Here, we draw attention to some unanswered questions and possible directions for future research.

3.1. Benchmarking methodology

Currently, there is no single accepted methodology for benchmarking the quality of embedding algorithms. In this work, we chose the benchmark measure of k-NN population label entropies and purities (see Section 4.2.3 for definition) over the more common measure of Kullback-Leiber (KL) divergence of distance distribution and various measures derived from similarity between k-neighborhoods of a datapoint in high-dimensional vs. embedding space, used e.g. by Becht et al. (2018). Alternative measures include e.g. the NPE and residual variance used by Konstorum et al. (2018).

We made this choice due to our focus on high-throughput cytometry data, for which neither the KL divergence minimization nor k-neighborhood preservation is the primary desired feature. Cells typically form relatively dense populations of a single cell type, the inner structure of which is either absent or biologically irrelevant (specifically, cells with a 0.1% difference in marker expression are usually considered the same). Therefore, algorithms should aim to separate the cells that are expected to belong to different populations, which is described by our k-NN measures, rather than attempt to preserve the irrelevant inner structure and exact global distances of such populations, which is required to produce a high similarity of k-neighborhoods and low KL divergence.

Even though the perception of 2-dimensional depiction and separation of individual populations is highly subjective, we believe that the similarity of Em-

bedSOM embeddings to the usual dot-plot projections used for manual gating will simplify interpretation of the results by scientists.

3.2. Population alignment

Although the idea of aligning populations in the embedding presented in Section 2.4 is not new, the two-stage design of EmbedSOM allows this alignment to be produced without algorithm modification or additional computational overhead. However, this works only if the same SOM is used for embedding of the subsequently added samples. Naïve re-training of the SOM is a pseudorandom process that is not guaranteed to produce the same result even on very similar data, which breaks the alignment in the case of any SOM updates. Despite this, much of the perceived non-determinism caused by SOM re-training can be removed by careful initialization, or by adding a bias that forces user-selected cell populations to favor user-selected SOM grid nodes (see Astudillo and Oommen (2014) for survey of relevant methods). Application of such techniques to cytometry workflows is a topic for future research.

3.3. Trajectories and noise

The smoothness and local linearity of the EmbedSOM projection are valuable aids in visualizing transitions between different cell populations and their states. As discussed in Section 2.3, this comes as a tradeoff — the embedding is unable to completely separate cell populations from surrounding noise and debris if these are not separable in high-dimensional space, but the same property causes it never to disrupt prolonged populations and cell development trajectories — a prominent example can be seen in the connection between $CD4^+CD8^-$ and double-positive $CD4^+CD8^+$ T cells in Figure 3a. At the same time, local linearity improves the depiction of cell densities (Figure 1c) that can be used as a guide for distinguishing populations and trajectories of interest from noise. Computational identification of such phenomena is a topic of recent research (Saelens et al., 2018; Wolf et al., 2018) that has already shown interesting results (Li et al., 2018). In the future, we aim to exploit the extra information available from EmbedSOM computation, such as the manifold projection distance, to improve the speed and precision of this process.

3.4. Augmented dissection of cell populations

The workflow demonstrated in Section 2.5 serves as a valuable alternative to the commonly used manual gating strategies. Apart from the improvement in precision and reproducibility, the augmentation avoids the need to manually

dataset	#events	dim.	labels	FlowRepository
`Levine-13`	167k	13	24	`FR-FCM-ZZPH` (Levine et al., 2015)
`Levine-32`	265k	36	14	
`Samusik-01`	86k	54	24	
`Samusik-all`	841k	54	24	
`iPSC`	234k	12	–	`FR-FCM-ZYDL` (Takeda et al., 2018)
`Pregnancy`	90.2M	55	–	`FR-FCM-ZY3Q` (Aghaeepour et al., 2017)

Table 2: Summary of the publicly available datasets used for demonstrations and benchmarking.

draw gates, which is especially convenient if applied to multiple samples at once and connected with automated analysis of their properties. For instance, the significance plots (Figure 3c) provide a compound view of data from many samples aggregated in an easy-to-inspect image of relevant statistical information, which can quickly guide the supervised analysis to the most interesting parts of the data. A similar presentation of the sample statistics has already been proposed in diffcyt (Weber et al., 2018) and successfully used for prediction of responses to immunotherapy (Krieg et al., 2018).

In the near future, we plan to release a graphical, user-friendly software for executing this workflow.

4. Experimental procedures and methods

4.1. Data and software availability

EmbedSOM is available from `https://bioinfo.uochb.cas.cz/embedsom`. Repositories with source code are hosted by GitHub.

The benchmarking data were selected from public-domain datasets that previously have been used for benchmarking other algorithms. The two datasets used for visualization of sample differences (`iPSC` and `Pregnancy`) were selected based on the availability of time-series and multi-sample data. A summary of all datasets used in this work is provided in Table 2.

The datasets were pre-processed as in the corresponding original articles. For the analysis of embedding quality, we reused the cell labels provided in the `Levine` and `Samusik` datasets. To compare the embedding against the results of unsupervised analysis, we used FlowSOM to generate the same number of metaclusters as of manually gated populations.

The benchmark of performance scaling was run on cells and markers that were sampled randomly from the `Pregnancy` dataset.

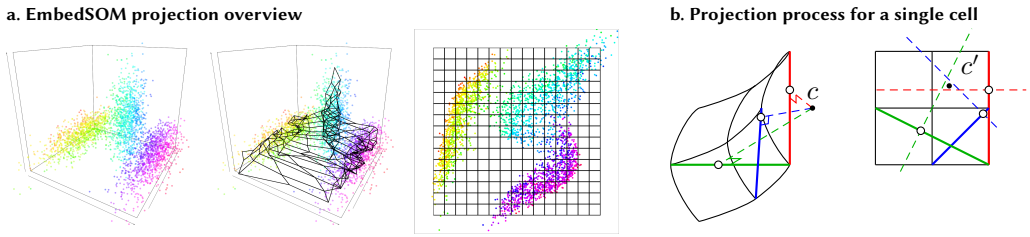

Figure 4: **a.** Overview of the two EmbedSOM stages, shown on three synthetic Gaussian clusters in 3-dimensional space: The cells in multi-dimensional space (left) are used to train a SOM that describes their distribution (middle) in the first stage; seconds stage smoothly bends the multi-dimensional space so that the contained SOM manifold becomes flat (right). **b.** The process of approximating the projection of a single cell to obtain 2-dimensional grid coordinates. Left: 3×3 SOM and the cell in the multi-dimensional space, white points represent orthogonal projections of c on three different lines g_{ij}. Right: corresponding flattened SOM and the cell projection in embedding space; c' is fitted to have minimal distance from the expected orthogonal projections to each corresponding g'_{ij}.

For demonstration of the augmented hierarchical dissection technique in Section 2.5, we used original, locally generated data from transgenic mouse spleens. A detailed description of the experiment and methods is available in section S1.

4.2. Method Details

4.2.1. Embedding algorithm

The task of the cell-projection stage of EmbedSOM is to find approximate positions of the projections of each cell onto the manifold that is implicitly defined by the SOM that was computed in the first stage of the algorithm (e.g., using FlowSOM). The projection process is separate for each cell, and depends on the set G of the positions of SOM grid vertices and several tunable parameters (described below). The following description is illustrated in Figure 4.

To embed the cell at position c, we first order the elements of G by their distance to c. Distance to the n-th closest element of G is used as σ of the Gaussian probability distribution centered at c; all elements of $g_i \in G$ are then assigned conditional probabilities p_i of the event that they would be selected from this distribution under the assumption that only the elements of G were selected (this usage of n is somewhat inspired by the perplexity parameter of tSNE). Next, a set $G \times G$ of elements g_{ij} is constructed with probabilities p_{ij} that both g_i and g_j would be selected from this distribution at the same time.

At this point, each g_{ij} corresponds to a pair of vertices of the SOM grid, which together define a line. We orthogonally project c to each such line in multidimen-

sional space, and obtain relative distances of the projected point $\text{proj}_{ij}(c)$ to each g_i and g_j, as $d_{ij} = \frac{|\text{proj}_{ij}(c) - g_i|}{|g_j - g_i|}$.

Distances d_{ij} are used to reconstruct the embedded cell position c' in 2-dimensional space. We set the embedded grid coordinates g'_i so that if g_i was on x-th row and y-th column of the grid, the position of g'_i is exactly (x, y). We define the embedded lines g'_{ij} and distances d'_{ij} accordingly for c'. Finally, we aim to position c' so that d_{ij} and d'_{ij} are as similar as possible for all i, j, which is accomplished by algebraically finding minimum of polynomial $\sum p_{ij} \|g'_i - g'_j\|^{-a} (d_{ij} - d'_{ij})^2$. Because d'_{ij} is linear in c', the solution is obtained as in the least-squares method. In the formula, the weights p_{ij} provide non-linearity, and the parameter a is used to lower the influence of non-local information on the approximation.

As an optimization, we reduce the amount of computation required for each cell from the original $\mathcal{O}(|G|^2)$ by truncating G to k elements that are nearest to c, which has negligible impact if only elements with low p_i are removed. Using this method, the total time required for embedding a set C of cells is at most $\mathcal{O}(|C| \cdot |G| \cdot k^2)$.

4.2.2. Algorithm parameter selection

The major parameters of EmbedSOM include the SOM-training settings (shared with FlowSOM) and projection parameters n, k, a. Correct setup of the SOM training has been previously discussed by Van Gassen et al. (2015), who recommend training 10×10 SOM. We recommend using a slightly larger SOM size to provide a smoother manifold for the projection approximation. Accordingly, we used 24×24 as a default throughout this work. In our experience, projection parameter values of $n \in [5, 30]$, $k \in [2n, 5n]$ and $a \in [0, 3]$ provided good results, and setting $n = 15$, $k = 40$, $a = 1$ was a good default for all data we tested. Visual differences between various parameter settings are shown in Figure S5.

The granularity of cell populations correctly embedded by EmbedSOM, unlike tSNE and UMAP, is limited by the size of the underlying SOM. In our experience, good embedding of a population requires that at least three SOM grid vertices be mapped to it. Hypothetical samples that contain 100, 1,000 and 10,000 different cell populations would thus require SOMs of at least 18×18, 55×55 and 174×174 vertices, respectively. Higher number of vertices would negatively impact EmbedSOM performance, because both algorithm stages scale linearly with the number of SOM vertices. Nevertheless, typical cytometry data do not contain such quantities of small populations, and more than 40×40 vertices (approximately $2.8\times$ slower than the recommended 24×24) was not required on any data

we tested.

4.2.3. Benchmark setup

To obtain a metric of the embedding quality, we measured how the layout of the cells in the embedding compares to the 'ground truth' classification of the cell populations. In the benchmark, the ground truth was obtained either from manual gating or unsupervised clustering. For comparison, we measured the entropy and purity of the cell population labels (i.e. cluster numbers) in the k-nearest neighborhoods of the embedded cells.

First, the data from each dataset were embedded by tSNE, UMAP and EmbedSOM, using information from all relevant markers. To compare with a linearity-preserving method, we also calculated 2-dimensional PCA projections.

We defined k-NN entropy as the standard information entropy of the population label values in a k-neighborhood, and k-NN purity as a probability that a random cell selected from a k-neighborhood belongs to the same population as the neighborhood center. These measures implicitly capture the amount of high-entropy noise and number of the misplaced cells in the embedding. Both measures were calculated for all neighborhoods of size $k = 100$ in a sample of 10,000 cells from each embedding. Individual values for the cell neighborhoods were plotted as reliability distributions to aid comparison.

The benchmarking computations were also used to collect speed measurements from all algorithms. Data were collected on an Intel® Core™ i7-4790K CPU@4.00GHz and nVidia® GeForce® GTX 1060 GPU. UMAP implementation from Python package `umap-learn` version 0.3.2 and tSNE implementation from R package `Rtsne` version 0.13 were used with default parameters: UMAP was run with `n_neigbors=15`, `min_dist=0.1`, `n_components=2` with 200 epochs and Euclidean metric. RtSNE was run with perplexity set to 30, $\theta = 0.5$, $\eta = 200$, on 1000 iterations with momentum scaling from 0.5 to 0.8. Unless noted otherwise, we used 24×24 grids to build the SOM; other EmbedSOM parameters were left at default values.

4.2.4. Use-case workflows

Embeddings with population alignment in Section 2.4 were produced by reusing the same trained SOM for multiple datasets. First, we aggregated all data to a single sample that was used to train the SOM, which was in turn used to embed the separate samples by EmbedSOM. UMAP was run on the aggregate sample, and the cells were then separated into a distinct plot for each input sample.

To generate significance plots (as seen in Figure 2b, Figure 3c), cell counts in the metaclusters in all samples (or, alternatively, FlowSOM pre-clusters) were normalized as percentages of the entire displayed population. The percentages were grouped according to the experiment (e.g. wild-type vs. transgenic samples) and both groups were subjected to two one-sided Mann-Whitney tests (R function `wilcox.test`) to test the hypotheses of lower and higher relative cell abundance in the sample groups. The resulting pairs of p-values were used for color-labeling of the corresponding clusters in the plot.

Manual data analysis used in Section 2.5 was performed using FlowJo software (Tree Star). The full gating strategy can be viewed in Figure S1. For EmbedSOM-based analysis, we first created a 20×20 SOM on a sample from all aggregated cells, embedded it to plot the marker expressions, and computed FlowSOM metaclustering with $k = 10$ to provide the clusters for user-based selection. At this point, the user examined the resulting plots and selected metacluster numbers for further exploration. The same analysis was then repeated for the selected cell subset, using a 16×16 SOM.

Acknowledgements

This work was supported by SVV project 260451. M.K. and J.V. were supported by ELIXIR CZ LM2015047 (MEYS). A.K. and K.D. were supported by the Grant Agency of the Charles University, GAUK 1610218. J.B., V.N. and R.S. were supported by RVO68378050 (CAS), LM2015040 (MEYS), OP RDE CZ.02.1.01/0.0/0.0/16_013/0001789 (MEYS and ESIF) and OP RDI CZ.1.05/1.1.00/02.0109 (MEYS and ERDF). K.F. was supported by NV18-08-00385 (AZV).

We are extremely grateful to Vladimír Vondruš for providing invaluable insight into Vulkan® API, to Alena Keprová for providing datasets for testing, and to Yvan Saeys and Sofie Van Gassen for benchmarking advice.

References

Aghaeepour, N., Ganio, E.A., Mcilwain, D., Tsai, A.S., Tingle, M., Van Gassen, S., Gaudilliere, D.K., Baca, Q., McNeil, L., Okada, R., et al., 2017. An immune clock of human pregnancy. Science immunology 2, eaan2946.

Astudillo, C.A., Oommen, B.J., 2014. Topology-oriented self-organizing maps: a survey. Pattern analysis and applications 17, 223–248.

Becht, E., McInnes, L., Healy, J., Dutertre, C.A., Kwok, I.W., Ng, L.G., Ginhoux, F., Newell, E.W., 2018. Dimensionality reduction for visualizing single-cell data using umap. Nature Biotechnology .

Belkina, A.C., Ciccolella, C.O., Anno, R., Spidlen, J., Halpert, R., Snyder-Cappione, J., 2018. Automated optimal parameters for t-distributed stochastic neighbor embedding improve visualization and allow analysis of large datasets. bioRxiv , 451690.

Beyrend, G., Stam, K., Höllt, T., Ossendorp, F., Arens, R., 2018. Cytofast: A workflow for visual and quantitative analysis of flow and mass cytometry data to discover immune signatures and correlations. Computational and Structural Biotechnology Journal 16, 435–442.

Chan, D.M., Rao, R., Huang, F., Canny, J.F., 2018. t-SNE-CUDA: GPU-accelerated t-SNE and its applications to modern data. arXiv preprint arXiv:1807.11824 To appear in HPML 2018 High Performance Machine Learning Workshop (Accepted, 2018).

Duò, A., Robinson, M.D., Soneson, C., 2018. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. F1000Research 7.

Gorban, A., Zinovyev, A.Y., et al., 2001. Visualization of data by method of elastic maps and its applications in genomics, economics and sociology .

Kimball, A.K., Oko, L.M., Bullock, B.L., Nemenoff, R.A., van Dyk, L.F., Clambey, E.T., 2018. A beginner's guide to analyzing and visualizing mass cytometry data. The Journal of Immunology 200, 3–22.

Konstorum, A., Vidal, E., Jekel, N., Laubenbacher, R., 2018. Comparative analysis of linear and nonlinear dimension reduction techniques on mass cytometry data. bioRxiv URL: https://www.biorxiv.org/content/early/2018/03/01/273862, doi:10.1101/273862, arXiv:https://www.biorxiv.org/content/early/2018/03/01/273862.full.pdf.

Krieg, C., Nowicka, M., Guglietta, S., Schindler, S., Hartmann, F.J., Weber, L.M., Dummer, R., Robinson, M.D., Levesque, M.P., Becher, B., 2018. High-dimensional single-cell analysis predicts response to anti-PD-1 immunotherapy. Nature medicine 24, 144.

Levine, J.H., Simonds, E.F., Bendall, S.C., Davis, K.L., ad D. Amir, E., Tadmor, M.D., Litvin, O., Fienberg, H.G., Jager, A., Zunder, E.R., Finck, R., Gedman, A.L., Radtke, I., Downing, J.R., Pe'er, D., Nolan, G.P., 2015. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell 162, 184–197. URL: http://dx.doi.org/10.1016/j.cell.2015.05.047, doi:10.1016/j.cell.2015.05.047.

Li, N., van Unen, V., Höllt, T., Thompson, A., van Bergen, J., Pezzotti, N., Eisemann, E., Vilanova, A., de Sousa Lopes, S.M.C., Lelieveldt, B.P., et al., 2018. Mass cytometry reveals innate lymphoid cell differentiation pathways in the human fetal intestine. Journal of Experimental Medicine 215, 1383–1396.

Linderman, G.C., Rachh, M., Hoskins, J.G., Steinerberger, S., Kluger, Y., 2017. Efficient algorithms for t-distributed stochastic neighborhood embedding. arXiv preprint arXiv:1712.09005 .

McInnes, L., Healy, J., 2018. UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 .

Pezzotti, N., Höllt, T., Lelieveldt, B., Eisemann, E., Vilanova, A., 2016. Hierarchical stochastic neighbor embedding, in: Computer Graphics Forum, Wiley Online Library. pp. 21–30.

Pezzotti, N., Lelieveldt, B.P., van der Maaten, L., Höllt, T., Eisemann, E., Vilanova, A., 2017. Approximated and user steerable tSNE for progressive visual analytics. IEEE transactions on visualization and computer graphics 23, 1739–1752.

Saelens, W., Cannoodt, R., Todorov, H., Saeys, Y., 2018. A comparison of single-cell trajectory inference methods: towards more accurate and robust tools. bioRxiv URL: https://www.biorxiv.org/content/early/2018/03/05/276907, doi:10.1101/276907, arXiv:https://www.biorxiv.org/content/early/2018/03/05/276907.full.pdf.

Saeys, Y., Van Gassen, S., Lambrecht, B.N., 2016. Computational flow cytometry: helping to make sense of high-dimensional immunology data. Nature Reviews Immunology 16, 449.

Takeda, M., Kanki, Y., Masumoto, H., Funakoshi, S., Hatani, T., Fukushima, H., Izumi-Taguchi, A., Matsui, Y., Shimamura, T., Yoshida, Y., et al., 2018. Identification of cardiomyocyte-fated progenitors from human-induced pluripotent stem cells marked with CD82. Cell reports 22, 546–556.

Tino, P., Nabney, I., 2002. Hierarchical GTM: Constructing localized nonlinear projection manifolds in a principled way. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 639–656.

Unen, V., Höllt, T., Pezzotti, N., Li, N., Reinders, M.J., Eisemann, E., Koning, F., Vilanova, A., Lelieveldt, B.P., 2017. Visual analysis of mass cytometry data by hierarchical stochastic neighbour embedding reveals rare cell types. Nature communications 8, 1740.

Van Der Maaten, L., 2014. Accelerating t-SNE using tree-based algorithms. The Journal of Machine Learning Research 15, 3221–3245.

Van Gassen, S., Callebaut, B., Van Helden, M.J., Lambrecht, B.N., Demeester, P., Dhaene, T., Saeys, Y., 2015. FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry Part A 87, 636–645.

Weber, L.M., Nowicka, M., Soneson, C., Robinson, M.D., 2018. diffcyt: Differential discovery in high-dimensional cytometry via high-resolution clustering. bioRxiv URL: https://www.biorxiv.org/content/early/2018/06/18/349738, doi:10.1101/349738, arXiv:https://www.biorxiv.org/content/early/2018/06/18/349738.full.pdf.

Weber, L.M., Robinson, M.D., 2016. Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. Cytometry Part A 89, 1084–1096.

Wolf, F.A., Hamey, F., Plass, M., Solana, J., Dahlin, J.S., Gottgens, B., Rajewsky, N., Simon, L., Theis, F.J., 2018. Graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. bioRxiv URL: https://www.biorxiv.org/content/early/2018/11/04/208819, doi:10.1101/208819, arXiv:https://www.biorxiv.org/content/early/2018/11/04/208819.full.pdf.

Xiao, Y., Feng, R.B., Han, Z.F., Leung, C.S., 2015. GPU accelerated self-organizing map for high dimensional data. Neural Processing Letters 41, 341–355.