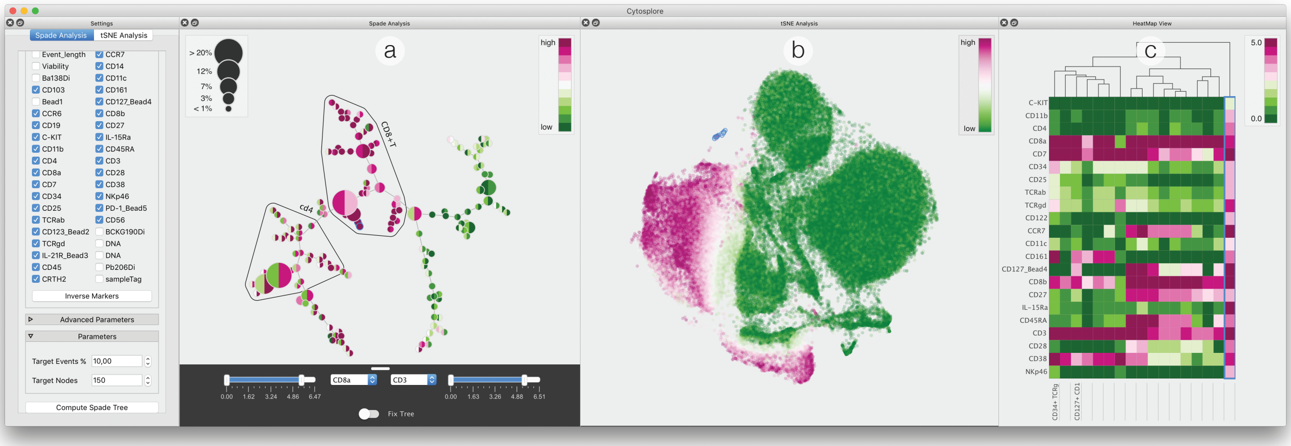
Eurographics Conference on Visualization (EuroVis) 2016
K.-L. Ma, G. Santucci, and J. van Wijk
(Guest Editors)

Volume 35 (2016), Number 3

Cytosplore: Interactive Immune Cell Phenotyping for Large Single-Cell Datasets

T. Höllt[1], N. Pezzotti[1], V. van Unen[2], F. Koning[2], E. Eisemann[1], B. Lelieveldt[1,2], and A. Vilanova[1]

[1]TU Delft, Delft, The Netherlands
[2]Leiden University Medical Center, Leiden, The Netherlands



Settings

Spade Analysis | tSNE Analysis

□ Event_length	☑ CCR7
□ Viability	☑ CD14
□ Ba138Di	☑ CD11c
☑ CD103	☑ CD161
□ Bead1	☑ CD127_Bead4
☑ CCR6	☑ CD8b
☑ CD19	☑ CD27
☑ C-KIT	☑ IL-15Ra
☑ CD11b	☑ CD45RA
☑ CD4	☑ CD3
☑ CD8a	☑ CD28
☑ CD7	☑ CD38
☑ CD34	☑ NKp46
☑ CD25	☑ PD-1_Bead5
☑ TCRab	☑ CD56
☑ CD123_Bead2	□ BCKG190Di
☑ TCRgd	□ DNA
☑ IL-21R_Bead3	□ DNA
☑ CD45	□ Pb206Di
☑ CRTH2	□ sampleTag

Inverse Markers

▷ Advanced Parameters

▽ Parameters

Target Events % 10,00

Target Nodes 150

Compute Spade Tree

Figure 1: **Cytosplore.** Screenshot of our system with four widgets (adaptive settings, overview (a), embedding (b) and heatmap (c)), representing the workflow. Views can be rearranged or additional views of these types added.

Abstract

To understand how the immune system works, one needs to have a clear picture of its cellular compositon and the cells' corresponding properties and functionality. Mass cytometry is a novel technique to determine the properties of single-cells with unprecedented detail. This amount of detail allows for much finer differentiation but also comes at the cost of more complex analysis. In this work, we present Cytosplore, implementing an interactive workflow to analyze mass cytometry data in an integrated system, providing multiple linked views, showing different levels of detail and enabling the rapid definition of known and unknown cell types. Cytosplore handles millions of cells, each represented as a high-dimensional data point, facilitates hypothesis generation and confirmation, and provides a significant speed up of the current workflow. We show the effectiveness of Cytosplore in a case study evaluation.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

The immune system primarily protects our body against bacterial, viral and parasitic infections. However, it may respond to harmless self antigens, leading to auto-immune diseases, e.g., type 1 diabetes or rheumatoid arthritis. Detailed knowledge of the immune system's functioning is required to understand the cause of immune-mediated diseases, which is an important step towards preventive or therapeutic measures. To mediate its function, the immune system utilizes both; humoral (soluble) and cellular constituents. The cellular immune compartment consists of a variety of cellular subsets, each with a distinct function and associated *phenotype*. The phenotype describes "*the observable physical or biochemical char-*

acteristics of an organism, as determined by both genetic makeup and environmental influences" [AHD06]. In the last decades a large number of phenotypically and functionally distinct subsets have been defined and, for some, a major role in disease processes has been found. For immune cells, the functionality mostly relates to a set of proteins expressed on the cells surface.

Recently introduced *mass cytometry* [OKB*08] at the moment allows the observation of 36 of these proteins at the same time, three times as many as the clinical standard. However, this number is still orders of magnitude smaller than the estimated 10,000 immune-system-wide available proteins, providing phenotypic information. Hence, specific panels of markers, corresponding to proteins of interest, need to be designed for each study. The composition of these panels if often unique to a study and it is not known beforehand, which combinations of proteins can be expected. Therefore, the identification of different phenotypes largely needs to be carried out in a data-driven fashion by studying data heterogeneity rather than applying prior knowledge.

The fine granularity of mass cytometry is usually not only used to increase detail but also to increase breadth, i.e., markers for different *cell lineages* can be tested simultaneously. A cell lineage describes a group of subsets, all derived from the same ancestry and sharing certain characteristics. Consequently, the data inherently provides multi-scale information; major lineages form clusters on a high-level scale, while lower-level scale clusters correspond to phenotypical subsets.

To ensure comparability of measurements of multiple blood or tissue samples the same marker panel needs to be applied. In addition, different batches of the same marker can produce different results. Therefore, experiments are usually run in large cohort studies, resulting in hundreds of samples containing millions of cells. These large sizes pose significant challenges during the analysis process.

We worked closely with immunohaematology experts to design a data-driven workflow for phenotype specification of cytometry data that we present in this paper. We are the first to specifically tackle the multi-scale properties of the data. To this extent, we combine and link two proven techniques for the analysis of single-cell data on different levels of detail. For both steps, we provide in-place and linked visualizations of the feature space to interact with and refine the automatically-generated classifications.

The major contributions of this paper are:

- Cytosplore: an integrated system to interactively explore large high-dimensional single-cell datasets and identify phenotypically distinct subsets in a data-driven fashion.
- An analysis workflow, supporting linking of multiple levels of detail to enable
  - rapid, data-driven phenotype specification (including for unknown cell types)
  - the discovery, pinpointing and fixing of mistakes over multiple levels of detail

2. Biological Background

To analyze heterogeneity of immune cell subsets, multiparameter analysis of immune cells at single-cell level is required. Flow cytometry has been the method of choice for this purpose, however, suffers from a limitation; it is restricted by the number of cellular markers that can be simultaneously analyzed, usually 10 to 12. Therefore studies employing flow cytometry are usually focused on very specific, known cell types. This limitation has been overcome by the introduction of mass cytometry.

Mass cytometry is a novel, *mass* spectrometry-based, technique for characterizing protein expression on cells (*cytometry*) at single-cell resolution. In short, antibodies, selected to bind to specific proteins of interest on the cell membrane, are conjugated with heavy-metal reporters. After staining, the cells are vaporized, atomized and ionized one by one and the remaining metals in the ion cloud can be measured in a mass spectrometer to quantify the selected proteins on a per-cell basis. Mass cytometry currently allows the simultaneous analysis of 36 markers, a number which is expected to rise to 100 in the near future. This allows much broader studies, for example to compare different diseases. Furthermore it allows the inclusion of markers that usually would not be expected in a certain group, possibly allowing the discovery of unknown cell types.

2.1. Data

Our partners use a prototypical non-integrated version of the workflow presented in this paper in a real world study of tissue- and disease-associated signatures of the human mucosal immune system [vULM*16]. They acquired a cohort data set consisting of 102 samples from 44 donors. During preprocessing, the acquired dataset was filtered for live cells, with a strong expression of the CD45 marker (indicating immune cells), resulting in 5.2 million high-dimensional data points. 32 markers were selected for the study to provide information regarding six expected major lineages.

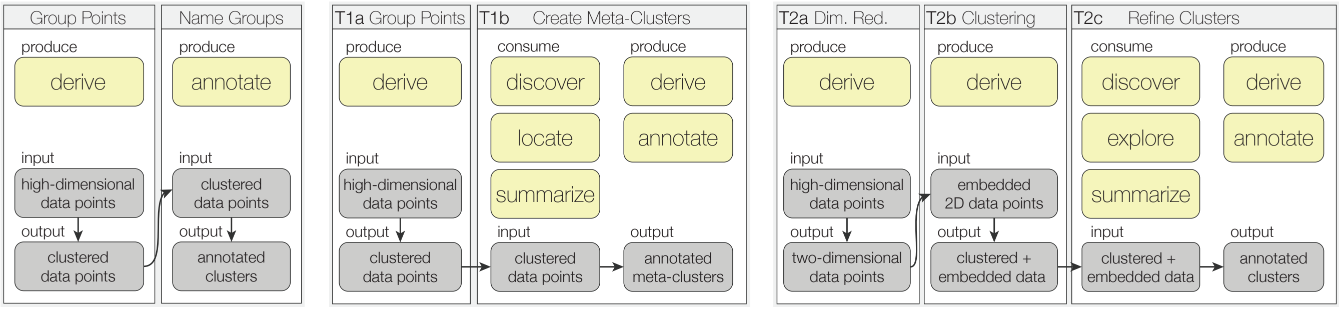
The resulting data is a table of cells and their expression profiles over all available markers. Each row in the table corresponds to a single cell and can be interpreted as a single high-dimensional data point. In abstract terms our input data consists of a large number of high-dimensional data points forming clusters on multiple scales (see Section 1).

2.2. Tasks

In this work we aim to tackle the first step of the data analysis process, namely the definition of the phenotype of every cell. In this process our collaborators need to

- Group similar cells, where similarity is defined based on the protein expression for each cell.
- Define for each group the type of cell, which can be unknown beforehand, and annotate the cells.

We provide an abstraction of these tasks, following Brehmer and Munzner's multi-level task typology [BM13] in Figure 2a and make use of their adaptions for the visualization of high-dimensional data [BSIM14]. We use a `monospaced font` throughout the paper, when we use their typology.



a) High Level Task

b) Major Lineage Delineation

c) Phenotypical Subset Exploration and Identification

Figure 2: **Abstraction** of the identified high-level tasks (a), consisting of grouping and naming points, as well as the detailed subtasks. b shows the major lineage delineation (high-level clustering) and c the phenotypical subset exploration and identification (low-level clustering).

3. Related Work

Recent years brought many computer-aided solutions for cytometry-data analysis. SPADE [QSB*11] visualizes high-dimensional data and was developed for (and is commonly used in) single-cell analysis [BSQ*11, BZF*12, LZN*15]. It clusters data in the high-dimensional space and then builds a minimum spanning tree. Flow MAP [ZLG*15] follows SPADE, but replaces the spanning tree by a *k*-nearest neighbor graph, which is laid out via a force-directed layout. The approach avoids SPADE's problem of placing similar nodes far apart, but creates visual clutter. Scaffold Maps [SGF*15] enable the user to drive the layout by defining landmarks of cell-type prototypes and by placing them in the visual space to build a scaffold in which similar clusters will be placed.

viSNE [ADT*13] introduces t-Distributed Stochastic Neighbor Embedding (tSNE) [vdMH08] to mass cytometry data and ACCENSE [SBDC14] uses tSNE as the basis for automatic clustering. Classification in viSNE is performed by manually gating on the scatterplot, while ACCENSE performs automatic clustering of the embedded data.

The tSNE-based techniques perform exceptionally well in embedding cytometry data and provide single-cell resolution. Nonetheless, due to a large computational cost, only limited interactivity is reached. In fact viSNE and ACCENSE both propose downsampling of large data for increased speed. Recently, Pezzotti et al. [PLvdM*16] introduced A-tSNE, a tSNE variant, which aims at minimizing precomputation times for high-dimensional neighborhoods. While the cluster-based techniques are reasonably fast, they do not allow inspection on a single-cell level, and overall do not retain the high-dimensional structure as well as tSNE.

A standard system for single-cell data analysis is the web-based service Cytobank [KKI10]. It offers SPADE and tSNE computations in a reasonably-easy way. However, it lacks integration and interactivity. As computations are queue-based, significant wait times of several hours can occur.

A multitude of visual analysis tools for *omics*-data have been proposed recently. The focus of the vast majority of these tools is on genomic data. Generally, these data are similar in structure, e.g., a cell can be represented by a high-dimensional expression vector. However, usually the goal of the analysis of these data are quite different. StratomeX [LSS*12] allows exploration of genomics data for cancer subtype characterization. They allow comparison of multiple groups using a ribbon-based visualization. The presented case study data consists of a few thousand data points, consisting of up to 6,000 genes (dimensions), each. MizBee [MMP09] is targeted at the exploration of syntenic blocks, blocks of features that appear in the same form on the same or multiple chromosomes. While the data only consists of dozens of chromosomes, the number of features reaches hundreds of thousands. invis [DHHH13] allows exploration of RNA sequences. Among others, the authors use dimensionality reduction, by means of PCA, and two-dimensional scatterplots to visualize the data. The presented data consist of 19,000 sequences with 186 dimensions. MulteeSum [MMDP10] is a tool for the visual analysis of gene expression data in cells, with the addition of spatial and temporal information. Here, a typical dataset consists of thousands of cells with 50 dimensions over 6 time points. For all these tools it becomes apparent that besides different analysis questions, the data differs in key properties, compared to cytometry data; instead of millions of data points a typical genomics dataset only consists of thousands of data points, but sometimes with thousands of dimensions.

4. Multilevel Phenotype Specification Workflow

We introduce a high-level task description in Section 2.2. In short, we need to `derive` groups of similar high-dimensional data points and `annotate` these groups. In Section 3, we present a number of tools that are available and commonly used for these tasks in single-cell analysis. However, none of these tools performs optimally on large cohort studies (Section 2.1) consisting of millions of cells. The de facto standard in terms of quality is a combination of tSNE [vdMH08] (i.e., viSNE [ADT*13]) with manual or automatic clustering in the embedding [SBDC14]. However, the computational complexity severely limits the applicability of tSNE for large data. Other tools, like SPADE [QSB*11] work with larger data but do not produce cluster separation of the same quality.

In this work, we propose a multilevel workflow that effectively reduces these problems; we use SPADE clustering to create a high-level partitioning of the data, coupled with a detail analysis of each partition via A-tSNE, reducing the input size of each embedding and making it feasible. The partitioning is a means to deal with large data sizes but also has a biological justification. The amount of markers in mass cytometry enables the design of marker panels covering multiple cell lineages at the same time. In this case, the expression of markers strongly vary between lineages, but are more subtle within a lineage. Using the increased number of markers to create breadth inherently creates multiple scales within the data, which we separate in our multilevel workflow.

In the following, we present an abstraction of the two levels of this workflow, following Brehmer and Munzner's multi-level task typology [BM13]. Similar to their extension for the visualization of high-dimensional data [BSIM14], we focus on the `why` and `what` in this section. We describe the `how` in Section 5.

4.1. Major Lineage Delineation

A major lineage of cells corresponds to a high-level cluster in the data (see Section 2.1). For the details of the biological background we refer the interested reader to a special issue of Immunological Reviews [Rot10]. While we do expect tens to hundreds of different cell types, the number of major lineages is limited. Since the marker panel is designed specifically to cover a set of lineages of interest, their number, as well as their discerning markers, are known beforehand. However, the boundaries between the clusters are not fixed and the discerning markers are not always unique for a single lineage. Therefore, we propose an interactive approach to defining the high-level clustering.

We present an abstraction of the major lineage delineation in Figure 2b. We propose a two step approach. In **T1a** we group points, `deriving` a set of clusters in the high-dimensional space. Even though we do know the number of expected high-level clusters, we propose to create more clusters here and combine them to high-level meta-clusters in **T1b**, to find the optimal boundaries. For **T1b**, we propose an interactive approach; since the target is known (based on the discerning markers) the user needs to `locate` the corresponding groups of clusters, `summarize` them to `derive` meta-clusters, and finally `annotate` those meta-clusters.

In summary, we need to provide the user with effective tools and visual encodings to:

- `derive` a predefined number of clusters, while preserving high-dimensional structures.
- `locate`, `summarize` and `derive` major lineages by their discerning markers using prior knowledge.

4.2. Phenotypical Subset Exploration and Identification

Exploration and identification of phenotypically-distinct subsets happens in the second step of our workflow. We define a phenotypically-distinct subset as a group of cells with similar marker expression profiles. The subsets can greatly vary in size, in fact small subsets, corresponding to rare cells, are often of major interest and must not be lost during the analysis. Since the high-dimensional space, corresponding to the marker panel, varies from study to study, subsets need to be created in a data-driven fashion. Other than with the discerning markers in the lineage delineation, here, all markers can be of interest. We also expect to find subsets not known before requiring an explorative analysis.

We propose an approach consisting of three steps as presented in abstract form in Figure 2c. We use dimensionality reduction in **T2a** to `derive` two-dimensional data points for visual inspection of the complete data, without clustering or downsampling. This assures that small subsets do not get lost in a larger cluster or during downsampling. For creating the subsets (**T2b**), we propose to `derive` clusters based on the structure of the dimensionality reduced data. Finally, for **T2c**, we propose to re-introduce the original high-dimensional data to `explore` and verify the clusters. If the clustering is too coarse, the user can go back to the previous step and `derive` a new set of clusters. If the clustering is too fine, she can `derive` new clusters in this step by merging. Once the user is satisfied with the clustering she can `annotate` the clusters based on the complete expression profile.

To recapitulate; the proposed system needs to provide effective means to:

- `derive` two-dimensional coordinates, based on the high-dimensional expression.
- `derive` clusters, based on the two-dimensional structure.
- `explore` and `summarize` the data at single-cell resolution and `derive` subsets with similar marker expression.

5. Cytosplore

We implemented Cytosplore, a complete system for our workflow respecting the identified tasks (Figure 1). Cytosplore provides a configurable environment with multiple linked views for the analysis. Here, we describe the implementation details, reasoning, and `how` we map the different workflow tasks presented in Section 4 to the actual visualization and analysis tools. Figure 3 shows the complete workflow, as implemented.

5.1. Major Lineage Delineation

Figure 2b shows a the abstraction of the major lineage delineation. We identified two major tasks, described in Section 4.1: **T1a**: grouping of points to clusters of similar expression and **T1b**: the creation of meta-clusters, clusters of clusters, that correspond to the major lineages. In the following, we describe `how` we support these tasks in our visual analysis tool.

T1a: Group Points. We use SPADE [QSB*11] for automatically grouping points to clusters of similar expression. In short, SPADE clusters data points based on their similarity in the high-dimensional space. It does so by downsampling the data, based on local densities, to avoid removing small distinctive groups. The downsampled data is then clustered and the data points, removed during downsampling, are added to the most similar cluster. The number of clusters needs to be predefined and should be set about an order of magnitude larger than the expected lineages to compensate for SPADE's lack of precision. Finally, a minimum spanning tree is constructed using the clusters' median expressions.

We chose SPADE, as it is well known in the domain and has been proven to be a valuable tool for single-cell analysis [BSQ*11, BZF*12, LZN*15]. Its lack of precision and the need to predefine the number of clusters are not an issue for the major lineage

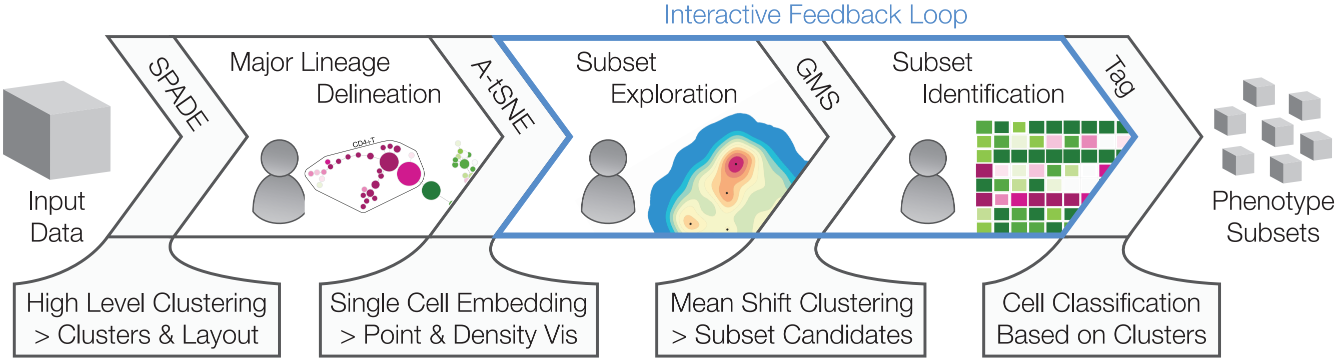
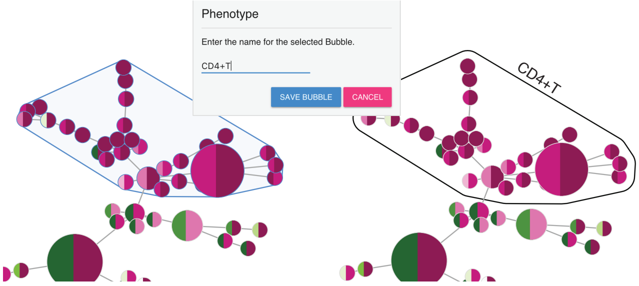

Interactive Feedback Loop
Input Data
SPADE
Major Lineage Delineation
CD4+T
A-tSNE
Subset Exploration
GMS
Subset Identification
Tag
Phenotype Subsets
High Level Clustering > Clusters & Layout
Single Cell Embedding > Point & Density Vis
Mean Shift Clustering > Subset Candidates
Cell Classification Based on Clusters

Figure 3: **Phenotype Specification Workflow** and its three major user-facing blocks; major-lineage delineation, subset exploration and identification. *SPADE*, *A-tSNE*, *GMS* and *Tag*-labeled blocks form the computational glue between user-driven parts. GMS requires a kernel-bandwidth definition, but is computed in real time, merging subset exploration and identification.

delineation. Here, we are only interested in high-level structures and, in case points are mis-classified, these can be fixed later in the pipeline. The number of major lineages expected in the data is inherently defined by the design of the marker panel and as such known beforehand. Therefore, the fact that SPADE requires the definition of the number of clusters beforehand does not pose a problem. To minimize the risk of clusters containing data points that belong to multiple lineages, the user simply selects a much larger number of clusters than expected as major lineages. These clusters are then grouped manually into *meta-clusters*, defining the major lineages.

T1b: Create Meta-Clusters. We visualize the SPADE tree using a node link diagram, where nodes correspond to the clusters and the links to the edges in the minimum spanning tree. The nodes are initially laid out using a force-directed layout but the user can `arrange` the layout as needed. Our partners are familiar with these types of diagrams and used them before to inspect the results of SPADE clustering, hence, we decided not to change this basic encoding of the data and rather focused on optimizing it for the task at hand.

The experts need to `locate` branches of the tree with a similar expression in a few markers (usually no more than three), corresponding to the known major lineages. To help the user `navigate` to and `select` these branches, we color code the nodes to show the median expression of one or more markers of the corresponding cluster. To show two or three different markers, we divide the node into segments of equal size. By default, we use the pink-to-green diverging color map from colorbrewer, as the expression is usually classified in low or high values, which here correspond to the ends of the diverging spectrum.


Phenotype

Enter the name for the selected Bubble.

CD4+T|

Figure 4: **SPADE Detail.** Meta-clusters can be selected by brushing (left) and annotated (dialog-box and right).

Once the user has identified a group of clusters with similar expression in the selected markers, she can simply brush in the diagram to `select` and `annotate` the selection via the context menu. A permanent meta-cluster is automatically `derived` from the annotated selection (Figure 4).

The described steps are usually sufficient to define the major lineages. In case the user wants to inspect the complete expression of a cluster, we provide a circular heatmap that opens around the node of interest by double-clicking.

5.2. Phenotypical Subset Exploration and Identification

We show the abstraction for the phenotypical subset exploration and identification in Figure 2c. The process is divided into three major parts, as presented in Section 4.2; **T2a**: dimensionality reduction, **T2b**: clustering and **T2c**: cluster refinement, as described below.

T2a: Dimensionality Reduction. Sedlmair et al. [SMT13] conclude that "*there is no one-and-only Dimensionality Reduction solution*". A-tSNE [PLvdM*16] is a variant of tSNE [vdMH08], which is designed to preserve local structure (i.e., clusters) in the high-dimensional space and is optimized to target two- or three-dimensional spaces for visualization [vdM09] and, therefore, fits our task very well. However, standard tSNE suffers from long computation times. We aim at fast computation of the detail visualization, as it will allow us to go back and forth between the high-level and detail visualizations to iron out mistakes in the high level selection. Therefore, we chose A-tSNE to `derive` two-dimensional data points. A-tSNE is specifically designed for such interactive settings. By approximating the high-dimensional neighborhoods

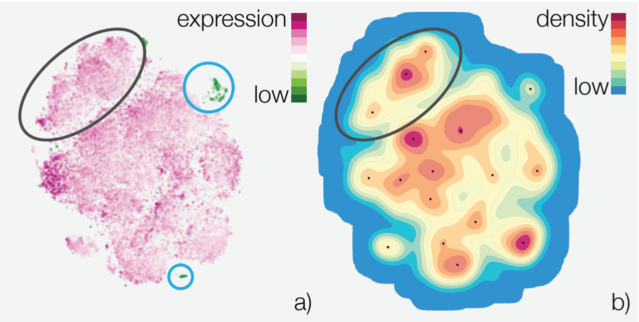


Figure 5: **tSNE Visualization** of a single lineage, as scatterplot (a) and as density plot (b). Erroneous selections can be identified in the scatterplot (blue circles) due to the low expression in the discerning marker for this lineage. Visual clusters can easily be distinguished in the density plot.

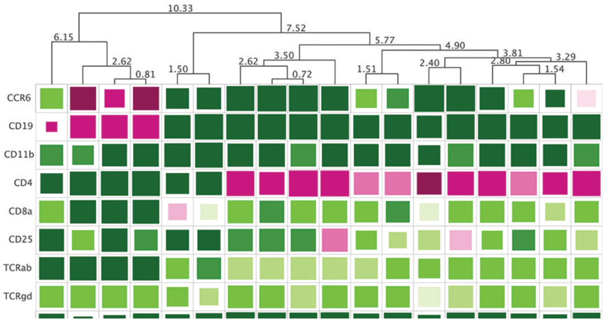


Figure 6: **Detail of the Heatmap View** showing marker expressions and variation. Variation is encoded in the amount of paint in each box. Columns are ordered by similarity as indicated by the dendrogram on top.

the startup time can be reduced by up to two orders of magnitude, when compared to the original implementation of tSNE. We use a conservative approximation parameterization, as described by Pezzotti et al. [PLvdM*16] to make sure that the resulting embedding faithfully represents the data without user interaction.

T2b: Clustering. Manual selection of visual clusters in the embedding to derive subsets is a tedious task. Previous work proposes to use automatic clustering of the embedding to specify the phenotypical subsets. In their work on ACCENSE [SBDC14], Shekhar et al. propose a technique for density-based clustering of tSNE maps in the context of cytometry. However, ACCENSE suffers from several problems. Most importantly, they use a proprietary clustering algorithm that typically clusters only around 50% of the data.

We decided to use Gaussian Mean Shift (GMS) clustering to create the subsets. GMS has proven to be a reliable tool for the analysis of complex data, is capable of creating arbitrarily-shaped clusters [CM02], will cluster all available data, and corresponds well with the visually-identified clusters. Similar to ACCENSE, GMS does rely on density computations and a kernel bandwidth needs to be specified. ACCENSE tries to find an optimal size automatically by inspecting the number of resulting peaks for a range of different values. In our tests, the results of this approach were questionable. Instead, we expose this parameter to the user, in combination with a linked feature-space view of the resulting clusters. Hereby we allow the user to make an informed decision on the kernel bandwidth. For an effective visual exploration, the data needs to be clustered at interactive rates. GMS is a rather complex algorithm and is therefore usually not employed in interactive settings. In Section 6.1, we describe a GPU-based, discrete GMS implementation that allows for interactive clustering of hundreds of thousands of data points.

T2c: Cluster Refinement. We support the user in the process of exploring the created clusters and deriving new clusters with three visual encodings. We use a scatterplot (Figure 5a) or a density plot (Figure 5b) to show the dimensionality-reduced data.

In the scatterplot (Figure 5a), subsets can be identified best by inspecting the actual marker expressions. Therefore, we use color coding to represent a single user-defined marker, using the same diverging colormap as described in Section 5.1. E.g., the user selects a discerning marker for the defined lineage from a dropdown menu to use for the color coding. Cells that show a high expression of the marker when low is required (or vice versa) can easily be identified in the scatterplot (see the blue circles in Figure 5a). The user can then go back and remove them from the defined lineage using the SPADE visualization, or simply handle them as outliers and create the correct annotation in the following steps. The density plot (Figure 5b) shows more detail within the groups. E.g., the group in the top left of the embedding (black highlight) seems relatively homogeneous in the scatterplot but shows three peaks in the density plot. However, in the density plot, we lose single-cell resolution and the marker expression. We couple the GMS clustering to the density plot and each cluster is represented by a black dot on the corresponding density peak for easy discovery.

The third visual encoding is a heatmap view (Figure 6), showing the median marker expression of the created clusters. A phenotypically-distinct subset is defined by a homogeneous unique marker expression of the contained cells. Consequently, we propose to use the homogeneity of the resulting clusters as a quality measure. We provide the standard deviation as a measure for the homogeneity. Inspired by Gove and Herzog's work [GH13], we encode the standard deviation in the amount of paint in each box in the heatmap. Here, a filled box means little standard deviation, whereas a box with a lot of white corresponds to large heterogeneity inside the cluster for the corresponding marker. The combination of the interactive clustering and the linked heatmap view, including information on the homogeneity of clusters allows the user to make an informed decision on when the automatic clustering is satisfactory.

Once the user has defined a suitable kernel bandwidth, she proceeds to refine the created clusters, i.e., by merging clusters with a similar expression. We provide quick interactions (directly in the heatmap view) to merge multiple clusters that belong to the same phenotypical subset. The user can select one or more clusters by clicking on the corresponding column in the heatmap. The cluster will be highlighted in the heatmap view and the embedding to indicate the correspondence to the spatial location. We provide different ways to `arrange` the heatmap for easy comparison. To organize columns by their overall similarity, we compute hierarchical clustering using the median cluster expressions and visualize the columns as leaves of the resulting dendrogram. Thus, similar columns are automatically placed next to each other, allowing fast `exploration` of the clusters and the corresponding feature space. In addition, the user can also sort the columns based on the values of a single row. To `derive` new clusters, the user can simply select multiple columns and merge them to a single cluster via the context menu. The dendrogram and sorting are automatically updated on such interaction. Finally, the refined clusters can be `annotated` directly in the heatmap and exported to separate files for further inspection and quantitative analysis in external tools.

6. Implementation

We implemented the core system of Cytosplore using C++ and Qt. For the visualization components, we use a combination of different rendering techniques, including D3 [BOH11] and hardware accelerated OpenGL [SSKLK13] with custom GLSL shaders [RLKG*09], depending on the amount of objects on screen. Even though we mix and match hardware-accelerated OpenGL-based visualization with slower web-based techniques, we would like to note that we strictly divide between pure visualization and intensive computational tasks. All heavy lifting, such as clustering, gradient descent and computation for A-tSNE is implemented in C++ or, if possible, on the GPU for maximum performance. When applicable, we only use a thin web layer for visualization.

6.1. GPU-based, Discrete Mean-Shift Clustering

One of the main drawbacks of the mean-shift algorithm is its computational complexity, making it not applicable in interactive scenarios with millions of data points. Therefore, we implemented a grid-based streaming version of the Gaussian Mean Shift algorithm based on work by Sirotkovic et al. [SDP13] for image segmentation. Instead of using the Improved Fast Gauss Transform [YDGD03], however, we use fast density estimation on the GPU [LH11] reducing the shift operation to a single lookup in a gradient table.

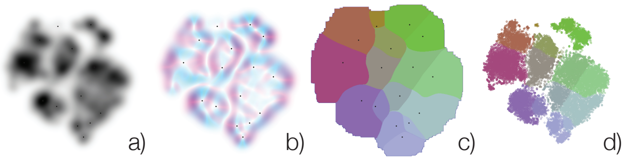


Figure 7: **GPU Mean-Shift Steps.** a shows the density map, with increasing density from white to black. b shows the corresponding (absolute) gradients, using the m and c channels of the cmyk color space to indicate the x and y components of the gradient vectors, respectively. c shows the final segmentation using unique colors for each partition. d shows the clustered points using the same coloring as in c.

In general, the mean-shift algorithm is a mode-seeking algorithm, taking each input data point and iteratively shifting it to the average of the data points in its neighborhood until convergence to a fixed location. To increase the performance, we map the clustering problem to a segmentation problem of the visual space used for the embedding, to be able to apply the algorithm presented by Sirotkovic et al. [SDP13]. As a result, the cost of the shift operation is dependent on the resolution of the visual space, rather than the number of input points. Additionally this approach maps nicely to the GPU, further increasing performance.

We use three render passes to compute the segmentation of the visual space. In the first pass, we compute the density profile (Figure 7a) in image space [LH11]. Based on the density, we compute the first derivative via central differences, resulting in the gradient at each grid position in the second render pass (Figure 7b). In the third pass, we follow the gradient map upwards until we find a local peak for each pixel with a non-zero density. We inscribe the found position as a color to the starting pixel, resulting in a map of constant colored partitions (Figure 7c). Finally, on the CPU, we set a unique id for each of these partitions. Assigning this id to each data point is then a simple look up in the resulting map using the point's position. Figure 7d shows the final clustered points.

Performance. Figure 8 shows computation times of the GPU mean-shift algorithm for different numbers of points, different grid sizes, and different kernel sizes from 10% to 40% of the image size. The computations were carried out using a 4 core intel core i7 processor, clocked at 4Ghz and an AMD Radeon R9 M295X with 4GB of GPU memory. Blue columns show measured times for $10,000$ data points, green columns for $50,000$ points and orange columns correspond to tests using $100,000$ data points. It can be seen that the performance mostly depends on the resolution of the grid, while kernel size and number of points have a smaller effect. However, for larger resolutions, the impact of these two factors is visible. Overall, it can be seen that for the 128^2 resolution, we easily achieve real-time update rates for all tested kernel and data sizes. We can keep interactivity even at 512^2 resolution and $100,000$ data points.

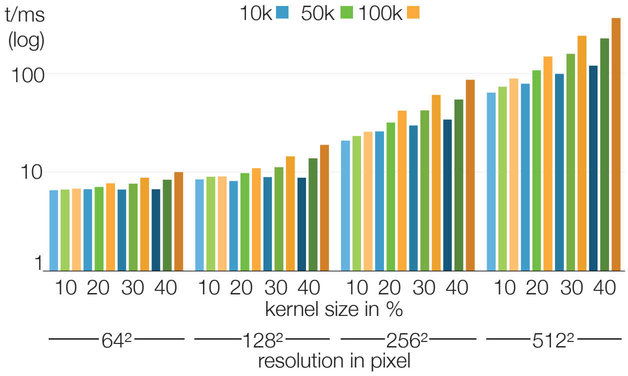


Figure 8: **Performance** of our Mean-Shift Clustering. The graph shows that the grid size has the biggest impact on performance, while the number of points and kernel size only contribute slightly.

7. Results

As described in Section 5, we focused on improving existing visual encodings and designing an integrated interactive workflow with the goal to improve efficiency. We conducted interviews with three experts from our collaborating institute to validate the choices we made to improve the visual encodings (Section 7.1). A prototypical version of the presented workflow, using separate tools, such as Cytobank, Matlab and custom R-scripts, is the basis for our collaborators complete study as presented in [vULM*16]. For detailed information on specific findings, especially how the workflow supports hypothesis generation, we refer to that work. In our case study (Section 7.2), we focus on how we improve the effectiveness of the analysis by creating an integrated interactive system.

The participants in our evaluation had different exposure to Cytosplore before the study. Participant A was our main partner when developing the workflow and had strong influence on the design process of the system. He tested the system since its inception and can be considered an expert user. Participant B is a close collaborator but was less involved in creating the system. She tests the system frequently but for her daily routine still relies on other tools. Participant C was presented with the final system just for this study and only had brief exposure to a very early prototype before. All participants are familiar with the available computational tools.

7.1. User Evaluation

We demonstrated the tool to the participants in a group session and installed it on their lab computers, including a short document, describing the most important features and how to access them. The participants had as much time as needed to familiarize themselves with the system. We followed this up with a structured interview, to find out which parts of the proposed system work and which could be improved.

The integrated nature of Cytosplore provides a strong improvement. Participant C specifically mentions the linking;*"to see which clusters in the heatmap are which cells in the tSNE* [...] *makes it easy to make adjustments in the beginning of the pipeline"* and *"makes it more reliable"*.

All participants agree that showing two markers at once in the SPADE visualization *"saves time"* (Participant B). Participant C mentions that she is fine with using a single marker in Cytobank, *"but with two markers, it is a lot faster to find subsets"*. Without knowledge that we tested more markers in an early design phase, she also states that *"more than two markers would probably* [...] *make me lose the overview."*. The circular heatmap received mixed reactions. Participant C states that *"it is not very helpful when a lot of markers are used in the panel"*. Hovering over each item to see the corresponding marker is time consuming, *"however, it is still faster than adjusting the color of the node one by one"*. Particpant A sums it up to *"looking at high detail for one node is a luxury but not a necessity"*, validating our choice to make it optional.

Participant B works with data that sometimes produces very small lineages (i.e., consisting of a few hundred cells). During testing, she was able to successfully define the subsets with this kind of data. With such small data, where the differences in the density of the embedding are rather subtle, *"we need the heatmap to combine our immune knowledge to define the kernel bandwidth."*. Before, this process completely failed with her standard workflow. Interactively defining the kernel density made Participant C much more confident in the results of the density-based clustering: *"Yes, this* [the linked heatmap view] *is very helpful. The variation display shows even more clearly whether more subsets need to be created."* Participant A praises the linking between clustering and the heatmap visualization of marker expressions: *"It immediately feedbacks the signatures revealing overall heterogeneity and homogeneity that often is the* ***unknown*** *for your data. It gives so much valuable simulteaneous information and you are flexible in changing parameters without having to do hours of computations again. I am really happy with it."* He does not, however, use the visualization of the standard deviation since markers without a clear low or high expression are hard to discern from the background due to the diverging colormap with a white center. We since changed the available colormaps in the heatmap view by removing the very light colored blocks, but did not conduct an updated evaluation.

7.2. Case Study

To measure the efficiency of our proposed system, we set up a small case study. The study consists of a single blood sample which was downsampled to 50,000 cells. The task was to specify the phenotypically distinct subsets within the dominant major lineage (CD4+T) within the sample.

We asked Participant A to create the subsets using his traditional workflow [vULM*16] as a benchmark, as well as our workflow for comparison. We chose Participant A because he is the most experienced user among our three participants. Table 1 shows the time it took to create the subsets with the traditional workflow compared to the time with our integrated solution. It can be seen that Cytosplore outperforms the traditional workflow roughly threefold. It should be noted that this small test case cannot completely capture the details of the workflow. E.g., as shown in Section 6.1, our implementation of the clustering for **T2b** scales very well with increasing data sizes, whereas the automatic clustering within ACCENSE often takes hours with real-world data sizes. However, it was necessary to use such a simple example, to allow the subset definition within a reasonable time frame.

SPADE and tSNE computations in Cytobank are done in the cloud. We assume they use distributed computing, as their conventional tSNE was computed in the same time as our A-tSNE. However, since Cytobank runs on shared hardware, SPADE and tSNE computations are queued for all users and wait times easily reach hours during peak times. We measured the time only after the job was started to make sure the comparison is fair.

Table 1: **Case Study Performance.** Time in minutes needed for the different steps in the workflow.

	Total	**T1:** Lineage Delineation	**T2a/b:** Subset Computation*	**T2c:**Subset Postprocessing
Traditional	108	27	29	52
Ours	39	13	11	15

* completely automatic in the traditional and interactive in our workflow

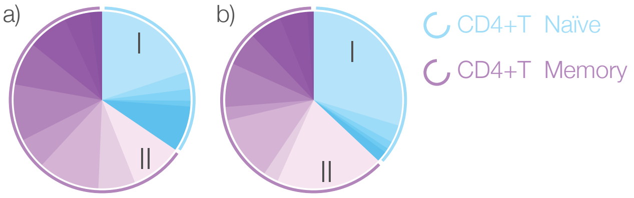


Figure 9: **Subsets Created in the Evaluation** by Participant A with the traditional workflow (a) and using Cytosplore (b). Note that a consists of only 54% of the cells assigned to the lineage, due to incomplete clustering using ACCENSE.

With our tool, clusters can be merged with a few clicks and be verified immediately. The most time is needed for the biological interpretation of the heatmap itself. We can see a large speed up in this step, due to the fact that this is the least integrated part in the original workflow and requires several different tools and sometimes multiple iterations for verification of the results.

Finally, we compared the subsets that were assigned to each cell, to make sure our results are comparable to the traditional workflow. In the SPADE tree 27,172 cells were assigned to the created CD4+T lineage with the traditional workflow, 26,591 with ours. Within the lineage, in all tests, the same 14 subsets were identified after merging 16 automatically-generated subsets in the traditional workflow and 19 with ours. The results are not directly comparable on a single-cell level, because ACCENSE only clustered 14,643 of the original 27,172 cells. Figure 9 shows the composition of the cells according to the subset specification during the evaluation. Except for the groups labeled I and II, where we found more cells using Cytosplore, the results were very similar; overall 14 subsets, 6 CD4+T Naïve (different shades of blue) and 8 CD4+T Memory (different shades of purple) were defined in all tests. After further investigation, we found out that the additional cells in group I and II were mostly from the regions that were not clustered using ACCENSE. It needs to be investigated further, whether the difference is due to a bias introduced by the incomplete clustering in ACCENSE, or if the greedy clustering using mean shift introduces cells into the subsets where the phenotype is uncertain.

To summarize, we were able to achieve comparable results using our interactive workflow, when compared to previous work [vULM*16]. Therefore, we assume that our framework allows for generating hypotheses in a similar fashion. However, it has the main advantage of significantly higher efficiency, when compared to the previous approach.

8. Conclusion and Future Work

We presented Cytosplore, an interactive integrated system and workflow for the specification of phenotypical subsets in large high dimensional cytometry data sets. We have shown the benefits of our approach in a case study evaluation. Participants found our integrated workflow useful and it allows them to produce results considerably faster than with their traditional workflow. The integrated nature of Cytosplore leads to much faster iteration during the subset specification.

Cytosplore allows us to go beyond data sizes currently possible to handle with other tools by effectively partitioning the input. However, scalability (in terms of data points) is still limited by the input size for A-tSNE. In our tests, tSNE is not only a limiting factor in terms of computational performance, but the embedding quality also quickly degenerates when going beyond a few million data points. We expect the number of dimensions to rise to around a hundred. For the computational tools presented in this work this will not be an issue. Cytosplore is also flexible enough to be employed in a basic clinical setting, e.g., to analyze the lower-dimensional flow cytometry data. If data are small enough, e.g., when analyzing a single blood sample, the overview generation using SPADE can be skipped entirely and the data can be analyzed using the embedding and heatmap, immediately.

For the analysis of the immune system as a whole, the specification of cell types is only the first step, followed by a quantitative analysis of the found subsets. In future work, we would like to integrate the quantitative analysis within Cytosplore.

Acknowledgements.

We would like to thank the anonymous reviewers for their constructive criticism and Baldur van Lew for narrating the supplemental video. This work received funding through the STW Project 12720, VAnPIRe.

References

[ADT*13] Amir E.-a. D., Davis K. L., Tadmor M. D., Simonds E. F., Levine J. H., Bendall S. C., Shenfeld D. K., Krishnaswamy S., Nolan G. P., Pe'er D.: viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature Biotechnology 31* (2013), 545–552. doi:10.1038/nbt.2594. 3

[AHD06] *The American Heritage Dictionary of the English Language*, 4 ed. Houghton Mifflin Harcourt, 2006. 2

[BM13] Brehmer M., Munzner T.: A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2376–2385. doi:10.1109/TVCG.2013.124. 2, 4

[BOH11] Bostock M., Ogievetsky V., Heer J.: **D**^{3}: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2301–2309. doi:10.1109/TVCG.2011.185. 7

[BSIM14] Brehmer M., Sedlmair M., Ingram S., Munzner T.: Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of ACM BELIV Workshop* (2014), pp. 1–8. doi:10.1145/2669557.2669559. 2, 4

[BSQ*11] Bendall S. C., Simonds E. F., Qiu P., Amir E.-a. D., Krutzik P. O., Finck R., Bruggner R. V., Melamed R., Trejo A., Ornatsky O. I., Balderas R. S., Plevritis S. K., Sachs K., Pe'er D., Tanner S. D., Nolan G. P.: Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science 332*, 6030 (2011), 687–696. doi:10.1126/science.1198704. 3, 4

[BZF*12] Bodenmiller B., Zunder E. R., Finck R., Chen T. J., Savig E. S., Bruggner R. V., Simonds E. F., Bendall S. C., Sachs K., Krutzik P. O., Nolan G. P.: Multiplexed mass cytometry profiling of cellular states perturbed by small-molecule regulators. *Nature Biotechnology 30* (2012), 858–867. doi:10.1038/nbt.2317. 3, 4

[CM02] Comaniciu D., Meer P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (2002), 603–619. doi:10.1109/34.1000236. 6

[DHHH13] Demiralp c., Hayden E., Hammerbacher J., Heer J.: invis: Exploring high-dimensional RNA sequences from in vitro selection. In *IEEE Biological Data Visualization (BioVis)* (2013), pp. 1–8. doi:10.1109/BioVis.2013.6664340. 3

[GH13] Gove R., Herzog B.: Visualizing uncertain critical paths in schedule management. Poster at the IEEE Conference on Visualization (VIS), 2013. 6

[KKI10] Kotecha N., Krutzik P. O., Irish J. M.: *Current Protocols in Cytometry*. 2010, ch. 10, Web-Based Analysis and Publication of Flow Cytometry Experiments. doi:10.1002/0471142956.cy1017s53. 3

[LH11] Lampe O. D., Hauser H.: Interactive visualization of streaming data with kernel density estimation. In *Proceedings of the IEEE Pacific Visualization Symposium* (2011), pp. 171–178. doi:10.1109/PACIFICVIS.2011.5742387. 7

[LSS*12] Lex A., Streit M., Schulz H., Partl C., Schmalstieg D., Park P. J., Gehlenborg N.: StratomeX: visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization. *Computer Graphics Forum 31*, 3 (2012), 1175–1184. doi:10.1111/j.1467-8659.2012.03110.x. 3

[LZN*15] Lujan E., Zunder E. R., Ng Y. H., Goronzy I. N., Nolan G. P., Wernig M.: Early reprogramming regulators identified by prospective isolation and mass cytometry. *Nature 521* (2015), 352–356. doi:10.1038/nature14274. 3, 4

[MMDP10] Meyer, Munzner T., DePace A., Pfister H.: MulteeSum: A tool for comparative spatial and temporal gene expression data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 908–917. doi:10.1109/TVCG.2010.137. 3

[MMP09] Meyer M., Munzner T., Pfister H.: MizBee: A multiscale synteny browser. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 897–904. doi:10.1109/TVCG.2009.167. 3

[OKB*08] Ornatsky O. I., Kinach R., Bandura D. R., Lou X., Tanner S. D., Baranov V. I., Nitz M., Winnik M. A.: Development of analytical methods for multiplex bio-assay with inductively coupled plasma mass spectrometry. *Journal of Analytical Atomic Spectrometry 23* (2008), 463–469. doi:10.1039/B710510J. 2

[PLvdM*16] Pezzotti N., Lelieveldt B. P. F., van der Maaten L., Höllt T., Eisemann E., Vilanova A.: Approximated and user steerable tSNE for progressive visual analytics. *IEEE Transactions on Visualization and Computer Graphics, under revision* (2016). arXiv:1512.01655. 3, 5, 6

[QSB*11] Qiu P., Simonds E. F., Bendall S. C., Gibbs Jr K. D., Bruggner R. V., Linderman M. D., Sachs K., Nolan G. P., Plevritis S. K.: Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature Biotechnology 29* (2011), 886–891. doi:10.1038/nbt.1991. 3, 4

[RLKG*09] Rost R. J., Licea-Kane B. M., Ginsburg D., Kessenich J. M., Lichtenbelt B., Malan H., Weiblen M.: *OpenGL Shading Language*. Addison-Wesley Professional, 2009. 7

[Rot10] Rothenberg E. V.: Lineage determination in the immune system. *Immunological Reviews 238*, 1 (2010), 5–11. doi:10.1111/j.1600-065X.2010.00965.x. 4

[SBDC14] Shekhar K., Brodin P., Davis M. M., Chakraborty A. K.: Automatic classification of cellular expression by nonlinear stochastic embedding (ACCENSE). *Proceedings of the National Academy of Sciences 111*, 1 (2014), 202–207. doi:10.1073/pnas.1321405111. 3, 6

[SDP13] Sirotkovic J., Dujmic H., Papic V.: Accelerating mean shift image segmentation with IFGT on massively parallel GPU. In *36th International Convention on Information Communication Technology Electronics Microelectronics (MIPRO)* (2013), pp. 279–285. 7

[SGF*15] Spitzer M. H., Gherardini P. F., Fragiadakis G. K., Bhattacharya N., Yuan R. T., Hotson A. N., Finck R., Carmi Y., Zunder E. R., Fantl W. J., Bendall S. C., Engleman E. G., Nolan G. P.: An interactive reference framework for modeling a dynamic immune system. *Science 349*, 6244 (2015). doi:10.1126/science.1259425. 3

[SMT13] Sedlmair M., Munzner T., Tory M.: Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2634–2643. doi:10.1109/TVCG.2013.153. 5

[SSKLK13] Shreiner D., Sellers G., Kessenich J. M., Licea-Kane B. M.: *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley Professional, 2013. 7

[vdM09] van der Maaten L.: Learning a parametric embedding by preserving local structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)* (2009), vol. 5, pp. 384–391. 5

[vdMH08] van der Maaten L., Hinton G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9* (2008), 2579–2605. 3, 5

[vULM*16] van Unen V., Li N., Molendijk I., Temurhan M., Höllt T., van der Meulen-de Jong A. E., Verspaget H. W., Mearin M. L., Mulder C. J., van Bergen J., Lelieveldt B. P. F., Koning F.: Mass cytometry of the human mucosal immune system identifies tissue- and disease-associated immune subsets. *Immunity 44*, in press (2016). doi:10.1016/j.immuni.2016.04.014. 2, 8, 9

[YDGD03] Yang C., Duraiswami R., Gumerov N., Davis L.: Improved fast Gauss transform and efficient kernel density estimation. In *Proceedings of Ninth IEEE International Conference on Computer Vision* (2003), pp. 664–671 vol.1. doi:10.1109/ICCV.2003.1238383. 7

[ZLG*15] Zunder E. R., Lujan E., Goltsev Y., Wernig M., Nolan G. P.: A continuous molecular roadmap to iPSC reprogramming through progression analysis of single-cell mass cytometry. *Cell Stem Cell 16*, 3 (2015), 323–337. doi:10.1016/j.stem.2015.01.015. 3