CyteGuide: Visual Guidance for Hierarchical Single-Cell Analysis

Thomas Höllt, Nicola Pezzotti, Vincent van Unen, Frits Koning, Boudewijn P.F. Lelieveldt, and Anna Vilanova

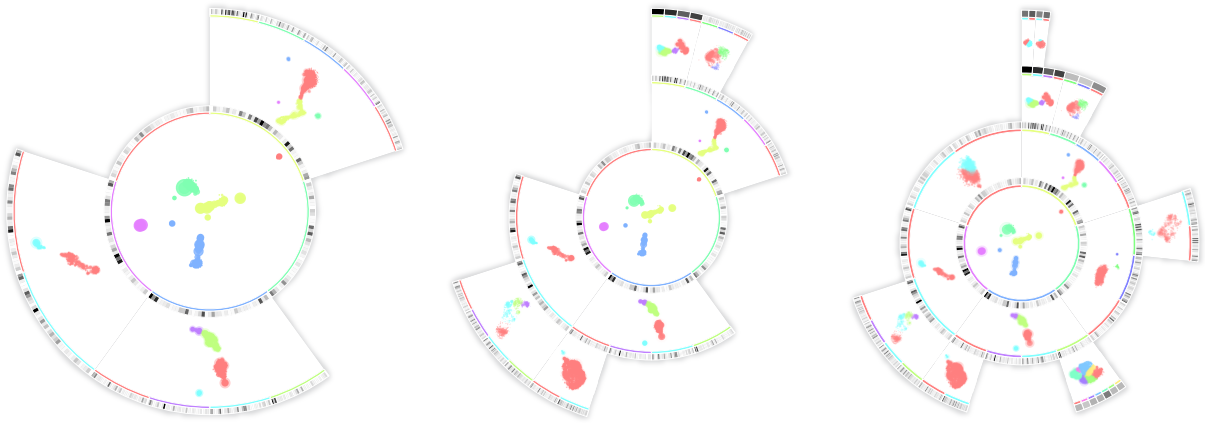


Fig. 1. **CyteGuideVisualizations** of different stages in the exploration of an HSNE hierarchy.

Abstract—Single-cell analysis through mass cytometry has become an increasingly important tool for immunologists to study the immune system in health and disease. Mass cytometry creates a high-dimensional description vector for single cells by time-of-flight measurement. Recently, t-Distributed Stochastic Neighborhood Embedding (t-SNE) has emerged as one of the state-of-the-art techniques for the visualization and exploration of single-cell data. Ever increasing amounts of data lead to the adoption of Hierarchical Stochastic Neighborhood Embedding (HSNE), enabling the hierarchical representation of the data. Here, the hierarchy is explored selectively by the analyst, who can request more and more detail in areas of interest. Such hierarchies are usually explored by visualizing disconnected plots of selections in different levels of the hierarchy. This poses problems for navigation, by imposing a high cognitive load on the analyst. In this work, we present an interactive summary-visualization to tackle this problem. CyteGuide guides the analyst through the exploration of hierarchically represented single-cell data, and provides a complete overview of the current state of the analysis. We conducted a two-phase user study with domain experts that use HSNE for data exploration. We first studied their problems with their current workflow using HSNE and the requirements to ease this workflow in a field study. These requirements have been the basis for our visual design. In the second phase, we verified our proposed solution in a user evaluation.

Index Terms—Hierarchical Data, HSNE, Single-Cell Analysis, Visual Guidance.

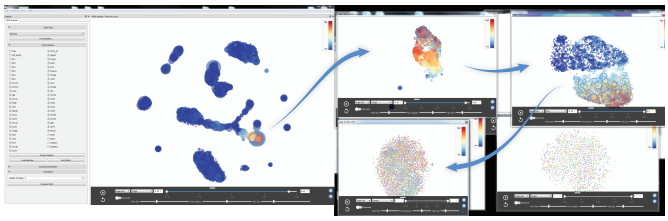
1 Introduction

Many different application and research areas, from text analysis [19] to life sciences nowadays produce high-dimensional data [21, 22, 40]. Exploratory visual analysis of such data is a challenging process but often necessary, when the contents of a dataset is unknown. Direct visualization techniques, such as parallel coordinates [13] and scatterplot matrices [10] do not scale to large numbers of dimensions or data points. In such cases dimensionality reduction techniques like PCA or t-SNE [38] are often applied to limit the number of dimensions to two or three for visualization, for example in a scatterplot. Hierarchical dimensionality reduction schemes, such as hierarchical stochastic neighbor embedding (HSNE) [24] or HiPP [23] were introduced to handle ever increasing amounts of data with limited visual space. Generally, such techniques group data points on higher, more abstract, levels of a hierarchy to provide an overview and present details on demand.

Single-cell data analysis is an example of high-dimensional data analysis with strongly increasing data amounts, where dimensionality reduction is commonly used. We recently implemented HSNE in our interactive, visual single-cell analysis framework Cytosplore [11]. By creating hierarchies of multiple levels, HSNE allows the interactive exploration of millions of data points, while preserving non-linear structures in the data even on the most abstract levels of the hierarchy. This hierarchy can then be explored by neighborhood embeddings similar to t-SNE. In these embeddings points are placed based on similarity, over all dimensions. In the basic HSNE implementation in Cytosplore, the user would start with a high level embedding, showing only the most representative cells, so-called landmarks. Each of these landmarks represents a group of cells of the next, more detailed level. The user can then zoom in selectively to see more details on demand.

- *Thomas Höllt, Nicola Pezzotti, and Anna Vilanova are with the Computer Graphics and Visualization Group, Delft University of Technology, The Netherlands. E-mail: {T.Hollt-1|N.Pezzotti|A.Vilanova}@tudelft.nl*
- *Thomas Höllt is with the Computational Biology Center, Leiden University Medical Center, The Netherlands.*
- *Vincent van Unen and Frits Koning are with the Department of Immunohematology and Blood Transfusion, Leiden University Medical Center, The Netherlands. E-mail: {V.van_Unen|F.Koning}@lumc.nl*
- *Boudewijn P.F. Lelieveldt is with the Division of Image Processing, Department of Radiology, Leiden University Medical Center, The Netherlands and the Pattern Recognition and Bioinformatics Group, TU Delft, Delft, The Netherlands. E-mail: B.P.F.Lelieveldt@lumc.nl.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx



(a) without CyteGuide

(b) with CyteGuide

Fig. 2. **Screenshots of HSNE Explorations** taken at the end of our user study. (a) Without CyteGuide our users (P1: top, P3: bottom) tried to arrange as many views as possible on two screens, usually reserving one screen for the main program window (left) containing the highest level embedding and place zoom-ins on the second screen. Eventually they would start overlapping and stacking the views. Identification of branches and corresponding views would rely mostly on proper naming of the views and the users memory. (b) The end of the exploration with CyteGuide as conducted by P1, all embeddings were kept as tabs on the left, while the CyteGuide was used as the main view. The blue arrows in (a) and (b) show the path that P1 had taken for the final zoom (Section 5, T4) in during the user study.

This approach tackles scalability issues of techniques like t-SNE in terms of data size and computational performance, however, at the cost of increased user interaction. By looking just at the unordered embeddings the user can easily lose the overview of the state of the exploration. To give an impression of a typical exploration with the original implementation, Fig. 2a shows exemplative screenshots of two small explorations conducted without CyteGuide. For comparison Fig. 2b shows an example with CyteGuide. We took these screenshots at the end of a user study we conducted to evaluate CyteGuide, presented in Section 5. Fig. 2a shows the exploration strategy conducted by two different study participants, while Fig. 2b shows the same participant as on the top in Fig. 2a.

Furthermore, HSNE does not provide any guidance for exploration, or overview of the state of the exploration beyond a set of unordered embeddings. Computing the embeddings is costly and can be unnecessary, if a higher level embedding already shows all features of interest. Therefore, guiding the user to regions of interest can save computation time. While our partners were able to create meaningful insight with HSNE in Cytosplore [39], especially due to its scalability, the exploration of the hierarchy became a challenging task.

The goal of this work is to ease the process of exploring the hierarchy by providing an overview of the current state of the exploration, but also by pointing the user to unexplored parts that could provide deeper insight in lower levels of the hierarchy. Therefore, we present the design and implementation of CyteGuide. CyteGuide, shown in Fig. 1, is an integrated visualization that summarizes the current state of the exploration of hierarchical data. It provides guidance on the necessity and direction of further exploration. While we designed CyteGuide for the application of single-cell analysis with HSNE, the concepts are applicable to the exploration of other hierarchically represented data as well as other analysis tools where similar concepts of selection, zooming, and data representation are applicable.

The main contribution of this work is the design and implementation of CyteGuide. Therefore, we first defined the requirements to ease the analysis of hierarchically represented data in the application domain of single-cell analysis during a field study and finally verified the complete design by a user study.

In the following Section 2 we specify the requirements for the proposed design. We present previous work related to this project in Section 3. The design and its underlying ideas are described in Section 4 and its effectiveness is evaluated in Section 5. Finally we conclude and discuss open problems for future work in Section 6.

2 Requirement Analysis

To define the requirements for our visualization, we conducted a field study with three collaborating domain experts who had been working with Cytosplore and HSNE for several months. Table 1 presents an overview of the participants of the field study and the evaluation presented in Section 5. Participant 1, also a co-author of this manuscript, has been involved in our efforts of applying HSNE to single-cell analysis, and has been testing early prototypes of our HSNE implementation without CyteGuide extensively.

Over several sessions, we watched the participants' interaction with CyteGuide and finally asked about their experience in a short, structured questionnaire (see http://cyteguide.cytosplore.org). The questionnaire consisted of three sections. One to define the requirements for guiding and navigating the exploration, one for summarizing the exploration, and one to quantify a typical exploration.

A typical HSNE exploration starts with the computation of the complete hierarchy. Previous work on using HSNE for single-cell analysis [39] proposes to create a hierarchy consisting of $\log_{10} N - 2$ levels for N cells in the input dataset. Typically for such mass cytometry single-cell datasets N is in the order of 10^5 to 10^7. After the hierarchy is computed the most abstract level of the hierarchy is visualized as a similarity embedding of the contained landmarks. To analyze the data, the user would now select regions in the high-level embedding, for example, based on visual clustering of the landmarks in the embedding or by inspecting the original values of the high dimensional space, also called marker expression. Then an embedding of the corresponding cells on the next more detailed level is requested for the selection. The blue arrows in Fig. 2 indicate such a zoom in from the highest level to the data level through multiple intermediate levels that one of the domain experts created during the user study (Section 5).

We asked whether the participants were able to navigate these hierarchies and what information is necessary to effectively do so. All participants said they cannot keep track of where they already zoomed in with the current way of only showing the actual embeddings in linked visualizations ("*where was I?*", "*where am I?*"). Especially the exploding number of partitions with increasing levels caused problems. One participant mentioned that he "skipped" levels, that is, he selected all landmarks in that level and embedded them directly in the next more detailed level, to reduce the number of views and by this the cognitive load of keeping track where he already explored the data. When asked about what they would need to keep track of the exploration process, participants uniformly mentioned that the most important thing would be to know which clusters they already zoomed into. One participant mentioned that he uses information of previous runs of the same data to reproduce those in his current workflow. Therefore, he would manually screenshot and annotate different levels of the hierarchy while exploring and then later would refer to the collected data.

Table 1. **Participants of the Evaluation.** The participants hold different positions from MSc Student to Lab Technician and work regularly with different computational tools for analyzing single-cell data. Participant 1 is a co-author of this manuscript and involved in the development of HSNE for single-cell analysis and a test user since our first efforts in that area. The other participants all actively used HSNE in Cytosplore for several months, at least weekly. For the study we asked them to rate their own expertise with Cytosplore and HSNE on a scale of 1 to 5. Order indicates whether the participants carried out the experiment first with or without CyteGuide.

Participant	Position	Department	Use HSNE for	Frequency	Expertise (1..5)	Order
P1*°	PhD Student	Immunology	10 months	3x/week	5	with/without CyteGuide
P2°	Research Technician	Immunology	4 months	weekly	3.5	without/with CyteGuide
P3°	MSc Student	Immunology	4 months	2x/week	3	with/without CyteGuide
P4	PhD Student	Immunology	2 months	weekly	4	without/with CyteGuide
P5	PhD Student	Parasitology	6 weeks	weekly	1	with/without CyteGuide

*Co-author of this manuscript. °Participant in the field study.

Zooming into more detail typically requires several seconds to minutes of computation until the embedding converges. Avoiding zooming into regions that will not provide any further information gain can speed up the data exploration significantly. Therefore, we asked the participants if they always zoomed into the hierarchy until they reached the data level. All participants answered yes, at least for the initial exploration of the data. We also asked about potential criteria to decide whether to zoom in or not. Generally, all participants would want to zoom in further if clusters still present heterogeneity within the original high dimensional space. Depending on the application, metadata (such as disease association of cells) and the corresponding cluster heterogeneity would be of interest. When asked about how they decide where to zoom in, and how they would prioritize, essentially all participants said that they would want to explore the complete dataset without any priorities, besides their "*favorite cell type*", but cases may exist, where only a specific cell type is of interest for a study.

Cytosplore allows for significant freedom when exploring the hierarchy. Within an embedding, areas for more detailed analysis are selected manually, either by brushing on the scatter plots or by selecting clusters generated using mean-shift clustering, where the kernel size is adjusted by the user, so that the clustering fits the desired granularity. The manual selection is completely unconstrained and, for example, allows overlapping selections. During the field study we found out that a lot of this freedom is unnecessary, or even actively avoided. All participants had a strong preference for zooming into clusters, instead of using manual selections. They deem this "*more reproducible*" and "*more precise*". When asked about whether they would like to inspect overlapping regions, which would not be possible with the clustering-based approach, one of the participants answered that he wants "*to actively avoid this*". Since the clustering-based approach always creates disjoint regions, he prefers to use it over manual selection. Typically, the participants would partition each HSNE embedding into one to five clusters. The special case of creating only a single cluster indicates that they would like to skip that level. Usually, more clusters were created in the final embedding of the data level. The participants generally follow the directive of $\log_{10} N - 2$ levels for N cells as described above. In summary, a typical exploration hierarchy for datasets of the order of 10^7 data points consists of up to 5 levels and and every embedding can be partitioned into approximately 1 to 5 disjoint clusters.

With regard to summarization of the hierarchy, we asked what information would be necessary to get an overview of the complete exploration and for reproduction. One participant mentions the hierarchy itself and more specifically the "*embeddings might be helpful*". Furthermore, all participants mentioned the size of the clusters ("*relative size of the data*", "*number of cells per subset*"). Also of interest are the marker expressions as well as composition statistics of the clusters. It should be noted that Cytosplore already offers a set of linked views such as heatmaps and table views that provide similar information, and as such participants might be influenced by what is already in the system. That is, they might not ask for something that is already in a different view or specifically ask for something they know to work.

Based on the observations and the answers to our questionnaire we formulate a set of requirements for a visualization that supports the exploration and summarizes the hierarchy. We divide these requirements in `general` requirements (`G1`), requirements for a visualization that effectively supports the `exploration` of the hierarchy (`E1–E3`), and requirements for a visualization that `summarizes` the hierarchy in a single visualization (`S1–S4`). We aim to design a single visual representation that can be used for both cases, to minimize the learning curve. The final visualization must support

- `G1` presenting the complete state of the exploration in a single view, consisting of up to
  - `G1a` 5 hierarchy levels
  - `G1b` 5 clusters per embedding in the next higher level of the hierarchy
- `E1` navigating the hierarchy, i.e. moving up and down in levels and finding and opening specific embeddings
- `E2` the identification of the clusters that are heterogeneous and as such need to be analyzed further. The heterogeneity can be
  - `E2a` structural, i.e. several clusters appear in the embedding
  - `E2b` functional, i.e. high variation in the marker expression
- `E3` the identification of clusters that have previously been analyzed in more detail
- `S1` the presentation of the size of each cluster
- `S2` the presentation of average marker expression for each cluster
- `S3` the presentation of the embeddings that lead to specific clusterings
- `S4` exporting the final result for reproduction of the workflow.

2.1 Data Abstraction

The main goal of CyteGuide is not the visualization of the original input data but rather to provide a meta-visualization to guide and summarize the exploration of that data. Therefore, the input data to CyteGuide is the exploration process itself, as described in Section 2. Here, we present an abstraction of that data. During the requirement analysis (Section 2) we found out that our target users explore the hierarchy purely based on zooming into disjoint clusters. This allows us to represent the exploration process as an acyclic, directed graph or a rooted tree. Each node of the tree contains a set of data points and their visual representation. The root node of the tree contains the highest level of the hierarchy in its entirety and therefore an abstract representation of the complete dataset. A child node represents a subset of the parent's data points and there is no overlap in data points between siblings. We define a level in the tree as the set of all nodes with the same number of links (connecting children to their parent) that need to be traversed to reach the root node. This corresponds directly to the original levels in the hierarchy. Finally, each node should be augmented with further information about the contained data points: heterogeneity for exploration guidance; number of contained cells; marker expression; and the corresponding neighborhood embedding for summarization.

3 Related Work

As described in the previous section, the exploration process itself can be abstracted as a tree structure, that is created by and augmented with information gathered through clustering and dimensionality reduction techniques. With treevis.net [28], Schulz provides an overview of available tree visualizations, including a categorization, based on dimensionality, representation of edges and alignment of nodes. Within this work the links within the tree only carry structural information, while the nodes contain rich additional information to be presented. Therefore, to maximize the visual space for the nodes, we focus on space-filling representations (Fig. 3) which are mostly categorized as implicit or hybrid edge representations [29]. The most notorious of these representations is probably the tree-map (Fig. 3b) and its' derivatives, introduced by Johnson and Shneiderman [14]. Tree-maps assign the whole available space to the root of the tree. Then, the space is divided into one slice per child and children are sliced recursively until the leaves of the tree are reached. In the final result, the complete visual space is used and each slice of the image corresponds to a leaf of the tree. In this representation, the root and intermediate nodes as well as the corresponding edges are identified by the combination of leaves, for example through alternating cut directions (also known as slice-and-dice), shade, etc. Some intermediate representations allow some space for showing the structure more explicitly. For example, nested tree-maps [14] indicate each node in the tree by a box, nesting the children with some margin inside this box. Tree-maps have been studied intensively and a plethora of extensions exist [2, 4, 30, 36, 41–43], all building on the same principle.

Other space-filling representations are icicles [6, 15], and their radial counterpart the sunburst layout [5, 16, 32], illustrated in Fig. 3c and Fig. 3d, respectively. As Stasko and Zhang indicate [32], space-filling "*is somewhat a misnomer*" for these techniques, as they usually leave some display space unoccupied, however, the space that is being used is populated densely. Icicles and sunburst diagrams, just like tree-maps do not show the edges of the underlying tree explicitly, but different to tree-maps they show all nodes of the tree. Therefore, icicles divide the visual space into segments for each layer in the tree. The root completely occupies the first segment. Nodes in every following layer are then placed next to their parent node, splitting up the visual space between all siblings. Sunburst graphs work the same way, but using polar, instead of euclidian coordinates. Usually the root of the tree is placed in the center and nodes of the following layers split up the space on concentric circles. A key difference between the two techniques is that the total area per level is constant for icicles, but grows in the sunburst diagram towards the outer rings. Since the number of items on each level grows towards the leaves that means that leaves in the sunburst diagram get relatively more space than leaves in the icicle diagram. Comparing tree-maps to sunburst diagrams for the analysis of hierarchical structures, Stasko et al. [33] suggest that "*SB [sunburst] is easier to learn than the TM [tree-map]*".

The tree that is built during exploration, as described in Section 2.1 is similar in concept to a compound graph. In addition to the implicit graph structure compound graphs, also called clustered graphs [7], consist of a hierarchy, grouping the nodes of the graph. Similarly, HSNE builds a hierarchy, but instead of clustering nodes in a graph data points are grouped, based on local similarities, that are typically visualized as neighborhood embeddings. Consequently, some of the techniques developed for the visualization of compound graphs inspired the development of CyteGuide, and, vice versa, CyteGuide could be adapted to the visualization of compound graphs. Sugiyama and Misue [35] and later Bertault and Miller [1] present automatic algorithms for drawing compound graphs. Raitner [25] extends these works for visual navigation. For the hybrid TreeMatrix visualization, Rufiange et al. [26] embed adjacency matrix views in nodes of a graph, which in turn is nested inside a tree-map. Our work is similar in spirit as it combines different visualization techniques to show the global hierarchy in combination with more detail in each node. A major difference in our work is that each node in the hierarchy is self contained and presents the complete information of one cluster, while compound graphs extend the previous level in the hierarchy and usually contain interactions between hierarchy levels. Dogrusoz et al. [8] adapt compound graphs for the visualization of biological pathways, by embedding subgraphs representing local pathway structure in a tree-map-like visualization. The approach focuses on the visualization of transitions within a single pathway at a time.

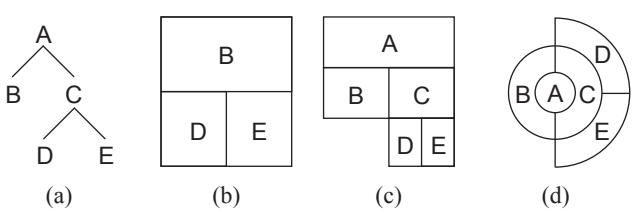


Fig. 3. **Different Representations of the Same Tree.** Node-link diagram (a), tree-map (b), icicle (c), and sunburst diagram (d).

4 CyteGuide

Here, we present the design and implementation of CyteGuide. The design was guided by the observation and discussion with three of our target users presented in the requirement analysis in Section 2.

4.1 Design

Fig. 2a illustrates how we experienced a typical exploration by our target users. Zooming into a dataset with more and more detail, using the Cytosplore system without CyteGuide would result in the users resizing and moving views between screens until eventually screen space ran out and they would overlap and stack views. The figure shows two screenshots taken after the evaluation (Section 5), each containing several embedding visualizations that correspond to different selections and levels in the hierarchy. The views are linked so that the analyst can reason about the origin of cells in a deeper level. While it is possible to keep the overview of a single linear zoom with a set of linked scatterplot views (blue arrows in Fig. 2a), in the real world the exploration quickly branches out and tens of embeddings are created. Observing our target users showed that they would typically use a second screen to organize the different visualizations of previous levels, while continuing the current path on the main screen.

4.1.1 Exploration

As described in Section 2, we expect hierarchies, consisting of tens to a few hundred nodes (Requirement `G1`), where each node will contain nested visualizations, for example, to indicate variation within the contained clusters. Links in the hierarchy are limited in number and only provide information on the stratification of the hierarchy. To make efficient use of the space and allow the presentation of embeddings (Requirements `E2a` and `S3`) inline, we opt for a space-filling representation of the overall hierarchy. Here, links are represented implicitly and the majority of the visual space is assigned to the nodes. We considered tree-maps, icicles, and sunburst diagrams as the basis for our visualization. Tree-maps are problematic, as they represent intermediate nodes of the hierarchy only implicitly, and therefore, provide limited or no visual space that could be used for nested visualizations (Requirements `E2a/b` and `S3`) to these nodes. Icicles and sunburst diagrams explicitly show all nodes in the tree. We built prototypes of CyteGuide in both variants, shown in Fig. 4. Both designs have advantages and disadvantages. On a rectangular screen the circular sunburst leaves space unused in the corners, while icicles make more efficient use of the visual space. However, the circular shape of the sunburst provides increasing space when moving away from the root, while the width of the icicle design is constant for all layers of the tree. Generally, leaves in the sunburst diagram will have more visual space, while nodes closer to the root can have more space in the icicle diagram (Fig. 4, center).

Since the number of nodes tends to grow quickly towards the more detailed levels of the hierarchy (up to five-fold, as per Requirement

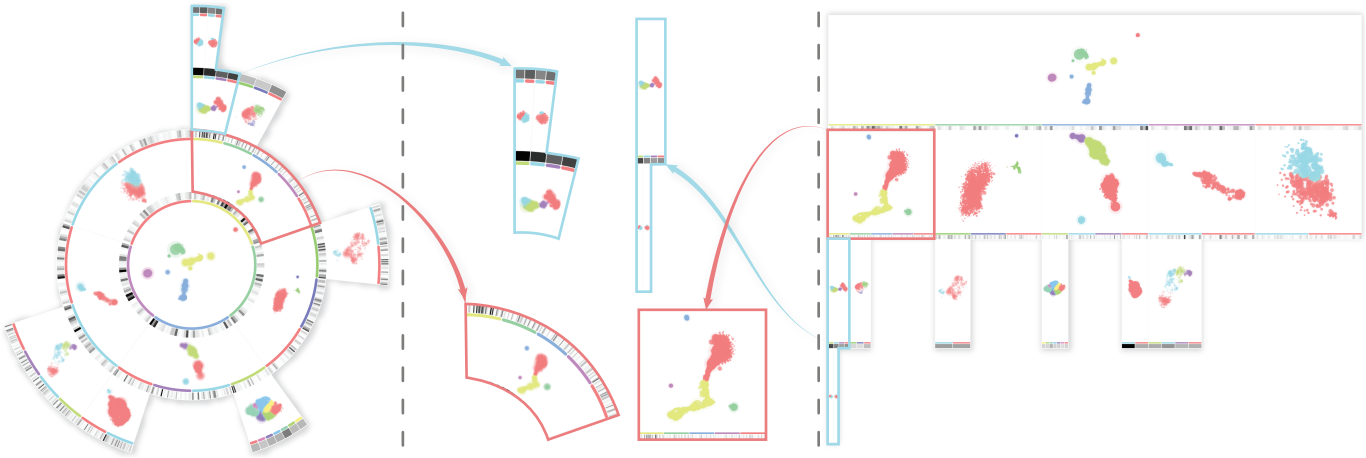


Fig. 4. **Icicle and Sunburst** visualizations of an exploration example. Comparisons of the same nodes from both visualizations in the center. Here, nodes close to the root provide more space for the embedded visualizations in the icicle visualization but nodes closer to the leaves provide more space in the sunburst visualization.

G1b), we chose the sunburst diagram for the final design. With up to five levels in the hierarchy (Requirement G1a), visual space for each level can become sparse for typical screen sizes. Therefore, we implemented an interactive version of the diagram, that allows limiting the visualization to the sub-tree of any node (Fig. 5a).

Interaction and Orientation

By clicking on a node, the user can limit the visualization of the hierarchy to the sub-tree of that node. This can be helpful to increase the visual space for the current branch of the exploration as shown in Fig. 5a. To support the user in keeping the orientation in such cases we provide a thumbnail, as well as label-based breadcrumbs, a visual representation that been proven effective for navigating in hierarchical structures in websites [17], leading to the current sub-tree (Fig. 5). The idea of the thumbnail (Fig. 5a) and breadcrumbs (Fig. 5b) is to show the path from the root of the complete tree to the root of the currently visualized sub-tree. The thumbnail is a reduced, smaller version of the complete exploration. All nodes are represented as light grey arcs, without any additional information. If only a sub-tree is being shown in the main view, the nodes from the root of the complete tree to the root of the sub-tree are shown in blue. The breadcrumbs show the names of the same nodes in linear fashion. For an interactive example please see http://cyteguide.cytosplore.org.

Both views are also used for navigation. The user can click on any node in the thumbnail or any name in the breadcrumbs path to show the sub-tree starting with this node in the main view.

Embeddings

To support Requirement E2a, structural information of the current cluster needs to be presented for each node. For expressiveness, we used a small version of the actual embedding scatterplot, corresponding to the current cluster. For better identification we use the same rectilinear coordinate system, though polar coordinates might help to use the sunburst arc space more efficiently. The result of the clustering of the current embedding, that is, the clusters that will be available to explore in more detail in the next level of the hierarchy are shown in different colors. Therefore, we use different, equally spaced hues in the hsv colorspace. Since we expect in the order of five clusters in any intermediate node the colors are well separated. We add a band of the same color to the inner arc of the corresponding segment in the next level of the hierarchy to ease the identification. Examples for both, colored clusters and the arcs, representing the cluster in the next, more detailed level in the hierarchy can be seen in Fig. 6. The top left shows a cutout of the root node, consisting of a cluster, colored in purple, that has already been embedded in more detail (attached arc). Three clusters have been created in the next level (red, blue, green) and space for the corresponding arcs has been reserved as indicated by the bands in the same color.

Heatmaps

Computing the standard deviation of markers within a cluster is much faster than creating the embedding for the next level and can already provide a reasonable impression of the variation that can be expected. Therefore, to allow direct identification of variation in the marker expression, that is, the feature space (RequirementE2b), we integrate a heatmap visualization showing the standard deviation. We show the heatmap on the inner arc of the segment corresponding to each cluster. Engle et al. [9] conducted a survey on non-traditional cluster heatmap visualizations including a sunburst visualization. While this type performed worse than others in tasks such as counting the number of clusters, the performance for tasks involving the identification of values in the heatmap, which is of main interest here, was similar to other techniques. To not interfere with the identification, we use the original order of the markers to order the heatmap.

As shown in Fig. 6, the space for the heatmap will automatically be reserved and populated, when the clusters of the preceding level are computed. In the standard setting for exploration, the heatmap shows the standard deviation for each marker for all cells contained in the corresponding cluster. By default, we use a continuous grey ramp from white to black to minimize interference with the cluster colors. If the

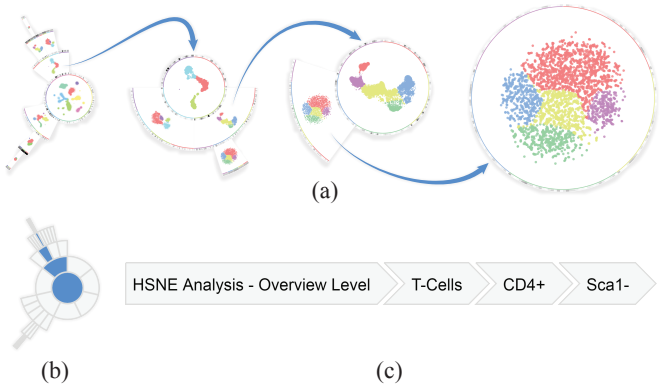


Fig. 5. **Sub-tree Visualization** (a) and Corresponding Thumbnail (b) and Breadcrumbs (c). Blue arrows point from the node that is used as the root of the new sub-tree to the new sub-tree. (b) and (c) show the icon and breadcrumbs, leading to the rightmost visualization. The blue arcs in the thumbnail and the breadcrumbs show the path from the tree root to the root of the currently shown sub-tree. All elements can be clicked to immediately jump to the corresponding sub-tree in the main view.

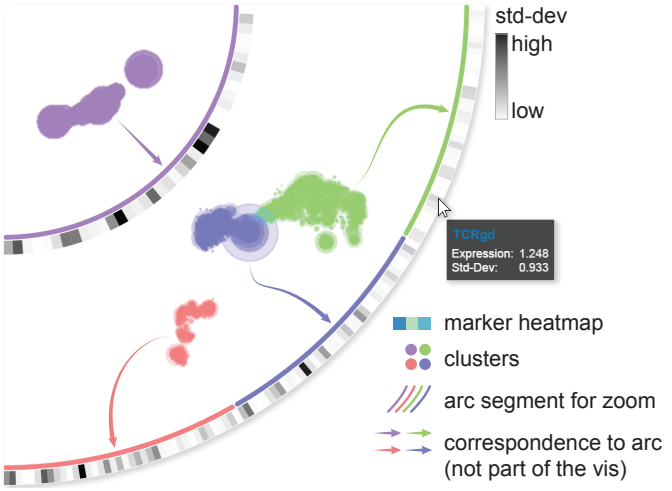


Fig. 6. **Detail of a Single Arc Segment.** Clusters are colored in the embedding visualizations and corresponding arc segments. The attached heatmaps indicate markers with high variation within a cluster.

variation within a cluster is low the heatmap will automatically fade into the background with a very light grey (green cluster in Fig. 6) while the dark segments highlight clusters with large variation (Fig. 6, purple cluster). If desired the colormap can be changed by the user.

Using the heatmap, the user might decide that, for example the green cluster in Fig. 6 is homogeneous enough to stop exploration, but the red and blue clusters need to be investigated further through more detailed embeddings that can then be computed on demand. The varying position of the heatmap can make it hard to identify a marker based on its position. Here, the number of markers that show large variation, as well as the amount of variation are more important, than which of the markers actually contribute to the variation. In case the user wants to identify a specific marker in the heatmap we provide the name, as well as the actual expression and standard deviation values in a popover that can be activated by hovering the mouse over the corresponding segment in the heatmap.

Responsive Design

During the exploration of the hierarchy the same scatterplot can be shown in vastly different sizes. For example, the root of the tree is basically screen-filling when the exploration is started, but will be much smaller, when the exploration is several levels deep. Therefore, we use three different representations for small, medium and large sizes and blend between them. Simply scaling the same representation to match the visual space (Fig. 7, top row) can lead to loss of information. Following some of the ideas Luana et al. [18] present for automatic perceptual optimization of scatterplot visualizations, we increase the size of the points while reducing their opacity in the plot for smaller versions (Fig. 7, bottom row).

Not unlike the scatterplot visualization, the heatmap needs a certain amount of visual space to be effective. When new layers are added to the hierarchy, or the view is resized, we compute the minimum extent of the arc segment assigned to each item in the heatmap, in screen space. If the size of the segment is less than two pixels at the smallest extent we switch out the full heatmap for a simple representation that only consists of a single value. Since any single marker with a high standard deviation indicates that there will be separation of the data when investigated with more detail, we use the marker with the maximum standard deviation in the corresponding cluster. As with the full heatmap we show marker name, expression, and standard deviation on mouse over. In addition we add the full heatmap to the popover to provide the complete information on demand.

Linking and Navigation

CyteGuide is not only being used as a passive view on the exploration. We also implemented several features to drive the exploration and navigate the scatterplots that are being created during the exploration to support Requirement `E1`. Without the hierarchy view, the user would select a group of points or a cluster directly in the scatterplot and request a more detailed representation. In CyteGuide, the user can also request the next hierarchy level through a right click on the arc, corresponding to an unexplored cluster. A scatterplot view for the new embedding will automatically be created and added to the main Cytosplore window. The corresponding arc in the hierarchy will be updated continuously, while the embedding is being computed. While the separate scatterplot view is not strictly necessary, it allows the presentation of additional information, such as the marker expression, through the color channel, which is reserved for the clustering information in CyteGuide. For an example see Fig. 8, showing the final CyteGuide design alongside linked heatmap and scatterplot views within Cytosplore.

Since the standard deviation of the markers is readily available whenever the user zooms in from the CyteGuide view, the marker with the highest value is automatically selected to be visualized by color overlay in the connected scatterplot view. Without CyteGuide, the user would select the marker to visualize through a drop-down menu in the view. By linking the CyteGuide view to the scatterplot view, the user can also select a marker by simply clicking on any marker in the heatmap. This will automatically bring the corresponding scatterplot view to the front and select the clicked marker. Similarly, clicking on one of the embeddings in CyteGuide will open the corresponding embedding in a separate scatterplot view.

4.1.2 Summarization

As described in Section 2, our goal is to use the same visualization for guiding and summarizing the exploration to avoid forcing users to learn two different representations. Requirement `S3` overlaps with Requirement `E2a` and is already addressed by showing the embedding in the exploration visualization.

To indicate the size of clusters (Requirement `S1`) we implemented an alternative mode, where the size of the segments in the sunburst is proportional to the number of cells in the corresponding cluster. The user can switch between the two modes on-the-fly, without leaving the view. At first glance, this mode might also be useful for the exploration, as one might expect larger clusters to be more heterogeneous, and therefore produce more nodes in their respective sub-trees. However, in single-cell analysis, small groups of cells, so-called rare subsets, are often of increased interest and the assigned visual space in the proportional mode is often too small. Therefore, we divide the arcs in equally sized segments by default and leave the proportional mode as an option, if the user is interested in size.

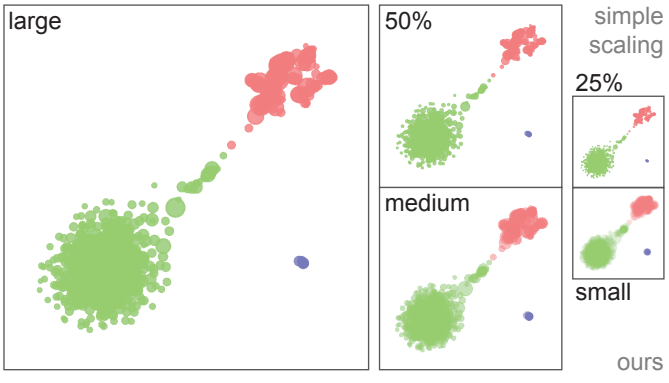


Fig. 7. **Responsive Scatterplot Renderings.** Embedding Scatterplots are rendered in large, medium, and small sizes (bottom row) for display, according to the available visual space. Separate details, such as the small purple cluster, as well as the overall structure is preserved better compared to simply scaling the large representation (top row).

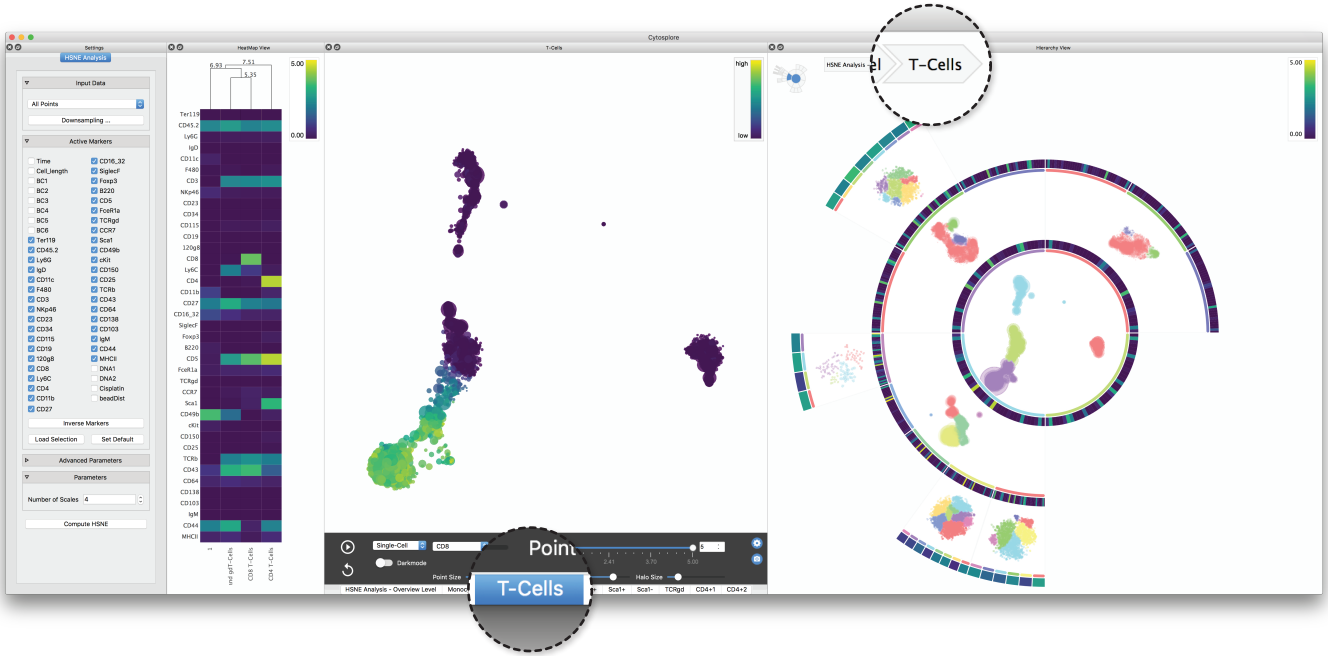


Fig. 8. **A Screenshot of Cytosplore with CyteGuide.** CyteGuide is integrated in the Cytosplore framework and linked to other available views. The views from left to right are: settings, heatmap, embedding and CyteGuide. As indicated in the breadcrumbs view, CyteGuide shows the subtree corresponding to the *T-Cells*, a major cell population that was selected on the highest level, with the corresponding embedding in the center. Marker expression was selected to be visualized in the embedded heatmaps. The embedding view also shows the *T-Cell* embedding with a marker overlaid using color-coding. The heatmap view shows the median expressions of the created clusters, each column corresponds to a cluster, each row to a marker. The matplotlib viridis colormap [12] is used to show marker expression in all views.

Once the exploration is finished the actual marker expression for the leaf clusters becomes more interesting than the variation (Requirement S2). The expression provides the user with information on the type of cell that is contained in a cluster. Heatmaps are commonly used to present these expression vectors and our target users already use separate heatmap visualizations for this task. Switching out the standard deviation heatmap to show the expression instead is therefore the natural choice. We also provide different colormaps to differentiate the marker expression from the standard deviation. Fig. 8 shows the marker expression in all views using matplotlibs viridis colormap [12]. We chose to use a different colormap, compared to showing the variation, to make the switch in context obvious, and decided for the viridis colormap, as we use it as default for the marker expression throughout Cytosplore. As for switching the segments between equal and proportional sizes, we provide the possibility to switch between the expression and standard deviation modes on-the-fly.

For presentation (Requirement S4), we provide the means to export the complete exploration visualization as a static svg image or as an interactive HTML site containing the complete exploration. The interactive version provides the same features as implemented in our system, except for the possibility to further drive the exploration or the linking with other views. It is still possible, however, to switch between the different modes for the heatmaps and segments sizes, as well as visualizing specific sub-trees. For illustration, we provide the explorations created during the course of our evaluation (Section 5) as interactive versions at http://cyteguide.cytosplore.org.

4.2 Generalization

The presented design can be applied more generally with regard to the embedded visualizations, as well as to the visualized data. In the original Cytosplore implementation [11], we presented a two-level hierarchical workflow using different visual representations on the levels of the hierarchy. That is, we used a graph-based representation on the abstract level and t-SNE plots on the detail level. In principal, CyteGuide is applicable to all such hierarchies where data is being partitioned with increasing granularity and a visual representation of each partition on every level of the hierarchy is of interest. With regard to the visualized data some limitations apply. In the presented application on single-cell data, a cell is represented as an abstract high dimensional data point. Researchers in this field of single-cell analysis are used to the abstract heat map representation of the feature space, we use to indicate the variation within clusters. For other types of data, for example the hyperspectral imaging data, presented in the original HSNE publication [24] other features, such as the spatial positions of points in a cluster might be more interesting and, therefore, each cluster should be presented as a binary mask in image space, as shown by Pezzotti et al. [24]. The limited space on the ring around the embedding in CyteGuide would be problematic in such cases. Generally, limitations in terms of data size apply, as described in Section 2. The heatmaps work well in the current setting with tens of dimensions. For visualizing very high dimensional data, such as RNA sequencing data, consisting of thousands of dimensions, the heatmap would need to adapt more fine grained than just showing all or a single marker, depending on the available space.

4.3 Implementation

We implemented CyteGuide using a combination of C++/OpenGL [31] and D3 [3] in the Cytosplore [11] software. The backend, preparing the data comprising the state of the exploration, is implemented in C++. To minimize the computational load on the visualization side, we render the scatterplots into an offscreen buffer using OpenGL and create png images in memory. This offloading was necessary as each scatterplot typically consists of thousands of objects, making direct rendering in D3 infeasible. The visualization of the exploration is implemented in D3 and data is exchanged with the backend through the QtWebkit Bridge API[1] as JSON objects. All views are linked interactively in the Cytosplore framework which allows updating the visualization as the hierarchy is explored as well as steering the exploration through CyteGuide. Fig. 8 shows a screenshot of the Cytosplore application with the integrated CyteGuide as well as traditional heatmap and scatterplot visualizations.

[1] http://doc.qt.io/qt-4.8/qtwebkit-bridge.html

5 Evaluation

We evaluated the effectiveness of CyteGuide by means of a userstudy with five domain experts. In addition to the three participants of the field study (Section 2), two more participants, who started using Cytosplore and HSNE only after we conducted the requirement analysis, were added to the user study. All participants are typical users in our target group, working on single-cell analysis using mass cytometry for different applications and used HSNE at least weekly for several weeks before the study. All were familiar with the standard functionality of Cytosplore and HSNE but were introduced to and used CyteGuide for the first time during the evaluation. To measure the performance of the data exploration with CyteGuide, we set up an example workflow which the participants would work on with our guidance (Section 5.1). After the performance evaluation, we conducted informal interviews to find out whether the other design goals were fulfilled (Section 5.2).

Before the performance evaluation, we presented CyteGuide to the participants in an interactive session, where we showed the main features by exemplative exploration of a small test dataset. The session took approximately 30 minutes and with the exception of participants 2 and 3, who had been introduced to CyteGuide together, was carried out in one-on-one fashion. During the session participants were free to interrupt for questions and to take over the software to test out features themselves at any time and did so on a few occasions. There was no training phase in which participants could get used to the tool. After the introduction the participants started right away with the exploration as described in the next section.

5.1 Performance

As a meta-visualization, CyteGuide is meant to improve the efficiency with which the user can explore the complex HSNE hierarchy (Requirement `E1`), rather than providing new insights or form hypotheses. Therefore, we prepared a small controlled experiment resembling the typical workflow as carried out by our participants in their day to day work. We defined a set of tasks to simulate an exploration and to quantify the efficiency we measured the time the participants needed to fulfill the tasks. To reduce possible bias introduced by knowledge of the data, we used the publicly available mouse bone marrow dataset provided by Samusik et al. [27] that, except for Participant 1, had not been investigated by any of the participants. The mouse bone marrow dataset consists of 841,644 data points, each representing a single cell as a 39-dimensional vector and we computed a four level hierarchy on the dataset. All participants carried out the same tasks using the same data, once with CyteGuide and once without. To account for learning effects, we asked three participants to first carry out the analysis with CyteGuide and then without, while the other two would go in reverse order (see Table 1 for the exact division).

The first given task (T1) was then to identify a group of cells, based on the expression of a given marker, zoom into that group and repeat that process for the zoomed in group. Then we asked the participants to go back to the first zoomed in group (up one level in the hierarchy) and repeat the process for a second group of cells (T2). At this point the participants would have opened five embeddings. To see if this already causes problems navigating the exploration, we asked here to point out the group on the highest level that they zoomed into in the very first step (T3). Finally, the participants were asked to separate another group of cells, based on two markers and visualize it on the data point level (T4). The tasks described above are typical tasks that commonly occur during this kind of analysis. The tasks were given to the participants at the start of the evaluation and were not known to any of the participants beforehand. We timed the completion of each task manually.

For the experiment we identified interesting groups of cells to zoom into beforehand and asked the participants to zoom into these groups, based on a given marker expression. This was necessary to make sure that the results between runs would be comparable. The identified groups and markers were typical examples that would also come up in a self-guided exploration, that is, groups of cells that formed a clear cluster in the embedding and markers that exhibited strong variation within these groups. To simulate a self driven exploration task T1 was initially split into four sub-tasks. Two zooms (T1a and T1b, supplemental) and the identification of the markers with the highest variation (TX, supplemental) after each of these zooms. These high variation markers would be a typical indicator to guide the exploration and the given marker for the next zoom was picked beforehand to be one of the highest variation markers. We removed the timings from the summarized results, shown in Fig. 9, as they heavily skewed the results in favor of CyteGuide, likely with an unrealistically large impact. We specifically added the standard deviation heatmap to CyteGuide to solve this task. Without CyteGuide participants used a table view containing several statistics or the regular separate heatmap view, where standard deviation can be encoded in size [11]. Both views were not designed for the task and are harder to read than the direct visualization. The timings can still be found in the raw data (http://cyteguide.cytosplore.org).

Fig. 9 shows the time each participant needed to complete the whole workflow (T1–T4, excluding TX) with and without CyteGuide, as well as the timings for tasks T1, T2, and T4. The computation of the embeddings can be quite lengthy and can vary strongly, depending on the selection for zooming in as well as the speed of the computer. Therefore, we removed the portions of the time, where participants were purely waiting for computations to finish. Cytosplore implements progressive visual analytics [20, 34, 37] techniques and thus stays responsive and shows meaningful intermediate results. Therefore, some of the computation time was used to interact with the data, for example to select markers to visualize, etc. This time is included in the measurements for the workflow with and without CyteGuide.

Generally the timings for T1, T2 and T4 were proportional to the number of contained zoom and navigation operations. We could see that CyteGuide reduced the number of steps, such as switching views or identifying the place in the hierarchy of the view, and the corresponding time. Except for Participant 5, T3 was instant for all participants in both settings as the corresponding views were still open and in the case of the exploration without CyteGuide, on the main screen without overlap, with the corresponding distinguishing marker still selected.

Fig. 9 shows that overall, except for Participant 4, all participants were able to navigate the hierarchy quicker with CyteGuide than without. We found out after the test that Participant 4 was not very used to the clustering options in Cytosplore, yet. Instead, she often used selections based on brushing in her workflow, which can be faster, especially if the clustering needs to be adjusted. As a result her timings without CyteGuide decreased due to her quicker selections and the timings with CyteGuide, where she was forced to use the clustering tools, increased. We discuss the implications and advantages of a clustering-based workflow in Section 5.3.

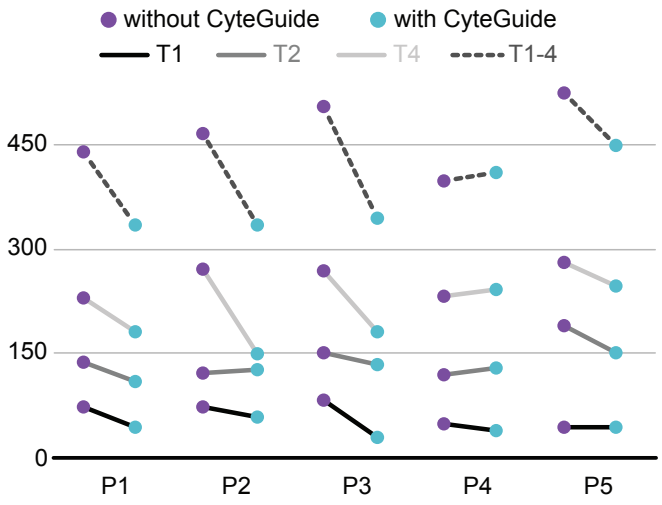


Fig. 9. **Performance Comparison** of the time in seconds (y-axis) for T1, T2, T4 and the complete evaluation. T3 is omitted for its insignificant contribution, please refer to the supplemental for the complete data. Purple dots correspond to the time without CyteGuide and blue dots to time needed with CyteGuide.

5.2 Feedback

We collected informal, qualitative feedback from the participants after the performance evaluation. Generally, all participants were quite enthusiastic about CyteGuide. P2: "*I am eager to start analyzing my data with CyteGuide.*" P3: "*I enjoyed analyzing the data.*" All participants said that the visualization helped them navigating the hierarchy. P1: "*It is also easier to switch to lower or higher levels of a hierarchy in a drilling strategy.*" P5: "*It's easier to have this hierarchy visualization to keep the overview of the different scales and clusters.*"

CyteGuide also made them more confident in exploring the hierarchy in complete detail. This is especially important, since skipping levels, as was practiced before by some of the participants to reduce the complexity of the exploration, will make embeddings on the data-level more crowded and might lead to loss of detail. P4: "*I like the function to study two or more different immune lineages in one running.*" P5: "*It's easy to explore and go back and forth between different scales, without the feeling of losing the overview.*"

We asked about the potential for guidance of the exploration, which is somewhat hypothetical, since the exploration was not completely self-guided, as described above. Generally the participants were quite positive about the potential. P1: "*It is very useful to know whether a cell cluster still contains variability, and if drilling is necessary. Also, the overview of which corresponding markers are diversely expressed is useful.*"

Keeping track of the state of the exploration "*was the strong part*" (P1). Participants 2 and 3 would like to see the given names of the clusters added to the hierarchy visualization. P3: "*Sometimes I was confused which population was visualized in the next hierarchy level, maybe the name you give to that population can pop up in the hierarchy visualization?*"

We did not specifically test the summarization mode in the user study. However, we showed the features to the participants in the first phase. Participants 1 and 4 specifically mentioned the value of the proportional mode as helpful to get an impression of cell frequencies. P1: "*[It] also gives me more sense of how small the populations are in the lower hierarchical levels.*" P4: "*I like the visualization of cell frequencies of the clusters.*"

When observing the participants while using the tool, we saw a few times that the participants tried to interact directly with the clustered embeddings within CyteGuide. Participant 2 mentioned in the open feedback that "*I am tempted to click in the hierarchy view to select clusters*". At the moment clicking on the embeddings opens the corresponding subtree. Participant 2. also brought up a second connected issue. She mentioned that "*colors in the ring are sometimes far away from subsets [clusters].*" Currently we do not optimize positions of the sunburst segments with respect to the positions of the corresponding clusters in the embedding. We are investigating this issue right now, and would also expect that solving this would reduce the urge to interact directly with the clusters.

As expected, the space on the sunburst can become quite limited, especially with many levels in the hierarchy. Participant 5 mentioned this in the open comments "*At the highest level the pictures become a bit small*" but is not concerned about it, as "*Of course you can zoom in on it.*" Finally, Participant 3 is happy that she can free up some space on her second screen "*It is good that you can use the program in one screen.*"

5.3 Discussion

While the general trend of the performance evaluation indeed indicates that CyteGuide helped the participants navigate the exploration, we can hardly claim statistical significance, due to the limited number of participants, but also the limited size of the experiment that was necessary to keep the total time to be invested by participants reasonable. Specifically the experiment could have been tackled with zooming into a cluster seven times, climbing the hierarchy three times and opening a total of eight embeddings. Typically, the given experiment would just be the beginning of a real world analysis. With the given experiment navigating the hierarchy was still rather simple, even without CyteGuide. We would argue that the difference would increase in favor of CyteGuide with larger experiments. The participants uniformly laid out as many views as possible on two screens and could therefore often directly switch between two open views in the given tasks. This became especially obvious in T3, where, except for Participant 5, all participants could immediately point to the cluster in CyteGuide but also without CyteGuide as the root embedding was still open in the main window. Eventually, with more and more views open, we expect the difference to become larger, as it would happen more often that views of interest are occluded or the analyst does not remember where a view was placed. Here, the navigational tools in CyteGuide, described in Section 4.1.1, will have a much larger impact.

As described in Section 5.1, Participant 4 was slightly slower with CyteGuide, than without. When we observed her during the evaluation we realized that she did not make use of the clustering tools for selection in the workflow without CyteGuide, but rather selected cells by brushing in the embedding. In a discussion afterwards, we found out that in her regular workflow she did not use the clustering tools but rather relied on manual brushing. We did not cover this case in the requirement analysis and, therefore, we did not consider manual selections as input to CyteGuide. To use CyteGuide Participant 4 had to adapt to the unfamiliar clustering tools, slowing her down in the analysis. In the future, we plan to adapt CyteGuide to also accept manual partitions as input. However, it should be noted that, as indicated in our field study, using automatic clustering increases the reproducibility of the analysis and can therefore be advantageous over manual selections.

Generally, the results indicate that CyteGuide indeed effectively supports the navigation of HSNE hierarchies. Participants in the study were more confident in their analysis and see potential for guidance. We also received valuable feedback for possible improvements, which will help us to iteratively refine CyteGuide .

6 Conclusions and Future Work

We presented the design and implementation of CyteGuide, an integrated visualization for guiding and summarizing the hierarchical exploration of large single-cell data. CyteGuide extends HSNE by providing effective navigation and visualization of the exploration hierarchy. We based our design on requirements gathered during a field study and verified the implemented CyteGuide in a user study. While we focused on the application in single-cell analysis using HSNE, CyteGuide can be applied to the analysis of other high-dimensional data as well as other hierarchical techniques.

We are in the process of making CyteGuide available to all users of our single-cell analysis framework Cytosplore and plan to further optimize it, based on the feedback we received during the user study and through continuous usage.

Acknowledgments

The authors would like to thank Jessica S. Suwandi, Karin de Ruiter, Na Li, and Sandra Laban for graciously providing their time for the evaluation of CyteGuide. Furthermore, we thank Baldur van Lew for narrating the supplemental video, and Julian Thijssen for proofreading the manuscript. This work received funding through the STW Project 12720, VAnPIRe.

References

[1] F. Bertault and M. Miller. An algorithm for drawing compound graphs. In *Proceedings of the 7th International Symposium on Graph Drawing*, pp. 197–204, 1999. doi: 10.1007/3-540-46648-7_20

[2] R. Blanch and E. Lecolinet. Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1248–1253, 2007. doi: 10.1109/TVCG.2007.70540

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

[4] M. Bruls, K. Huizing, and J. J. van Wijk. Squarified treemaps. In *Proceedings of the Eurographics / IEEE VGTC Symposium on Visualization*, 2000. doi: 10.2312/VisSym/VisSym00/033-042

[5] Y. Chen, X. Zhang, Y. Feng, J. Liang, and H. Chen. Sunburst with ordered nodes based on hierarchical clustering: a visual analyzing method for associated hierarchical pesticide residue data. *Journal of Visualization*, 18(2):237–254, 2015. doi: 10.1007/s12650-014-0269-3

[6] F. Chevalier, D. Auber, and A. Telea. Structural analysis and visualization of c++ code evolution using syntax trees. In *Ninth International Workshop on Principles of Software Evolution: In Conjunction with the 6th ESEC/FSE Joint Meeting*, pp. 90–97, 2007. doi: 10.1145/1294948.1294971

[7] G. Di Battista and F. Frati. Efficient c-planarity testing for embedded flat clustered graphs with small faces. In *Proceedings of the 15th International Conference on Graph Drawing*, pp. 291–302, 2008. doi: 10.1007/978-3-540-77537-9_29

[8] U. Dogrusoz, E. Giral, A. Cetintas, A. Civril, and E. Demir. A compound graph layout algorithm for biological pathways. In *Proceedings of the 12th International Conference on Graph Drawing*, pp. 442–447, 2004. doi: 10.1007/978-3-540-31843-9_45

[9] S. Engle, S. Whalen, A. Joshi, and K. S. Pollard. Unboxing cluster heatmaps. *BMC Bioinformatics*, 18(2):63, 2017. doi: 10.1186/s12859-016-1442-6

[10] J. A. Hartigan. Printer graphics for clustering. *Journal of Statistical Computation and Simulation*, 4(3):187–213, 1975. doi: 10.1080/00949657508810123

[11] T. Höllt, N. Pezzotti, V. van Unen, F. Koning, E. Eisemann, B. P. F. Lelieveldt, and A. Vilanova. Cytosplore: Interactive immune cell phenotyping for large single-cell datasets. *Computer Graphics Forum (Proceedings of EuroVis)*, 35(3):171–180, 2016. doi: 10.1111/cgf.12893

[12] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55

[13] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985. doi: 10.1007/BF01898350

[14] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the IEEE Conference on Visualization*, pp. 284–291, 1991. doi: 10.1109/VISUAL.1991.175815

[15] J. B. Kruskal and J. M. Landwehr. Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168, 1983. doi: 10.2307/2685881

[16] G. Li-Wei, C. Yi, Z. Xin-Yue, and S. Yue-Hong. A hierarchical data visualization algorithm: Self-adapting sunburst algorithm. In *Proceedings of the International Conference on Virtual Reality and Visualization*, pp. 185–190, 2013. doi: 10.1109/ICVRV.2013.36

[17] C. A. Maldonado and M. L. Jlesnick. Do common user interface design patterns improve navigation? *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 46(14):1315–1319, 2002. doi: 10.1177/154193120204601416

[18] L. Micallef, G. Palmas, A. Oulasvirta, and T. Weinkauf. Towards perceptual optimization of the visual design of scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, in press, 2017. doi: 10.1109/TVCG.2017.2674978

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations (ICLR)*, pp. 1310–1318, 2013.

[20] T. Mühlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1643–1652, 2014. doi: 10.1109/TVCG.2014.2346578

[21] O. Ornatsky, D. Bandura, V. Baranov, M. Nitz, M. A. Winnik, and S. Tanner. Highly multiparametric analysis by mass cytometry. *Journal of Immunological Methods*, 361(1–2):1–20, 2010. doi: 10.1016/j.jim.2010.07.002

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

[23] F. V. Paulovich and R. Minghim. HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1229–1236, 2008. doi: 10.1109/TVCG.2008.138

[24] N. Pezzotti, T. Höllt, B. P. F. Lelieveldt, , E. Eisemann, and A. Vilanova. Hierarchical stochastic neighbor embedding. *Computer Graphics Forum (Proceedings of EuroVis)*, 35(3):21–30, 2016. doi: 10.1111/cgf.12878

[25] M. Raitner. Visual navigation of compound graphs. In *Proceedings of the 12th International Conference on Graph Drawing*, pp. 403–413, 2004. doi: 10.1007/978-3-540-31843-9_41

[26] S. Rufiange, M. J. McGuffin, and C. P. Fuhrman. Treematrix: A hybrid visualization of compound graphs. *Computer Graphics Forum*, 31(1):89–101, 2012. doi: 10.1111/j.1467-8659.2011.02087.x

[27] N. Samusik, Z. Good, M. H. Spitzer, K. L. Davis, and G. P. Nolan. Automated mapping of phenotype space with single-cell data. *Nature Methods*, 13:493–496, 2016. doi: 10.1038/nmeth.3863

[28] H.-J. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, 2011. doi: 10.1109/MCG.2011.103

[29] H.-J. Schulz, S. Hadlak, and H. Schumann. The design space of implicit hierarchy visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):393–411, 2011. doi: 10.1109/TVCG.2010.79

[30] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS)*, pp. 73–78, 2001. doi: 10.1109/INFVIS.2001.963283

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

[32] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS)*, pp. 57–65, 2000. doi: 10.1109/INFVIS.2000.885091

[33] J. T. Stasko, R. Catrambone, M. Guzdial, and K. McDonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International Journal of Human-Computer Studies*, 53:663–694, 2000. doi: 10.1006/ijhc.2000.0420

[34] C. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1653–1662, 2014. doi: 10.1109/TVCG.2014.2346574

[35] K. Sugiyama and K. Misue. Visualization of structural information: automatic drawing of compound digraphs. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(4):876–892, 1991. doi: 10.1109/21.108304

[36] S. Tak and A. Cockburn. Enhanced spatial stability with hilbert and moore treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 19(1):141–148, 2013. doi: 10.1109/TVCG.2012.108

[37] C. Turkay, E. Kaya, S. Balcisoy, and H. Hauser. Designing progressive and interactive analytics processes for high-dimensional data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):131–140, 2017. doi: 10.1109/TVCG.2016.2598470

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

[39] V. van Unen, T. Höllt, N. Pezzotti, N. Li, M. J. T. Reinders, E. Eisemann, F. Koning, A. Vilanova, and B. P. F. Lelieveldt. Interactive visual analysis of mass cytometry data by hierarchical stochastic neighbor embedding reveals rare cell types. *under review*, 2017.

[40] V. van Unen, N. Li, I. Molendijk, M. Temurhan, T. Höllt, A. E. van der Meulen-de Jong, H. W. Verspaget, M. L. Mearin, C. J. Mulder, J. van Bergen, B. P. F. Lelieveldt, and F. Koning. Mass cytometry of the human mucosal immune system identifies tissue- and disease-associated immune subsets. *Immunity*, 44(5):1227–1239, 2016. doi: 10.1016/j.immuni.2016.04.014

[41] J. J. van Wijk and H. Van de Wetering. Cushion treemaps: visualization of hierarchical information. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS)*, pp. 73–78, 1999. doi: 10.1109/INFVIS.1999.801860

[42] R. Vliegen, J. J. van Wijk, and E.-j. van der Linden. Visualizing business data with generalized treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):789–796, 2006. doi: 10.1109/TVCG.2006.200

[43] M. Wattenberg. Visualizing the stock market. In *Extened Abstracts of the Conference on Human Factors in Computing Systems (CHI)*, pp. 188–189, 1999. doi: 10.1145/632716.632834