The end of gating? An introduction to automated analysis of high dimensional cytometry data

Florian Mair*, Felix J. Hartmann*, Dunja Mrdjen, Vinko Tosevski, Carsten Krieg and Burkhard Becher

Institute of Experimental Immunology, University of Zurich, Zurich, Switzerland

Ever since its invention half a century ago, flow cytometry has been a major tool for single-cell analysis, fueling advances in our understanding of a variety of complex cellular systems, in particular the immune system. The last decade has witnessed significant technical improvements in available cytometry platforms, such that more than 20 parameters can be analyzed on a single-cell level by fluorescence-based flow cytometry. The advent of mass cytometry has pushed this limit up to, currently, 50 parameters. However, traditional analysis approaches for the resulting high-dimensional datasets, such as gating on bivariate dot plots, have proven to be inefficient. Although a variety of novel computational analysis approaches to interpret these datasets are already available, they have not yet made it into the mainstream and remain largely unknown to many immunologists. Therefore, this review aims at providing a practical overview of novel analysis techniques for high-dimensional cytometry data including SPADE, t-SNE, Wanderlust, Citrus, and PhenoGraph, and how these applications can be used advantageously not only for the most complex datasets, but also for standard 14-parameter cytometry datasets.

Keywords: Citrus · CyTOF · Data analysis · Flow cytometry · Mass cytometry · PSM · PhenoGraph · SPADE · t-SNE · Wanderlust

Year 2015 marked the 50-year anniversary of the publication of the first cell sorter, a device that was able to separate cells in suspension based on their difference in volume [1], as well as the first cytometry-based cell analyzer [2]. Shortly after that, the first sorter that could discriminate cells based on fluorescence was developed [3, 4], and this seminal work marked the advent of flow cytometry as a widely used, single-cell analysis technique driving the identification of all major immune cell subsets known today [3, 5] (for an overview, see Fig. 1, top). These days, many laboratories are equipped with flow cytometers capable of detecting 10–20 parameters [6], revealing an ever-increasing diversity within established cellular subsets, such as CD4+ T-helper cells or professional APCs. Quite recently, an alternative cytometry-based technique has been developed that relies on antibodies labeled with heavy-metal isotopes instead of fluorophores, detecting the resulting signals using a time-of-flight detector as is done in atomic mass spectrometers [7–9]. This method has been termed “mass cytometry” and became commercially known as the “CyTOF” (cytometry by time-of-flight), allowing the theoretical detection of more than 100 parameters per cell.

While both fluorescence-based flow cytometry as well as mass cytometry provide a technological platform to interrogate the immune system at a previously unprecedented level (for a comparison of the two techniques see [10]), scientific progress depends on our ability to analyze and comprehend the resulting data in a meaningful way. Historically, flow cytometric data were—and still is—analyzed using a series of 2D plots and manual “gating,” i.e. drawing regions of interest (gates) on a plot and either

*These authors contributed equally to this work.
examining that subset on a different bivariate plot, or reporting a certain qualitative measure (cell subset is present or absent, or its relative proportion) or quantitative measure (Fig. 2A). While this approach is still adequate for many experimental settings, it suffers from several serious shortcomings. First, it is not easily scalable, as the number of 2D plots increases exponentially with the number of measured parameters (i.e. an 18-parameter experiment would require a total of 153 2D plots to display every marker combination) and thus cannot be easily comprehended anymore. Second, it has been shown by many multicenter studies, such as the Human Immune Genome Project, that manual gating is one of the largest variables in the outcome of a flow cytometry-based experiment [11]. Third, every manual gating approach relies on the researcher’s prior knowledge, thus introducing a bias toward “expected” results. Due to this focus, much of the potentially relevant and novel information within the dataset might not be recognized and could end up being ignored.

Setting the stage for alternatives

The above-mentioned problems were already recognized some years ago, and several independent groups have started to develop computational methods that automatically identify populations in multidimensional flow cytometry data. The first reports on automated cytometry data analysis date back as early as 2007 [12–14], and opened up the field for future developments (Fig. 1, bottom). With the rapid increase of available tools, in 2010 the first FlowCAP (Flow Cytometry: critical assessment of population identification) challenge was initiated and revealed that a large number of these automated algorithms perform equally well or even better than a human expert with respect to population identification as well as sample classification [15]. The more recent FlowCAP challenges (http://flowcap.flowsite.org/) have seen a number of additional promising automated analysis approaches, among them several novel methods for clustering such as SWIFT [16] and immunoCLUST [17] (for an overview of available tools and their dedicated functions see references [15, 18–22]). Nevertheless, few or none of these methods have reached the wider immunological community yet, probably due to the hesitation in adopting new technologies when existing approaches seemingly do the job adequately, a perceived difficulty in mastering the new methods, as well as the fact that the publication of many of these computational tools has been limited to bioinformatics journals.

Also, as the development of these tools is work in progress, the available packages often lack a user-friendly graphical user interface and instead require at least basic knowledge of one of the programming languages such as R, Python, Java, or Matlab. Although possibly sounding like a barrier to any bench scientist, the use of tools written in these programming languages is facilitated by the existence of free, open source repositories containing well-documented and fully functional pieces of code (packages, add-ons, modules) that perform a particular task. Perhaps the best known open source repository is Bioconductor (www.bioconductor.org), a software repository originally developed for the analysis and comprehension of high-throughput genomic data [23]. It uses the statistical programming language R and currently contains 38 packages that deal with the various aspects of cytometry data handling. Within the scope of this review we would like to discuss only some of the most prominent data analysis algorithms that aim at the visualization and exploration of high-dimensional data, focusing on their purpose and practical usability, accompanied by an example to illustrate this.

Early work in analyzing high-dimensional cytometry data often deployed principal component analysis (PCA), a statistical method...
for reducing the dimensionality of complex datasets to two or three dimensions [24–26]. PCA takes all chosen parameters of the higher dimensional data and calculates a smaller number of variable parameters (called “principal components”) that best preserve most of the variability of the original data and can be more easily displayed (reviewed in [27]). However, PCA is a computational approach that assumes linear relationships between the measured parameters, which is not the case for many biological datasets. As a result, within the past years a variety of new algorithms have been developed that are based on nonlinear dimensionality reduction and thus are adequate for the visualization and exploration of single-cell cytometry data.

**SPADE — Building a tree of relationships**

One of the early tools used to computationally analyze and visualize high-dimensional mass cytometry data is spanning-tree progression analysis of density-normalized events (SPADE) [28]. SPADE depicts cellular populations in a branched tree structure (dendrogram), thus visualizing high-dimensional data in an intuitive, 2D manner (Fig. 2B). The SPADE tree consists of connected nodes that represent clusters of cells, thereby also providing information about the relationship of cell types. Typically, the branches of the resulting SPADE tree are manually annotated to give an immediate overview of the main cellular lineages. Further, a third dimension of color can be overlaid onto the tree to depict expression levels of any given marker or fold changes between multiple sample groups. Since the high-dimensional data are visualized as nodes that have undergone a clustering step, and therefore comprise multiple cells, the single-cell resolution of the data is lost at this point and any applied coloring represents subset characteristics.

To create the SPADE tree, the algorithm follows a multistep process. Briefly, SPADE first performs density-dependent downsampling. Thus, cells with few similar neighbors in the high-dimensional space will be more likely to be included in the downsampled dataset than cells with plenty of neighbor cells. This ensures a similar representation of rare and abundant cell types in the downsampled dataset, and thus facilitates the identification of rare cellular subsets. Second, the algorithm performs hierarchical, agglomerative clustering, grouping cells with its closest neighbors until a user-defined number of clusters is reached. Next, SPADE constructs a minimum spanning tree by randomly choosing an already connected subgraph and adding an edge to the cluster or subgraph with the minimum distance to the selected subgraph. This process is iterated until all nodes are connected in one minimum spanning tree. Lastly, for all cells in the initial dataset SPADE determines the closest neighbor in the downsampled data and assigns the cell to the corresponding cluster.

The user-defined parameters that are required to construct the SPADE tree, in addition to the choice of cellular markers and the desired number of clusters, are the outlier density and target density. Outlier density is used to exclude cells with very few neighbor cells while the target density determines the outcome of the downsampling step and therefore modifies the threshold of rare populations that can still be identified. Since SPADE includes steps involving random decisions, it is of a nondeterministic nature. As a consequence, several runs of SPADE will result in differently organized trees, which should be kept in mind when applying SPADE. While the organization of the tree at each individual run might appear distinct, the identified populations have been found to be comparable across several runs.

SPADE has been used in several publications [25, 28–30], mostly to give an instant overview of different immune populations and their expression of surface markers or intracellular signaling molecules. SPADE is available as a Bioconductor package (CytoSPADE), which can be used through a standard command line interface or through an interactive graphical user interface available as a plugin for the Cytoscape Network Visualization Platform (www.cytospace.org). Alternatively, the SPADE analysis algorithm is an integral part of the Cytobank online single-cell analysis platform (www.cytobank.org), although only for the registered users of the paid, premium version.

**t-SNE — Visualizing the similarity landscape**

Another powerful tool to visualize multiparameter cytometry data is nonlinear dimensionality reduction via t-stochastic neighbor embedding (t-SNE) [31, 32]. t-SNE visualizes high-dimensional similarities of cells in an easily understandable 2D or 3D scatter plot, the so-called t-SNE map (Fig. 2C). The proximity of cells in the t-SNE map reflects their distances in the high-dimensional space. Cells that are similar in their analyzed protein-expression pattern will be located closely together in the t-SNE map, thus enabling the visualization of different cellular subpopulations. Importantly, t-SNE has been shown to successfully identify small cellular subpopulations, as low as those comprising 0.25% [32], and is able to recognize high-dimensional associations of markers that might be missed when performing conventional, 2D gating.

Briefly, the computational steps underlying t-SNE are: first, the calculation of a pairwise similarity matrix for all data-points, based on their high-dimensional distances. Next, a low-dimensionality similarity matrix is calculated based on initially random locations for each cell in the two (or three) t-SNE dimensions. In an iterative process, the algorithm then tries to minimize the difference between the high- and low-dimensional similarity matrices, thereby adjusting every cell’s position in the 2D space. Recent improvements to t-SNE make use of the Barnes–Hut algorithm [33] that uses a tree-like structure to reduce the number of pairwise similarities that actually have to be calculated. This now allows for a higher number of cells to be analyzed and shorter computation time.

Similarly to SPADE, t-SNE incorporates random seeds and thus the results from multiple runs may visualize the data differently. On the resulting t-SNE map, different immune subpopulations will visually appear as distinct clusters, however t-SNE does not per se assign single cells to defined clusters. This can be accomplished either by subsequent manual gating on the t-SNE map or...
by using automated clustering algorithms. Prominent examples of such clustering methods are ACCENSE (automatic classification of cellular expression by nonlinear stochastic embedding) [34] or DensVM [35], both employing a density-peak detection approach on the 2D t-SNE map. Additionally, the stand-alone ACCENSE application also provides the alternative option of using a k-means clustering method.

The usability of t-SNE to identify immune subpopulations has been demonstrated recently for both flow cytometry and mass cytometry data [32, 35, 36]. Various implementations of t-SNE exist. viSNE is the implementation developed in Dana Pe’ers’ lab and distributed as a part of CYT, an interactive visualization tool based on MATLAB that is free for academic use and contains a graphical user interface [32]. Cytobank has sublicensed the viSNE implementation and made it available as an integrated tool in their premium version. ACCENSE (www.cellaccense.com) is a freely downloadable stand-alone application that performs the t-SNE dimensionality reduction followed by the aforementioned clustering step, thus automating the classification of cells into sub-populations. t-SNE has also been directly implemented in various programming languages (C++, Matlab, Python, R) that are freely available for download.

Wanderlust and probability state modeling — Revealing developmental paths

Another tool to analyze high-dimensional single-cell data in the context of cellular differentiation and development has been termed “Wanderlust” [37]. Starting from a snapshot of a developmental process at a single time point (such as found in the bone marrow or thymus), the Wanderlust algorithm explores developmental relationships between cells and constructs a trajectory that resembles the developmental path of a given cellular lineage. It is therefore especially suited to capture and interrogate temporal differentiation processes in the immune system. The Wanderlust trajectory orders the single cells from their most immature to the most mature state, usually using a provisional scale from 0–1, often referred to as a progression or developmental scale. This in situ predicted trajectory can then be used to follow the regulation of all other markers across the developmental path.

Construction of the trajectory in Wanderlust is graph based. In a nutshell, the high-dimensional data are first converted into a k-nearest neighbor graph (k-NNG), which connects similar cells (neighbors) via edges that reflect their similarity in the high-dimensional space. Developmental distances between cells are then defined as shortest paths on the graph, and each cell is ordered according to its distance from an early cell. This initiator cell has to be chosen by the user and is required to determine the direction of the trajectory. In order to make the calculation of the trajectory more robust against noise, Wanderlust includes some random waypoint cells and iterates the calculation of the shortest path over several smaller subgraphs. The number of subgraphs, l, and number of nearest neighbors, k, have to be defined by the user, however it has been shown that the algorithm calculates similar results for a certain range of k and l values. In order to be able to choose these parameters as well as the starting cell, some initial biological knowledge of the system is important to validate the Wanderlust output. Again, owing to random elements in the algorithm, results from different runs will result in slightly different outputs.

The algorithm makes several assumptions about the data, which should be kept in mind when applying Wanderlust. First, the sample has to include cells spanning the complete developmental process in order to be able to calculate a continuous trajectory. Connected to this, a second assumption is that changes in protein expression levels are gradual and all cellular intermediates exist. Third, the algorithm maps cells onto exactly one trajectory and currently does not allow for branching events in the differentiation process. However, computational solutions may well integrate such functionality in the future.

Wanderlust has been successfully applied to mass cytometry data in order to define new key regulatory points in B-cell development [37]. Currently, Wanderlust can be run within the CYT tool built on MATLAB, available for download from Dana Pe’ers’ lab webpage [32].

Of note, another tool with the same intention as Wanderlust, i.e. the analysis of developmental trajectories, is commercially available as GemStone (Verity Software House) and is based on probability state modeling (PSM) [reviewed in (38)]. PSM was first reported in 2007 by Bagwell and colleagues in order to model...
high-dimensional cytometry data (Fig. 1, bottom), and has in the meantime successfully been used for modeling CD8+ T-cell differentiation [39] as well as human B-cell development [40]. Future studies will reveal whether PSM or the Wanderlust algorithm provides a more robust means to delineate developmental trajectories.

**Citrus — Identification of stratifying subpopulations**

Visualization of multiparameter cytometry data is a helpful first step in making high-dimensional data accessible to scientists for further investigation. In many cases, such subsequent questions involve the comparison of two sets of samples in search of cellular subpopulations with distinctive abundance or expression levels. The algorithm named “Citrus” provides an unsupervised and automated process—meaning that the application is not guided by any user input save the size of the smallest cluster—that combines computational identification of cellular subpopulations with various association models to reveal stratifying clusters and cellular responses that are best predictive, or best correlated with the experimental endpoint [41]. For example, such experimental endpoints can be affiliation to a certain group, good, or poor clinical outcomes or even survival time. As an input, Citrus receives cytometry data of two or more sets of samples associated with distinct groups, e.g. patient and control groups or different stimulation regimes. Citrus then automatically defines cellular populations and calculates cluster characteristics. After applying a chosen association model, Citrus provides the user with a list of clusters behaving differentially in the two sample groups, as well as characteristics of these clusters represented in histograms and bar plots (Fig. 2E). Additionally, Citrus provides a model to predict the affiliation of samples to one of the input groups using regularized classification.

Shortly, the Citrus algorithm combines all samples into one aggregate dataset before identifying cell populations by hierarchical clustering of phenotypically similar cells. Importantly, only clusters of populations that are more abundant than the user-defined minimum cluster-size threshold will be included in the subsequent analysis. Next, cells are assigned back to the individual samples and cluster characteristics, such as median expression level for all markers, are calculated. Subsequently, Citrus employs a regularized classification model to identify stratifying clusters which are used to predict the group of each sample. Citrus was introduced and applied in [41] as well as in [42–44]. It is available as an R package including a graphical user-interface.

**PhenoGraph — Clustering in high-dimensional space**

While both t-SNE and SPADE facilitate the visualization of phenotypically similar cell subsets in high-dimensional datasets, these analysis approaches do not directly assign cells to distinct clusters. Also, the density-peak detection algorithms mentioned above (ACCENSE and DensVM) [34, 35] do not take the entire dimensionality of a given dataset into account. To do so, a novel algorithm was recently developed and termed “PhenoGraph” [45].

Similar to Wanderlust, PhenoGraph models the high-dimensional space using a k-NNG, in which each cell is depicted as a node that is connected to its neighbors by edges. In this graph, phenotypically similar clusters of cells will be represented as sets of highly interconnected nodes. These can be seen as “neighborhoods” or “communities” of cells, and can be partitioned in high-dimensional space using similar community-detection algorithms that are being used for the analysis of social networks. The resulting clusters can then be visualized on a t-SNE map, which often corresponds well with the PhenoGraph clustering, or alternatively on a heat map, which will show the expression levels of selected marker across all found clusters (Fig. 2D).

The reported application of PhenoGraph to high-dimensional data of human bone marrow derived from healthy donors as well as acute myeloid leukemia (AML) patients [45] suggests that this method is superior to previously available clustering methods, and can resolve subpopulations as rare as 1 in 2000 cells. In fact, one advantage distinguishing PhenoGraph from previous dimensionality reduction approaches is the addition of a second iteration process using the so-called Jaccard similarity coefficient. This computational step aids the identification of small cellular subsets (e.g. hematopoietic stem cells) that could potentially be obscured either by noise or by other larger cell populations. PhenoGraph has also been made available in the CYT plugin for MATLAB [45].

**Practical considerations for automated data analysis**

The first practical aspect when starting to analyze high-dimensional datasets in an automated manner is to perform appropriate quality control (QC) and data preprocessing. While proper QC of both the instrument and acquired data should always be part of any standard cytometry analysis workflow, it becomes even more relevant with increasingly complex experiments. In brief, QC should at least include validation of both the instrument and reagents used (e.g. nonspecific binding of antibody, spillover due to metal impurity), and for fluorescence-based flow cytometry, the assessment of signal stability over time and especially potential artifacts arising from improper compensation (reviewed in [46]).

In addition, automated analysis of high-dimensional data asks for dedicated preprocessing steps, especially data transformation. Epitope abundances measured by cytometry often follow normal distributions on a logarithmic scale (so called log-normal distribution) with the variance of different cell populations depending on their intensity. Therefore, obtaining a suitable representation of the data across its entire intensity range is crucial for visualization and particularly for automated analysis. To do so, different transformation methods can be employed (compared and reviewed in [47] and [48]). In the context of the aforementioned algorithms, the data are frequently transformed using a logicle (or arcsinh) transformation [49, 50]. This transformation was introduced for flow cytometry data to allow negative values, which originate, on the one hand, from baseline subtractions and, on
the other hand, from spectral overlap correction (compensation), while maintaining logarithmic like scaling for large parameter values. Functions to perform such data transformations are commonly integrated into the respective analysis tools (R packages, CYT, Cytobank, ACCENSE).

Another important preprocessing step when performing automated analysis is data normalization. This standardization can be employed on multiple levels. In general, we here refer to normalization as the process of using internal or external control values to express the acquired data on a common scale, thus improving comparability between multiple measurements. A first level of normalization is often to adjust measurements to differences that might occur in instrument sensitivity (over time or across different machines), and thus might influence measured values. In both, flow and mass cytometry, the addition of standardized beads with known signal intensity can be used for this purpose [46, 51].

Further, in settings in which multiple samples with technical intersample variation (e.g. due to different cytometers, sample handling, staining process, etc.) should be included in the same automated analysis, per-channel basis normalization can help automated algorithms to identify respective cellular populations across many samples. Different approaches have been developed for this purpose, such as alignment of prominent landmark features or features of defined subpopulations [52, 53]. In addition, to help match populations from multiple samples, normalization can also be applied in order to reduce differences in the contribution of markers and antibodies with different dynamic ranges. Large separations between positive and negative populations do not necessarily reflect the biological differences between those populations, therefore adjusting the range to a common scale might be desired. This can be done by normalizing the data on a per-channel basis, e.g. to the z-score (also called standard score), thus expressing measured intensities as numbers of standard deviations above or below the mean. Alternatively, data points can also be scaled to any chosen channel-specific percentile of the dataset, thus expressing them in relation to the spread of the respective channel. Scaling options are integrated into ACCENSE and Citrus or can be easily implemented using the R-environment.

End of gating — A practical example

To provide a comparative example of the results obtained from some of the above-mentioned computational tools (Table 1, Wanderlust was not applicable in this scenario) we used a standard 14-parameter fluorescence-based panel to probe a well-studied system, namely the lung-resident immune compartment of mice deficient for the GM-CSF receptor (GM-CSFR) (Fig. 2). It has been firmly established that the GM-CSFR is essential for the genesis of alveolar macrophages, and is also involved in the development of some tissue-resident DC subsets and eosinophils [35, 54, 55]. While full analysis of this 14-parameter dataset using manual gating would require at least 91 bivariate plots, a meaningful analysis using prior knowledge still employs 15–30 plots, during which the focus of the analysis is usually biased by previous experiences of the researcher. As shown in Fig. 2A, a series of bivariate plots (pregated on CD45+ live events) can be used to identify most major subpopulations, which however requires previous knowledge of the subset surface phenotype. Differences in the abundance of alveolar macrophages and eosinophils in WT and GM-CSFR−/− mice can then be identified visually by comparing corresponding plots or by extraction of corresponding frequency values.

Analysis of the very same dataset (also after pregating on CD45+ live events and appropriate transformation as discussed above) using SPADE (Fig. 2B) or t-SNE (Fig. 2C) provides the researcher with an immediate overview of the major immune populations present in the different sample groups, of course limited by the parameters available in the panel. Importantly, the graphical representation both of t-SNE and SPADE immediately reveals the absence or reduction of two cellular populations in GM-CSFR−/− mice, which can then be identified as alveolar macrophages and eosinophils by displaying the expression level of the analyzed markers (shown here, Siglec-F). Notably, single cell resolution of the data is maintained in the t-SNE map, allowing further analysis and display (e.g. dot plot overlays with manual gates, etc.) in standard analysis software packages such as FlowJo (www.flowjo.com) or Cytobank. In contrast, the output of SPADE are small clusters of cells arranged in branches in a dendrogram, which has proven to be useful for the visualization of differences in the expression level of functional markers between different sample groups (e.g. phospho-epitopes, not used in this example) [29]. Additionally, these two methods provide a more unbiased way to analyze and compare this high-dimensional cytometry dataset, as in contrast to manual analysis they are not dependent on previous knowledge.

Next, we ran Phenograph for the same dataset to perform automated clustering of phenotypically distinct populations (Fig. 2D). The resulting clusters can then be visualized in the form of a color-dimension overlaid on a t-SNE graph or in form of a heat map. Within our 14-parameter flow cytometry experiment, Phenograph identified 21 phenotypically distinct cellular clusters (depending on the input parameters). Cluster frequencies can thus directly be compared, without manual gating on the SPADE tree or t-SNE map as performed previously, thus further automating and debiasing analysis. This makes Phenograph particularly well suited for the identification of potentially novel cellular clusters in unknown samples, as has been shown for AML patient samples [45].

Lastly, to determine the stratifying population characteristics in WT versus GM-CSFR−/− mice in a fully automated manner, we applied Citrus to our dataset (Fig. 2E). The phenotypic marker expression pattern of these stratifying subpopulations can be displayed as histograms, in this case revealing that one of the stratifying clusters is characterized as Siglec-Fhigh, CD11c+, CD11b+, and CD24+, corresponding to alveolar macrophages, while another characteristic cluster denotes eosinophils. Since these clusters were determined in a fully automated manner, Citrus promises to be particularly well suited for the analysis of previously unknown differences, e.g. between patient groups and healthy subjects or between two genotypes of mice.
Table 1. Overview of the described computational methods and their coverage of the typical cytometry data analysis workflow

<table>
<thead>
<tr>
<th>Method</th>
<th>Data visualization</th>
<th>Automated population identification</th>
<th>Availability</th>
<th>Main application</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPADE</td>
<td>Yes (clustering)</td>
<td>No</td>
<td>Script for R, Matlab; Cytobank</td>
<td>Finding fold differences in expression levels between samples</td>
<td>No single cell resolution, manual extraction of some cluster features</td>
</tr>
<tr>
<td>t-SNE</td>
<td>Yes (dimensionality reduction)</td>
<td>No ACCENSE - Yes</td>
<td>Script for Matlab, Python, R; <a href="http://www.cellaccense.com">www.cellaccense.com</a>; CYT; Cytobank</td>
<td>Direct 2D visualization of relationships in high-dimensional space</td>
<td>No automated sample comparisons</td>
</tr>
<tr>
<td>Wanderlust</td>
<td>Yes (dimensionality reduction)</td>
<td>Yes, assigning cells across developmental trajectory</td>
<td>Cyt</td>
<td>Visualization of continuous cellular developmental trajectories</td>
<td>Developmental trajectory assumed to be nonbranched, start-, and end-points must be known</td>
</tr>
<tr>
<td>PSM</td>
<td>Yes (dimensionality reduction)</td>
<td>Yes, assigning cells across developmental trajectory</td>
<td>GemStone (Verity Software House)</td>
<td>Visualization of continuous cellular developmental trajectories, can handle branching</td>
<td>At least one transition direction must be known</td>
</tr>
<tr>
<td>CITRUS</td>
<td>Yes (clustering)</td>
<td>Yes</td>
<td>Script for R</td>
<td>Finding cell clusters that differ between groups and correlate or predict experimental outcomes</td>
<td>No single cell resolution, nontrivial interpretation, and population assignment of clusters</td>
</tr>
<tr>
<td>PhenoGraph</td>
<td>Yes (clustering)</td>
<td>Yes</td>
<td>Cyt</td>
<td>Very good performing clustering algorithm and unique population identification approach</td>
<td>No automated population comparison</td>
</tr>
</tbody>
</table>

Concluding remarks

While the pace of technological advances in cytometry has for some time been ahead of the evolution of data analysis, the past years have brought fascinating novel approaches to thoroughly analyze complex cytometry data. High-dimensional flow and mass cytometry datasets can now be analyzed in a fast, automated, and more unbiased manner than previously possible through bivariate gating. Algorithms like SPADE and t-SNE can be used to visualize the data in intuitive low-dimensional representations, PhenoGraph automatically assigns cells to phenotypically different clusters and Citrus identifies populations that characterize different groups of samples. Additionally, Wanderlust as well as PSM facilitate the investigation of developmental paths in high-dimensional datasets. Of note, this list is not exhaustive and includes mostly tools that allow for exploratory data analysis.

While manual gating will continue to allow us to test our ideas and hypotheses, these computational methods for analyzing complex cytometry data have the potential to deepen our understanding of the immune system in health and disease to previously unprecedented levels. Further collaboration between biologists and computational scientists will be needed for this process, and is likely to generate even more tools to answer the questions asked by immunologists in the future.

Acknowledgments: This work was supported by grants from the Swiss National Science Foundation (SNF, accession numbers 316030_150768, 310030_146130, and CRSII3_136203), the
European Union FP7 project TargetBrain, NeuroKine, and ATECT and the University Research Priority Project: Translational cancer research.

Conflict of interest: The authors declare no financial or commercial conflict of interest.

References

B cell is expanded in multiple myeloma.


Abbreviations: AML: acute myeloid leukemia · GM-CSFR: GM-CSF receptor · k-NNG: k-nearest neighbor graph · PCA: principal component analysis · PSM: probability state modeling · SPADE: spanning-tree progression analysis of density-normalized events · t-SNE: t-stochastic neighbor embedding · QC: quality control

Full correspondence: Prof. Burkhard Becher, Institute of Experimental Immunology, University of Zurich, Zurich, Switzerland e-mail: becher@immunology.uzh.ch

Received: 27/7/2015
Revised: 15/10/2015
Accepted: 3/11/2015
Accepted article online: 9/11/2015

© 2015 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim www.eji-journal.eu