Comparison of Clustering Methods for High-Dimensional Single-Cell Flow and Mass Cytometry Data

Lukas M. Weber[1,2], Mark D. Robinson[1,2,*]

[1] Institute of Molecular Life Sciences, University of Zurich, Switzerland
[2] SIB Swiss Institute of Bioinformatics, University of Zurich, Switzerland

[*] mark.robinson@imls.uzh.ch

Abstract

Single-cell analysis techniques are frequently used to characterize and detect cell populations in biological samples. The latest advances in multicolor flow cytometry and mass cytometry enable measurement of dozens of parameters per cell in thousands of cells per second. This has led to bottlenecks in data analysis, since traditional gating techniques are insufficient for these large, high-dimensional data sets. To address this, major efforts have been made to develop automated analysis methods. A key step in automated analysis is the use of clustering methods to detect high-dimensional clusters representing cell populations. Here, we have performed an up-to-date, comprehensive, extensible comparison of clustering methods for detecting cell populations in high-dimensional flow and mass cytometry data, including several new methods that were not yet available during previous comparisons. We evaluated clustering methods using four publicly available data sets containing major immune cell populations as well as specific rare populations, with population identities available from manual gating. The comparisons revealed that FlowSOM (with optional meta-clustering but without automatic selection of the number of clusters) performed well across all data sets, and had among the fastest runtimes. We recommend FlowSOM as a first choice for analyzing new data sets. In particular, the fast runtimes enable interactive, exploratory analyses on a standard laptop. Several methods were sensitive to random starts when detecting rare cell populations, indicating that multiple random starts should be used in these cases. Our results provide a guide for researchers deciding between clustering methods for analyzing data sets from high-dimensional flow and mass cytometry experiments. R scripts to reproduce all analyses are available from GitHub (https://github.com/lmweber/cytometry-clustering-comparison), and preprocessed data files are available from FlowRepository (FR-FCM-ZZPH), allowing our comparisons to be extended to include new clustering methods and reference data sets.

Author Summary

Detecting cell types or populations is an important part of many biological experiments and medical diagnostic procedures. For example, in cancer diagnostics, tumor subtypes may be identified by the presence of certain cell types, while in vaccine research, the success of a vaccine may be inferred from the frequency of activated immune cells. Flow cytometry and mass cytometry are technologies used to identify cell populations by measuring the expression of characteristic proteins on cell surfaces or within cells. With the latest advances, dozens of proteins can be measured for each cell, precisely characterizing the diversity of cell types present. However, this leads to challenges in data analysis. Recently, significant efforts have been made to develop automated analysis methods, many of which rely on clustering methods to detect groups of similar cells. Here, we have compared the performance of available clustering methods, using several publicly available data sets as benchmarks. The results provide guidance to researchers trying to decide which method to use when analyzing data sets. By helping researchers choose appropriate data analysis methods, this work will improve confidence in reported experimental results, and ultimately contribute to broader adoption of these state-of-the-art technologies for applications in biology and medicine.

Introduction

Single-cell analysis techniques are frequently used to study cell populations, which may be defined as groups of cells with consistent phenotypic or functional characteristics. The identification or characterization of cell populations is a key analysis step in a broad range of studies in biology and biomedicine. Examples of design settings include: identification of known cell populations of interest in a biological sample, such as disease biomarkers; characterization of previously unknown cell subpopulations; and differential abundance analysis of cell populations between samples in different biological conditions, such as diseased and healthy.

Flow cytometry (fluorescence activated cell sorting, or FACS) is a widely used technology for measuring expression levels of surface and intracellular proteins at the individual cell level. In many cases, cell populations can then be defined according to their protein expression profiles. In the field of immunology, this procedure has seen enormous success over the past several decades, for example leading to a detailed framework of cellular differentiation during hematopoiesis [1].

Modern multicolor flow cytometers are capable of measuring up to 10-15 parameters per cell (in some cases up to 20), at a throughput rate of greater than 10,000 cells per

second [2]. The resulting multidimensional data sets allow cell populations to be characterized in great detail. However, the number of parameters (protein expression levels) that can be measured per cell is ultimately limited by technical characteristics of flow cytometry, in particular spectral overlap and autofluorescence [2].

Mass cytometry (also known as cytometry by time-of-flight, or CyTOF) allows a greater number of parameters to be measured per cell [3]. In mass cytometry experiments, antibodies are labeled with elemental isotopes selected from the transition metals, which are rare in normal biological systems, instead of the fluorescent molecules used in flow cytometry. Cells are then atomized and ionized in a high-temperature plasma and passed through a time-of-flight mass spectrometer, which detects the presence or absence of the metal tags for each cell. By using rare metal isotopes instead of fluorescent molecules to tag antibodies, mass cytometry largely eliminates the problems of spectral overlap and autofluorescence. However, throughput rates are somewhat lower than in flow cytometry, and it is not possible to collect cells after an experiment, since they are destroyed during ionization. Currently, mass cytometry systems are capable of measuring around 40 parameters per cell, at throughput rates of hundreds of cells per second [3].

Using multicolor flow cytometry or mass cytometry systems, experimentalists can now routinely create large data sets (up to millions of cells) with high dimensionality (dozens of parameters). This has resulted in bottlenecks during data analysis. Traditionally, flow cytometry data sets have been analyzed by "manual gating", consisting of visual examination of two-dimensional scatterplots to identify defined cell populations. However, manual gating may be insufficient for high-dimensional data sets for several reasons: there are too many possible two-dimensional projections to reliably analyze; any multidimensional structure in the data is ignored; it is difficult to detect previously unknown cell populations; and the analysis can be highly subjective. Therefore, major efforts have been made to partially or fully automate the data analysis procedures [4].

A key step in many automated analysis pipelines is the use of clustering tools to automatically detect or define cell populations. However, due to the "curse of dimensionality", standard clustering techniques such as k-means and hierarchical clustering generally do not perform well in high-dimensional data spaces [5]. In order to address this, several research groups have developed specialized clustering methods designed specifically for high-dimensional flow and mass cytometry data sets. These methods take a variety of theoretical approaches to solve the problems of high dimensionality, with the common aim of detecting distinct populations of cells corresponding to known biology. Several recent studies have provided overviews of available clustering methods [6, 7], and performance comparisons using standardized benchmark data sets [8–10] (including the FlowCAP challenges [10]).

In this study, we carry out an up-to-date, comprehensive, extensible performance comparison of clustering methods for high-dimensional flow and mass cytometry data, including several new methods that were not yet available at the time of the FlowCAP challenges. To evaluate the methods, we have selected four publicly available data sets with moderate to high dimensionality, where cell population identities are known from manual gating. The data sets include major immune cell populations as well as specific rare populations, in well-characterized biological systems where manual gating is likely to be reliable. Unlike the FlowCAP comparisons, we do not include low-dimensional flow cytometry data sets, since we are interested in the performance of clustering algorithms in high-dimensional settings. Our aim is to provide guidance to researchers faced with decisions about which clustering tools to use when analyzing new data sets from high-dimensional cytometry experiments. Code and preprocessed data files are available from GitHub (https://github.com/lmweber/cytometry-clustering-comparison) and FlowRepository (repository FR-FCM-ZZPH), allowing our analyses to be easily reproduced or extended to include new clustering methods and reference data sets.

Methods

Clustering methods

We compared a total of 13 clustering methods. The methods are listed in Table 1, along with key information and references to original publications. Most of the methods are designed specifically for analyzing high-dimensional cytometry data, however some (`FLOCK`, `flowMeans`, and `SamSPECTRAL`) were originally developed for lower-dimensional flow cytometry data, but have been found to still perform well in higher-dimensional settings. Two of the methods (`FlowSOM_meta` and `immunoClust_all`) are variations of others in the list. We have also included `k-means` as a basic benchmark. Additional details including package versions and parameter settings are included in Supp Table S1.

We included only methods with freely available software implementations, since our aim is to provide practical recommendations to researchers performing data analyses. The methods are based on a range of different theoretical approaches, which are briefly described in Table 1. For comprehensive explanations of the methodological approaches, we refer to the original references for each method (Table 1).

Table 1. Overview of clustering methods compared in this study.

Method	Environment and availability	Short description	Ref.
ACCENSE	Standalone application with graphical interface	Nonlinear dimensionality reduction (t-SNE) followed by density-based peak-finding and clustering in two-dimensional projected space. Subsampling recommended for large data sets.	[11]
DensVM	R package (cytofkit) from Bioconductor	Initial steps similar to ACCENSE, followed by support vector machine to classify remaining uncertain points. Subsampling recommended for large data sets.	[12]
FLOCK	C source code (also available in ImmPort online platform)	Partitioning of each dimension into bins, followed by merging of dense regions, and density-based clustering.	[13]
flowMeans	R package from Bioconductor	Based on k-means, with merging of clusters to allow non-spherical clusters.	[14]
FlowSOM	R package from Bioconductor	Self-organizing maps (a type of neural network with a single layer of nodes arranged in a topology-preserving, two-dimensional lattice structure).	[15]
FlowSOM_meta	R package from Bioconductor	Same as FlowSOM, with an additional "meta-clustering" step using hierarchical clustering to merge clusters.	[15]
immunoClust	R package from Bioconductor	Iterative clustering based on finite mixture models, using expectation maximization and integrated classification likelihood. Uncertain cells are removed.	[16]
immunoClust_all	R package from Bioconductor	Same as immunoClust, with additional step to classify uncertain cells.	[16]
k-means	R base packages (stats)	Standard k-means clustering.	
PhenoGraph	Application with graphical interface (cyt) launched from MATLAB (Python implementation also available)	Construction of nearest-neighbor graph, followed by partitioning of the graph into sets of highly interconnected points ("communities").	[9]
Rclusterpp	R package from CRAN	Large-scale implementation of standard hierarchical clustering, with improved memory requirements.	[17]
SamSPECTRAL	R package from Bioconductor	Spectral clustering, with modifications for improved memory requirements.	[18]
SWIFT	Application with graphical interface launched from MATLAB	Iterative fitting of Gaussian mixture models by expectation maximization, followed by splitting and merging of clusters using a unimodality criterion.	[19]

Additional details including package versions and parameter settings used for each clustering method are provided in Supp Table S1.

Table 2. Summary of data sets used to evaluate clustering performance.

Data set	Flow or mass cytometry	Clustering task	No. of cells	No. of dimensions	No. of manually gated cell populations of interest	No. of manually gated cells	Biological sample description	Ref.
Levine_2015_marrow_32	Mass cytometry	Detection of major immune cell populations	265,627	32 surface marker proteins	14	104,184 (39%)	Healthy human bone marrow mononuclear cells (BMMCs) from 2 individuals	[9]
Levine_2015_marrow_13	Mass cytometry	Detection of major immune cell populations	167,044	13 surface marker proteins	24	81,747 (49%)	Healthy human BMMCs from 1 individual	[9]
Nilsson_2013_HSC	Flow cytometry	Detection of a single rare cell population	44,140	13 surface marker proteins	1 (hemato-poietic stem cells, HSCs)	358 (0.8%)	Healthy human BMMCs from 1 individual	[20]
Mosmann_2014_activ	Flow cytometry	Detection of a single rare cell population	396,460	14 surface and intra-cellular proteins	1 (activated cytokine-producing memory CD4 T cells)	109 (0.03%)	Healthy human peripheral blood mononuclear cells (PBMCs) from 1 individual, exposed to influenza antigens	[19]

Data sets

We evaluated the performance of the clustering methods on four publicly available data sets, from experiments in immunology using mass cytometry or high-dimensional flow cytometry. We selected these data sets to allow testing of two distinct clustering tasks: detection of all major immune cell populations (data sets `Levine_2015_marrow_32` and `Levine_2015_marrow_13`); and detection of a single rare cell population of interest (data sets `Nilsson_2013_HSC` and `Mosmann_2014_activ`). Key characteristics of the data sets are summarized in Table 2.

For each of these data sets, manually gated cell population identities are available, either directly within the original data files or easily reproducible from gating diagrams published in the original papers. We have used these manually gated cell population labels as the "truth" against which the automated clustering methods can be evaluated. All of the data sets are from well-characterized biological systems, where manual gating is likely to be reliable despite the high dimensionality. However, due to the limitations of manual gating, labels are only available for a subset of cells in each data set. For `Levine_2015_marrow_32` and `Levine_2015_marrow_13`, the authors note that 61% and 51% of cells remain unassigned [9]. For `Nilsson_2013_HSC` and `Mosmann_2014_activ`, we use only labels for the rare cell population of interest. In each case, we run the clustering algorithms

on all cells (i.e. including unassigned cells), and subsequently evaluate performance on the cells with manually gated population labels.

The data sets were distributed by the original authors through FlowRepository [21] and Cytobank [22]. Repository identifiers and links to experiment data pages can be found in the original publications referenced in Table 2. Note that the data sets `Levine_2015_marrow_32` and `Levine_2015_marrow_13` are referred to as "benchmark data set 2" and "benchmark data set 1" in the respective publication [9].

Data preprocessing and availability

For the mass cytometry data sets (`Levine_2015_marrow_32` and `Levine_2015_marrow_13`), we first combined the original FCS files into one file per data set and added the manually gated population labels. Next, we applied an arcsinh transformation with a standard cofactor of 5 for mass cytometry (i.e. $arcsinh(x/5)$) [3]. We did not include any additional standardization or normalization of dimensions. This was unnecessary since the arcsinh calculation already transforms all dimensions to comparable scales; and in addition, since we perform clustering using all dimensions, unnecessary standardization of dimensions that do not contain a strong signal could amplify the effect of noise, adversely affecting clustering performance. We also did not include any additional normalization between samples, since the authors of the data sets already performed sample normalization (and the second data set contains only a single sample) [9].

For the flow cytometry data sets (`Nilsson_2013_HSC` and `Mosmann_2014_activ`), additional preprocessing was required. We performed pre-gating in Cytobank to exclude doublets, debris, and dead cells, following the published gating hierarchies displayed in Figure 2 in [20] and Figure 4A in [19]. The remaining single, live cells were kept for further analysis. Compensation was performed automatically in Cytobank using the spillover matrices provided with the original FCS files. Next, we generated cell population labels for the rare populations of interest by reproducing the remainder of the published gating hierarchies (Figure 2 in [20] and Figure 4A in [19]). Finally, we applied an arcsinh transformation with a standard cofactor of 150 for flow cytometry (i.e. $arcsinh(x/150)$) [3], and exported a new FCS file for each data set. As previously, we did not include any additional standardization or normalization of dimensions. Additional normalization between samples was also not required, since each data set contains only a single sample.

Our preprocessed data files are available from FlowRepository (repository identifier FR-FCM-ZZPH). Manually gated cell population labels are included within the FCS files. For each data set, we provide files with and without the arcsinh transformation, since some

Table 3. Number of clusters selected for each method.

Method	Data set				Selection options		
	Levine_32	Levine_13	Nilsson	Mosmann	Automatic	Indirect parameters	Manual
ACCENSE	38	41	52	47	✔	✔*	
DensVM	13	10	13	10	✔		
FLOCK	32	20	22	9	✔		
flowMeans	40	40	40	40	✔		✔*
FlowSOM	100	100	100	400			✔
FlowSOM_meta	40	40	40	40	✔		✔*
immunoClust	174	136	49	91	✔	✔*	
immunoClust_all	174	136	49	100	✔	✔*	
k-means	40	40	40	40			✔
PhenoGraph	32	26	29	26	✔*	✔	
Rclusterpp	40	40	40	40			✔
SamSPECTRAL	22	14	20	14	✔	✔*	
SWIFT	1553	347	95	144	✔		

Options available to select the number of clusters for each method are indicated with check marks (✔). Stars indicate the option used if more than one option is available (✔*). Additional details including the number of clusters for all available options are provided in Supp Table S1.

clustering algorithms apply automatic transformations. The gating hierarchies used to select the rare cell populations in the `Nilsson_2013_HSC` and `Mosmann_2014_activ` data sets are reproduced in Supp Figs S1–S2.

Number of clusters and other parameter settings

The number of clusters is a key parameter for many clustering methods. Some methods provide an option to select this number automatically, while others leave it as an input for the user. Many methods with an automatic option also allow the user to adjust the number indirectly by changing related parameters.

We used the automatic choice for methods where this was available and gave reasonable results, and 40 clusters for each data set if a manual input was required. For methods where an automatic choice was available but did not perform well, we adjusted parameters to give a number close to 40. However, since several methods only allowed the number to be adjusted through indirect parameters, it was not always possible to obtain exactly 40 clusters. For some methods, it was not possible to adjust the automatic number at all. The choice of 40 clusters was designed to be conservative, in the sense of selecting too many clusters rather than too few. All four data sets contain fewer than 40 true (manually gated) populations

(Table 2); selecting fewer clusters tended to merge important clusters, especially for the data sets containing rare cell populations.

Table 3 summarizes the number of clusters used for each method, and the options available to select the number. For methods where more than one option is available, the option we used for the main results is indicated. For three methods (`DensVM`, `FLOCK`, `SWIFT`), it was not possible to adjust the automatic number of clusters, even if it was sub-optimal. Some methods are designed to return large numbers of clusters (`immunoClust`, `immunoClust_all`, `SWIFT`); so we did not attempt to reduce these (see Discussion). Similarly, the large number of clusters from `FlowSOM` are intended to be merged in a separate "meta-clustering" step, which we have labeled as the method `FlowSOM_meta` (see Table 1).

Additional details including the number of clusters for all available options, as well as all other parameter settings for each method, are provided in Supp Table S1.

Evaluation criteria

We evaluated the clustering methods using F1 scores, precision, and recall. For a given true (manually gated) cell population and a corresponding detected cluster from an algorithm, the precision measures the proportion of cells in the detected cluster that are from the correct true population (i.e. avoiding false positives); the recall (or sensitivity) measures the proportion of cells from the true population that are assigned to the detected cluster (i.e. avoiding false negatives); and the F1 score is the harmonic mean of precision and recall.

For the data sets with multiple cell populations of interest (`Levine_2015_marrow_32` and `Levine_2015_marrow_13`), we calculated unweighted means of the F1 score, precision, and recall across true populations. The use of unweighted means ensures that large and small populations are represented equally. (Additional results with weighting by true population size are included in Supp Figs S3–S4.) For the data sets with a single rare population of interest (`Nilsson_2013_HSC` and `Mosmann_2014_activ`), we used the F1 score, precision, and recall for the rare population. In all evaluations, we used the manually gated population labels as the "truth" against which the F1 scores, precision, and recall were calculated. Clusters were matched to true populations by maximizing the F1 score; in other words, for each true population, we assigned the cluster giving the highest F1 score.

We investigated the protein expression profiles of detected clusters by comparing them against the true (manually gated) populations using hierarchical clustering. In addition, we recorded runtimes for each method, and evaluated the stability of the clustering algorithms by running them multiple times with different random starts.

Ensemble clustering

We performed ensemble clustering (consensus clustering) to test whether performance may be improved by aggregating results from multiple clustering algorithms, as in previous comparisons (for example, the FlowCAP challenges [10]). We used the `clue` R package [23] to combine clustering results from the top five methods for each data set, according to mean F1 score or F1 score in the main results. However, the following methods were excluded: methods that required subsampled data (`ACCENSE`, `DensVM`, `Rclusterpp`) or removed outliers (`ACCENSE`, `immunoClust`, `SamSPECTRAL`), since consensus clustering requires the same data for each method; and methods that returned large numbers of clusters (`FlowSOM`, `immunoClust`, `immunoClust_all`, `SWIFT`), since this greatly increased runtime. For the `Nilsson_2013_HSC` data set, the small size of the data set (see Table 2) meant that runtime remained fast even with large numbers of clusters, so these methods were not removed. Methods that were excluded were replaced with the next lower-ranked methods, so that five methods were still included for each data set.

Stability across random starts

To evaluate the stability of the clustering algorithms across random starts, we ran methods 30 times with different random seeds, generating empirical distributions of the main evaluation criteria (F1 score, precision, and recall) as well as runtime. Where possible, we ran the methods in parallel on our multi-core server. We restricted the stability analysis to methods that were available as R packages or command-line programs to allow scripting; this excluded `ACCENSE` and `SWIFT`, which could only be accessed through graphical interfaces. For `PhenoGraph`, we used the alternative Python command-line implementation (with multiple processor cores) instead of the graphical interface (see Table 1). We also excluded `DensVM`, which was difficult to parallelize since results are written to output files that may be overwritten; and `Rclusterpp`, which required too many processor cores for each instance.

Results

Detection of major cell populations

Fig 1 summarizes the results of the comparison of clustering methods for data set `Levine_2015_marrow_32`. This is a 32-dimensional mass cytometry data set, where we are interested in the ability to detect 14 major immune cell populations (see Methods and Table 2). Panel A shows the mean F1 score across the 14 populations, which we use as

our main evaluation criterion. The best-performing method in terms of mean F1 score is `FlowSOM_meta`, followed by `flowMeans`, `FLOCK`, and `PhenoGraph`. Panel B provides additional detail on the distribution of F1 scores across populations; further results according to population size are provided in Supp Fig S5, revealing that among the top-performing methods, the lowest F1 scores tend to occur for the smallest populations. Panel C displays the mean precision and mean recall, showing that the relatively poor performance of `immunoClust` and `SWIFT` is due to low recall, while `Rclusterpp` suffers from low precision. Panel D displays the true (manually gated) population sizes, which range from 304 to 26,366 of the 265,627 total cells. Panel E summarizes the runtimes, which vary over several orders of magnitude between methods. Apart from `k-means`, `FlowSOM` and `FlowSOM_meta` have the fastest runtimes, followed by `FLOCK`. Two methods (`Rclusterpp` and `SWIFT`) make use of multiple processor cores (the number of cores for each method is included in Supp Table S1). Panel F plots runtime against mean F1 score, where the best-performing methods can be seen in the bottom-right corner, combining fast runtimes with high mean F1 scores.

Fig 2 displays the same results for data set `Levine_2015_marrow_13`, which is a 13-dimensional mass cytometry data set (see Methods and Table 2). Compared to the first data set, this data set is lower-dimensional, and cells are divided into a greater number of major immune cell populations (24 instead of 14). The populations also span a larger range of relative population sizes (Panel D). We are interested in the ability of clustering algorithms to detect all 24 major immune cell populations. The top method in terms of mean F1 score (Panel A) is again `FlowSOM_meta`, followed by `PhenoGraph`, `flowMeans`, and `k-means`. However, for all methods, there are some populations with very low F1 scores (Panel B); among the top-performing methods, the lowest F1 scores tend to occur for the smallest populations (Supp Fig S6). As previously, the poor performance of `immunoClust` and `SWIFT` is due to low recall, while `Rclusterpp` has low precision (Panel C). The fastest runtimes are observed for `k-means`, `FlowSOM`, `FlowSOM_meta`, and `FLOCK` (Panel E). In Panel F, the best-performing methods can again be seen toward the bottom-right, combining fast runtimes with high mean F1 scores.

To further investigate the quality of the clustering results, we used heatmaps to compare the protein expression profiles of detected clusters and true (manually gated) populations. Fig 3 displays an example of these results, for method `FlowSOM_meta` with data set `Levine_2015_marrow_32`. The heatmap shows the median expression intensities of each protein marker (in columns), for each detected cluster or true population (in rows). Values are arcsinh-transformed, and scaled between 0 and 1 for each protein marker. Rows and columns are sorted by hierarchical clustering (Euclidean distance, average linkage), and rows are labeled to indicate whether they represent true populations or detected clusters. We observe

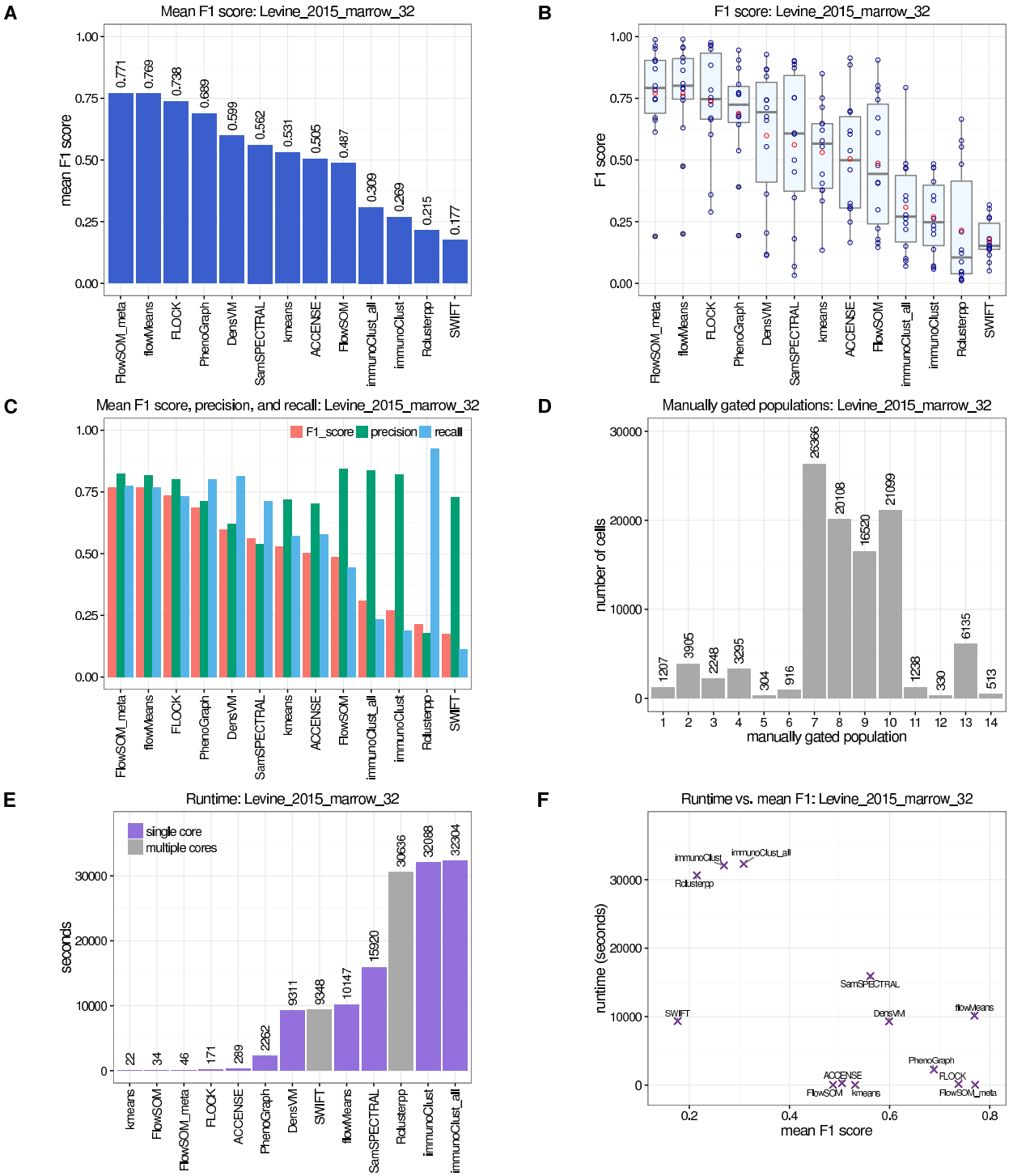

Figure 1. Results of comparison of clustering methods for data set Levine_2015_marrow_32. (A) Mean F1 score across cell populations. (B) Distributions of F1 scores across cell populations. The box plots show medians, upper and lower quartiles, whiskers extending to 1.5 times the interquartile range, and outliers, with means shown additionally in red. (C) Mean precision and mean recall, together with mean F1 scores. (D) Number of cells per true population. (E) Runtimes, with methods that make use of multiple processor cores indicated in gray. (F) Runtime vs. mean F1 score, where the best-performing methods combining fast runtimes with high mean F1 scores are seen toward the bottom-right.

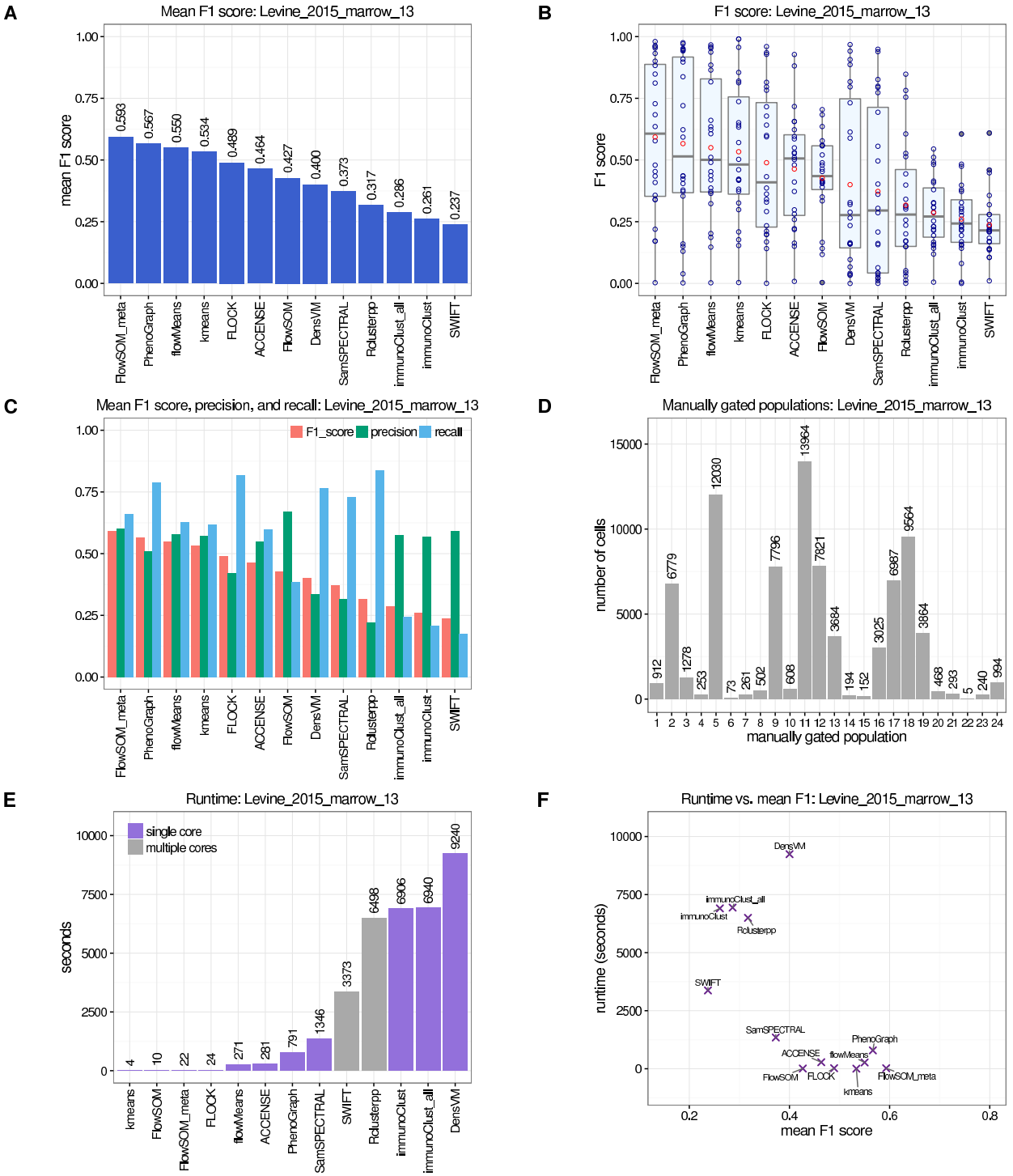

Figure 2. Results of comparison of clustering methods for data set Levine_2015_marrow_13. (A) Mean F1 score across cell populations. (B) Distributions of F1 scores across cell populations. The box plots show medians, upper and lower quartiles, whiskers extending to 1.5 times the interquartile range, and outliers, with means shown additionally in red. (C) Mean precision and mean recall, together with mean F1 scores. (D) Number of cells per true population. (E) Runtimes, with methods that make use of multiple processor cores indicated in gray. (F) Runtime vs. mean F1 score, where the best-performing methods combining fast runtimes with high mean F1 scores are seen toward the bottom-right.

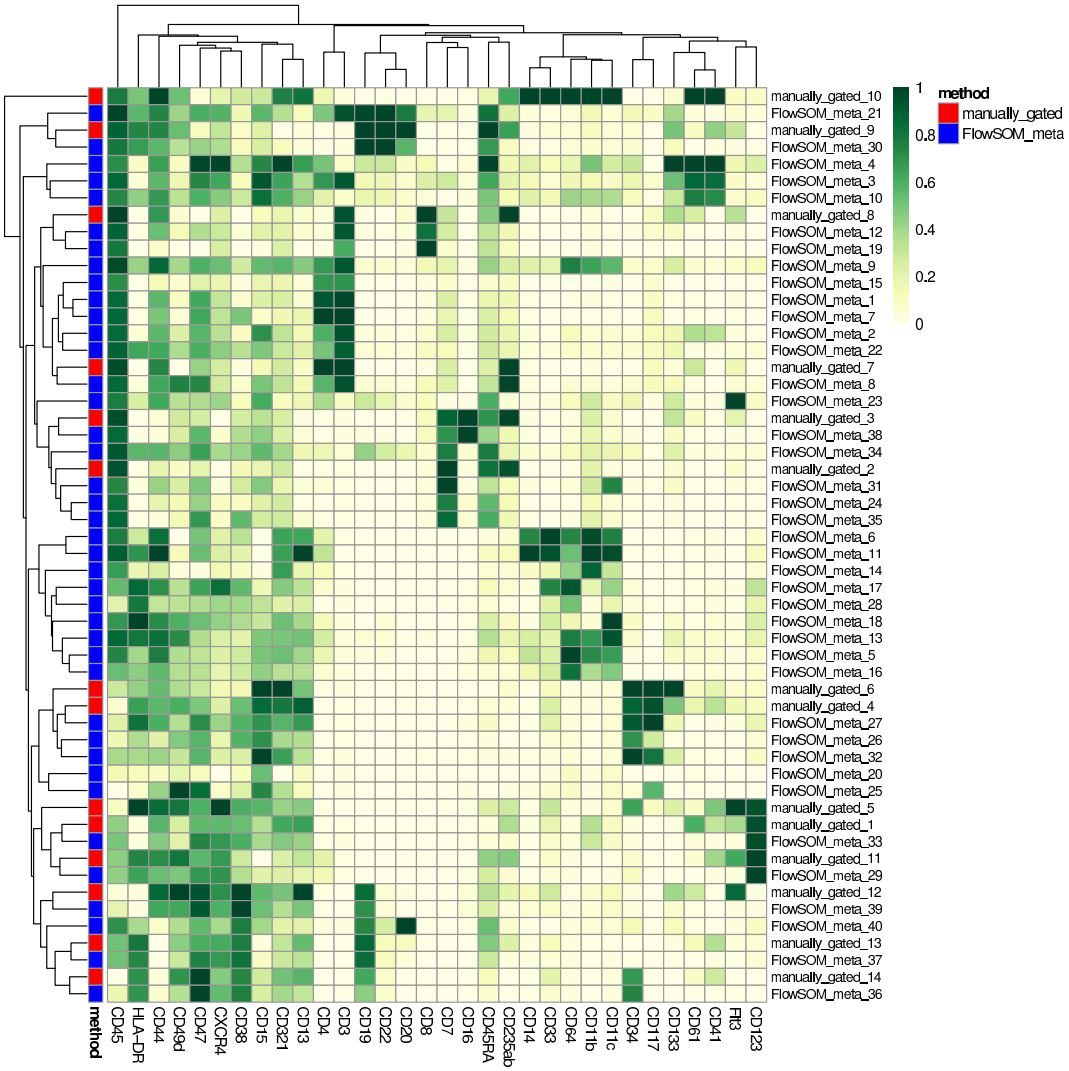

Figure 3. Median expression intensities of clusters detected by FlowSOM_meta and true (manually gated) populations, data set Levine_2015_marrow_32. Heatmap shows median expression intensities of each protein marker (columns), for each detected cluster or true population (rows). Values are arcsinh-transformed, and scaled between 0 and 1 for each protein marker. Rows and columns are sorted by hierarchical clustering (Euclidean distance, average linkage). Cluster and population indices are included in the row headings. Red labels indicate rows representing true (manually gated) populations, and blue labels indicate rows representing clusters detected by `FlowSOM_meta`. Additional figures for all other clustering methods as well as the second data set are provided in Supp Figs S15–S42.

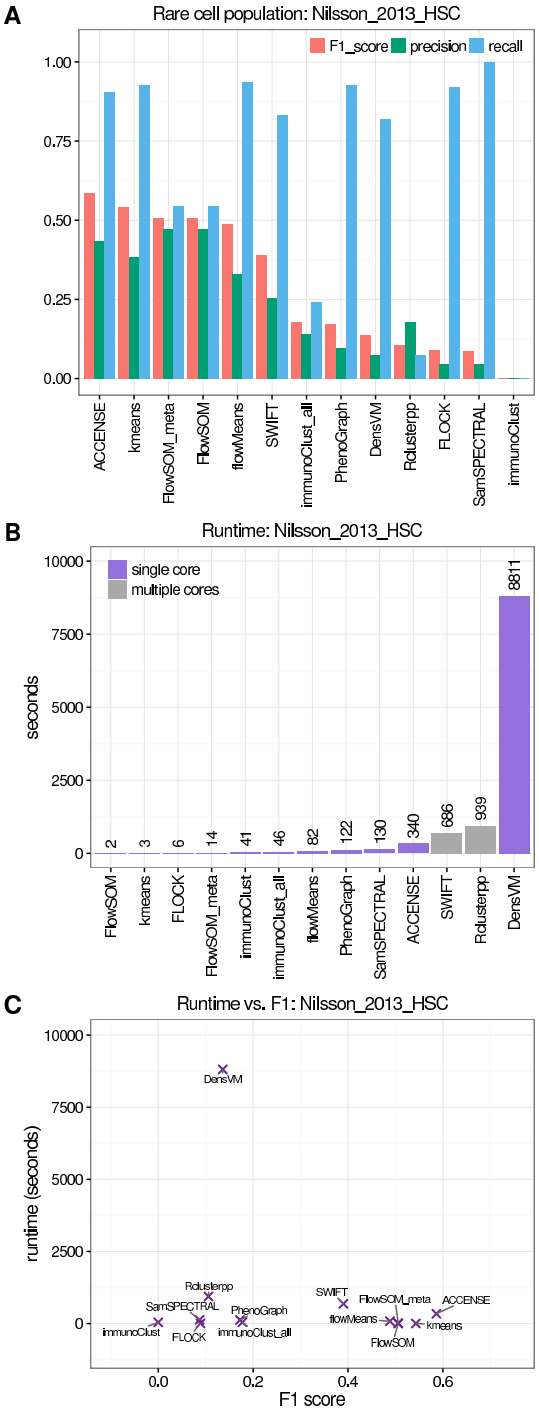

Figure 4. Results of comparison of clustering methods for data set Nilsson_2013_HSC. (A) F1 score, precision, and recall for the rare cell population of interest. Clustering methods are ordered by decreasing F1 score. The rare cell population consists of hematopoietic stem cells (HSCs), and contains 358 manually gated cells, or around 0.8% of total cells in the data set. (B) Runtimes, with methods that make use of multiple processor cores indicated in gray. (C) Runtime vs. F1 score, where the best-performing methods combining fast runtimes with high F1 scores are seen toward the bottom-right.

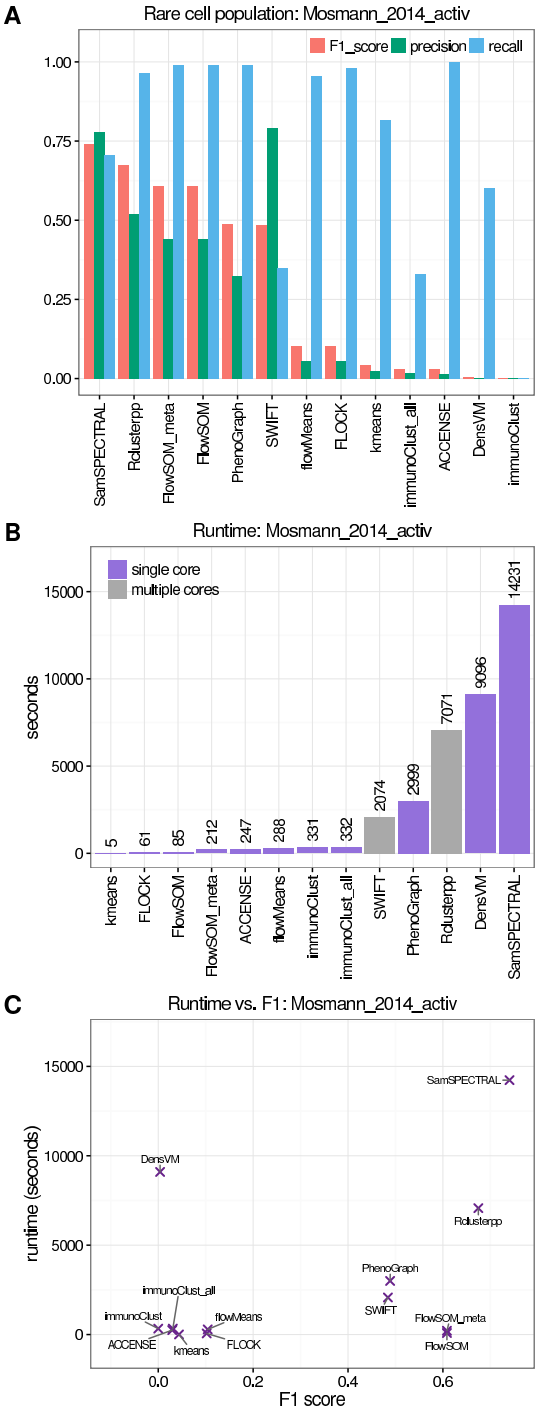

Figure 5. Results of comparison of clustering methods for data set Mosmann_2014_activ. (A) F1 score, precision, and recall for the rare cell population of interest. Clustering methods are ordered by decreasing F1 score. The rare cell population consists of activated (cytokine-producing) memory CD4 T cells, from a sample of healthy human peripheral blood mononuclear cells (PBMCs) exposed to influenza antigens. The rare population contains manually gated 109 cells, or around 0.03% of total cells in the data set. (B) Runtimes, with methods that make use of multiple processor cores indicated in gray. (C) Runtime vs. F1 score, where the best-performing methods combining fast runtimes with high F1 scores are seen toward the bottom-right.

that for most of the true populations (red rows), at least one cluster detected by `FlowSOM_meta` (blue rows) matches closely to the expression profile of the true population. This indicates that `FlowSOM_meta` has correctly detected clusters corresponding to these populations, with some additional splitting of populations. However, the separation and matching is not perfect; some true populations (red rows) group together before matching to any detected clusters. For this data set, `FLOCK`, `immunoClust`, and `immunoClust_all` achieve the smallest number of mismatches (Supp Figs S18 and S22–S23). For data set `Levine_2015_marrow_13`, which contains a greater number of true populations, `FlowSOM` achieves the smallest number of mismatches (Supp Fig S34). Additional figures for all clustering methods and both data sets are provided in Supp Figs S15–S42.

Detection of a single rare cell population

In Fig 4, we present results for data set `Nilsson_2013_HSC`. This is a 13-dimensional flow cytometry data set, where we are interested in the ability to detect a single rare cell population. The rare population consists of hematopoietic stem cells (HSCs), and contains 358 manually gated cells, or around 0.8% of total cells in the data set (see Methods and Table 2). Our main evaluation criteria are the F1 score, precision, and recall for the rare cell population. These results are displayed in Panel A, with clustering methods ordered by decreasing F1 score. The best-performing methods in terms of F1 score are `ACCENSE` and `k-means`, followed by `FlowSOM_meta` and `FlowSOM`. Several of the lower-ranked methods achieve high recall (sensitivity) but low precision. Runtimes are summarized in Panel B; for this data set, `DensVM` is considerably slower than all other methods. Panel C plots runtime against F1 score, with methods that achieve both high F1 scores and fast runtimes shown toward the bottom-right.

Fig 5 displays the results for data set `Mosmann_2014_activ`. This is a 14-dimensional flow cytometry data set, where the interest is again in detecting a single rare cell population. Here, the rare population consists of activated (cytokine-producing) memory CD4 T cells, from a sample of healthy human peripheral blood mononuclear cells (PBMCs) exposed to influenza antigens. Compared to the previous data set, the rare population represents a much smaller proportion (109 manually gated cells, or around 0.03%), and the data set contains a greater total number of cells (see Methods and Table 2). Panel A displays the F1 score, precision, and recall for the rare population, with methods ordered by decreasing F1 score. For this data set, the best-performing methods in terms of F1 score are `SamSPECTRAL` and `Rclusterpp`, followed by `FlowSOM_meta` and `FlowSOM`. `PhenoGraph` and `SWIFT` also perform well; after these, there is a distinct drop-off in performance for the remaining methods. `SWIFT` achieves the highest precision, but relatively low recall. Panel B reveals that the

two top-performing methods (`SamSPECTRAL` and `Rclusterpp`) have very slow runtimes, with `Rclusterpp` also requiring multiple processor cores; by contrast, `FlowSOM_meta` and `FlowSOM` have among the fastest runtimes. In Panel C, methods that combine high F1 scores with fast runtimes can again be seen toward the bottom-right.

Optimal number of clusters

For methods where a manual selection of the number of clusters was required, we used 40 clusters for each data set (see Methods and Table 3). To confirm that this was an appropriate choice, we performed additional analysis to calculate results over a range of values for the number of clusters, using `FlowSOM_meta`. Figure 6 displays the mean F1 score (data sets with multiple populations) or F1 score (data sets with a single rare population), across a range of values for the number of clusters. The results confirm that 40 clusters (marked in red) is an optimal choice for this method across all data sets.

Ensemble clustering

Results of the ensemble (consensus) clustering are displayed in Supp Figs S7–S10. Unlike previous studies (for example, the FlowCAP challenges [10]), ensemble clustering did not give any improvements in performance compared to the best-performing individual methods. For each data set, the top individual method achieves a higher mean F1 score across populations (`Levine_2015_marrow_32` and `Levine_2015_marrow_13`) or F1 score for the rare population (`Nilsson_2013_HSC` and `Mosmann_2014_activ`) than the ensemble clustering.

Stability across random starts

Results of the stability analysis are displayed in Fig 7 (`Mosmann_2014_activ`) and Supp Figs S11–S14 (all data sets; including runtimes). Each clustering method was run 30 times with different random starts to generate distributions of the evaluation criteria (see Methods). For the data sets with multiple populations (`Levine_2015_marrow_32` and `Levine_2015_marrow_13`; Supp Figs S11–S12), the variation in mean F1 scores, mean precision, and mean recall is relatively small for all methods. Several methods show zero variation, which in some cases is due to the methods automatically assigning fixed random seeds. The variation in runtimes is small for all methods.

However, for the data sets with a single rare population (`Nilsson_2013_HSC` and `Mosmann_2014_activ`; Fig 7 and Supp Figs S13–S14), we observe significant variation in F1 scores, precision, and recall. This includes several of the top-performing methods for

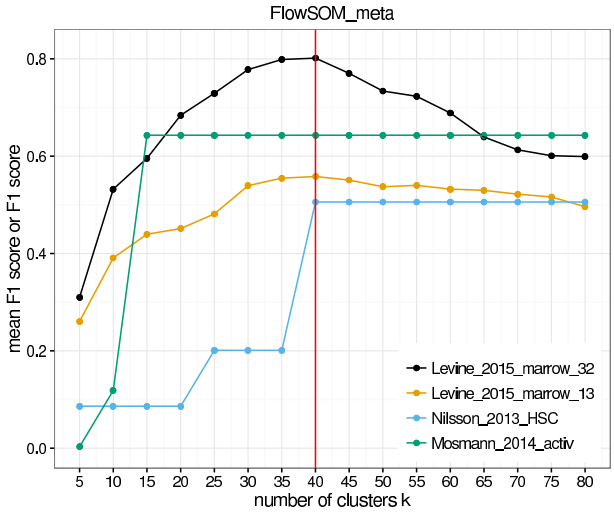

Figure 6. Optimal choice of number of clusters for FlowSOM_meta. Mean F1 scores (data sets with multiple populations; `Levine_2015_marrow_32` and `Levine_2015_marrow_13`) and F1 scores (data sets with a single rare population; `Nilsson_2013_HSC` and `Mosmann_2014_activ`) across a range of values for the number of clusters k. The number of clusters was varied between 5 and 80, in steps of 5. The optimal choice of 40 clusters for this method is indicated with a red vertical line.

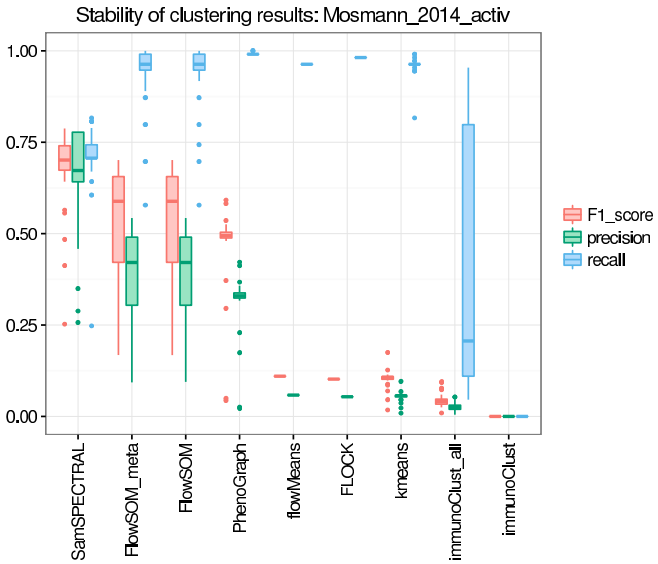

Figure 7. Stability of clustering results across random starts for data set Mosmann_2014_activ. Distributions of F1 scores, precision, and recall for the rare cell population in data set `Mosmann_2014_activ`, generated by running each clustering method 30 times with different random starts. The box plots show medians, upper and lower quartiles, whiskers extending to 1.5 times the interquartile range, and outliers. Additional results for all data sets (including runtimes) are provided in Supp Figs S11–S14.

these data sets, including SamSPECTRAL, FlowSOM_meta, FlowSOM, PhenoGraph, and k-means. In particular, there are a number of outlier runs for these methods, where performance is considerably better or worse than average. Some methods again show zero variation. The variation in runtimes is highest for PhenoGraph and SamSPECTRAL (for the Nilsson_2013_HSC data set), and small for all other methods.

Discussion

The automated detection of clusters representing cell populations in high-dimensional flow and mass cytometry data sets is a difficult computational task, since standard clustering techniques generally do not perform well in high-dimensional data spaces. In this study, we have compared the performance of a range of leading clustering methods for high-dimensional cytometry data (Table 1). We evaluated the methods using four publicly available data sets from experiments in immunology (Table 2). The data sets were selected to test two distinct clustering tasks: detection of all major cell populations, and detection of individual rare populations. The clustering methods include several new methods that were not yet available at the time of earlier comparisons, such as the FlowCAP challenges [10].

FlowSOM (with meta-clustering but without automatic number of clusters) gives superior performance and fast runtimes. Our comparisons showed that FlowSOM_meta was the best-performing method in terms of mean F1 score for the data sets with multiple cell populations (Levine_2015_marrow_32 and Levine_2015_marrow_13; Figs 1–2). FlowSOM_meta also achieved a good separation and matching between detected clusters and true populations (Fig 3). For the data sets with a single rare cell population of interest (Nilsson_2013_HSC and Mosmann_2014_activ), FlowSOM_meta was the third-best performing method in terms of F1 score for both data sets (Figs 4–5).

In addition, FlowSOM_meta had extremely fast runtimes, running in less than one minute for each of the first three data sets and less than four minutes for Mosmann_2014_activ; and required only a single processor core (Figs 1–2 and 4–5). This placed FlowSOM_meta among the fastest methods overall. By contrast, the slowest methods for each data set took several hours to complete. In particular, for Mosmann_2014_activ, the two methods that outperformed FlowSOM_meta in terms of F1 score (SamSPECTRAL and Rclusterpp) were extremely slow; Rclusterpp also required multiple processor cores and subsampling (see below) to complete in a reasonable amount of time.

The modeling approach taken by FlowSOM and FlowSOM_meta is based on self-organizing maps, followed by an optional "meta-clustering" step using hierarchical clustering

to merge clusters. In this study, we have used the name `FlowSOM_meta` to indicate that the meta-clustering step is included (Table 1). The implementation of the meta-clustering includes an option to select the final number of clusters automatically; however, we found that this performed relatively poorly. Since it was also possible to select the number manually, we instead used 40 clusters for each data set, the same number as for the methods where no automatic option was available (Table 3). We confirmed that the default choice of 40 clusters was appropriate by calculating additional results over a range of alternative values (Fig 6).

Based on these results, we recommend the use of `FlowSOM` with meta-clustering (but without automatic selection of the number of clusters) as a first choice for analyzing data sets from high-dimensional flow and mass cytometry experiments. In particular, the fast runtimes and minimal computational requirements make this method well-suited for analyzing large data sets, and make it possible to perform interactive, exploratory analyses on a standard laptop. The availability as an R/Bioconductor package (Table 1) also enables easy integration into analysis pipelines, including downstream statistical analysis and plotting.

Other high-performing methods. Several other clustering methods also performed well for these data sets. For the data sets with multiple cell populations, `flowMeans`, `FLOCK`, and `PhenoGraph` consistently performed well. However, runtimes for `flowMeans` and `PhenoGraph` were significantly slower than `FlowSOM_meta`. Among these, `flowMeans` is available as an R/Bioconductor package, while `PhenoGraph` includes a graphical user interface and convenient visualization tools within its associated MATLAB tool `cyt`, making it accessible for users without command-line or programming expertise. A Python command-line implementation of `PhenoGraph` is also available. `FLOCK` requires compilation from C source code, which is a difficult procedure for most users; however, an online implementation is also available in the ImmPort analysis platform (Table 1).

For the data sets with a single rare cell population, the sets of top methods differed between the two data sets. For `Nilsson_2013_HSC`, the two best-performing methods were `ACCENSE` and `k-means`; but these methods performed poorly for `Mosmann_2014_activ`. The two top methods for `Mosmann_2014_activ` were `SamSPECTRAL` and `Rclusterpp`, but these suffered from slow runtimes and heavy computational requirements, as mentioned above. `FlowSOM_meta` and `FlowSOM` were the third and fourth-ranked methods for both data sets. `PhenoGraph` and `SWIFT` also performed well for `Mosmann_2014_activ`; in particular, `SWIFT` achieved the highest precision for this data set. This may be an important consideration for some applications if precision (avoiding false positives) is judged to be more important than recall (sensitivity, or avoiding false negatives). The difference in rankings between these two data sets may be related to the relative sizes of the rare populations. For `Nilsson_2013_HSC`,

the rare population of hematopoietic stem cells accounted for 0.8% of total cells, while for `Mosmann_2014_activ`, the activated memory CD4 T cells represented only 0.03% (Table 2). In many cases, detecting a smaller proportion is a more difficult computational task, making the rankings for `Mosmann_2014_activ` more informative in practice.

Unlike previous comparisons (for example, the FlowCAP challenges [10]), ensemble clustering did not give any improvements in performance compared to the best-performing individual methods for each data set (Supp Figs S7–S10).

Parameter inputs to manually select the number of clusters are more important than automatic options. In many applications of clustering, the expected number of clusters is unknown in advance. Several clustering methods in this study included options to determine the number of clusters automatically, while others left it as an input for the user. We used the automatic numbers where available, and a consistent choice of 40 clusters for each data set if a manual selection was required. If the automatic choice did not perform well, we also attempted to use 40 clusters; however, for some methods, it was not possible to change the number, while for others it was only possible to adjust an indirect parameter (see Methods and Table 3).

We found that the automatic options frequently performed poorly. Several methods selected too few clusters, tending to merge important clusters, which negatively affected performance as well as the interpretability of the results. Overall, considerable fine-tuning was required to ensure that each method selected an adequate number of clusters. In general, while the ability to select the number of clusters automatically is an appealing theoretical feature of clustering algorithms, our experiences indicate that its absence is not a severe limitation for studies of this type. On the contrary, an inability to easily adjust the number of clusters was a significant limitation for several methods where the automatic options performed poorly (for example, `DensVM` and `FLOCK`; see Table 3).

In practice, it is often straightforward for experienced immunological researchers to make a reasonable initial choice for the number of clusters, and to merge multiple clusters from a known cell population during later computational analysis. An optimal strategy may be to generate somewhat too many clusters, to ensure that small populations are not merged into larger clusters; and subsequently merge some of the excess clusters manually. However, this strategy is only feasible if it is possible to manually specify the desired number of clusters. For studies of this type, a simple, direct parameter input to manually select the number of clusters is a highly desirable feature of clustering algorithms.

Unweighted averages ensure equal representation of large and small populations. For the data sets with multiple cell populations, we used unweighted means of the F1 score, precision, and recall across populations as our evaluation criteria. The use of unweighted averages ensures that large and small populations are represented equally. The relative population sizes in these data sets span several orders of magnitude (Figs 1D and 2D), so calculating averages weighted by population size would largely ignore the contribution of the smaller populations. This was especially a concern since we observed that, for many clustering methods, the F1 scores for individual populations were considerably worse for smaller populations than larger ones (Supp Figs S5–S6). Additional results with averages weighted by population size showed that, for many methods, the weighted mean F1 scores were higher than the unweighted ones, as expected due to the better performance among larger populations; this also influenced the final rankings of the methods (Supp Figs S3–S4).

Several clustering methods are sensitive to random starts when detecting rare cell populations. To investigate the stability of the clustering algorithms across random starts, we ran methods multiple times with different random seeds. For the data sets with a single rare population of interest, this revealed significant variability for several methods. In particular, we observed a number of outlier runs, where performance was considerably better or worse than average. The variability was large enough that the rankings of the top methods for these data sets would change significantly for certain random seeds. The strongly affected methods included several of the best-performing methods for these data sets, including `FlowSOM_meta`, `PhenoGraph`, and `SamSPECTRAL` (Fig 7 and Supp Figs S13–S14). By contrast, for the data sets with multiple populations of interest, variability was relatively minor (Supp Figs S11–S12). These results indicate that the sensitivity of clustering algorithms to random starts can be an important consideration, especially when the aim is to detect rare cell populations. In these cases, we recommend running clustering methods multiple times with different random starts.

Results depend on the choice of benchmark data sets. Throughout this study, we have used manually gated cell population labels as the "truth" against which the clustering methods were evaluated. All of the data sets we used are from well-understood biological systems, where manual gating is likely to be reliable despite the high dimensionality. However, the effectiveness of this evaluation strategy clearly depends on the accuracy and completeness of the manual gating. If the manually gated populations are poorly defined, it will be difficult to accurately determine clustering performance. In fact, the authors of the data sets `Levine_2015_marrow_32` and `Levine_2015_marrow_13` noted that 61% and 51% of cells in

these data sets were not assigned to any of the manually gated populations (see Methods and Table 2); [9]. For each data set, we ran clustering methods on all cells (including unassigned cells), to approximate typical analysis settings where prior knowledge about cell population identities is unavailable. Subsequently, we evaluated performance using the subset of cells with manually gated population labels. Potentially, inaccuracies in these labels, such as missing or incomplete populations or incorrectly labeled cells, could significantly affect the results. Despite this, the analysis for these data sets seems reliable, since the biological samples were sourced from healthy individuals and all major immune cell populations were present.

All four data sets used in this study were derived from real, experimental data sets. We did not include any synthetic data sets, due to the difficulty of designing accurate simulations in high-dimensional settings. Accurate synthetic data sets can be highly informative since the exact "truth" is known for every data point, allowing precise calculations of evaluation criteria such as the F1 score. Conversely, the advantage of real data sets is that they fully reflect the high-dimensional complexity of actual experimental settings. Since manually gated cell population labels are an accurate proxy for true labels in well-understood biological systems such as those considered here, we have chosen to focus on real data sets only. These four benchmark data sets cover a range of possible study designs in high-dimensional flow and mass cytometry, making our results generalizable for future experiments.

Some methods are designed for different or complementary analysis tasks. Throughout this study, we have compared methods by evaluating their performance on the task of fully automated detection of high-dimensional clusters representing cell populations. However, some methods are not strictly intended for performing fully automated analysis in this manner, or include additional or complementary analysis capabilities that we have not tested. For example, the authors of `SWIFT` describe a semi-automated analysis pipeline, where `SWIFT` is initially used to group cells into a large number of small clusters, and these clusters are then further analyzed by gating. This strategy is intended to be particularly effective for detecting rare populations [19], which is consistent with our results (Fig 5). Similarly, `immunoClust` is designed to return a relatively large number of clusters [16]. In our evaluations, `immunoClust` and `immunoClust_all` split large populations into multiple clusters, which negatively affects performance for these populations (Supp Figs S5–S6) as well as overall performance (Figs 1–2; see Supp Figs S22–S23 and S36–S37 for illustrations of the cluster splitting). In addition, `immunoClust` includes further analysis capabilities to compare cell populations between different biological samples, which is a central task in many experimental settings [16].

Three methods (ACCENSE, DensVM, and Rclusterpp) required the use of subsampling. For ACCENSE and DensVM, this is recommended by the authors [11, 12]; for Rclusterpp, it is not recommended directly but was required due to the heavy computational requirements (Figs 1–2 and 4–5). The numbers of subsampled cells we used for each data set are indicated in Supp Table S1. Although subsampling can be performed in a density-dependent manner to preserve smaller populations (as in ACCENSE, [11]), there remains a risk that for extremely rare populations, a significant proportion of cells may be discarded. Therefore, we do not recommend using these methods for detecting rare cell populations.

Similarly, several methods removed outliers or cells with uncertain cluster assignments (ACCENSE, immunoClust, and SamSPECTRAL). In the case of immunoClust, uncertain cells are removed by default, but can be classified in an additional step, which we have labeled as the separate method immunoClust_all (Table 1). This behavior explains why the F1 score for immunoClust is zero for the data sets with rare cell populations (Figs 4–5); the rare populations of interest are completely removed as uncertain cells. The additional classification step in immunoClust_all then assigns them to other populations. In general, as for subsampling, we do not recommend using these methods to detect rare cell populations. Despite this, we observed that ACCENSE performed very well for the Nilsson_2013_HSC data set (Fig 4), and SamSPECTRAL for the Mosmann_2014_activ data set (Fig 5), so there may be some cases where the removal of outliers does not significantly harm performance.

We also note that ACCENSE and DensVM rely on an initial nonlinear dimensionality reduction to two dimensions (using the t-SNE algorithm [24]), followed by the use of a density-based peak-finding algorithm to detect clusters in the projected space. Since the clustering effectively occurs in the two-dimensional projected space, this methodology may discard a significant amount of information, which could explain why these methods generally did not perform as well as some of the other approaches.

Recommendations. We have performed an up-to-date, comprehensive, extensible comparison of clustering methods for automated detection of cell populations in high-dimensional flow and mass cytometry data. We compared 13 clustering methods (11 distinct methods and two variations), using four publicly available data sets from experiments in immunology. Based on our results, we recommend the use of FlowSOM (with the optional meta-clustering step but without automatic selection of the number of clusters) as a first choice for analyzing data sets from mass cytometry and high-dimensional flow cytometry experiments. This method gave the best performance for the data sets with multiple cell populations, and was ranked third for the data sets with a single rare cell population of interest. In addition, runtimes were extremely fast and computational requirements minimal, making

`FlowSOM` well-suited for analyzing large data sets and performing interactive, exploratory analyses on a standard laptop. Other methods that performed well across several data sets, with reasonable runtimes, included `PhenoGraph`, `flowMeans`, and `FLOCK`. We also found that several clustering methods were sensitive to random starts when detecting rare cell populations, so for these data sets we recommend the use of multiple random starts to investigate variability. R scripts and preprocessed data files to reproduce our analyses are available from GitHub (https://github.com/lmweber/cytometry-clustering-comparison) and FlowRepository (repository FR-FCM-ZZPH), allowing our comparisons to be extended to include new clustering methods and reference data sets.

Supporting Information

S1 Supplementary Information

Supplementary Figures S1–S42. Document containing all supplementary figures referenced in the text.

S1 Table

Supplementary Table S1. Spreadsheet containing additional details on clustering methods.

Acknowledgments

We thank Charlotte Soneson, Stéphane Chevrier, Vito Zanotelli, Bernd Bodenmiller, Felix Hartmann, and members of the Robinson, Baudis, and von Mering labs at the University of Zurich for helpful discussions and feedback on earlier drafts of the manuscript. Figures were generated using the R packages `ggplot2`, `reshape2`, `ggrepel`, `cowplot`, `pheatmap`, `RColorBrewer`, and `colorspace`.

References

1. Orkin SH, Zon LI. Hematopoiesis: An evolving paradigm for stem cell biology. Cell. 2008;132(4):631–644.

2. Bendall SC, Nolan GP, Roederer M, Chattopadhyay PK. A deep profiler's guide to cytometry. Trends in Immunology. 2012;33(7):323–332.

3. Bendall SC, Simonds EF, Qiu P, Amir ED, Krutzik PO, Finck R, et al. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. Science. 2011;332:687–696.

4. Finak G, Frelinger J, Jiang W, Newell EW, Ramey J, Davis MM, et al. OpenCyto: An open source infrastructure for scalable, robust, reproducible, and automated, end-to-end flow cytometry data analysis. PLOS Computational Biology. 2014;10(8):e1003806.

5. Kriegel HP, Kröger P, Zimek A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Transactions on Knowledge Discovery from Data. 2009;3(1):1–58.

6. Chester C, Maecker HT. Algorithmic tools for mining high-dimensional cytometry data. Journal of Immunology. 2015;195:773–779.

7. Diggins KE, Ferrell PBJ, Irish JM. Methods for discovery and characterization of cell subsets in high dimensional mass cytometry data. Methods. 2015;82:55–63.

8. Wiwie C, Baumbach J, Röttger R. Comparing the performance of biomedical clustering methods. Nature Methods. 2015;12(11):1033–1038.

9. Levine JH, Simonds EF, Bendall SC, Davis KL, Amir ED, Tadmor MD, et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. Cell. 2015;162:184–197.

10. Aghaeepour N, Finak G, Hoos H, Mosmann TR, Brinkman R, Gottardo R, et al. Critical assessment of automated flow cytometry data analysis techniques. Nature Methods. 2013;10(3):228–238.

11. Shekhar K, Brodin P, Davis MM, Chakraborty AK. Automatic classification of cellular expression by nonlinear stochastic embedding (ACCENSE). Proceedings of the National Academy of Sciences of the United States of America. 2014;111(1):202–207.

12. Becher B, Schlitzer A, Chen J, Mair F, Sumatoh HR, Wei K, et al. High-dimensional analysis of the murine myeloid cell system. Nature Immunology. 2014;15(12):1181–1189.

13. Qian Y, Wei C, Lee FEH, Campbell J, Halliley J, Lee Ja, et al. Elucidation of seventeen human peripheral blood B-cell subsets and quantification of the tetanus response using a density-based method for the automated identification of cell populations in multidimensional flow cytometry data. Cytometry Part B. 2010;78B(Suppl. 1):S69–S82.

14. Aghaeepour N, Nikolic R, Hoos HH, Brinkman RR. Rapid cell population identification in flow cytometry data. Cytometry Part A. 2011;79A:6–13.

15. Van Gassen S, Callebaut B, Van Helden MJ, Lambrecht BN, Demeester P, Dhaene T, et al. FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry Part A. 2015;87A:636–645.

16. Sörensen T, Baumgart S, Durek P, Grützkau A, Häupl T. immunoClust - An automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. Cytometry Part A. 2015;87A:603–615.

17. Linderman M. Rclusterpp: Linkable C++ clustering. R package version 0.2.3.; 2013.

18. Zare H, Shooshtari P, Gupta A, Brinkman RR. Data reduction for spectral clustering to analyze high throughput flow cytometry data. BMC Bioinformatics. 2010;11:403.

19. Mosmann TR, Naim I, Rebhahn J, Datta S, Cavenaugh JS, Weaver JM, et al. SWIFT - Scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, Part 2: Biological evaluation. Cytometry Part A. 2014;85A:422–433.

20. Nilsson AR, Bryder D, Pronk CJH. Frequency determination of rare populations by flow cytometry: A hematopoietic stem cell perspective. Cytometry Part A. 2013;83A:721–727.

21. Spidlen J, Breuer K, Rosenberg C, Kotecha N, Brinkman RR. FlowRepository: A resource of annotated flow cytometry datasets associated with peer-reviewed publications. Cytometry Part A. 2012;81A:727–731.

22. Kotecha N, Krutzik PO, Irish JM. Web-based analysis and publication of flow cytometry experiments. Current Protocols in Cytometry. 2010;(July):10.17.1–10.17.24.

23. Hornik K. A CLUE for CLUster Ensembles. Journal of Statistical Software. 2005;14(12).

24. van der Maaten L. Accelerating t-SNE using tree-based algorithms. Journal of Machine Learning Research. 2014;15:1–21.