Exploring cell differentiation-inducing agents for AML with high-throughput flow cytometry and novel drug response analyses

# Abstract

Advancement of methods such as mass spectrometry and high-throughput flow cytometry has led to ever-increasing demand for interactive platforms and tools for analyzing large multiparametric datasets. We developed a tool that allows fast clustering of flow and mass cytometry data based on spatial similarity in multi-dimensional space. In detail, our algorithm, COMPARE, utilizes dimension division where multi-parametric space is divided into hypercubes and the similarity scores between samples is computed by comparing event densities inside the cubes. Parallel comparison of well pairs yields a similarity matrix that is used for sample clustering. Data analysis pipeline has been built into a graphical user interface (GUI) that allows easy and interactive exploration of clustered data. Using high-throughput flow cytometric ex vivo drug screening data we demonstrate that COMPARE can cluster samples into biologically relevant, logical groups and aid in discovery of drugs that induce distinct cell subtype -specific responses.

# Introduction

Single-cell technologies such as flow cytometry and mass cytometry are powerful methods to explore cellular heterogeneity and to identify and define different cell types in very fine detail and as such have greatly advanced our understanding of complex biology of cells and tissues in health and disease [ref]. As single-cell methods have become more advanced, more attainable to researchers and more high-throughput, it has become feasible to process large sample cohorts resulting in wealth of multi-parameter multi-sample datasets in need of scalable analysis methods. Traditionally, two markers have been examined at a time for flow cytometry data. However, as the number of parameters increases, the number of possible dimension pairs rapidly becomes overwhelming. In practice, several dimension pairs are left unexplored and as such valuable information is lost. In addition, biologically meaningful multivariate relationships may not be distinguishable in two dimensions.

Computational tools such as FLOCK[ref and for all coming softwares], SPADE and FlowSOM have been developed for multi-dimensional flow cytometry data to address these problems. However, they focused on the relational hierarchies of cells within individual samples (right? We have to be very careful for this sentence). As the number of data increases, the demand for comparing among samples has risen. PhenoGraph and CITRUS enabled comparison across multiple samples to build relational hierarchy among them. These approaches, however, loses single-cell resolution of the data by simply comparing the centroids of the identified clusters among samples. Moreover, obtaining centroid and clustering them still requires human-intervention. Tools developed for single cell RNAseq (scRNAseq) data such as Seurat equip with sophisticated algorithm to use transcriptome of individual cells. However, they also are suitable for finding relationships across hundreds of flow cytometry data.

An automated pipeline to integrate flow cytometry data and find their relationships will be highly useful. It is more obvious for recent high-throughput flow cytometry (HTFC) machines where hundreds of flow cytometry data are generated in a plate.

A challenge in performing clustering across samples is to remove the technical noise in each sample, called batch effect. A number of batch effect removal for single cell RNAseq (scRNAseq) data cannot readily be used for flow cytometry data with limited number of fluorescence. Moreover, common batch effect removal algorithm cannot be applied for hundreds of samples.

To address this gap, we developed a tool, CompaRe, which in combination with available computational tools provides investigators with a strong toolbox for comparative analyses of large sets of single-cell multiparametric data files. (why Compare works better than the others? Brief description of the algorithmic advantage will be very helpful. 1)hypercube 2) batch-removal) For effective processing, Compare use the distribution of the data in a multi-dimensional hyper-cube of the two samples. A hyper-cube is obtained after dividing the multi-dimensional data space into sub-space. Can you tell me more why it works well?. A hypercube is a way to consider the distribution of markers while effectively reducing the computing cost. To remove batch effect in HTFC, Compare studies intrinsic biases in the HTFC data, while it has a way to perform batch removal if control samples are provided.

The tool allows clustering of flow and mass cytometry samples, flow cytometry standard (FCS) files, instead of cells/events and is based on mass-aware (portion of space occupied by data points) dimension division (not to be confused with reduction). It robustly finds clusters of samples which are spatially similar and the clustered data is visualized through a graphical user interface (GUI). Here we apply the tool to interpret flow (and mass cytometry data) derived from healthy and leukemic human samples and AML mouse models.

# Results

**Measuring similarity between fluorescence cytometry samples**

COMPARE efficiently divides a higher dimensional space formed by fluorescent markers into a specific number of subspaces called hypercubes. It measures the proportion of data points from either dataset within each hypercube. The difference between the two proportions is indicative of similarity within that hypercube so that for two similar datasets this difference is close to zero in all hypercubes. Therefore, averaging these differences across all the hypercubes shows the amount of similarity between the two datasets. **Fig** shows an actual AML dataset with three surface markers dissected by COMPARE wherein each distinct color corresponds to one hypercube.

To generate a similarity matrix of input samples, COMPARE is run in parallel for samples in the upper-triangular submatrix using multithreading architecture. The similarity matrix is then used for identifying clusters of samples such as drugs with similar dose responses.

To cluster high-throughput flow cytometry samples (wells) we developed an in-house graphical clustering algorithm. Initially, all nodes (wells) are connected forming a weighted complete graph wherein edges represent similarity between nodes. This graph is then pruned for removing false positive edges, that is, nodes that are not similar enough. After constructing the graph, clustering is tantamount to finding maximal cliques (complete subgraphs that cannot be extended). In addition to maximal cliques, it also reports dense regions known as communities or hotspots.

We also developed **a GUI**, an interactive tool for visualization of output. The **GUI** has multiple features including plotting the clusters, coloring cells by marker expression, sample or subtype, and gating. **Fig** shows different modules of this pipeline.

**Quantifying similarity between healthy human bone marrow and pediatric AML samples**

To evaluated COMPARE’s precision of similarity measurement, we tested COMPARE on a diverse set of phenotypically heterogeneous and homogeneous samples diligently generated by **[REF]**. The set includes 21 mass cytometry samples of bone marrow aspirates from 16 pediatric AML patients obtained at diagnosis and 5 healthy adult donors. The samples are labeled with metal-isotope-conjugated antibodies specific for 16 surface markers.

We first reproduced their interpatient similarity heatmap using PhenoGraph as they described (**Figure 1A**). When applied to this dataset with the same hierarchical clustering method, COMPARE generated a heatmap with a rather different hierarchy (grouping) of samples (**Figure 1B**). To validate the new hierarchy, we compared it to an independently derived hierarchy of the samples using Jensen-Shannon divergence (JSD) as suggest in **[REF]** that displayed almost complete agreement (**Figure 1C**).

To compare the samples even further, we plotted the heatmap of median fluorescence intensity (MFI) (**Figure 1D**). More noticeably, samples SJ1, SJ2, SJ3, SJ4, SJ5, SJ7 and SJ14 grouped together expressing mid to high levels of CD34, CD38, CD44, CD123 and HLA-DR that is an expression profile more abundant in hematopoietic stem and progenitor cells (HSPCs), specifically myeloid progenitor cells. Interestingly, when we plotted the similarity graph of the dataset to reveal relations between the samples, the same groups of samples also appeared as connected components (**Figure 1E**). This observation was also in agreement with the result in [**REF**] wherein these samples are the only ones containing HSPCs with the aforementioned expression profile (**Figure 3D** ibid).

COMPARE runs in parallel and can process several multiple samples with numerous markers in a short time. While PhenoGraph and JSD took a couple of days to process only a subsample of this dataset (100,000 cells of each sample) on an average workstation, it took COMPARE a couple of minutes on the entire dataset.

We compared its output to PhenoGraph which is a cell clustering method and JSD which in information theory is a method of measuring the similarity between two probability distributions. Although they basically performed very similarly, sample grouping by JSD displayed almost complete agreement with COMPARE’s… (it is from Discussion. But, it should be in the results. Please kill more and provide a measurement of the comparisoin so that a person can EASILY see the advantages of using Compare.

**Cell differentiation screen identifies cell differentiation inducing agents in mouse model**

To do

**Cell differentiation screen identifies cell differentiation inducing agents in human**

**Exploring cancer heterogeneity across multiple leukemic individuals**

To do

# Discussion

……

We tested COMPARE’s accuracy in similarity quantification on 21 mass cytometry samples of bone marrow aspirates from 16 pediatric AML patients obtained at diagnosis and 5 healthy adult donors. We compared its output to PhenoGraph which is a cell clustering method and JSD which in information theory is a method of measuring the similarity between two probability distributions. Although they basically performed very similarly, sample grouping by JSD displayed almost complete agreement with COMPARE’s. This discrepancy between these two and PhenoGraph stems from the unnatural approach, meta-clustering, PhenoGraph should use to be able to compare samples because essentially it was designed to cluster cells not samples. In a general sense, for a typical cell clustering method like PhenoGraph to be able to compare multiple samples, it needs to meta-clusters centroids of cell clusters across all of them. Tacitly, it assumes centroids are enough to represent respective cell clusters which does not hold true for heterogenous clusters with multiple subclusters or morphologically different clusters but with spatially close centroids. In addition, subsampling and the choice of *k* for constructing the *k*-nearest-neighbor graph plays a determinant role in forming cell clusters. Likewise, subsampling, binning and dimension reduction steps needed to adapt JSD to compare samples make it susceptible to failure. However, it can be shown that a meta-cluster in PhenoGraph, a bin in JSD and a hypercube in COMPARE are somehow equivalent spatial entities so that in 2-dimesnional space, JSD and COMPARE essentially do the same work (**methods**).

Because of the steps mentioned above, PhenoGraph and JSD took a couple days to process a subsample of this dataset on an average workstation, whereas it took COMPARE a couple of minutes to finish processing. This is because COMPARE runs in parallel and can therefore process multiple large datasets with numerous markers in a short space of time. There are several features in addition to parallelism that COMPARE benefits from, making it substantially fast (**methods**).

…..

# Methods

**Applying COMPARE to mass cytometry AML samples.** To do by Morteza.

**High-throughput flow cytometry of AML mouse models**. Leukemic splenic cells sorted for c-Kit positivity (sorting schema in suppl.fig?) from *Npm1*+/cA ; *Flt3*+/ITD;*Tet2-/-*(reference) and *Npm1*+/cA ; *Flt3*+/ITD; *Dnmt3a*-/- (reference) moribund mice were kindly provided by Dr. Helin and Dr.Rasmussen. (Authenticity certificate?, approval of animal procedures?). Shortly, c-Kit+ splenic cells were expanded in cytokine rich Stemspan 34 media (media ingredients into suppl.table) *ex vivo* for two passages with complete media change every two/three days. Aliquots of one million cells were frozen down in 90% media 10% DMSO. Frozen aliquots were taken up and expanded for one week before drug screening. 5000 cells in 25 µl of media per well was seeded into 384-well plates (Greiner, #) containing a library of 132 compounds (suppl.table?) in a five point concentration range. After 72 h incubation at 37°C, 15 µl of media was aspirated from wells and antibodies (suppl.table) were added to drug plates using acoustic dispensing. Plates were incubated 40 min at RT, covered from light. Next, viability dye (7-AAD, ) was added and samples were read using a high-throughput flow cytometer iQue+ (Intellicyt). Live cell -gated data (gating schema as suppl.fig?) was used in all the further analyses.

**High-throughput flow cytometry of human AML.** Monocyte Isolation: Monocytes were isolated from donated (permit here) bone marrow aspirates. Briefly the aspirates in Li-Heparin were diluted 50/50 with PBS (w/o Ca+ Mg+ from Thermo), they were filtered on a 70µm filter (Sarstedt) and monocytes separated using prefilled LeucoSep tubes (Greiner Bio-One) as directed. The monocytic layer was red blood cell lysed with, (ACK – RH hospital) washed, and live frozen in 10% DMSO (Merck), 40% IMDM (Thermo) and 50% Serum (Thermo - #10270106) media. Tubes were stored in a vapour phase liquid nitrogen before being thawed and used.

Assay Preparation and Media: MNCs were thawed by quickly adding them to warm (37C) RPMI (Thermo) media containing 20% serum (Thermo - #10270106), 1% penicillin/streptomycin (Thermo) and 50U/ml of TurboNuclease (Abnova) at a ratio of 1ml MNC:10ml media. The suspension was spun 10min at 500g, decanted and re-suspended in 5ml of the thawing media and again spun. After decanting the pellet was re-suspended in 3ml of media without TurboNuclease, counted and spun 5min at 500g, decanted, and ready for suspension in assay media.

MNCs were allowed to rest overnight in assay media: StemSpan II-SFEM (StemCell), 100U/ml penicillin/streptomycin (Thermo), the following human recombinant cytokines from Preptech, unless otherwise stated, 50ng/ml Flt3 (StemCell), 10nl/ml IL3, 10ng/ml IL-1beta, 20ng/ml IL6, 20ng/ml G-CSF, 20ng/ml GM-CSF, and 10ng/ml SCF, and the following compounds diluted in DMSO (Merck) 1µM UM729 (Selleckchem) and 500nM StemRegenin 1(SR1) (MedChemExpress). Before being counted and re-suspended in fresh assay media at a density of 0.5E6 cells/ml and 20µl/well plated in 384 well conical bottom plates (Greiner Bio-One) containing 5µl of compounds (in DMSO). Assay plates with pre-aliquoted compounds were purchased from FIMM.

**Flow cytometry of AML and MDS patients.** To do by Krister’s.

**Comparing samples.** To measure similarity between samples, we developed COMPARE. It measures the similarity between two datasets (e.g. wells) with any number of dimensions (e.g. cell markers) and observations (e.g. cells/events).

COMPARE efficiently divides a higher dimensional space formed by variables into a specific number of subspaces called hypercubes. It measures the percentage of data points from either dataset within each hypercube. The difference between the two percentage values is indicative of similarity within that hypercube so that for two identical datasets this difference must be zero in all hypercubes. Therefore, averaging these differences across all hypercubes shows the amount of similarity between the two datasets.

COMPARE provides two strategies to divide the space. In the first mode, COMPARE merges two datasets into one and iteratively divides the entire dataset into three subsets using tertiles of each variable distribution. Therefore, the merged dataset of two datasets with variables will be divided into *at most* hypercubes. In the second mode, hypercubes are formed for each dataset separately, and similarity is measured within corresponding hypercubes. The second strategy is elucidated in the section related to handling signal drift.

COMPARE uses a two-level optimization to speed up the process. At first level, it is a mass-aware approach to avoid empty areas. The rationale behind using quantiles for dimension division is to form comparison regions (hypercubes) only around concentrations of data points disregarding areas devoid of data points. This conscious approach, compared to fixed regionalization, substantially speeds up the process by saving CPU time making it feasible for datasets with numerous variables. For example, dividing each dimension blindly into three fixed regions yields more than one billion regions to consider for a dataset with as few surface markers as 19. However, in practice, it turns out many of these regions are empty so using tertiles instead of fixed regions improves the time complexity from to . Even if no region is empty, COMPARE uses a second level of optimization, dynamic programming, to finish quickly. The second level is further explained in the section related to handling signal drift.

To generate a similarity matrix of input samples, COMPARE is run in parallel for samples in the upper-triangular submatrix using multithreading architecture. The similarity matrix is then used for identifying clusters of samples such as drugs with similar dose responses.

COMPARE is not restricted to only florescence cytometry data. It can be called by a wrapper to process any type of numerical data such as FCS data. It is written in R and available as an R package on GitHub.

**Clustering samples.** To cluster high-throughput flow cytometry samples (wells) we developed an in-house graphical clustering algorithm. Initially, all nodes (wells) are connected forming a weighted complete graph wherein edges represent similarity between nodes. This graph is then pruned for removing false positive edges, that is, nodes that are not similar enough. After constructing the graph, clustering is tantamount to finding maximal cliques (complete subgraphs that cannot be extended). In addition to maximal cliques, it also reports dense regions known as communities.

To decide whether two nodes are similar enough to remain connected (a true positive edge), we needed to assume a similarity threshold (minimum acceptable similarity). Knowing that controls (DMSO wells in our setting) are supposed to be similar enough to each other, heuristically we learned that an optimal threshold was near the maximum value for which the control subgraph remained still a tree. In simple terms, the cutoff is set so that control nodes remain connected to each other after pruning. As an option for the investigator, the threshold can be also set manually.

The cutoff and consequently the clustering can be inaccurate if some of the controls are less similar to other controls showing properties of outliers. The cutoff turns out to be good enough when: (i) there is a sufficient number of controls, (ii) spread properly across the plates, (iii) and they are all well similar to each other. In addition to control outliers, there are other phenomena such as auto-fluorescent drugs, edge effect, carryover effect, signal drift and fluidics blockage that may significantly affect the clustering. We followed several quality checks to ensure a good clustering.

Our clustering method is not limited to flow cytometry data and is applicable to any problem space provided it can be seen and understood in the context of presence of some samples as controls.

**Sample dispersion map.** Depending on the type of a screen, it may turn out several samples (e.g. non-responders) are immediately connected to control samples (e.g. DMSO) after pruning for a given cutoff. These are samples that are morphologically similar enough to controls. Therefore, we were interested in investigating how non-control samples disperse around controls. Starting from the control node (e.g. DMSO wells plus non-responders), a tree of samples (nodes) is constructed with respect to the control node (root) in which each sample is attached to its most similar one as the graph branches off to the leaves, and edge color represents similarity value. While this graph does not necessarily show sample derivation, it unfolds different potential directions from the root caused by various combinations of marker expressions. For example, staining samples with each MFI, while a crude way, could be visually useful. We have integrated several visualization tools into **the GUI** providing the investigator with useful insights into why samples go their separate ways.

**Signal drift handling in fluorescence cytometry data**

Fluorescence cytometry data are difficult to standardize between batches analyzed days or months apart, because cytometer settings can change with time, or reagents may fade. Imperfect protocol adherence may also lead to changes in staining intensity (signal drift) or machine settings. Such variations need to be identified, and where possible corrected. In experiments in which batch effects occur due to variability in staining or cytometer settings, algorithms for reducing this variation by channel-specific normalization have been developed but they require subpopulations to be already exposed by manual or automated pregating. For example, fluorescence minus one (FMO) controls are important when building multicolor flow cytometry panels as they will help one determine where their gates should be set. This is particularly important when identifying a positive from a negative population. However, batch effects due to other causes may be more difficult to correct.

To circumvent batch effect left uncorrected, for any reason, we generalized COMPARE so that the similarity metric can be computed even in the presence of batch effect provided the investigator believes what makes two samples different is the number and/or morphologies of cell subpopulations rather than marker expression levels. As explained earlier, COMPARE can be run in the “separate” mode wherein hypercubes are formed for two samples separately. In this way, if there is a drift in the signal (staining intensities), two identical samples still will have 100% similarity. Conversely, in the “merge” mode, their similarity will be less than 100%. Therefore, one should normally COMPARE samples in the “merge” mode if there is no significant signal drift or it is corrected for using available methods such as the one explained in the next section which is specific to HTFC screening.

In COMPARE’s algorithm, at each iteration, the input dataset (a sample) is divided into three subsets (children) based on the tertiles of each variable (column or dimension): (i) observations less than 1st tertile, (ii) within 1st and 2nd tertiles, and (iii) greater than 2nd tertile. **Fig** shows the expansion tree of a dataset with 3 variables after the final iteration (3rd). To measure similarity between two samples in separate mode correctly, we need to measure it within corresponding hypercubes (leaves after final iteration). Therefore, it is crucial to have a uniform numbering system for branches (divisions) across samples as the tree expands. Here, the leftmost branch is first numbered. In this setting, even if some of the corresponding hypercubes are absent in one of the samples (when there were no data points in those divisions), the numbering system ensures consistency. For example, if in the end, sample 1 has hypercubes and sample 2 has hypercubes, since hypercubes with the same number are ensured to be corresponding (from the same spatial positions), we are allowed to compare the two samples in the *union* of all their hypercubes. For hypercubes present in only one of the samples, similarity is 0%. **Fig** shows an actual AML dataset with three surface markers dissected by this process wherein each distinct color corresponds to one hypercube.

To ensure consistency in numbering, the number of child node must be calculated from its parent’s. Rewriting the branch numbers to include more information, it reveals that if is the number of parent node, child node’s number will be where is the current iteration (aka level or dimension) at which the child node is, is the number of families behind, and is the number of siblings behind. Therefore, to calculate for current child, we first need to calculate and of its parent as follows:

It can be noticed that and are always known, , and is actually the largest branch number after th iteration. Therefore, the problem we need to dynamically solve for each child at each iteration is:

COMPARE in merge mode employs the same algorithm as in separate mode except that since the two input samples are first merged, there is only one set of hypercubes instead of a union, and for hypercubes in which only one of the samples has data points, similarity is 0%.

Changing the order of columns and increasing the number of quantiles have no effect and little effect on the similarity metric respectively. The former is due to the commutative nature of dimension division, and the latter just changes the scale of values while mostly preserving relative differences.

COMPARE is written using dynamic programming in which an optimization problem is iteratively solved. Programs developed in this way benefit from shorter processing time compared to those using recursion in which solving a problem depends on solutions to smaller instances (e.g. measuring similarity within one hypercube) of the same problem.

**Signal drift correction in high-throughput flow cytometry**

Signal drift is an intrinsic property of HTFC and therefore a source of bias within and across plates. Depending on the order and the protocol wells are aspirated by the flow cytometer, time may become a major concern so that wells read later may have higher MFIs than expected. **Fig** shows three plates of an AML mouse model screen in which wells were read column-wise form right to left. Plotting MFIs vs pseudo-time (i.e. the order wells were read) reveals an increasing trend toward the bottom left of the plate. Also, since there is a time difference between reading the plates, there is also a batch effect between the plates.

To correct for this bias, we employed a two-step correction: intra-plate correction and inter-plate correction. For a given plate, we first fit a linear regression model and then vertically translate points (MFIs) with respect to the leaned line as it rotates to slope zero. This is because the relative distance between the points must be preserved as much as possible, and no point must be translated to the negative- quadrant after correction. In this way, a point at is translated to after intra-plate correction. The coefficient is called correction coefficient and derives from the ratio of y-coordinates of any point on the regression line before and after translation: where is translated , is slope and is intercept of the line. This ratio holds true for all other points in the -plane.

After correcting for intra-plate signal drift, inter-plate signal drift is corrected by simply aligning MFI medians of the plates, that is, translating to a common baseline. Let’s say is the baseline, and is the median of corrected MFIs in a plate, then the inter-plate correction coefficient is given by , and a point at is translated to .

**Quality control in HTFC**

High-throughput flow cytometry technologies can analyze a massive number of samples in a single experiment. However, no screening technology is perfect, and each instrument will generate different types and amount of errors and bias. These are due to the technical limitations of each screening platform. Therefore, it is necessary to understand, identify and exclude them that may impact the interpretation of downstream analysis. Screen quality control is therefore an essential first step in the analysis. We followed several steps to catch potential errors and bias.

Plotting heatmaps of wells’ MFIs in each plate helps reveal possible problems occurred during screening. However, since naturally an HTFC screen has wide dynamic range of MFIs, even average MFIs are obscured by very high MFIs in an unscaled heatmap. This necessitates scaling heatmaps to reveal phenomena such as signal drift, autofluorescence and carryover effect. Nonetheless, scaling will also distort original MFIs making it impossible to visually compare wells. Therefore, we developed a specialized heatmap whose color palette is spaced according to the density distribution of MFIs. In this approach, we took advantage of the interquartile range which is often used to find outliers in data. Outliers here are defined as observations that fall below the lower whisker or above the upper whisker. We then assigned shades of different distinct colors to values (i) less than the lower whisker, (ii) between the lower whisker and the first quartile, (iii) within the interquartile range, (iv) within the third quartile and the upper whisker and (v) greater than the upper whisker. We could also identify control outliers by plotting the heatmap of similarity values of only control samples. This way, samples less similar to others shows properties of outliers. These steps are included in our pipeline.

# Data Availability

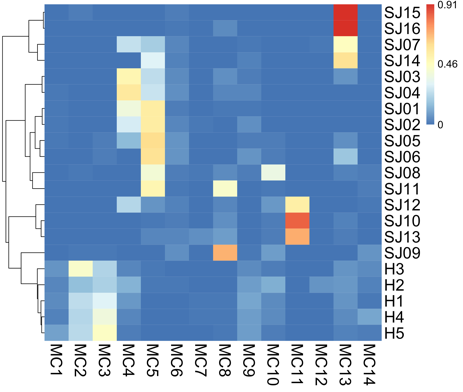
Human and mouse flow cytometry data generated for this paper is available for download from FlowRepository: <https://flowrepository.org/>.

# Code Availability

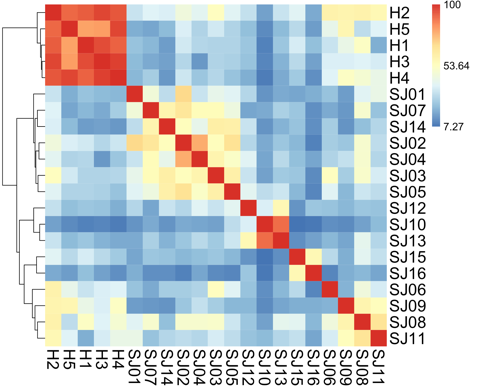
The HTFC pipeline is entirely integrated in **the GUI** available for download from **HERE**. The command line version of the pipeline is also available for download from GitHub: <https://github.com/morchalabi/HTFC.git>. The pipeline internally uses the COMPARE package which is also available for download as a stand-alone application from GitHub: <https://github.com/morchalabi/compaRe.git>. The COMPARE package includes COMPARE and our clustering modules.

# References

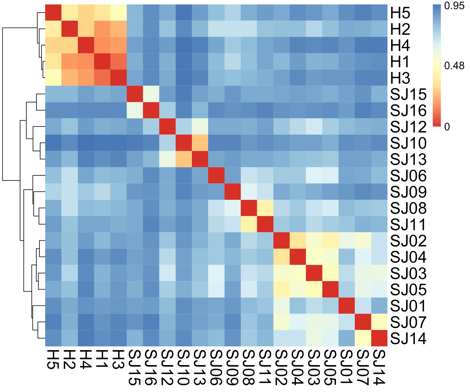
Figures & Tables



**A**

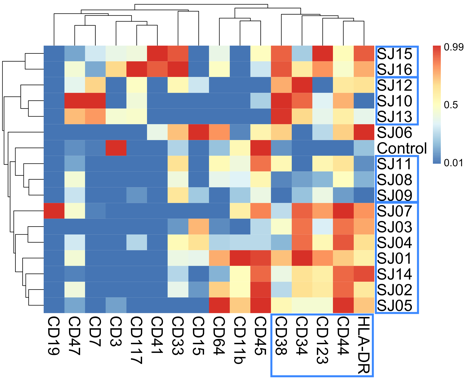
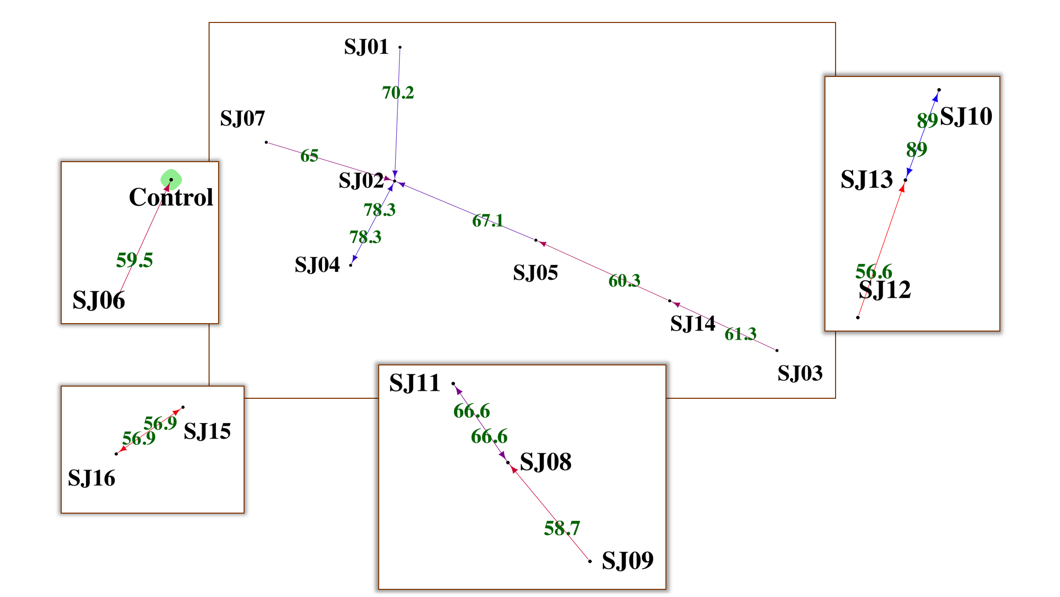


**B**



**C**

**E**



**D**

Figure 1 **COMPARE precisely measures similarity between high-dimensional samples**

(A) Inter-patient similarity heatmap of 21 mass cytometry samples of bone marrow aspirates from 16 pediatric AML patients (SJ) obtained at diagnosis and 5 healthy adult donors (H) reproduced from **[REF]**. The samples are labeled with metal-isotope-conjugated antibodies specific for 16 surface markers. To compare samples, subpopulations identified by PhenoGraph from each patient were represented by their centroids. PhenoGraph was used for a second time to group centroids into meta-clusters (MCs), identifying 14 MCs.

(B) Similarity heatmap generated by COMPARE of the same dataset with the same hierarchical clustering method. Although grouping of samples in both the heatmaps looks very similar, there are some differences as seen in SJ1, SJ2, SJ3, SJ4, SJ5 and SJ6. To validate these discrepancies, we compared our hierarchy with an independently derived hierarchy using Jensen-Shannon divergence (JSD) as explained in **[REF]**.

(C) Similarity heatmap generated by JSD with the same hierarchical clustering method. In information theory JSD is used to measure similarity between two probably distributions. We used the same approach suggested in **[REF]** to convert each sample to a probability distribution. The hierarchy retuned by JSD completely agreed with COMPARE’s.

(D) Median fluorescence intensity (MFI) heatmap. To compare the samples further, we compare them across the MFIs of the 16 cell surface markers. Hierarchical clustering of samples and MFIs revealed groups of samples with similar expression profiles of specific markers. For example, SJ1, SJ2, SJ3, SJ4, SJ5, SJ7 and SJ14 expressed mid to high levels of CD34, CD38, CD44, CD123 and HLA-DR indicating expression profiles of hematopoietic stem and progenitor cells. This agreed with the result shown in **[REF]** (Figure 3D ibid).

(E) Similarity heatmap revealing relations between samples. Interestingly, when we plotted the similarity graph of the samples, the same grouping of MFI heatmap was observed as connected components in the graph (trees). For better visualization, each component of the entire graph is shown in a separate pane.

Supplementary Methods

# Experimental methods

# Computational methods

Letter to Editor

By KJ and Krister¶