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

# Abstract

To do by Krister’s.

# Introduction

Personalized medicine (PM) has the potential to tailor therapy with the best response and highest safety margin to ensure better patient care. By enabling each patient to receive earlier diagnoses, risk assessments, and optimal treatments, PM holds promise for improving health care while also lowering costs. In PM, diagnostic testing is often employed for selecting appropriate and optimal therapies, such as appropriate drug doses, based on the context of a patient’s genetic content or other molecular or cellular analysis. For example, increasing the dose of a drug with a small therapeutic index (ratio of the concentration in which the drug is toxic to that in which the drug is effective) increases the probability of toxicity (adverse effects) or ineffectiveness of the drug. Therefore, drug-response analysis is necessary to determines the required dose and frequency for a drug in a population.

Emerging single-cell technologies such as high-throughput flow cytometry (HTFC) and mass cytometry, powerful tools for drug response analysis, have enabled researchers to uncover new and potentially unexpected biological discoveries relative to traditional profiling methods in drug development and personalized medicine. HTFC has enabled the study of a set of parameters (e.g. antibody-labeled surface proteins) in tens of thousands of single cells of hundreds of samples simultaneously.

Single-cell data are often examined in two dimensions at a time. Nevertheless, as the number of parameters increases, the number of pairs becomes factorially large. A typical cytometry dataset may allow numerous pairwise combinations. Besides, a pairwise viewpoint could miss biologically meaningful multivariate relationships that may not be distinguishable in two dimensions. Several very good computational tools such as FLOCK, SPADE and FlowSOM have been developed to address these problems. However, while these approaches are powerful for analyzing the makeup of cell populations and sometimes their relational hierarchies within individual samples, they are not applicable to multiple-sample comparisons and consequently not immediately useful for analyzing drug responses by HTFC, for example. In HTFC, the aim is typically to search many treated samples for treatments targeting specific cell subpopulations, inducing differentiation modifying proportions of different subpopulations and even inducing new subpopulations. For this, tools that cluster a large number of samples and identify their hierarchical similarities in parallel are needed instead of the aforementioned methods that essentially cluster cells. PhenoGraph and CITRUS serving a similar purpose seem to be two good candidates to consider. However, approaches such as PhenoGraph, whether automated or not, principally cluster cells and examine the average (centroids) of each cluster across multiple samples to form meta-clusters, resulting in the loss of single-cell resolution of the data by assuming centroids are enough to represent heterogeneous clusters. Also, since CITRUS is based on statistical learning, it needs distinct groups of samples like control vs treatment or healthy vs cancer with a minimum of eight samples per group beforehand in order to have enough statistical power to be confident in the trained model. This means CITRUS entails prior knowledge about heterogeneity of samples. Thus, CITRUS is also not appropriate for comparing unknown samples even if they are sufficiently heterogeneous.

To address this gap, we developed a tool, CompaRe, which in combination with available computational tools provides investigators with a strong toolbox for HTFC data analysis and other comparative analyses of large sets of single-cell multiparametric data files. 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. Its output provides visual representations of clusters as well as a table of drugs with their assigned phenotypic groups. Here we apply the tool to interpret flow and mass cytometry data derived from healthy and leukemic human and mouse models.

# Results

**Precise measurement of similarity between high-dimensional fluorescence cytometry samples**

To do

**Surveying phenotypic landscapes across multiple healthy and AML samples**

To test the effectiveness of CompaRe, we evaluated its ability to correctly arrange phenotypically heterogeneous samples. We analyzed a previously published mass cytometry dataset of single-cell proteomic profiles of cryopreserved bone marrow aspirates from 16 pediatric AML patients obtained at diagnosis and 5 healthy adult donors. The samples had been labeled with metal-isotope-conjugated antibodies specific for 16 surface markers and 14 intracellular phosphorylation markers, thus allowing simultaneous measurement of surface phenotype and signaling behavior in single cells. However, we only used the parameters measuring the surface markers. The tool correctly arranged labeled samples, displaying superior precision and robustness.

Having ensured conformity between intra-patient heterogeneity results, we sought kinships between the AML and healthy samples. To generate the network graph, we used the similarity threshold automatically inferred from the healthy bone marrow samples. Interestingly, the AML samples (SJ07, SJ10, SJ12, SJ13, SJ15 and SJ16) reported to make significant contributions to malignant cell populations resided far from the healthy samples, and the AML samples (SJ05, SJ06 and SJ11) with significant contributions to non-malignant populations were found in close proximity to the control node.

**Exploring cancer heterogeneity across multiple leukemic individuals**

To do

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

To do

# Discussion

* A conclusion on the tool and what we found

# Methods

**Applying COMPARE to mass cytometry AML samples**

**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.** To do by Krister’s.

**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 samples under consideration, 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 factors such as autofluorescent drugs, edge effect, carryover effect, signal drift and fluidics blockage that may significantly affect the clustering. We have integrated several quality checks into the tool, and the investigator is strongly advised to follow these steps for a good clustering.

Our clustering method surely 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 graph.** 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 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.

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.

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.

**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 . 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 pipeline.**

**Plate heatmap.**

**Control heatmap.**

# Data Availability

To do by Krister’s.

# Code Availability

To do by Morteza.

# References

Figures & Tables

Supplementary Methods

# Experimental methods

To do by Krister’s.

# Computational methods

Letter to Editor

By KJ and Krister¶