Computational and Structural Biotechnology Journal 19 (2021) 3160–3175

ELSEVIER

COMPUTATIONAL ANDSTRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

Data processing workflow for large-scale immune monitoring studies by mass cytometry

Paulina Rybakowska [a], Sofie Van Gassen [b,c], Katrien Quintelier [b,c,d], Yvan Saeys [b,c], Marta E. Alarcón-Riquelme [a,e,*], Concepción Marañón [a,*]

[a] GENYO, Centre for Genomics and Oncological Research, Pfizer, University of Granada, Andalusian Regional Government, PTS Granada, Spain
[b] Department of Applied Mathematics, Computer Sciences and Statistics, Ghent University, Gent Belgium
[c] Data Mining and Modeling for Biomedicine, VIB Center for Inflammation Research, Gent, Belgium
[d] Department of Pulmonary Diseases, Erasmus MC, Rotterdam, the Netherlands
[e] Institute for Environmental Medicine, Karolinska Institute, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 22 January 2021
Received in revised form 14 May 2021
Accepted 20 May 2021
Available online 21 May 2021

Keywords:
Mass cytometry
Semi-automated data preprocessing
Immune phenotyping
Data normalization
Quality control

ABSTRACT

Mass cytometry is a powerful tool for deep immune monitoring studies. To ensure maximal data quality, a careful experimental and analytical design is required. However even in well-controlled experiments variability caused by either operator or instrument can introduce artifacts that need to be corrected or removed from the data. Here we present a data processing pipeline which ensures the minimization of experimental artifacts and batch effects, while improving data quality. Data preprocessing and quality controls are carried out using an R pipeline and packages like *CATALYST* for bead-normalization and debarcoding, *flowAI* and *flowCut* for signal anomaly cleaning, AOF for files quality control, *flowClean* and *flowDensity* for gating, *CytoNorm* for batch normalization and FlowSOM and UMAP for data exploration. As proper experimental design is key in obtaining good quality events, we also include the sample processing protocol used to generate the data. Both, analysis and experimental pipelines are easy to scale-up, thus the workflow presented here is particularly suitable for large-scale, multicenter, multibatch and retrospective studies.

1. Introduction

Mass cytometry (cytometry by time-of-flight, CyTOF, MC) is a single cell proteomics technology that allows the measure of up to 50 markers per cell. Currently it is extensively used in fundamental biology studies [1], but due to researchers' effort in technology and protocol optimizations it is gaining interest for the analysis of clinical samples [2,3]. The CyTOF success is due mainly to its multiparametric capacity, the ease of panel design owing to minimal spill-over issues and the facility to stain multiple samples in one single tube using barcoding approaches [4].

On the other hand, the maximal acquisition capacity of CyTOF devices is limited to 1000 cells/s [5], while a rate of 400 cells/s is recommended to avoid cell aggregation and doublet formation [6,7]. Because of this, it becomes problematic to acquire multiple samples per day, particularly if several tubes of complex tissues (like blood or liquid biopsies) are acquired to detect rare cell populations. A solution to this is to split the samples in multiple batches and acquire them on different days. Nevertheless, this approach requires an optimized experimental workflow that limits technical variation, including a single antibody cocktail and the inclusion of a reference sample in every batch. In addition, the analysis pipeline should contain tools for normalizing the data and removing experimental and day-to-day detector variation [8].

Although the samples are fixed after staining, they are usually acquired in water in the case of the narrow bore (NB) sample injector. Thus, due to the prolonged water exposure and long acquisition times, samples are degraded and lose their tags. To limit the exposure time, samples can be split and acquired in multiple aliquots [9] (as also presented in this manuscript), but still some differences in signal intensity can occur due to variations in detector yield. Furthermore, clogging in the capillary introduction system in cytometry devices alters the flowrate and signal quality over time [10]. In order to obtain high quality data, fcs files should be

* Corresponding authors at: GENYO, Centre for Genomics and Oncological Research Pfizer, University of Granada Andalusian Regional Government, PTS Granada, Spain.

E-mail addresses: marta.alarcon@genyo.es (M.E. Alarcón-Riquelme), concepcion.maranon@genyo.es (C. Marañón).

https://doi.org/10.1016/j.csbj.2021.05.032
2001-0370/© 2021 The Authors. Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology.

Table 1
Protocol for high quality data preparation.

Before starting the protocol	Staining and acquisition protocol
1. Calculate the number of samples that will be analyzed and the number of batches that will be acquired on CyTOF. Comment: Estimate the number of samples required to prove or reject the hypothesis. 2. Prepare all the reagent needed for the whole protocol using a limited number of lots. Aliquot those reagent requiring freezing. Comment: As the reagent expiration date might be a problem, check batch-to batch equivalence. 3. Design the antibody panel, conjugate antibodies if necessary, check cellular epitopes resistance to fixation if necessary, titrate each probe using the same experimental conditions [31]. Comment: Titrate the antibody using a fixed number of cells, this way it will be easily scalable to experiments involving barcoding. 4. Prepare a single big batch of antibody cocktails, stimulation agents and reference sample, aliquot and freeze at −80 °C, as described [6,19]. Comment: at least one extra aliquot should always be prepared. In the case of multicenter studies we recommend preparing all reagents in one central laboratory and ship them on dry ice to the rest of the centers. 5. To limit experimental variability use barcoding when multiple samples are run in one batch. Comments: Different barcoded approaches can be used [4,32–35]. Titrate barcoding reagent according to the number of cells used for staining. 6. Design each acquisition batch ensuring even distribution of biological groups. Comments: If multiple patient groups or culturing conditions are studied, each batch should contain a representation of each group, including healthy controls.	1. Thaw selected blood samples assigned to the staining batch as described [6] a. Count cells b. Aliquot equal number of cells for each sample, in this protocol 1.5*10^6 cells/sample 2. Barcode samples, in this protocol with palladium based barcoding a. Wash with Barcoding Perm buffer (BPB) b. Stain with barcoding reagent resuspended in 500 µl of BPB, 20 min RT c. Wash with CSB d. Pool all the samples in a single tube 3. Stain surface antigens with previously thawed surface antibody cocktail a. Resuspend cells at 5*10^7 cells/ml with surface cocktail b. Incubate for 30 min 4 °C 4. Stain intracellular antigens with previously thawed functional antibody cocktail a. Wash with Perm-S buffer b. Resuspend cells at 5*10^7 cells/ml with the intracellular cocktail c. Incubate for 30 min 4 °C 5. Stain DNA with Iridium solution a. Wash with CSB b. Incubate 20 min RT with 5 µM Ir in Fix and Perm buffer 6. Store in 2% PFA O/N a. Wash with CSB b. Resuspend in freshly prepared 2% PFA at 6*10^6 cells/ml 7. Acquire on CyTOF2/Helios instrument in aliquots (if cells are acquired in water,or if reduction of data size is of interest) a. Take an aliquot of 250 µl b. Wash with CSB c. Wash with acquisition solution (AS) d. Resuspend at 8*10^5/ml in AS e. Acquire in CyTOF2/Helios. Adjust the flow rate to the AS and the type of cells

screened and cleaned from any abnormalities. Hence, extensive data preprocessing should be included in the analysis pipeline.

Due to the *pros* and *cons* described above, mass cytometry experiments require a special and careful experimental design and an extensive analysis pipeline that allows automatic preprocessing of the data and performs proper quality control, especially when hundreds of files need to be analyzed. Although much effort was put to develop automated gating strategies including clustering and dimensional reduction algorithms [11–16] or quantitative analysis [17,18], less was developed in the field of data cleaning and preparation.

Here we present a semi-automated, R-based, CyTOF analysis pipeline that performs data preprocessing and quality control. It spots and removes potential artifacts introduced during sample preparation and acquisition, like clogs, changes in signal intensities upon acquisition and batch effects, thus improving data quality. This analysis pipeline gathers known tools used in both flow cytometry (FC) and MC and adapts them to large and multibatch MC studies, providing also solutions for data visualization. For data preprocessing, steps like bead-based normalization, debarcoding, file aggregation, and automated gating using Gaussian parameters, DNA and live/dead staining for intact and live cell selection are included. Furthermore, we implement additional quality control steps to remove bad quality events or to identify and correct batch effects using a reference sample. We provide full access to the data set used in this work, so the users can reproduce the data processing and analysis steps. Good data quality starts with a proper sample processing minimizing experimental bias and eliminating bad quality events. Therefore, we also provide the protocol of the experimental setting used to generate the data analyzed in this work and show how to scale it up and some important tips.

This workflow contains all the necessary steps to obtain high quality events in an semi-automated way. Importantly to note, it requires knowledge of basic R language programming. It is especially suited for researchers performing multicenter or retrospective studies involving collection of hundreds of biological samples, but is equally suitable for small-scale studies.

2. Material and methods

2.1. Study participants and whole blood processing

Two human healthy donors were enrolled under a protocol approved by the Ethical Committee of Centro Granada (CEI-Granada) according to the Helsinki declaration of 1975, as revised in 2013. All donors signed an informed consent according to the ethical protocol of the Andalusian Biobank and the PRECISESADS project. The Granada node of the Andalusian Health System Biobank collected whole blood samples. Samples were processed following the protocol described in [6]. Briefly, 10 ml of blood from two healthy donors was collected using EDTA-K3 vacutainer tubes, and 1.5 ml blood was diluted 1:1 with RPMI (Gibco) and stimulated with four different Toll-like receptor agonists: R848 (resiquimod, RSQ, 1.25 µg/ml, Invivogen), R837 (imiquimod, IMQ, 2.5 µg/ml, Invivogen), lipopolysaccharide (LPS, 0.05 µg/ml, Invivogen), ODN2006 (CpG, 2 µM, Invivogen) and medium alone (UNS). The stimulations were performed for 6 h in the presence of Protein Transport Inhibitor Cocktail 1X (Thermo Fisher Scientific) to prevent intracellular cytokine exocytosis. After stimulation, live/dead staining was performed using cisplatin (CisPt, 5 µM), for 5 min at RT. Blood cells were fixed with 4.2 ml of Proteomic Stabilizer

(Smart Tube Inc) and 3 aliquots of 2.4 ml of each stimulation were frozen and kept at −80 °C until staining. Cell staining and acquisition were carried out as detailed in Supplementary Methods and Table 1.

The reference sample consisted in 2 ml of whole blood of donor 2 stimulated with RSQ and processed as described above. Four aliquots of 2.4 ml were frozen. The RSQ agonist was chosen as reference stimulation since it induced the expression of all the studied cytokines across multiple cell populations of interest in previous analysis (laboratory data).

2.2. Data analysis

The fcs files were pre-processed using an in-house R script pipeline built through the assembly of several algorithms. The R script, functions and packages necessary for its installation can be found in github CyTOF_analysis_piepline (https://github.com/prybakowska/CyTOF_analysis_Pipeline1), and the example fcs files are deposed on flowrepository [20] with the accession id FR-FCM-Z3YR. Additionally, metadata (*meta_data.csv*) and FlowJo gating workspace (*gating_strategy.wsp*) can be found in the *Attachments* section in flowrepository experiment, and can be downloaded to reproduce the results.

.fcs files were first normalized using the `bead_normalized()` function, which uses `normCytof()` function from the *CATALYST* package [21]. Firstly, a baseline file was generated by the aggregation of 25,000 cells per file using the in-house function `baseline_file()`. This file was further used to compute baseline beads values to which all the files were normalized. The following settings for `normCytof()` were used: *dvs* beads, that were set to be removed post-normalization, *norm_to* was set to the baseline flow frame generated above, transformation was set to FALSE and plot to TRUE. Plots for marker visualization across all normalized files were generated using the `plot_marker_quantiles`() function, which takes advantage of *ggplot2* package [22].

Flow rate examination and signal cleaning were done using `clean_flow_rate()` and `clean_signal()` functions, respectively. For flowrate cleaning we used functions from *flowAI* [10] package with the TIMESTEP adaptation to 1 bin per 10 s and alpha set to 0.01. For signal cleaning the *flowcut* package (https://github.com/jmeskas/flowCut) was used with 1000 segments *MaxPercCut* set to 0.5. The parameters *UseOnlyWorstChannels*, *AlwaysClean* and *AllowFlaggedRerun* were set to TRUE.

The outlier detection was done using the wrapper function `file_quality_check()`. This function calls the FlowSOM clustering [14], which serves as an input for the Average Overlap Frequency (AOF) algorithm [23]. The FlowSOM was built using a 10x10 grid and 10 metaclusters. We calculated AOF scores per aliquot and batch using `greedyCytometryAof()` function, with the default parameters from the *cytutils* package (https://github.com/ismms-himc/cytutils), and Scaled and Quality AOF using the formula presented in [24]. Next, we estimated the mean of Quality AOF scores taking into consideration the scores calculated for all the fcs files. We considered as outliers those files with Quality AOF scores > mean + 3 SD. Heatmaps were generated using the *pheatmap* package.

For file debarcoding we applied the `debarcode_files()` function, which uses *CATALYST* debarcoding functions (`assignPrelim()`, `estCutoffs()`, and `plotYields()`), although we introduced the possibility to include minimal separation thresholds.

File aggregation was done using an in-house function called *`aggregate_files()`* and gating for Gaussian parameters and event length were performed using *CyTOFClean* package (https://github.com/JimboMahoney/cytofclean). Additionally, intact and viable cells were gated using `gate_intact_cells()` and `gate_live_cells()` functions, which use *flowDensity* package with `deGate()` function [25]. The parameters depend on the markers gated and are discussed below.

CytoNorm package [26] was used to normalize files using a reference sample, and data were normalized without FlowSOM clustering using 5 and 95 percentile and *limit* parameter set to 0 and 8.

The batch effects were visualized using `plot_batch()` function, which performs UMAP dimensional reduction [12], from *uwot* package and *ggplot2*. Prior to dimensional reduction, files were aggregated using 1000 cells per file to reduce data size and speed up the analysis time. The *`umap()`* function was applied with default settings. Additionally `plot_marker_quantiles()` function was used to visualize the differences before and after normalization. Cell population frequencies and MSI of phenotyping and functional markers were extracted using FlowSOM algorithm [14] and used to track batch effects, as well. Briefly, 50,000 cells per file were randomly selected, aggregated and arcsine transformed. FlowSOM was built using default parameters for grid and 35 metaclusters. Phenotyping channels (see Supplementary Table 2) were used to build the model. Next, cell frequencies and MSI for phenotyping and functional markers were extracted for clusters and metaclusters. Zero-imputation was used for MSI values of clusters without cells as shown before [27]. Only the MSI values with CV > 0.2 per marker and cluster/metacluster were used. The data were further analyzed with UMAP and visualized with *ggplot2* package.

UMAP was also used for data exploration with 5000 cells aggregated per file. To map cell labels on UMAP, the aggregated file was manually gated using FlowJo software (10.0.7). FlowJo workspace was next read in R using *CytoML* [28], *flowWorkspace* [29] and *OpenCyto* [30] packages.

The figures can be reproduced using the data uploaded to FlowRepository and the code is deposited in github, however we noticed some differences when running the script either on Linux or Windows operating systems. This was due to the differences in the floating numbers generated after the 14th decimal. The R session information with package versions is attached to this manuscript as a supplementary pdf file named *Rybakowska_et_al_RSession_Information.pdf*.

3. Results and discussion

3.1. Analysis workflow

The analysis pipeline presented herein includes all the major steps necessary to process and clean collected MC data. However, it should be noted that although the presented tools significantly improve data quality, they cannot fully fix the improper design of the experiments, according to the rule *garbage-in, garbage-out*. We emphasize the importance of well-designed experiments and therefore we also present the experimental workflow (**Supplementary** Fig. 1) and protocol (Table 1), used to generate the fcs files for this manuscript. A deeper description of the sample processing protocol can be found in the supplementary methods section.

The example data set used in this work contains whole blood samples collected from two individuals (p1 and p2). EDTA-K3 blood from each donor was stimulated with 4 different stimuli (RSQ, IMQ, LPS, CpG) or left unstimulated, as described in methods section. In total 10 samples (5 per donor) were aliquoted in triplicates, fixed and frozen to generate 3 staining batches. Every batch included 10 stimulated samples plus one aliquot of the reference sample, and was barcoded, resulting in 11 samples per staining and acquisition (**Supplementary** Fig. 1). After pooling the barcoded samples, they were stained with a cocktail of antibodies recognizing surface markers, permeabilized and stained with an cocktail of antibodies recognizing intracellular cytokines (see **Supplementary Table 2**). The samples were acquired the following day, as described above. Therefore, in total 3 acquisition batches were analyzed (day 1 to day 3). This experimental workflow can be scaled up for multibarcoded, multibatch and multicenter studies (**Supplementar**y Fig. 2). Each barcoded experiment was acquired in aliquots, and 11–12 fcs files were generated each day using an NB sample injector.

For data curation we built an R-based pipeline. Although R studio and basic programming skills are required, we will point out which steps can be performed with standalone or user-friendly programs. The data analysis workflow can be seen in Fig. 1. The pipeline starts with the aliquots collected for each acquisition batch. In total, this example data set contains 35 aliquots obtained in 3 acquisition batches (12 aliquots for day1, 12 aliquots for day2, 11 aliquots for day3). These aliquots are not aggregated until step 5 (Fig. 1), as one-by-one file processing is more beneficial in the context of algorithms and computer capacity. However, if data are acquired in one big aliquot using a wide bore injector (WB), or in the case of FC or spectral cytometry (SC) experiments, and the computer resources are limited, the data can be split upfront into aliquots to generate a set of smaller files. The function to do this is provided and is called `split_big_flowFrames`. Alternatively they can be analyzed as a single big fcs file.

As a first step, a bead-based normalization is performed, followed by flow rate and signal cleaning. Next, the bad quality aliquots are detected and removed from the data set. Cleaned files are then debarcoded, resulting in generation of 11 (# barcodes) × 35 files, and at this point files from the same barcode and experimental day are aggregated. The gating of the fcs files is performed to remove doublets and dead cells and afterwards batch effect detection and normalization using the reference sample is performed. The data are further explored using dimensional reduction methods. To build this pipeline we gather already published algorithms and put them in the sequential order, that we believe is optimal for CyTOF data curation. We have also modified some steps to adjust these tools for CyTOF data or to improve their performance. These modifications will be highlighted along the workflow and some of them will be illustrated in the figures. We also provide wrapper functions for an easier use of the code. The workflow steps are prepared for CyTOF data, however as it is built in blocks, some of them can be skipped or adapted to FC or SC data. This will be also highlighted along the manuscript.

3.2. Bead-based normalization

The first step of data preprocessing is the bead-based normalization, performed for all collected aliquots. This step uses the signal of the EQ four element calibration beads acquired together with the sample [36] and the *CATALYST* package [21].

In the *CATALYST* package the function `normCytof` allows for the selection of the parameter *norm_to*, that requires an fcs file as an input. This file will be used as a baseline, hence the rest of the files will be normalized according to its values. The selection of baseline file is important for proper data normalization and can be difficult when hundreds of files need to be normalized to the same intensities. Therefore, in this pipeline we propose to aggregate all the collected fcs files in order to obtain the mean bead intensities across experimental aliquots and batches, as shown **in** Fig. 2**A**. To avoid the generation of a big object, we used `baseline_file` function that first aggregated 25,000 randomly selected events per aliquot to obtain at least 200 beads per fcs file (the number of aggregated cells is user-defined). Next, using this aggregated file we created a baseline file for which bead mean intensities were calculated and used for aliquot normalization. In this way all the files were normalized to the global mean of the aggregated file, Fig. 2**A**.

As an outcome, normalized files with suffix *beadNorm.fcs* were generated in a new subfolder called *BeadNorm*. The diagnostic plots for one aliquot were plotted in Fig. 2**B,C**. Beads were gated as negative for Ir and highly positive for the bead channels 140Ce, 151Eu, 153Eu, 165Ho, 175Lu (Fig. 2**B**). These events were further used for the estimation of the bead-derived normalization factor that will be applied to the channels selected by the users. The bead gate area can be adjusted by changing parameter *trim*, however for our samples the default parameter was good enough, as it properly gated all the necessary events. As shown in Fig. 2**C**, normalized bead intensities became higher and more stable along the acquisition time.

Another important and novel step that we introduced is the verification of the behavior of the cell markers after bead-based normalization. This possibility is provided using the `plot_marker_quantiles()` function, giving a good insight of the homogeneity of the aliquots and the batch effects present in the data. Thus, to have an insight into normalization quality and its artifacts we recommend to plot all the markers that were chosen to be normalized and verify their expression before and after normalization as shown in Fig. 2**D** for CD45RA. For this marker a slight signal decrease with the time of acquisition was observed, which was corrected by bead-based normalization. No significant differences were observed between different acquisition days for this particular marker. This part of the code does not include a user-friendly interface, although data could be also normalized using the *premessa* package (https://github.com/ParkerICI/premessa/). It should be noted that this option is not automatic, and hence the user will need to define bead gates manually.

In FC, the data are not usually acquired with spike-in beads, and thus this step is not useful for FC users. However, the advantage of using Rainbow 8-peak beads (acquired just before samples acquisition) was recently published [37] in the context of day-to-day instrument variation correction for PRECISESADS [38] project. Therefore, this step could be introduced here and is beneficial for FC data quality. Alternatively the package *flowBeads* could be also used [39,40].

3.3. Flow rate and signal cleaning

Sample acquisition using CyTOF instruments can suffer from clogs or sudden changes in the flow rate, which affect the quality of the data. Therefore, it is important to detect and clean these abnormalities. To do this we used two algorithms: *flowAI* [10] to spot flow rate irregularities and *flowCut* [41] to track signal instability. Since these two packages were initially created to clean FC data, this step can be also applied to fluorescence data with their default parameters.

The functions from these packages were adapted to the nature of the MC data and were written to wrapper functions `clean_flow_rate()` and `clean_signal()` for *flowAI* and *flowCut* respectively, although the user can still modify the original parameters.

The time resolution is different for FC/SC and MC data, thus we modified the *TIMESTEP* parameter of *flowAI* to 1 bin per 10 s. As the acquired files contained high number of events per file (around 700,000) we also increased the *Segment* parameter (the number of events per bin) from 500 to 1,000 in *flowCut* setting. In this way we could spot bigger changes in the mean but still analyze sufficient number of segments to obtain a good statistic. The settings of these parameters are sensitive to the file size (number of events collected), and should be adjusted according to the user needs. If less cells were acquired per file, less events per segment

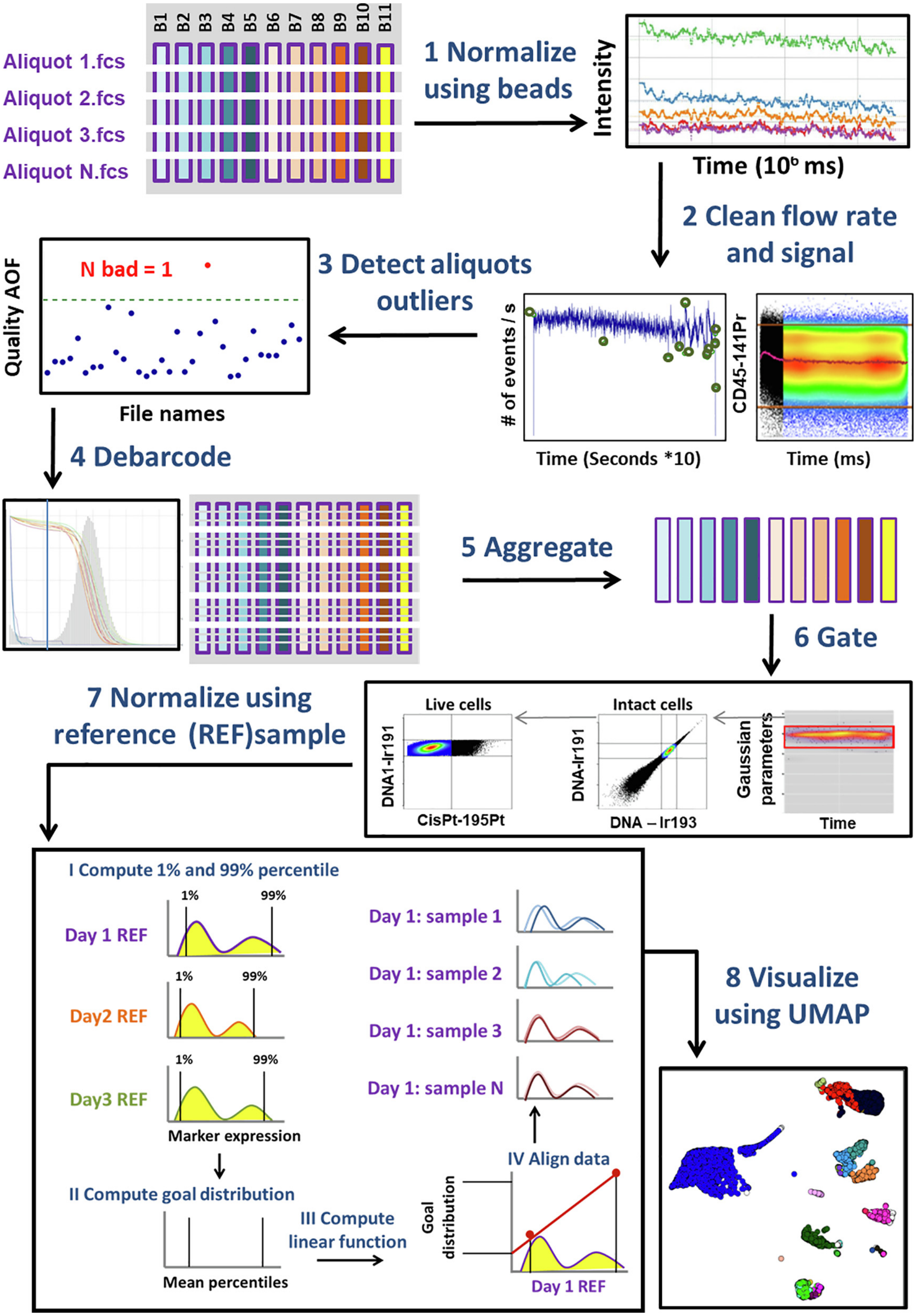

Fig. 1. Computational pipeline. Each barcoded batch was acquired in aliquots resulting in multiple .fcs files obtained from each acquisition batch. Therefore, in this figure fcs files are called "Aliquot.fcs". We represent aliquots as 1…N for clarity. The colored rectangles indicate the aliquots together with their origin and are numbered with barcodes (B) from B1-B11 (as shown in Supplementary Figure 1). The blue shade comes from p1 sample, orange shades from p2 and yellow rectangle represents the reference sample. The purple border of each rectangle denotes Day 1. Each aliquot file is processed individually until the aggregation step. Steps 1 – 8 represent the blocks used in the R pipeline. First, bead normalization (Step 1), followed by flow rate and signal cleaning (Step 2) are performed. Next, the aliquot outliers are detected by calculation of the Quality AOF score, and files with high score are discarded from further analysis (Step 3). Files are then debarcoded (Step 4) and aggregated (Step 5). Gating of Gaussian parameters and intact, live cells is performed (Step 6), and files are normalized using the reference samples collected in each batch (Step 7). In the data exploration (Step 8) cell populations are visualized using UMAP dimensionality reduction.

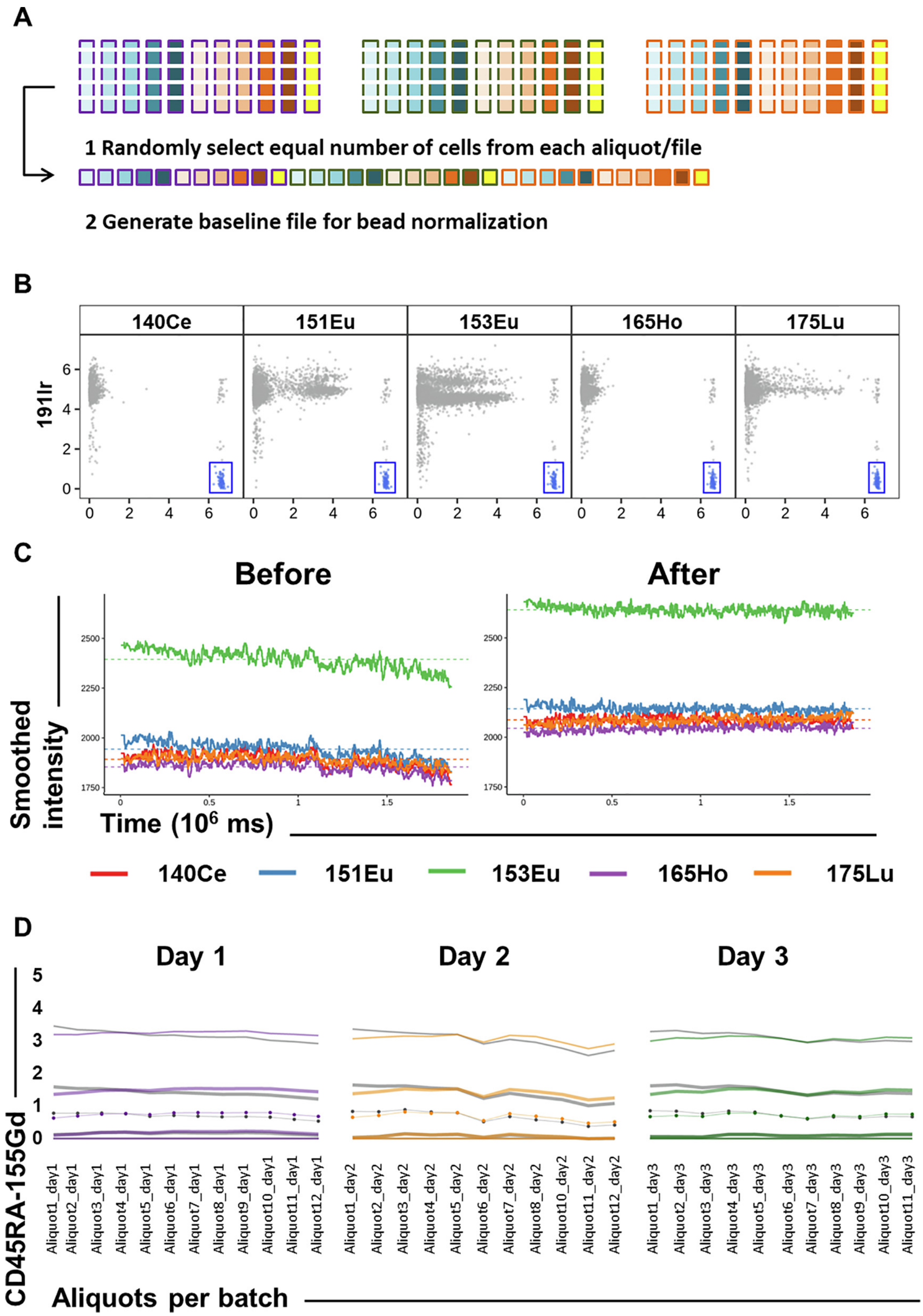

Fig. 2. Bead-based normalization. (A) Randomly selected cells from each aliquot and each acquisition day are aggregated to generate baseline files. Files are then normalized to the mean intensities of baseline file. (B) Dot plots generated using *CATALYST* package for one representative aliquot with arcsine transformed DNA channel (191Ir) versus all bead channels. The beads in the blue region are used for normalization. (C) Smoothed bead intensities in their positive channels along the time of acquisition for one representative aliquot. "Before" (left) and "After" (right) normalization data are plotted. (D) All aliquots are plotted for each acquisition day for CD45RA-155RA. The lines represent percentiles, circles represent median values, while thicker lines represent 25 and 75 percentiles. The thinner lines represent the 1 and 99 percentiles. Grey lines represent the percentiles before normalization, and colored lines after normalization.

should be analyzed. The same is applied for the removal of small or big changes in the mean signal, the bigger the segment the bigger mean changes can be removed. Thus, this parameter is data-specific and should be carefully adjusted in the case of barcoded or big fcs files. We also increased the *MaxPercCut* (the maximum percentage of cells to be removed) from 0.3 to 0.5 in order to ensure that most of the bad quality events were be removed. This parameter setting depends on the quality of the data but also on the number of events that users can tolerate to lose, and therefore it should be adjusted carefully. We set the parameter *UseOnlyWorstChannel*, and *AllowFlaggedRerun* to TRUE. This is because in the multiparametric MC data bad quality events in some channels can be missed when taking into consideration the statistic across all the channels (as it is the case in *flowCut*). To be stricter in the cleaning, we enabled this parameter. As it can occur that some other channel will have signals severely affected and will not get cleaned when a predefined channel is selected, we allow the algorithm to re-run the cleaning after the first bad quality events are removed. The setting of these parameters will depend on the data quality and the number of markers used. We also forced the algorithm to always clean the data, by setting *AlwaysClean* parameter to TRUE, as we did not observe any excessive cleaning in high-quality files; instead some small signal disturbances were efficiently cleaned. We did not observe any benefits in changing other parameters, and therefore the rest of them were set as default. These tools were firstly created for FC studies, thus same reasoning about the parameter setting can also be applied to high-dimensional fluorescent cytometry data. If FC or SC data are run the *data_type* parameter should be changed to ‘‘FC”. This will automatically switch algorithms parameters to the original FC-optimal setting.

This step generated a new folder called *Cleaned,* containing clean files with suffix *cleaned.fcs*. Additionally, two subfolders were created called *FlowRateCleaning* and *SignalCleaning* having one plot per file for both flow rate and signal cleaning, as shown in Fig. 3. These plots are convenient to check the quality of the signal across the markers and flow rate, and also the level of the cleaning achieved. Therefore, we recommend verifying if all the low-quality events are removed from the data, and to re-adjust the parameters when required. Example plots for flow rate and signal cleaning in one sample aliquot are shown in Fig. 3**A and 3B**. It can be observed that the anomalies occurred just at the beginning of the acquisition, when probably the flowrate was still unstable, and also at the end of the acquisition when more fluctuation typi-

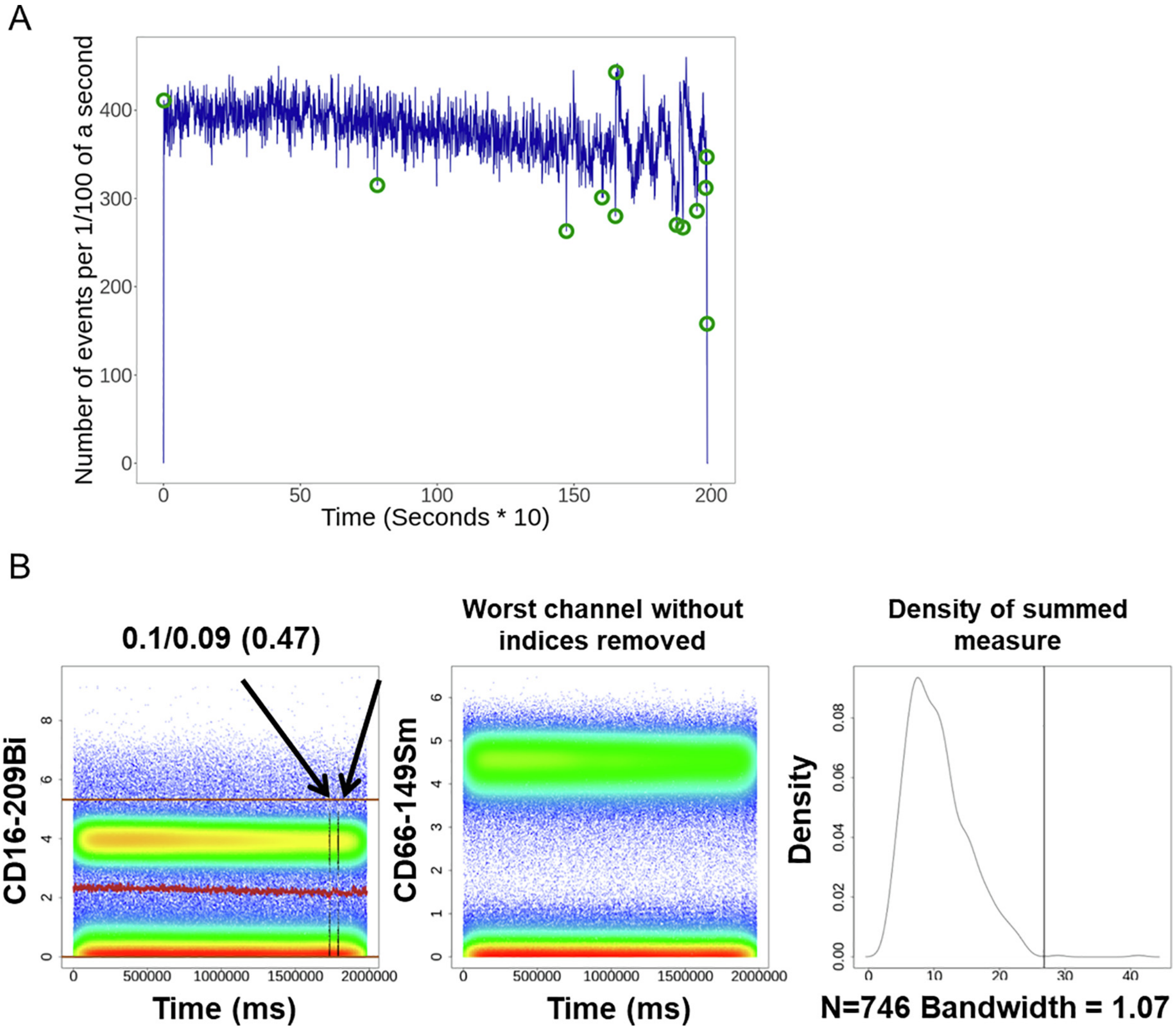

Fig. 3. Signal and flow rate cleaning. (A) Flow rate of one representative aliquot using *flowAI*. The green circles indicate the outliers that were detected and removed from further analysis by the *flowAI* algorithm. (B) Signal cleaning for the same aliquot using *flowCut*. The first two dot plots show the arcsine transformed intensity for cleaned CD16-209Bi and CD66ace-149Sm, detected as the worst channel. For CD16 plots, the arrows indicate the indices removed from the analysis, and the numbers above the plots the mean change before and after cleaning and the maximum mean change for this aliquot. On the right panel, histogram representing the summed measures of mean, median, several percentiles, skewness, and variation of the flow signal, parameters used by the *flowCut* algorithm. The vertical line represents the threshold for event removal.

cally occurs. This can be a sign of some cell clogging, sample degradation or the tube getting empty. As can be seen in Fig. 3**B**, overall a good quality signal was observed even in the worst channel (CD66ace). The indices removed at the end of the signal correspond with the flow rate abnormalities, confirming some problems at the end of the aliquot acquisition.

flowAI can be run in a user-friendly mode, and therefore no programming skills are required. On the other hand, *flowCut* can only be used through R language. Alternatively, *flowAI* can be used as a signal cleaning tool, however in our experience it is too stringent for MC data because it removes too many acceptable events (data not shown).

It can be argued that some events can be removed from the data due to their original properties. However, cells acquired either in MC or in FC/SC are homogenously distributed in the sample and thus across the "Time" parameter during acquisition. As those algorithms divide data into bins, and each bin represent the mixture of all cells, it is highly unlikely that some specific type of cells will be removed from the file.

3.4. Aliquot outlier detection

As mentioned before, in order to obtain a sufficient number of high-quality events during long acquisitions, barcoded data are acquired in aliquots. For this reason, sudden changes in detector sensitivity or problems with instruments like unexpected shutdown can happen, requiring additional tuning. Thus, it is advisable to verify the quality and channel intensities of the different aliquots acquired in each acquisition batch. To do this, we took advantage of AOF algorithm [23] that detects potential staining problems [24].

It is recommended to use AOF algorithm on gated or clustered events as input. In order to allow for automated quality check, we used FlowSOM [14] algorithm (**Supplementar**y Fig. 3A) to cluster the cells. Upon clustering (using the phenotyping markers shown in **Supplementary Table 2**), the AOF scores were calculated for each marker as shown in **Suppleme**ntary Fig. 3B and scaled to obtain Scaled AOF scores as illustrated in Fig. 4**A** and described in [24]. These scores were summed for each aliquot, and the Quality AOF score was obtained for each fcs file as shown in Fig. 4**B**. To detect outlier aliquots the mean for all the Quality AOF scores was calculated per all files. The files with Quality AOF score > mean + 3SD were considered as outliers and removed from further analysis.

If the data were acquired as single big barcoded samples we recommend to divide these files into segments and calculate the AOF and Quality score for each segment in order to spot signal decrease issues. If the user decides not to split the data, we advise to carefully inspect the files with high quality scores and markers with high AOF score across time. It can happen that upon the long acquisition only a portion of the data is affected, thus some good quality part of the data can be used for further analysis, as shown before [24].

In this setting we use AOF to detect abnormalities in the aliquots. All the aliquot files have exactly the same cell distribution, and thus removing one outlier aliquot from the data does not remove any specific cell subpopulation. However, this algorithm can also be used to spot staining discrepancies (missing antibodies, lower staining index) across samples from different individuals or acquisition batches [23], and in this case the results should be carefully checked to make sure that the observed low quality scores are due to technical problems and not to sample-specific or cell-specific phenotypes. We also focused on phenotyping markers and not functional, cytokine markers, as the performance of this algorithm is optimal when using bimodal markers. By doing so we assumed that if some phenotyping markers are affected and sample was scored high, most of the markers will have poor resolution and will be difficult to analyze. For FlowSOM we used a small number of metaclusters (N = 10) in order to detect and validate the marker expression in the main leucocyte populations across homogenous aliquots. We reasoned that the less metaclusters, the less aliquot-specific clustering will occur, and thus all the cells will be assigned to the same metacluster and the AOF calculation will be performed exactly for the same group of cells. FlowSOM parameters can be adjusted according to the sample diversity and user needs. To verify the quality of clustering, it can be useful to plot the FlowSOM tree, as shown in **Supplementary** Fig. 3A. The threshold can also be adjusted by the users, if necessary. We chose FlowSOM as a clustering method as it is the fastest and most accurate algorithm [42], however users are free to choose their own clustering algorithm to partition the data using the `greedyCytometryAOF()` function from *cytutils* package and next to calculate scores with functions `scaled_aof_scores()` and `file_outlier_detecion()` from our pipeline.

This pipeline block is only available as R code. FC/SC data can also be analyzed, however, the arcsine transformation should be changed to a cofactor of 150 in the function `fsom_aof` and *aof_scoring*.

3.5. File debarcoding

The samples included in this data set were barcoded with 11 different palladium-based barcodes. Therefore, in order to recover the individual sample information, we performed a data debarcoding step, using debarcoding functions from *CATALYST* package with the automated separation threshold identification setting. The generated files were stored in a new subfolder called *Debarcoded* with the suffix *debarcoded.fcs*. Additionally, we introduced a minimal separation cutoff of 0.18, which ensured a safe separation threshold for files with unclear barcode distribution. File names with detected threshold values lower than 0.18 can be stored in an RDS file called *files_with_lower_debarcoding_threshold.RDS* if the parameter *less_than_th* is set to TRUE. The minimal separation threshold can be modified if necessary.

To verify the quality of debarcoding two plots were generated for every debarcoded sample using *CATALYST* package, as shown in Fig. 5. The visualization of the separation between events positive or negative for the different Pd isotopes (see Fig. 5**A**) is key to assess the correct sample assignation during debarcoding. The second plot allows the monitoring of the yield and cell count obtained with the chosen minimal separation cutoff (Fig. 5**B**). We recommend reviewing all the plots generated upon debarcoding especially if the file names are stored in the *.RDS* file due to the assignation of separation threshold below to the minimum established.

In this protocol we used a 6-choose-3 scheme, resulting in maximum of 20 barcodes, and for the purpose of this work presentation we used 11 barcodes. However other approaches could also be taken, like 7-choose-3 (35 barcodes), CD45-based cadmium barcoding scheme recently offered by Fluidigm, or the combination of commercial barcoding with monocisplatin isotopes (60 or even more barcodes) [32]. The parameter *bc_key* that defines the barcoding scheme would need to be re-designed by the users and adjusted to new isotopes. It should be noted that Zunder-based algorithm used in *CATALYST* package will correctly deconvolute the data if positive and negative populations are present, thus barcoding with serial dilution of amine reactive fluorescence dyes commonly used in FC multiplexing will not work in this case. For the FC/SC users we recommend to deconvolute the data by manual gating or using *flowClust* algorithm as previously described [43].

In our experience when using the barcoded approach presented here some barcode intensity problems can be noticed when dead

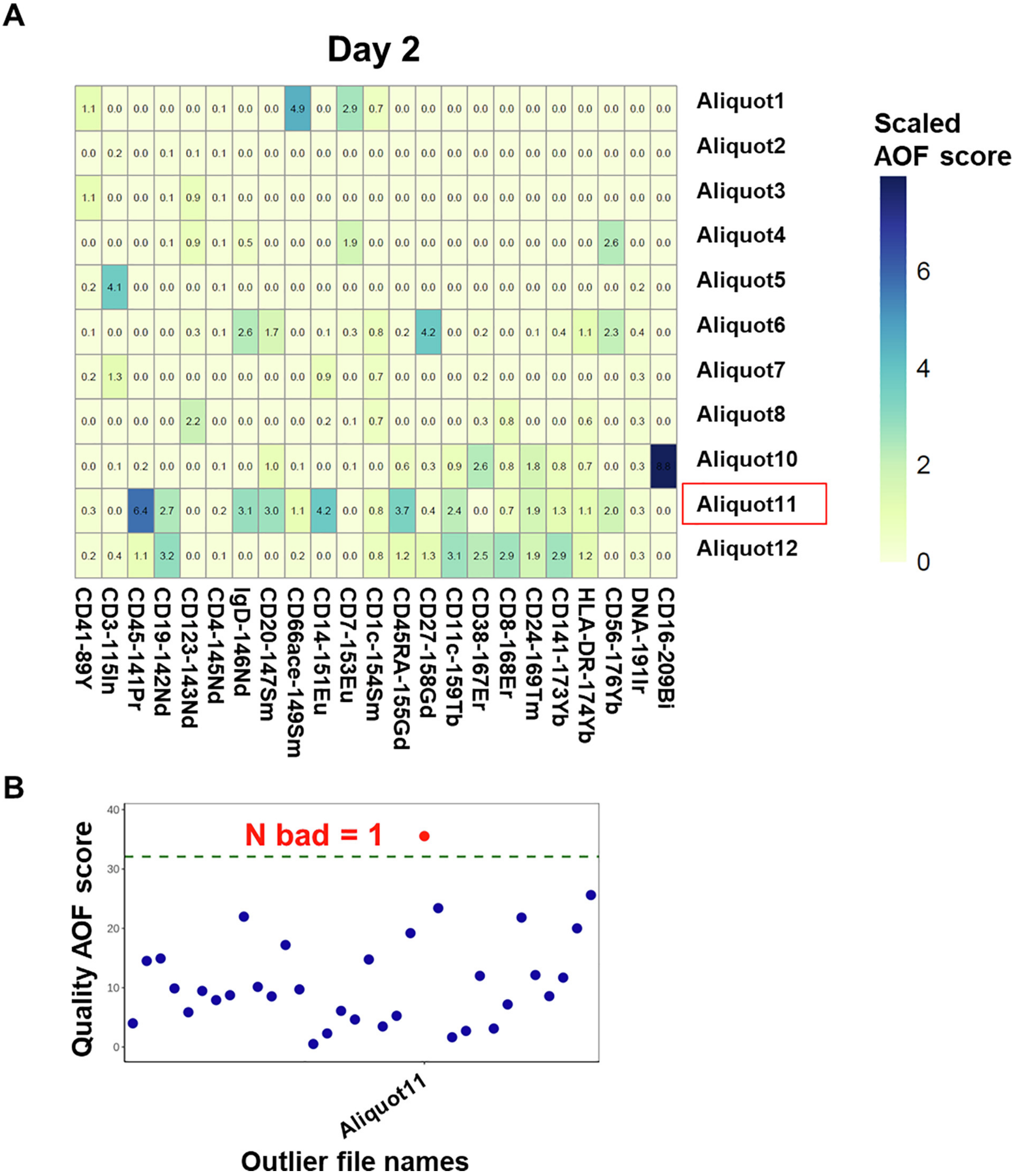

Fig. 4. Outlier aliquot detection. (A) Heatmap representation of Scaled AOF scores for each surface marker across all the aliquots collected on day 2. (B) Outlier aliquot detection. On the x-axis the file names marked as an outlier (aliquot 11 from day 2) and on the y-axis Quality AOF scores. The green dotted line represents the threshold for outlier definition. Dots represent scores for each aliquot, red dot is a file above the threshold.

cells and debris were present in excess (data not shown), a fact also observed before [33]. As reported [44], small polymer-palladium conjugates have high affinity to dead cells. Furthermore, a high amount of debris capture barcoding reagents and reduce the amount of complexes able to stain the cells of interest [4]. It is noteworthy that labeling with this barcoding reagent is sensitive to the number of cells, thus titration with the exact cell number is required. Accordingly, if a large amount of debris is present in the sample a reduction of the total amount of cells is recommended [4], alternatively if a lot of cells are available samples can be cleaned by a dead cell removal kit before barcoding. Additionally, the use of saponin-based reagents requires sample fixation, which could be problematic when live cells are studied or using fixation-sensitive markers. Alternatively, palladium barcoding using surface markers can also be considered [33–35]. In case of low separation between barcoded samples, manual gating could also be performed, alternatively clustering with cluster number equal to barcode number could also be tested [24].

A user-friendly application for debarcoding is provided by Fluidigm as a part of CyTOF software. However, it should be noted that

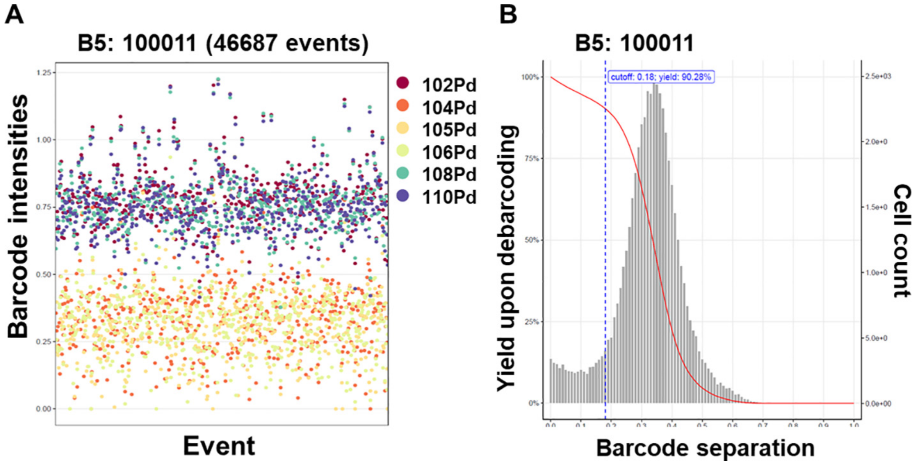

Fig. 5. Debarcoding quality plots. (A) All cell events assigned to barcode B5 (100011) are shown, with each cell event represented as a dot. The intensities for each palladium isotype are displayed with different colors. (B) Distribution of the events by separation distance (grey histogram) and the cell yield (red line) are displayed as a function of the separation threshold. The vertical, blue dotted line represents the cutoff for the separation threshold. The data for one representative aliquot is shown.

this software estimates the separation threshold for all the files after aggregation, in contrast to the method proposed in this workflow. This global threshold is not always correct for all the samples and can lead to precious event loss. Additionally, it is only dedicated to palladium, as barcoding channels cannot be manually defined. Instead the *premessa* package (https://github.com/ParkerICI/premessa/) could be used, allowing the definition of barcodes in a user-friendly interface.

3.6. File aggregation

We performed file aggregation of the debarcoded aliquot files using an in-house function called `aggregate_files()`. To do this, the names of the files containing metadata (barcoding identity and staining batch) need to be provided as shown in the Supplementary Table 1. The resulting aggregated files were stored in a folder called *Aggregated* and they contained the sample-specific names provided in the metadata. If files were barcoded and acquired as one big aliquot (for example while using the WB injector or in FC or SC) and the computer power resources are limited we recommend to split them into smaller fcs files, preprocess, and then aggregate after the debarcoding step, as shown in this workflow. If the data do not require aggregation this part of the code can be skipped.

3.7. Cell gating

The aggregated files contained a mixture of events including dead cells, debris and doublets, which should be removed from further analysis. To do this we took advantage of *CyTOFClean* package (https://github.com/JimboMahoney/cytofclean) and *flowDensity* [25].

CyTOFClean is a user friendly R package that excludes doublets using event length and 4 Gaussian parameters: Center, Offset, Residual and Width (see Fig. 6**A**), as described in [45]. Using the *flowDensity* package, the gating of intact cells is also provided with `gate_intact_cells()` function, and is based on the expression of DNA1 and DNA2 marker. Additionally the live/dead cells gating was applied by calling the `gate_live_cells()` function. Both gating strategies are shown in Fig. 6**B**. In this example CisPt and ^{195}Pt channel were used to detect dead cells, but users can change these parameters accordingly to the live/dead marker chosen. This part of the code does not have graphical user interface (GUI) implementation, however the fcs files can be analyzed using standard softwares as FlowJo, and the resulting files could be exported to R. Alternatively, the packages *flowWorkspace*, *CytoML* and *openCyto* can be used to import the gating settings into R [29]. As an output *CC_gated.fcs* files and control plots were generated for each file in the *Gated* folder.

It is important to note that *CyTOFClean* is a closed GUI app and therefore no parameter can be adjusted without changing the application code. Inversely, all the parameters can be adjusted for *flowDensity*-based gating. The parameters that we modified for the analysis of this example dataset were *upper* and *use.upper* that define which lower or upper inflection points of the density curve are analyzed and also *alpha* and *tinypeak.removal*, that specify the significance of the change in the slope being detected and the inclusion/removal of tiny peaks in the density distribution respectively. The change in *alpha* parameter will affect the tightness of the gate, and thus can be useful if a less strict gate for the Ir channels is necessary.

In this workflow we introduced the common gating strategy used for nucleated cells stained with Ir. However, the gating strategy could change depending on the type of cells acquired and the antibody panel configuration. This statement especially applies to the parameter “Event length” and “Ir” channels. The event length of each cell depends on the amount of metal that the cell is loaded with, thus the non-nucleated cells with only one probe have lower event length than nucleated cells bound to a high number of probes. *CyTOFClean* package uses a density function to apply the cutoff for both event length and Gaussian parameters, thus it will gate on the cells where the majority of events are located as recommended for these gates [46] and rather undergate than overgate. Therefore it is suitable for NB and WB injectors despite their known differences in the event length [47]. Additionally, it introduces the thresholds 10 and 50 to exclude events below and upon these values respectively. Most of the cells are typically located in between these values, unless some contamination or low metal content cells are present. Therefore the users should verify the event length of the cells of interest before the gating. In the case of non-nucleated cell analysis, the gating strategy for intact and live/dead cells should be changed and re-designed to fit the users’ needs. The gating presented here is especially useful for human leukocytes stained with Ir, however it could also be applied to other type of cells like mouse splenocytes.

The gating using “Event length” and Gaussian parameters will remove the doublets caused by fusion of ion clouds. However,

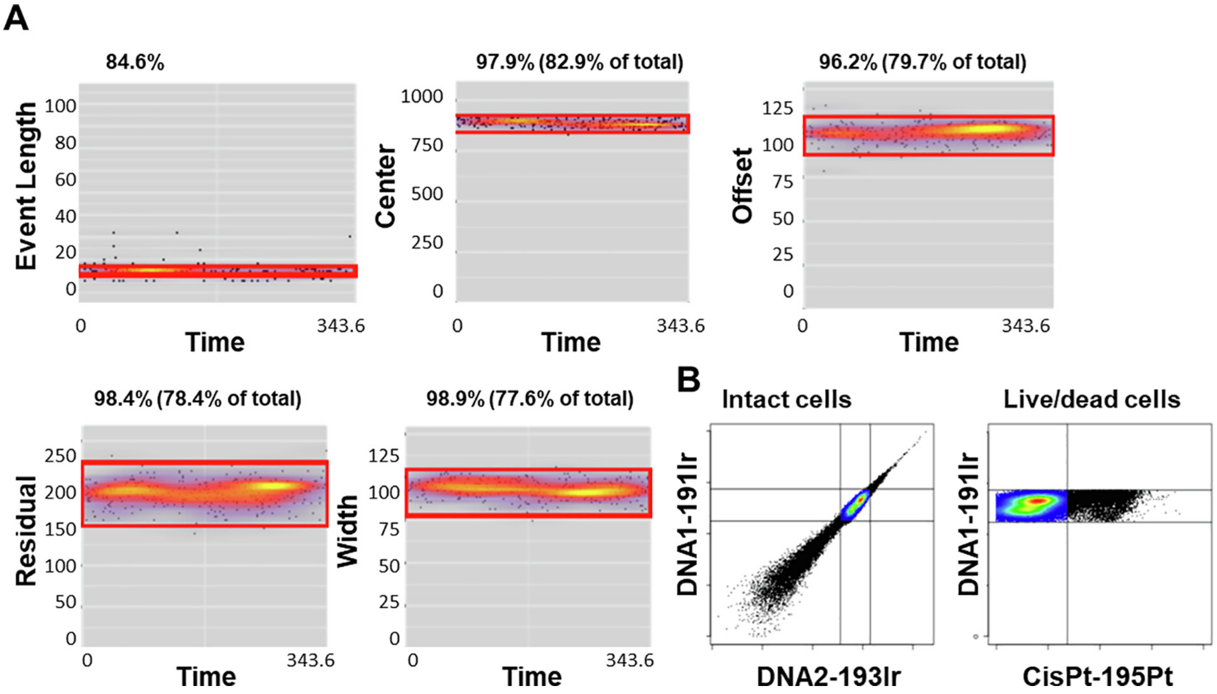

Fig. 6. Cell gating. (A) Doublet exclusion with *CleanCyTOF* package. Event length and Gaussian parameters are represented against "Time" parameter. Events that fall outside of the red gates are discarded form further analysis. (B) Gating for single intact cells and live/dead cells using DNA and CisPt parameters and *flowDensity* package. The black events are removed from further analysis. One representative file is shown.

the true cell aggregates will still remain in the fcs files. Barcoding with more than 2 probes per sample improves the quality of the data and helps to remove cellular aggregates when cells are coming from different barcoded samples. Unfortunately, aggregates within the same sample will not be cleaned, neither will be debris. Thus, to deal with this we used "Intact cell" gate to remove Ir^{low} events (debris) and Ir^{hi} events (aggregates). Cells late in the S, G_2 and M phases of the cycle can fall in the region of Ir^{hi} cells, due to their higher DNA content [48] and thus this algorithm can result in undesired cycling single cell removal. To avoid it we use a less strict gate at the upper part of Ir intensity. The use of markers specific to cell cycle in the antibody cocktail and automated clustering could also be helpful [48]. When using a less strict gate, care needs to be taken when analyzing the data, as some cell aggregates can still persist in the data and could be spotted upon gating or clustering, together with marker expression analysis. This gating is specific for MC data, and thus FC/SC users would need to apply their own gating strategy at this point. A good example of how to do it using R is published [49]. As mentioned before, manual gating can be alternatively done using programs like FlowJo or FCS Express and then the population of interest can be imported into R. However, this can be time–consuming, especially when hundreds of files are processed. It is also known that manual analysis can be biased [50], and hence it is faster and more robust to use automated approaches. Nevertheless, we recommend to always revise the quality of the gating and optimize the parameters if necessary and possible.

3.8. Batch normalization using reference samples

Although bead-based normalization helps to normalize different acquisition batches, it uses the information of a limited set of channels. The correction factor is calculated using 5 bead channels and then extrapolated to the rest of the markers, hence there is a risk that some markers and cell-specific changes are not precisely corrected. Even more, normalization using beads does not consider the experimental variation introduced during sample staining or manipulation (batch effects). Therefore, normalization with a reference sample that is present during sample staining and spans the whole spectrum of the channels used, is advisable. Because of this, as a last step in the preprocessing and quality control pipeline, we performed batch normalization using the *CytoNorm* package [26].

In contrast to *CytoNorm* default setting, instead of using quantile normalization (parameter *quantileValues* = 101) we computed 5% and 95% percentiles across the reference samples for each marker and obtained the marker-specific goal distributions using the function `QuantileNorm.train()`. This was because quantiles normalization introduced artifacts (data not shown), as previously reported [51]. In our experience this setting was good enough for the correction needed, although in some particular cases the distribution of the signal will not be modelled adequately if only two quantiles are used. In this case the users can change *quantileValues* parameter accordingly. Additionally, the normalization was performed on unclustered data, as clustering can be biased by the variation potentially introduced in the raw data. If a big batch effect is observed upfront in the markers used for clustering, improper cluster assignation can also introduce artifacts. We set the parameter *limit* to 0–8, as recommended by the authors, to avoid the introduction of extreme values, like negative values that are normally absent in MC data. We normalized our data to the "mean" marker quantiles of reference sample, using *goal* parameter. However, this can be changed by the users and data can be normalized to one of the batch values or to specific quantile values. The goal distributions obtained in the first step were then used for marker-specific batch normalization using the function `QuantileNorm.normalize()`. The scheme of the normalization process can be seen in Fig. 1**, step 7** and Fig. 7**A**.

As outputs, fcs files with the new prefix *Norm* were generated together with several diagnostic plots. These files were stored in the new subfolder *CytofNormed*. An example of generated plots are shown in Fig. 7 for the marker CD4 for the three days of acqui-

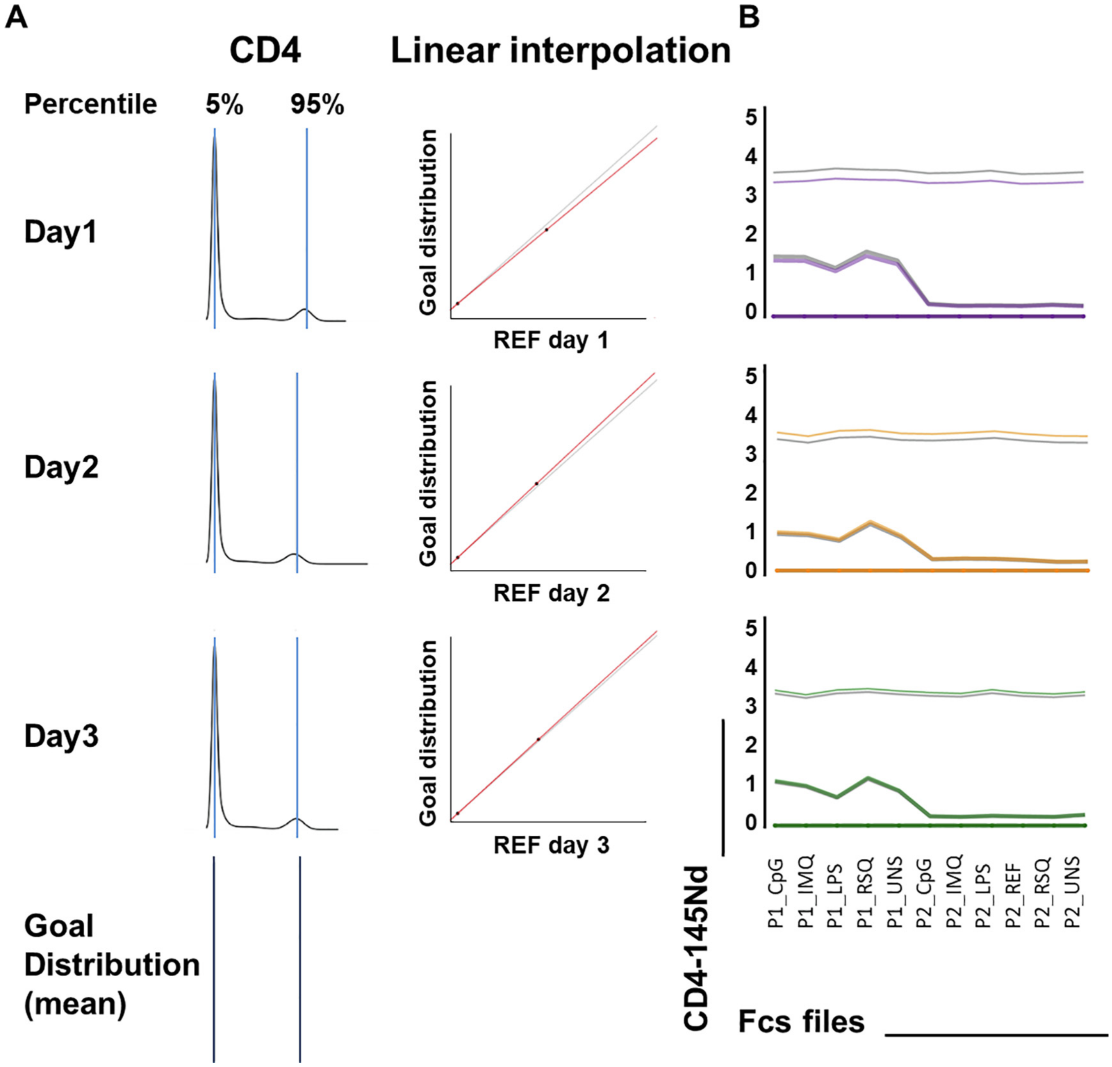

Fig. 7. Normalization based on reference samples. (A) Distribution of the CD4 marker in the reference sample on each staining day. Left panel, the 5% and 95% percentiles are represented by the blue vertical lines and the computed means for 3 experimental days are represented below in dark blue lines (goal distribution). Right panel, the linear function that transforms the result is represented with a red line and the grey line represents the identity function. If the red line is below the grey one, the values in the batch are decreased, and vice versa. (B) CD4 expression in the individual samples acquired in the three staining days are represented as in **Figure 3C**, 99% and 75% percentile lines can be seen. The rest of the quantiles have 0 values and thus are located at point 0 on the y-axis.

sition. A shift in the 95% peak position can be seen especially on day 1. The mean for each intensity peak was estimated using data from the 3 days. This mean is called a goal distribution and was used to obtain normalized data distribution as shown Fig. 7**A**. Furthermore, we provide a plot showing the expression of CD4 across samples by day of acquisition (Fig. 7**B**). A downshift of the 95% percentile colored line can be seen for day 1 when compared to the 95% grey line (normalized). In contrast, a slight upshift can be noticed for day 2 and 3, giving comparable median intensities after normalization when looking for all 3 days. These correspond with the shifts seen for the density peaks. In the same way the plots for IL-6 are shown in Supplementary Fig. 4A, as an example for a functional marker normalization.

Batch effects are commonly monitored using dimensionality reduction methods, which allow the visualization of the cell distribution in each file in a single plot. In our pipeline we also provide the possibility to run UMAP dimensional reduction [12] as shown in **Supplementary** Fig. 4B. It can be appreciated that the cells from day1 were unevenly distributed before normalization, however the homogeneity of the cell distribution was improved after normalization and cells become uniformly mixed between acquisition days. The same can be observed when cell frequencies for clusters or metaclusters were extracted using FlowSOM, followed by dimensional reduction analysis (see **Supplementary** Fig. 4C and 4D). It can be noticed that the batch effect is stronger at the cluster than at the metacluster levels, thus if researchers want to perform sample comparison in more detail, special care needs to be taken for batch effect adjustment. In general, no strong batch effect was observed in UMAP or cell frequency analysis where only phenotyping markers were used. To take a deeper look into functional markers and features, we extracted median marker intensities (MSI) per cluster and metacluster and visualized sample distribution by dimensional reduction, Fig. 8**A (**all the markers**), B** (functional markers) and **Supplementary** Fig. 4E. In Fig. 8**A** it can be appreciated that the day 1 samples were differentiated from the rest, including even the reference sample, but upon normalization they got intermixed with the corresponding donor samples. The references also got into closer proximity. In this experiment, the blood was treated with 5 different conditions, hence sample grouping in accordance to the stimulation and not only to donor is

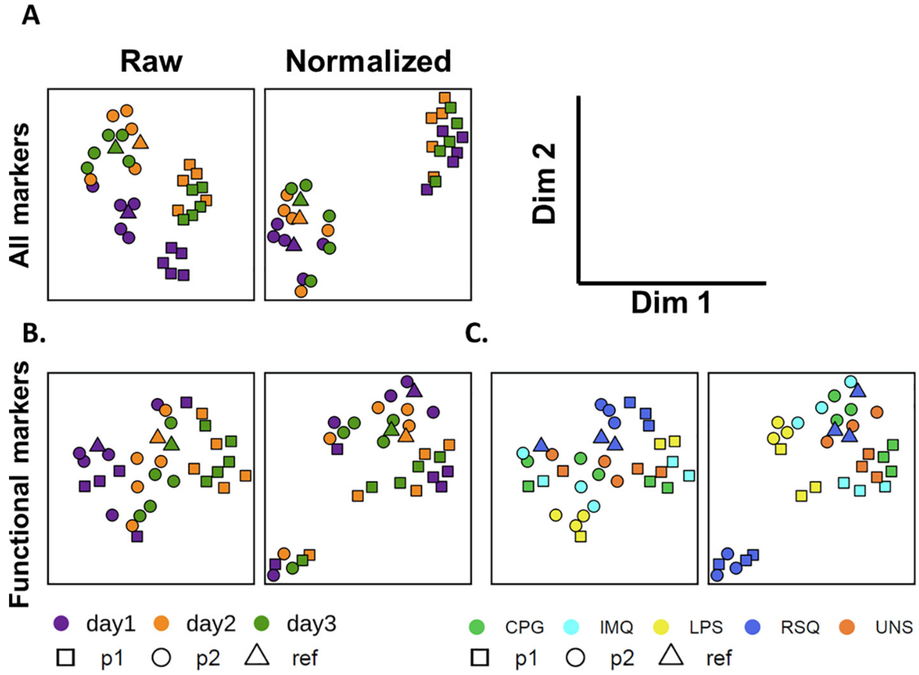

Fig. 8. Batch effect visualization using MSI values of phenotyping and funcional markers. Data were clustered using FlowSOM and MSI for all phenotyping and functional markers were extracted across all metaclusters. Next, dimensional reduction using UMAP was performed to verify sample distribution and data were represented using scatter dot plot where x and y-axis represents 1st and 2nd dimension of UMAP. (A) MSI for all the markers, (B, C) MSI for functional markers colored either by the day of acquisition or by the stimulation, respectively. Samples are colored by staining batch (A) and (B) or by stimulation (C). Donors and reference sample are represented with shapes.

expected. In panel B, grouping according to the donor and to the stimulation was observed after the normalization, but not before. These results showed that MSI is more sensitive to the experimental variation as expected, and it can be tracked and corrected when reference samples are included.

These results underscore that batch effect correction is a necessary step when multiple batches are acquired, even in well controlled experiments, and especially when fine phenotyping and MSI values are of interest. In this example we used a reference sample that was always barcoded and stained along with the samples, but spike-in cells could also be used [52]. However, it should be noted that spike-in cells should be processed using the same experimental protocol and should be stained with one specific channel or antibody to allow for their identification, thus the sacrifice of one channel should be taken into consideration. If this solution is considered, additional gating steps would need to be performed to identify and split the reference file. The reference sample is not commonly used in FC or SC, however this normalization could be also applied to fluorescence data and should improve the quality of the data in high-scale and longitudinal studies. The transformation cofactor should be changed in the *transformList* parameter and the *limit* should be set to default settings or adjusted to FC/SC data.

The *CytoNorm* function can be used as a plugin in FlowJo, as well as UMAP. Alternatively tSNE [53] can also be used for data exploration and detection of batch effects using platforms like Cytosplore [54]. The values for quantiles can be adjusted if necessary and FlowSOM clustering could also be introduced when no strong batch effect on phenotyping markers is observed. The extraction of cell population frequencies and MSI is a good way to track the differences between acquisition batches. In this example we used FlowSOM clustering, but any other tool or manual gating can be used.

3.9. Data exploration

Finally, we explored the cleaned and normalized data using a dimensional reduction method. In this example we choose UMAP due to its good performance and ability to handle a large number of events in a relatively short time [55]. To speed up the analysis we aggregated 5000 cells per fcs file and performed a dimensional reduction using the phenotyping markers as input (see **Supplementary Table 2**). In total 165,000 cells were used for the analysis. This allowed us to track marker expression across the studied individuals as shown in **Supplementary Fig. 5**A and to map them to the manually-gated populations as shown in **Supplementary Fig. 5**B. The manual gating strategy used can be found in **Supplementary Fig. 6**. Due to this approach we could explore the differences in sample frequencies (Fig. 9**A**) and MSI of cytokine markers (**B, C**). In panel A the density of the different populations can be visualized, showing differences in the frequency of several subsets between p1 and p2. Additionally, differences in cytokine expression could be detected. For instance, after RSQ stimulation MIP1β was more expressed in p2, and the expression was brilliant in monocytes and pDC and moderate in CD16$^+$CD66ace$^+$ granulocytes (Fig. 9**A**). TNFα was expressed in pDC and some subsets of the monocyte compartment, and the intensity of TNFα was higher in pDC than in monocytes for p1, while this ratio was inversed in p2 (Fig. 9**B**).

This basic data exploration can already give an insight into the internal diversity of the samples and give a first idea of the individual differences, although further analysis should be performed to verify the hypothesis generated before the experiments and upon data exploration. Data analysis and interpretation is dependent on the biological question raised, as was already discussed in different publications [17,54].

4. Preconditions, limitations and conclusions

Here we report an R-based data curation workflow that cleans collected data and corrects the experimental variation introduced during the sample preparation, staining and acquisition. This pipeline is semi-automated and optimized for large studies involving

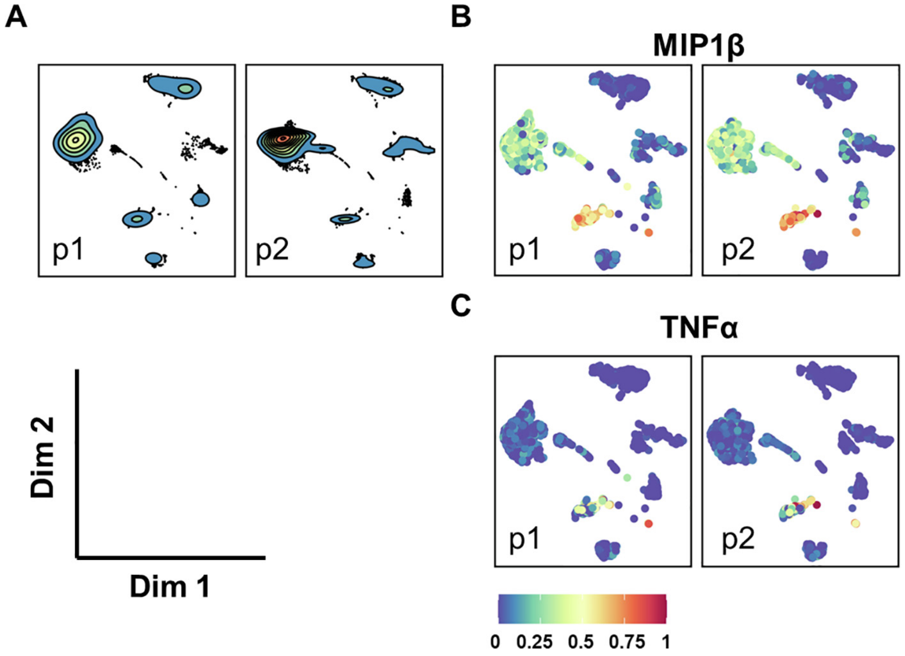

Fig. 9. Data exploration by dimensional reduction using UMAP. One thousand cells per file were aggregated. The UMAP was built with default parameters using phenotyping markers and scale set to TRUE. Samples stimulated with RSQ are visualized for 2 donors on day 1. (A) Equal number of cells are visualized by density plot for donors p1 and p2. (B, C) Dot plot showing 0-1 normalized, arcsine transformed expression of the cytokines MIP1β (B) and TNFα (C) for p1 and p2.

human blood surface phenotyping together with functional markers. To our knowledge this is the first data preprocessing pipeline that gathers and optimizes all the tools necessary for the processing of MC data. It is especially useful for multibatch studies and can be also applied to multicenter settings. Since our experimental setting uses fixed and frozen whole blood samples, this pipeline is useful for retrospective multicenter studies, as discussed in [6]. The example data set and the code are provided, thus each step can be reproduced. Although the data set provided here is limited in size, it is big enough to show the advantage and usefulness of each tool.

The analysis is performed in R environment without a user-friendly interface (except for part of the gating), thus basic programming skills are necessary and knowledge of R environment is mandatory. Although the workflow is presented as a script, we gathered all the tools into easy-to-use functions with detailed descriptions, thus we believe that inexperienced R users will be able to follow the steps and analyze their own data.

The function parameters are optimized for cellular studies, especially whole blood immune phenotyping studies by MC, but different samples, such as bone marrow aspirates, PBMC or mouse/human splenocytes can also be analyzed. This might require parameter adjustment that can be performed by the users. As in our gating strategy we are using the Ir parameter to detect genomic DNA, only nucleated cells are analyzed. For the batch correction it is mandatory to use the same reference sample stained and acquired on each acquisition batch together with the samples. Otherwise the normalization proposed in this manuscript cannot be achieved, and precious information contained in MSI should not be analyzed. This pipeline is specifically dedicated to MC data, however, some steps can also be used for FC and SC experiments with the adjustment suggested along this manuscript. Some tools can be sensitive to the size of the fcs files processed, therefore we recommend to acquire data in aliquots, especially when the acquisition is performed in low ionic environments like water, or split them just before the preprocessing. This pipeline gives some insight about data exploration, but does not fully cover deep data analysis. The final interpretation of the data will depend on the question raised in every individual project.

CRediT authorship contribution statement

Paulina Rybakowska: Conceptualization, Methodology, Investigation, Writing - original draft, Visualization. **Sofie Van Gassen:** Methodology, Investigation, Writing - review & editing, Visualization. **Katrien Quintelier:** Investigation, Writing - review & editing. **Yvan Saeys:** Writing - review & editing. **Marta E. Alarcón-Riquelme:** Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Concepción Marañón:** Conceptualization, Methodology, Resources, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could influence the work reported in this paper.

Acknowledgments

PR, MA, CM acknowledge support from the IMI2-JU project GA No 831434 (3TR) and IMI-JU project GA No 115565 (PRECISESADS). The JU receives support from the European Union's Horizon 2020 Research and Innovation Programme and EFPIA. PR received support from EMBO (7966) short-term fellowships and from Consejería de Salud y Familias de Junta de Andalucía (EF-0091-2018) to perform 3 and 2 month internships, respectively, at the VIB-UGhent. The authors also acknowledge funding from Consejería de la Salud y Familias de la Junta de Andalucía (PIER-0118-2019) and Instituto de Salud Carlos III (PI18/00082), partly supported by European FEDER funds. These results are part of the the PhD thesis in Biomedicine at the University of Granada of PR. We are grateful to Olivia Santiago and Jose Díaz Cuéllar for technical support as a Cytometry Core Facility in Genyo research center.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2021.05.032.

References

[1] Leylek R, Alcántara-Hernández M, Lanzar Z, Lüdtke A, Perez OA, Reizis B, et al. Integrated Cross-Species Analysis Identifies a Conserved Transitional Dendritic Cell Population. Cell Reports 2019;29(11):3736–3750.e8. https://doi.org/10.1016/j.celrep.2019.11.042.

[2] Shi W, Liu X, Cao Q, Ma P, Le W, Xie L, et al. High-dimensional single-cell analysis reveals the immune characteristics of COVID-19. Am J Physiol-Lung Cell Mol Physiol 2021;320(1):L84–98. https://doi.org/10.1152/ajplung.00355.2020.

[3] Neidleman J, Luo X, Frouard J, Xie G, Hsiao F, Ma T, et al. Phenotypic analysis of the unstimulated in vivo HIV CD4 T cell reservoir. ELife 2020;9:e60933. 10.7554/eLife.60933.

[4] Zunder ER, Finck R, Behbehani GK, Amir E-AD, Krishnaswamy S, Gonzalez VD, et al. Palladium-based mass tag cell barcoding with a doublet-filtering scheme and single-cell deconvolution algorithm. Nat Protoc 2015;10:316–33. 10.1038/nprot.2015.020.

[5] Bendall SC, Nolan GP, Roederer M, Chattopadhyay PK. A deep profiler's guide to cytometry. Trends Immunol 2012;33(7):323–32. https://doi.org/10.1016/j.it.2012.02.010.

[6] Rybakowska P, Burbano C, Gassen SV, Varela N, Aguilar-Quesada R, Saeys Y, et al. Stabilization of Human Whole Blood Samples for Multicenter and Retrospective Immunophenotyping Studies. Cytometry Part A n.d.;n/a. 10.1002/cyto.a.24241.

[7] Jang JS, Juran BD, Cunningham KY, Gupta VK, Son YM, Yang JD, et al. Single-cell mass cytometry on peripheral blood identifies immune cell subsets associated with primary biliary cholangitis. Sci Rep 2020;10(1). https://doi.org/10.1038/s41598-020-69358-4.

[8] Rybakowska P, Alarcón-Riquelme ME, Marañón C. Key steps and methods in the experimental design and data analysis of highly multi-parametric flow and mass cytometry. Comput Struct Biotechnol J 2020;18:874–86. https://doi.org/10.1016/j.csbj.2020.03.024.

[9] Warth S, Kunkel D. Setting Up Mass Cytometry in a Shared Resource Lab Environment. Methods Mol Biol 2019;1989:3–11. https://doi.org/10.1007/978-1-4939-9454-0_1.

[10] Monaco G, Chen H, Poidinger M, Chen J, de Magalhães JP, Larbi A. flowAI: automatic and interactive anomaly discerning tools for flow cytometry data. Bioinformatics 2016;32(16):2473–80. https://doi.org/10.1093/bioinformatics/btw191.

[11] Amir E-A, Davis KL, Tadmor MD, Simonds EF, Levine JH, Bendall SC, et al. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. Nat Biotechnol 2013;31(6):545–52. https://doi.org/10.1038/nbt.2594.

[12] McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform Manifold Approximation and Projection. J Open Sour Software; 2018. 10.21105/joss.00861.

[13] Belkina AC, Ciccolella CO, Anno R, Halpert R, Spidlen J, Snyder-Cappione JE. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. Nat Commun 2019;10:1–12. https://doi.org/10.1038/s41467-019-13055-y.

[14] Gassen SV, Callebaut B, Helden MJV, Lambrecht BN, Demeester P, Dhaene T, et al. FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry Part A 2015;87:636–45. https://doi.org/10.1002/cyto.a.22625.

[15] Qiu P, Simonds EF, Bendall SC, Gibbs KD, Bruggner RV, Linderman MD, et al. Extracting a Cellular Hierarchy from High-dimensional Cytometry Data with SPADE. Nat Biotechnol 2011;29(10):886–91. https://doi.org/10.1038/nbt.1991.

[16] 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;162:184–97. 10.1016/j.cell.2015.05.047.

[17] Beyrend G, Stam K, Höllt T, Ossendorp F, Arens R. Cytofast: A workflow for visual and quantitative analysis of flow and mass cytometry data to discover immune signatures and correlations. Comput Struct Biotechnol J 2018;16:435–42. https://doi.org/10.1016/j.csbj.2018.10.004.

[18] Chen H, Lau MC, Wong MT, Newell EW, Poidinger M, Chen J, et al. A Bioconductor Package for an Integrated Mass Cytometry Data Analysis Pipeline. PLoS Comput Biol 2016;12(9):e1005112. https://doi.org/10.1371/journal.pcbi.1005112.

[19] Schulz AR, Baumgart S, Schulze J, Urbicht M, Grützkau A, Mei HE. Stabilizing Antibody Cocktails for Mass Cytometry. Cytometry Part A 2019;95:910–6. https://doi.org/10.1002/cyto.a.23781.

[20] Spidlen J, Breuer K, Brinkman R. Preparing a Minimum Information about a Flow Cytometry Experiment (MIFlowCyt) Compliant Manuscript Using the International Society for Advancement of Cytometry (ISAC) FCS File Repository (FlowRepository.org). Current Protocols in Cytometry 2012;61:10.18.1-10.18.26. 10.1002/0471142956.cy1018s61.

[21] Crowell HL, Zanotelli VRT, Chevrier S, Robinson MD, Bodenmiller B. CATALYST: Cytometry dATa anALYSis Tools. Bioconductor version: Release (3.11); 2020. 10.18129/B9.bioc.CATALYST.

[22] Wickham H, Chang W, Henry L, Pedersen TL, Takahashi K, Wilke C, et al. ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. 2020.

[23] Amir ED, Guo XV, Mayovska O, Rahman AH. Average Overlap Frequency: A simple metric to evaluate staining quality and community identification in high dimensional mass cytometry experiments. J Immunol Methods 2018;453:20–9. https://doi.org/10.1016/j.jim.2017.08.011.

[24] Amir E-A, Lee B, Badoual P, Gordon M, Guo XV, Merad M, et al. Development of a Comprehensive Antibody Staining Database Using a Standardized Analytics Pipeline. Front Immunol 2019;10. https://doi.org/10.3389/fimmu.2019.01315.

[25] Malek M, Taghiyar MJ, Chong L, Finak G, Gottardo R, Brinkman RR. flowDensity: reproducing manual gating of flow cytometry data by automated density-based cell population identification. Bioinformatics 2015;31:606–7. https://doi.org/10.1093/bioinformatics/btu677.

[26] Van Gassen S, Gaudilliere B, Angst MS, Saeys Y, Aghaeepour N. CytoNorm: A Normalization Algorithm for Cytometry Data. Cytometry Part A 2020;97(3):268–78. https://doi.org/10.1002/cyto.v97.310.1002/cyto.a.23904.

[27] Emmaneel A, Bogaert DJ, Van Gassen S, Tavernier SJ, Dullaers M, Haerynck F, et al. A Computational Pipeline for the Diagnosis of CVID Patients. Front Immunol 2019;10. https://doi.org/10.3389/fimmu.2019.02009.

[28] Finak G, Jiang W, Gottardo R. CytoML for cross-platform cytometry data sharing. Cytometry Part A 2018;93(12):1189–96. https://doi.org/10.1002/cyto.a.v93.1210.1002/cyto.a.23663.

[29] Finak G, Jiang M. flowWorkspace: Infrastructure for representing and interacting with gated and ungated cytometry data sets. Bioconductor version: Release (3.9) 2019. https://doi.org/10.18129/B9.bioc.flowWorkspace.

[30] 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 Comput Biol 2014;10(8):e1003806. https://doi.org/10.1371/journal.pcbi.1003806.

[31] Thrash EM, Kleinsteuber K, Hathaway ES, Nazzaro M, Haas E, Hodi FS, et al. High-Throughput Mass Cytometry Staining for Immunophenotyping Clinical Samples. STAR Protocols; 2020:100055. 10.1016/j.xpro.2020.100055.

[32] McCarthy RL, Mak DH, Burks JK, Barton MC. Rapid monoisotopic cisplatin based barcoding for multiplexed mass cytometry. Sci Rep 2017;7:3779. https://doi.org/10.1038/s41598-017-03610-2.

[33] Lai L, Ong R, Li J, Albani S. A CD45-based barcoding approach to multiplex mass-cytometry (CyTOF). Cytometry Part A 2015;87(4):369–74. https://doi.org/10.1002/cyto.v87.410.1002/cyto.a.22640.

[34] Schulz AR, Mei HE. Surface Barcoding of Live PBMC for Multiplexed Mass Cytometry. Methods Mol Biol 2019;1989:93–108. https://doi.org/10.1007/978-1-4939-9454-0_7.

[35] Mei HE, Leipold MD, Schulz AR, Chester C, Maecker HT. Barcoding of live human PBMC for multiplexed mass cytometry. J Immunol 2015;194:2022–31. https://doi.org/10.4049/jimmunol.1402661.

[36] Finck R, Simonds EF, Jager A, Krishnaswamy S, Sachs K, Fantl W, et al. Normalization of mass cytometry data with bead standards. Cytometry A 2013;83A(5):483–94. https://doi.org/10.1002/cyto.a.v83a.510.1002/cyto.a.22271.

[37] Le Lann L, Jouve P-E, Alarcón-Riquelme M, Jamin C, Pers J-O. Standardization procedure for flow cytometry data harmonization in prospective multicenter studies. Sci Rep 2020;10(1). https://doi.org/10.1038/s41598-020-68468-3.

[38] Barturen G, Beretta L, Cervera R, Van Vollenhoven R, Alarcón-Riquelme ME. Moving towards a molecular taxonomy of autoimmune rheumatic diseases. Nat Rev Rheumatol 2018;14(2):75–93. https://doi.org/10.1038/nrrheum.2017.220.

[39] Pontikos N. flowBeads: flowBeads: Analysis of flow bead data. Bioconductor version: Release (3.12) 2021. https://doi.org/10.18129/B9.bioc.flowBeads.

[40] Dendrou CA, Fung E, Esposito L, Todd JA, Wicker LS, Plagnol V. Fluorescence Intensity Normalisation: Correcting for Time Effects in Large-Scale Flow Cytometric Analysis. Adv Bioinformatics 2009;2009:1–6. https://doi.org/10.1155/2009/476106.

[41] Meskas J, Wang S. Precise and Accurate Automated Removal of Outlier Events and Flagging of Files Based on Time Versus Fluorescence Analysis. Github Repository: Https://GithubCom/Jmeskas/FlowCut 2019.

[42] Weber LM, Robinson MD. Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. Cytometry A 2016;89(12):1084–96. https://doi.org/10.1002/cyto.v89.1210.1002/cyto.a.23030.

[43] Giudice V, Fantoni G, Biancotto A. Fluorescent Cell Barcoding for Immunophenotyping. Methods Mol Biol 2019;2032:53–68. https://doi.org/10.1007/978-1-4939-9650-6_3.

[44] Majonis D, Ornatsky O, Kinach R, Winnik MA. Curious results with palladium- and platinum-carrying polymers in mass cytometry bioassays and an unexpected application as a dead cell stain. Biomacromolecules 2011;12 (11):3997–4010. https://doi.org/10.1021/bm201011t.

[45] Bagwell CB, Inokuma M, Hunsberger B, Herbert D, Bray C, Hill B, et al. Automated Data Cleanup for Mass Cytometry. Cytometry Part A 2020;97 (2):184–98. https://doi.org/10.1002/cyto.v97.210.1002/cyto.a.23926.

[46] Olsen LR, Leipold MD, Pedersen CB, Maecker HT. The anatomy of single cell mass cytometry data. Cytometry Part A 2019;95(2):156–72. https://doi.org/10.1002/cyto.v95.210.1002/cyto.a.23621.

[47] Lee BH, Kelly G, Bradford S, Davila M, Guo XV, Amir E-A, et al. A Modified Injector and Sample Acquisition Protocol Can Improve Data Quality and Reduce Inter-Instrument Variability of the Helios Mass Cytometer. Cytometry Part A 2019;95(9):1019–30. https://doi.org/10.1002/cyto.a.v95.910.1002/cyto.a.23866.

[48] Rein ID, Notø HØ, Bostad M, Huse K, Stokke T. Cell Cycle Analysis and Relevance for Single-Cell Gating in Mass Cytometry. Cytometry Part A 2020;97 (8):832–44. https://doi.org/10.1002/cyto.a.v97.810.1002/cyto.a.23960.

[49] Gassen SV, Vens C, Dhaene T, Lambrecht BN, Saeys Y. FloReMi: Flow density survival regression using minimal feature redundancy. Cytometry Part A 2016;89:22–9. https://doi.org/10.1002/cyto.a.22734.

[50] Saeys Y, Van Gassen S, Lambrecht BN. Computational flow cytometry: helping to make sense of high-dimensional immunology data. Nat Rev Immunol 2016;16(7):449–62. https://doi.org/10.1038/nri.2016.56.

[51] Schuyler RP, Jackson C, Garcia-Perez JE, Baxter RM, Ogolla S, Rochford R, et al. Minimizing Batch Effects in Mass Cytometry Data. Front Immunol 2019;10. https://doi.org/10.3389/fimmu.2019.0236710.3389/fimmu.2019.02367.s001.

[52] Kleinsteuber K, Corleis B, Rashidi N, Nchinda N, Lisanti A, Cho JL, et al. Standardization and quality control for high-dimensional mass cytometry studies of human samples. Cytometry A 2016;89(10):903–13. https://doi.org/10.1002/cyto.a.v89.1010.1002/cyto.a.22935.

[53] van der Maaten L, Hinton G. Visualizing Data using t-SNE. J Mach Learn Res 2008;9:2579–605.

[54] Höllt T, Pezzotti N, van Unen V, Koning F, Eisemann E, Lelieveldt B, et al. Cytosplore: Interactive Immune Cell Phenotyping for Large Single-Cell Datasets. Comput Graphics Forum 2016;35(3):171–80. https://doi.org/10.1111/cgf.12893.

[55] Becht E, McInnes L, Healy J, Dutertre C-A, Kwok IWH, Ng LG, et al. Dimensionality reduction for visualizing single-cell data using UMAP. Nat Biotechnol 2019;37(1):38–44. https://doi.org/10.1038/nbt.4314.