Flow Cytometry Data Analysis Pipeline

Data Quality Control Tool Development and Biomarker Discovery

by

Sherrie (Xue) ^cWang

B.Sc., The University of British Columbia, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Bioinformatics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Flow Cytometry Data Analysis Pipeline-Data Quality Control Tool Development and Biomarker Discovery

submitted by **Xue Sherrie Wang** in partial fulfillment of the requirements for the degree of **Master of Science** in **Bioinformatics**.

Examining Committee:

Ryan R. Brinkman, Medical Genetics; *Supervisor*

Maxwell W. Libbrecht, Computer Science; *Supervisory Committee*

Sara Mostafavi, Medical Genetics and Statistics; *Supervisory Committee*

Peter Lansdorp, Medical Genetics and Medicine; *Additional Examiner*

Abstract

Technical complications occurring during the data acquisition process can impact the quality of the cytometry data and its analysis results. Clogs can cause spikes in the data sets in the time domain. Other issues, such as changing machine acquisition speed, can result in a shift in means of the populations analyzed. The outliers can potentially bias the downstream analysis if left unchecked and as such should be identified and removed. To address this need, I developed flowCut is an R package for automated detection of anomaly events and flagging of files for flow cytometry experiments. Results are on par with manual analysis, and it outperforms the existing approaches in data quality control. flowCut has the highest F1 scores in two types of evaluations used in this study and has zero crash rate on all files tested.

I also studied the bone marrow regeneration pattern of acute myeloid leukemia patients after chemotherapy by applying state of the art automated methods. I identified cell populations and biomarkers that are uniquely present in relapsed patients when comparing to normal bone marrow data. I also identified cell populations that have different regeneration dynamics between relapsed and non-relapsed patients.

Lay Summary

Flow cytometry is used widely in clinics and research for measuring blood cells. Its primary purpose is to quantify cell population compositions for diagnosis or studying immunological characteristics of diseases. Technical issues of cytometers during data acquisition can result in an inaccurate measurement of cells, which can cause an erroneous analysis of cell populations. My research focused on developing a data quality assessment tool and comparing the performance of current approaches. I also used several state of the art automated data analysis methods to identify cell populations in acute myeloid leukemia patients who relapsed after undergoing chemotherapy.

Preface

The flowCut tool (included in Chapter 2) was written by Justin Meskas and myself. The tool was written in R programming language and is available at

`https://github.com/jmeskas/flowCut`

My significant contribution to the package was writing the function for identifying and removal of outlier events based on the density of summed measures (section 2.2.2). I wrote the preliminary quality checking step. I was involved in testing the tools and optimizing parameters in section 2.3. Sibyl Drissler was also engaged in tool testing. Contents in chapters 2-3 are part of a paper to be submitted.

Project in chapter 4 was a collaboration between the department of laboratory medicine, Institute of Biomedicine, Sahlgrenska Academy at University of Gothenburg. Patients' data were collected from multiple medical centers in Gothenburg, Copenhagen, Israel, and Umea by Linda Fogelstrand, who is the head of the department of laboratory medicine at the University of Gothenburg.

Table of Contents

Appendix

List of Tables

List of Figures

Acknowledgements

I would like to thank Dr. Ryan Brinkman, Justin Meskas for their support during the course of my graduate work.

Chapter 1

Introduction

Flow cytometry is a technique for studying the physical and chemical characteristics of cells using light-emitting antibodies. Cells stained with antibodies have light-emitting fluorophores attached to them. Each cell type has some antigens on them. Biologists design antibodies that specifically bind to these antigens. Stained cells then flow past one or multiple laser light sources in ideally a single file manner[6]. The emitted light from the cells, which is proportional to the antigen density, can be converted to electric signals and analyzed on a 2D plot. The light signals consist of three types: forward scattering, side scattering, and fluorescence emission signals. Forward scattering and side scattering measure the physical attributes of the cells, whereas fluorescence signal measures the functional characteristics of cells[6]. For example, T cells will present CD3 antigens, and B cells will present CD19 antigens.

One critical step in flow cytometry data analysis is partitioning cells into types based on marker expressions. The cells of the same type are grouped and selected on a 2D plot with a bounded area called gate. We can identify subtypes of the selected cells with the addition of new markers. For example, T cells can have CD4+ T cells and CD8+ T cells subtypes and can be gated on CD4 and CD8 markers. Traditionally, the process of identifying cell populations is done manually. Researchers analyze two parameters at a time through visual inspection. The typical process starts by first removing dead cells and doublets. From live singlets cells, researchers can go down a path of finding targeted cell populations by following a specific partitioning (gating) strategy. The gating strategy indicates the sequence of markers to be analyzed to reach the target cell types.

Yet, manual analysis has many problems. Individual analysts can introduce subjectivity and bias into the gating analysis[12]. The presence of subjectivity makes cross center comparative studies difficult and hinders reproducible research. Besides, recent instrumental advances and reagents expansion allow for measuring tens of surface and intracellular markers simultaneously, and allow for the generation of 20-dimensional data. Traditional manual analysis of two markers at a time cannot cope with the amount of data received. Manual analysis is time-consuming and can be ineffective at analyzing high dimensional data [12].

There has been a surge in the production of computational tools for flow cytometry analysis in the past decade to address the challenges of manual analysis.

1.1 R/Bioconductor

More than 50 computational approaches are available for the analysis of flow cytometry data [3, 17], with a majority of the tools developed and released as free, open-source tools using R programming language [19]. These tools have been developed for high throughput workflows, and are not generally amenable to graphical user interface and manual interaction with individual files during the analysis process. However, they can be integrated into commercial tools FlowJo [FlowJo Bioinformatics Inc., Ashland OR] that are familiar to users.

A majority of the approaches have been released through the Bioconductor repository [20], which enforces strict requirements on cross-platform compatibility and functional documentation. Each package generally addresses one single step in the analysis pipeline, allowing users to substitute new approaches to the same challenge as the field advances.

The required core infrastructure widely used by other packages provided by the flowCore R/Bioconductor package [9] implements a computationally efficient data structure for reading and saving FCM data and provides systematic FCS file parsing. The flowCore infrastructure encourages new algorithms development and the use of combinations of tools in complex workflows [9].

The workflow involved in this study includes data compensation, transformation, quality control, automated gating, and biomarker identification.

Compensation and transformation Data needs to be properly compensated, transformed, and normalized to ensure the accuracy of any subsequent gating analysis. Compensation is necessary to correctly account for the contribution of each fluorochrome to each channel in conditions of spectral overlap [17]. A well-used transformation facilitates population gating, visualization, and downstream analysis. The often-used transformation methods that handle negative values and display normally distributed cell types are logicle, hyperlog, and arcsine [11].

1.2 Data quality assessment

One goal of data quality control is to assess the stability of signal acquisition over experimental time. We can visually check the signal stability by plotting fluorescence channels against time. A stable signal acquisition shows a consistent distribution of fluorescence intensity values over time. This is the expected behavior based on the assumption that cells from a heterogeneous sample are randomly measured at any time point [16]. Changes in fluorescence intensity values in the time domain are indicative of technical variability. Abnormal events can possess a unique space/cluster in a 2D dimension [16] and potentially get mislabeled as biologically significant events. Therefore, these events should be removed or flagged before passing to the gating analysis.

The manual inspection process can be time-consuming and subjective [7]. For removing spikes, users need to zoom in on the time channel to identify the boundaries of slivers accurately. Even so, the actual placement of boundaries can still be subjective. It is therefore necessary to develop automated methods that could remove human subjectivity and speed up the quality control process. The current approaches addressing this problem are flowClean [7] and flowAI[15].

1.2.1 Current Approaches

flowClean detects the abnormal changes in the compositions of cell populations over time. It partitions each marker into a high or low expression using median values and tracks the representation of resulting 2^Q phenotypes across equally split time bins [7]. flowAI detects changes in means and variances of fluorescence intensity in the time domain [16].

Both flowClean and flowAI utilize some versions of "multiple change points detection" algorithm implemented by the "changepoint" package [10]. Specifically, flowClean uses "pruned exact linear time (PELT)" algorithm, and flowAI "binary segmentation".

Binary segmentation Binary segmentation is a computationally fast approximation algorithm that repeatedly splits data into two groups at a time by repeating the single change point test over the split data sets until no change points are found in any part of the data [10].

Pruned exact linear time PELT uses dynamic programming and pruning to produce the exact segmentation. It is computed based on the assumption that the number of change points increases linearly as the data set grows [10]. Namely, change points will spread throughout the entire data rather than confined to one portion.

A common problem with flowClean is it has a long run time, average 1-4 minutes per file, and often misses anomalies 1.1. flowAI, on the other hand, is very fast, 3-5 seconds per file, due to the efficiency of binary segmentation.

However, it tends to be overly intolerant, removing large portions of normal regions, as shown in Fig 1.1.

Killick et al. (2014) note that both PELT and binary segmentation can be overly sensitive. In a normally distributed data set with three constructed change points, the PELT algorithm reported six change points while binary segmentation reported four [10]. We noted that flowAI, which uses binary segmentation, tends to be more sensitive than flowClean. The underreporting of change points by flowClean could be due to challenges in the analysis of phenotype compositions.

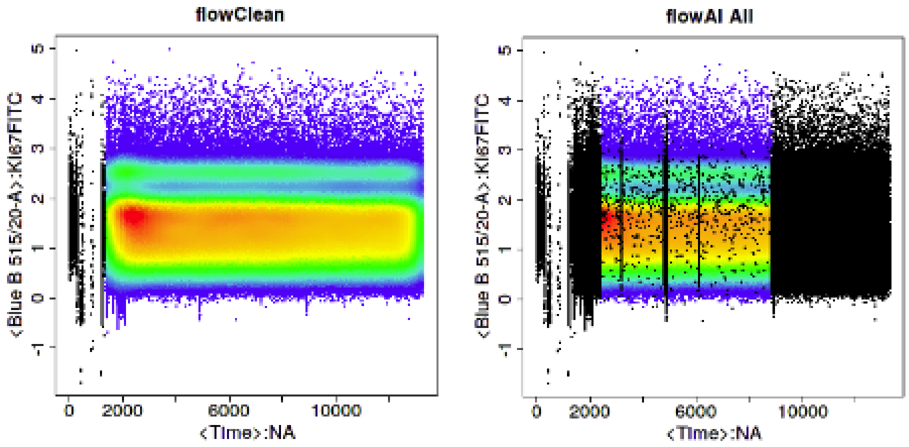


Figure 1.1: Quality control by flowClean (left) and flowAI (right) on the same file. The colored regions are fluorescence intensity signals. Red is the most dense, followed by yellow, green, and purple. Black are the removed regions.

1.3 My Research

Despite the current efforts of flowAI and flowClean, the challenge in data quality control remains. My research focused on developing a tool that addresses the ongoing problem more effectively. I hypothesized that a segment-wise statistical analysis could effectively identify outlier events. To prove this, I compared the performance of all three algorithms. In chapter 2, I described the algorithm development. Chapter 3 detailed the method and results for evaluating all three algorithms. In chapter 4, I covered the process and outcomes of using current tools for biomarker discovery.

Chapter 2

flowCut - a data quality control tool

2.1 Workflow

As shown in Fig 2.1, we start with already processed FCS files, which are files after compensation and transformation. flowCut first checks the quality of the time channel, flagging files with repeated time intervals or having a majority of events occurring in a short burst of time, usually at the beginning of a file. This step is to catch a problem that can potentially crash the algorithm. Second, flowCut removes low-density sections, which are regions with less than 1% of the range of data. Third, we begin data segmentation and score calculation for each segment. We then check if the scores are below a specific trigger threshold. If it is, then the files are up to standard. Otherwise, the bad data trigger flowCut to remove them based on a score distribution. After this, flowCut does a second quality control check and flags files with remaining problematic regions.

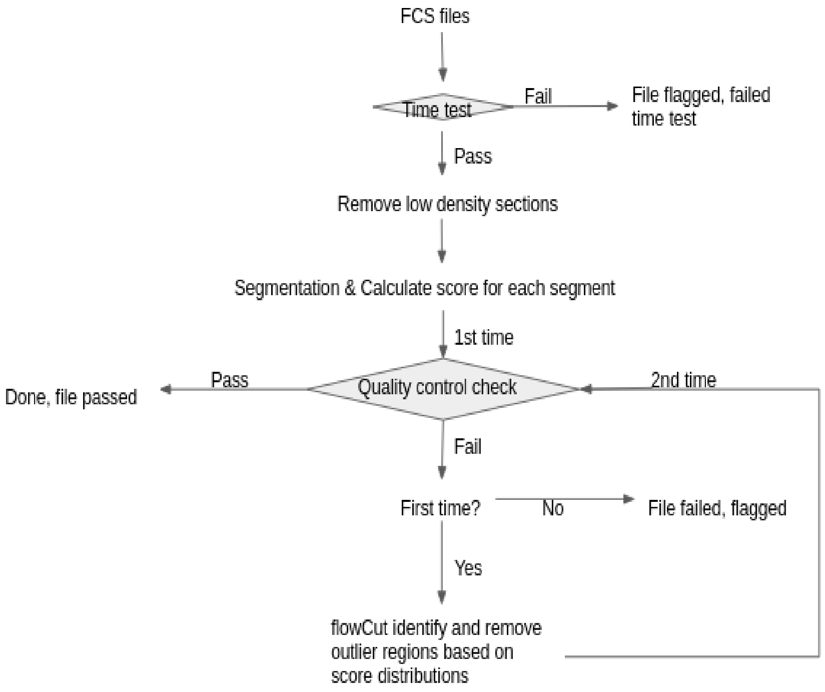

FCS files
Time test
Fail
File flagged, failed time test
Pass
Remove low density sections
Segmentation & Calculate score for each segment
1st time
Quality control check
2nd time
Pass
Done, file passed
Fail
First time?
No
File failed, flagged
Yes
flowCut identify and remove outlier regions based on score distributions

Figure 2.1: The figure summarizes workflow of flowCut algorithm. Overall, flowCut does a maximum of three quality checks. The first one pertains to the time channel. The second and third are checking the stability of the fluorescence signals.

2.2 Methodology

We hypothesized that abnormal events are statistically different than the normal events in the fluorescence versus time analysis. Naturally, we track the statistics of the time domain data to find abnormalities.

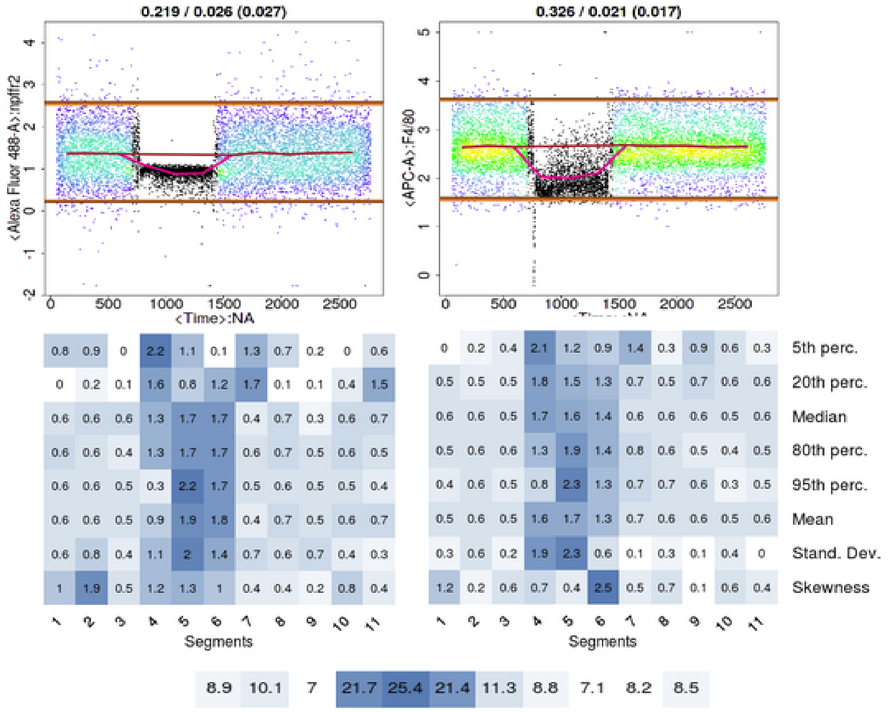
Standard score

In statistics, the standard score, also called Z score, is the signed fractional number of standard deviation and is used to study the deviation of data points from the mean value. We adapt from this concept and used absolute Z scores 2.1 for this purpose as we are only interested in the differences from the mean but not the direction.

$$|Z| = |\frac{x-\mu}{\sigma}| \tag{2.1}$$

2.2.1 Segmentation and calculate Z scores

We divided each fluorescence channel into equally populated segments, with 500 events per segment for a typical FCS file (less than 20MB in size). Fig 2.2 shows a two-channel (Alexa Fluor 488-A and APC-A) FCS file that is divided into 11 segments. We calculated eight statistical measures for each segment according to equation 2.1. Because we use absolute Z scores, the differences calculated are all accumulative. Segments with high Z scores indicate substantial deviations from the mean.



1	2	3	4	5	6	7	8	9	10	11
0.8	0.9	0	2.2	1.1	0.1	1.3	0.7	0.2	0	0.6
0	0.2	0.1	1.6	0.8	1.2	1.7	0.1	0.1	0.4	1.5
0.6	0.6	0.6	1.3	1.7	1.7	0.4	0.7	0.3	0.6	0.7
0.6	0.6	0.4	1.3	1.7	1.7	0.6	0.7	0.5	0.6	0.5
0.6	0.6	0.5	0.3	2.2	1.7	0.5	0.6	0.5	0.5	0.4
0.6	0.6	0.5	0.9	1.9	1.8	0.4	0.7	0.5	0.6	0.7
0.6	0.8	0.4	1.1	2	1.4	0.7	0.6	0.7	0.4	0.3
1	1.9	0.5	1.2	1.3	1	0.4	0.4	0.2	0.8	0.4

1	2	3	4	5	6	7	8	9	10	11	
0	0.2	0.4	2.1	1.2	0.9	1.4	0.3	0.9	0.6	0.3	5th perc.
0.5	0.5	0.5	1.8	1.5	1.3	0.7	0.5	0.7	0.6	0.6	20th perc.
0.6	0.6	0.5	1.7	1.6	1.4	0.6	0.6	0.6	0.5	0.6	Median
0.5	0.6	0.6	1.3	1.9	1.4	0.8	0.6	0.5	0.4	0.5	80th perc.
0.4	0.6	0.5	0.8	2.3	1.3	0.7	0.7	0.6	0.3	0.5	95th perc.
0.5	0.6	0.5	1.6	1.7	1.3	0.7	0.6	0.6	0.5	0.6	Mean
0.3	0.6	0.2	1.9	2.3	0.6	0.1	0.3	0.1	0.4	0	Stand. Dev.
1.2	0.2	0.6	0.7	0.4	2.5	0.5	0.7	0.1	0.6	0.4	Skewness

8.9	10.1	7	21.7	25.4	21.4	11.3	8.8	7.1	8.2	8.5

Figure 2.2: The top and bottom lines in the top two plots represent 98th and 2nd percentiles. The differences between these two lines define the range of data. The lines in the middle are the segments' means. Pink is before cleaning. Brown is after cleaning. flowCut divides a two-channel FCS file into 11 segments. It calculates eight statistical measures, including 5th, 20th, 80th, 95th percentiles, median, mean, standard deviation, and the third moment for skewness. Summing eight statistics across all channels results in a vector. Each element in the vector represents one segment. The most significantly different sections are in dark blue.

2.2.2 Removal of abnormal events

The removal of outlier segments is based on the density distribution of the score vector, shown in Fig 2.3. We want to find a data dependant threshold that separates outliers from normal cells. Any segments with values higher than this threshold are outliers for removal. To do this, we adapted the **deGate** function in the flowDensity R package [14]. The **deGate** function returns a gate line on a 1D density profile. The original purpose was to separate cell populations. We utilized it in our outlier detection methodology. We manipulated the **deGate** function so that it always returns a gate line that lies on the right side of the density distribution because we are only interested in removing the cells that are most different.

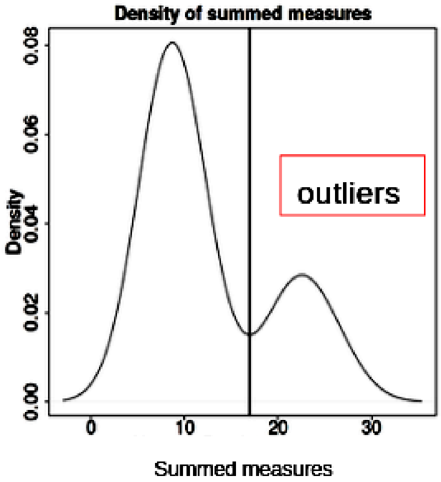


Figure 2.3: Density of summed Z scores. Outlier segments lie on the right side of the distribution and are separated from the rest by a natural cutoff line. Segments with scores higher than the cutoff will be removed.

The shape of the density profile of each marker can vary naturally. We allow natural variations of the data as long as they pass the quality control check. Otherwise, we define the cutoff line to minimize overcutting and undercutting. I developed a set of rules for finding an ideal cutoff line 2.1.

Table 2.1: Algorithms for finding cutoff lines based on density distribution

1. flowCut first finds all the peaks ($p = 1, 2, ...n$) in the density distribution.
2. if $p = 1$, it uses **deGate** function to find a natural point along the upstream of the density distribution to remove significantly different segments.
3. if $p >= 2$, flowCut checks each peak and calculates the valley height between the adjacent peaks, if it is less than 1% of the maximum peak, flowCut ignores the lower peak. If there are still more than 2 peaks left and the population to be removed is less than or equal to a user specified amount, flowCut removes the significantly different population.

2.3 Parameters

2.3.1 Quality control parameters

flowCut uses three thresholds to determine if a file passes or fails a quality control check, namely, the maximum allowable mean range, the average of this range across all channels, and the maximum continuous jump between adjacent segments. Fig 2.4 shows the maximum allowable mean range and maximum one-step jump. If any of the parameters calculated is higher than a threshold value, flowCut starts the cleaning process. The user can adjust the stringency of the algorithm and all by changing these parameters.

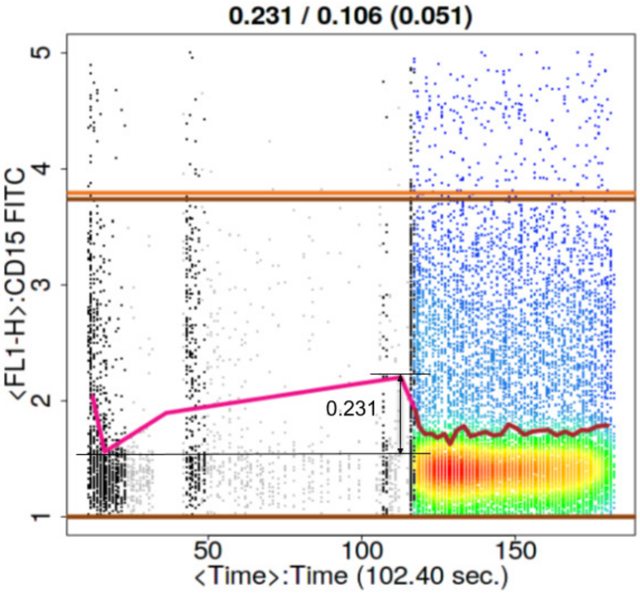


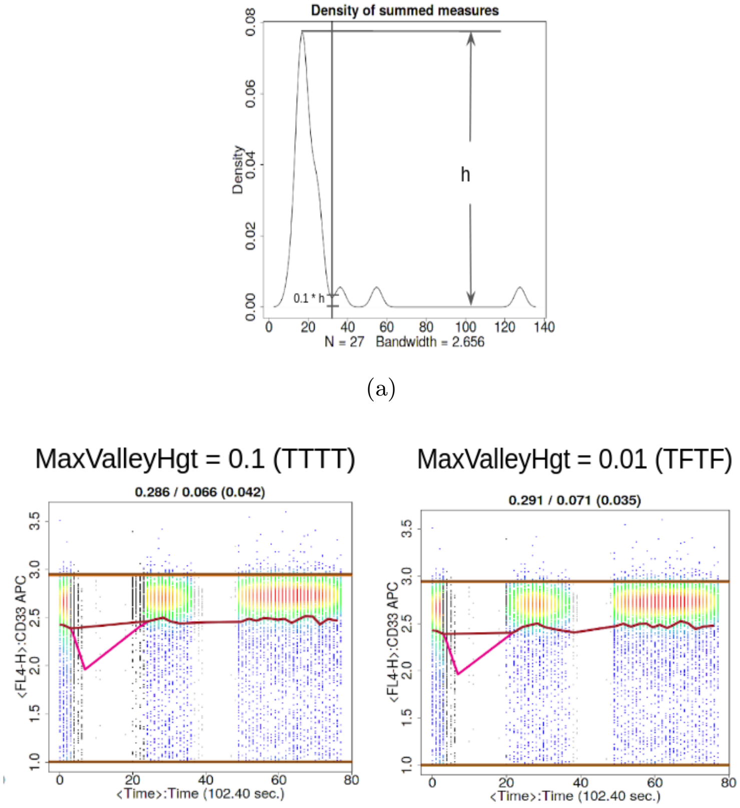
Figure 2.4: An example file shows the range and mean of data before and after cleaning. The data before cleaning is bounded by 98th and 2nd percentile indicated by yellow (before cleaning) or dark brown (after cleaning) lines. The pink line in the middle is the connected segments means before cleaning. The maximum range of these means is the first number on top. The second number is this range after cleaning. The number in the bracket indicates the maximum one-step change after cleaning.

2.3.2 Cutoff line parameters

Users can set two parameters, one that defines the maximum percentage of events for removal, one that sets the thresholds to be generally higher or lower. These two parameters can wiggle the cutoff line on the density distribution.

Maximum percentage of removal The default value is 30%. If the outlier populations in Fig 2.3 exceeds this amount, then the cutoff line will be moved further to the right, and nothing will be removed.

Maximum valley height See Fig 2.5. It sets the maximum height of the tail on the distribution, defaulted to be 10% of the tallest peak. This parameter determines how aggressive the cutting will be. If a user sets a value larger than the default, then the user allows for generally more aggressive cutting. In this case, the height of the valley is higher, and the cutoff line has a smaller value. Smaller threshold values allow the removal of more segments. For less aggressive cutting, the parameter will be lower, and the cutoff line moves further to the right. However, this could mean an insufficient removal of abnormal events, and the file is not likely to pass the second quality control check.



(a)

(b) (c)

Figure 2.5: By default, the threshold is placed at the valley with a height of approximately 10% of the tallest peak, shown in a) and b). If users are to decrease this value, the threshold is moving further to the right and is at the second valley. In this case, fewer segments will be removed c).

Chapter 3

Algorithms Comparison

I evaluated the performance of all three algorithms against manual analysis for selected files. I obtained these files from a public repository, FlowRepository [22].

3.1 Method

3.1.1 Selection of files for evaluation

I followed the following protocol when selecting files for evaluation:

- Randomly download 1071 files from FlowRepository.
- Eliminate corrupt files (83) that cannot be read, compensated, or transformed.
- Eliminate files crashed by any of the algorithms (145).
- Keep any that required cleaning by visual inspection (50) or identified with problematic regions by at least two algorithms (5). If a file had no visually identifiable regions and only got cleaned by only one algorithm, it was put aside and not counted toward the evaluation.

3.1.2 Manual vs algorithm analysis

For each of the selected 55 files, I visually identified problematic regions, then ran each algorithm on these files with their default settings. Examples are shown in Fig 3.1.

Manual analysis procedure I plotted each marker channel versus time and visually identified problematic regions. Each removal region had two boundaries. I created a spreadsheet for storing boundaries for each of the 55 files. Each row (file) has a series of an even number of boundaries that define the regions for cutting. For example, if there are four numbers, the first and second numbers are the beginning and end of the first region removed. And the third and fourth numbers are the beginning and end of the second region removed. See Appendix A.

3.1.3 F1 score as a measure for comparison

F1 score, in equation 3.1, is the harmonic mean of precision and recall. F1 score was used for judging algorithms' performance in FlowCAP studies [1, 2, 5]. We borrowed the idea here in this evaluation.

$$F1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 * \frac{precision * recall}{precision + recall} \tag{3.1}$$

The data selected by an algorithm can contain some portions of true positives and false positives. And the non-selected regions include some false negatives and true negatives. Precision is the proportion of events selected by the algorithm that are true positives, that is, overlapping with manually

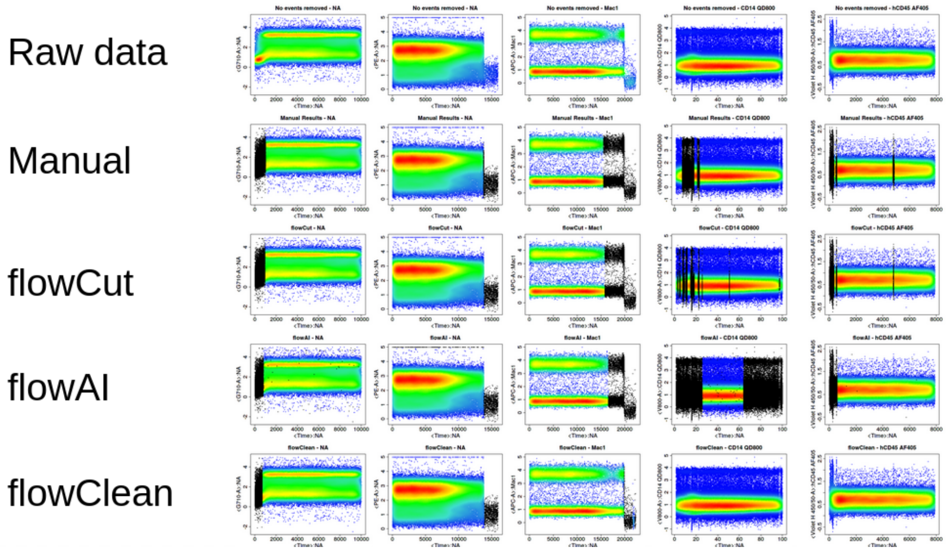


Figure 3.1: 5 exemplary files of raw data, manual analysis, flowCut, flowAI, and flowClean analysis are shown.

chosen regions. Recall is the proportions of cells selected by the manual analysis, which are also identified by the algorithm. I used the results of the manual gating as the standard for computing the F1 scores. Manual analysis took approximately 5-10 minutes per file.

3.1.4 File based evaluation

Noting that manual analysis can be subjective, I subdivided the 55 files into three categories based on the subjective confidence of the manual analysis, shown in Fig 3.2. The first category includes 17 files that have removal regions with clearly defined boundaries such as discontinuity, low density regions, and large spikes. The second category has 35 files that have fuzzy boundaries for removal regions. Examples include small spikes, boundaries in fluorescence drifting regions. This category also includes files that have overlapping problematic regions detected by at least two algorithms but not by manual analysis. The third category has three files in which manual analysis is arbitrary. The files look abnormal, but the regions for removal are not clear. For example, the first category three file in Fig 3.2 contains three shifting regions with different means. It is unclear which region(s) should be removed. Category three files contain no region of truth and are eliminated from the evaluation. For the remaining 52 files, I calculated F1 scores for categories one and two files for each algorithm. We subsequently calculated F1 score for random removing, with the percentage of removal similar to that of the manual removal, as a baseline for comparison.

3.1.5 Problem based evaluation

To analyze the problem-based performance of each algorithm, I categorized manual identified regions into four distinctive problem types, as shown in Fig 3.3. Each type has its unique characteristics. I evaluated algorithms' performance on dealing with these four types of problems by calculating F1 scores for all regions of each type. A single file can contain multiple types of issues. I calculated F1 scores for each of the identified problematic regions and clean regions, and their proportions in a file.

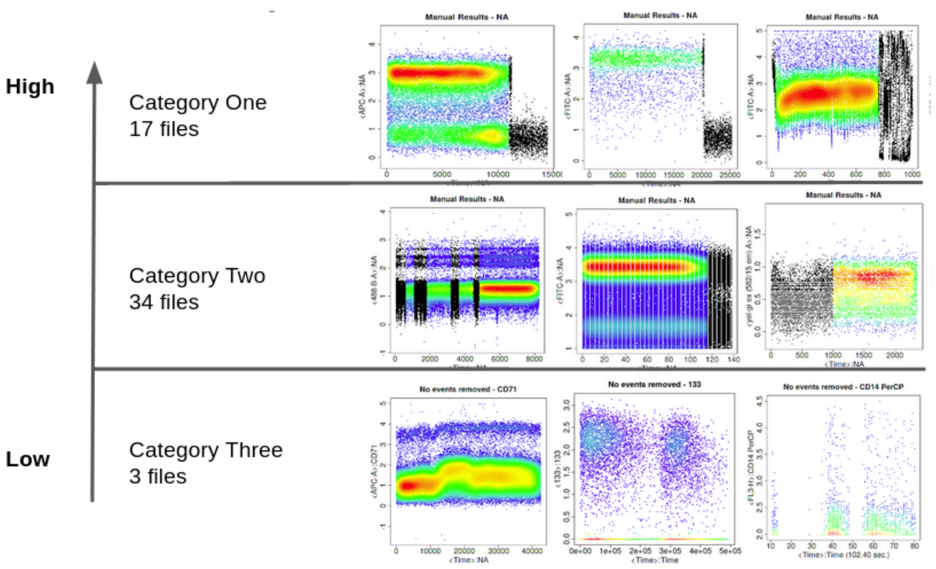


Figure 3.2: Files are divided into three categories based on confidence in manual analysis. From category three to one there is an increase in confidence levels for manual analysis. Category one (17 files) problematic regions contain distinctive discontinuity, large spikes regions. Category two (34 files) problematic region contain small spikes, intensity shifting regions. Category three (3 files) problematic regions are hard to defined, therefore, no manual analysis is performed on these files.

3.2 Result

3.2.1 File based evaluation results

flowCut has the highest F1, precision, and recall score, as shown in table 3.1. Run time is a close second to flowAI. flowAI removes most events for category two files yet has a low F1, indicating that it's removing a large amount of false positives. flowClean has overall lowest F1 scores, and its run time is several magnitudes longer than that of flowAI and flowCut.

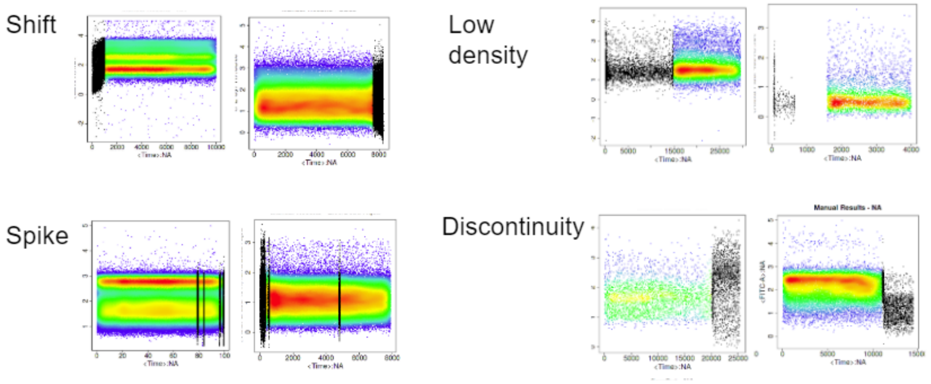


Figure 3.3: Categorizing problematic regions into four types. Two examples of each category are shown.

Table 3.1: F1 scores, precision, recall, run times, and percentages of removal of each algorithm were calculated for 17 category one and 35 category two FlowRepository exemplary files for three algorithms and a random removal. Computed on an Intel Xeon E5-2630 CPU with 128 GB RAM.

Category One:

	F1 scores	Precision	Recall	Run times (s)	% removed
flowCut	0.79	0.75	0.90	4.8	14.0%
flowAI	0.42	0.43	0.68	3.7	7.9%
flowClean	0.32	0.30	0.44	54.5	4.6%
random	0.16	0.16	0.16	-	15.6%

Category Two:

	F1 scores	Precision	Recall	Run times (s)	% removed
flowCut	0.5	0.5	0.71	7.18	7.5%
flowAI	0.18	0.3	0.23	4.4	10.0%
flowClean	0.15	0.16	0.33	190.25	1.01%
random	0.12	0.12	0.12	-	12.0%

Category Three files Although I eliminated category three files from score calculation, it is necessary to see how each algorithm deals with these files. As shown in Fig 3.4, flowCut flagged two files to users. The flagging contains information of why a file fails quality check. flowCut flagging has four letters of either T or F, checking 1) if the events are monotonically increasing with time, 2) abrupt changes in fluorescence, 3) large gradual shift of fluorescence signals in 3) all channels and 4) one channel. For category three first file, flowCut indicated a large gradual change of fluorescence in one channel. The second file had both abrupt and large gradual changes of fluorescence in one channel. The third file failed the time test with a fraudulent time channel, as described in section 2.1. flowAI detected some problems in the second file, but the removal regions can not be verified by manual at this stage. flowClean identified no problems in the first file, and couldn't process the second and third files due to too few cells to calculate phenotypes compositions. Note, these files did not crash flowClean.

3.2.2 Problem based evaluation results

Table 3.2 shows the mean F1 scores for each problematic type with their normalized percentage of events in a file. The normalized proportions is the sum of all regions by type, divided by the total number of files (52). I calculated a weighted F1 score for each algorithm according to $\sum_{i=problemtype}^{5} meanF1 \times normalizedPercentages$. flowCut was 0.93, flowAI 0.86, flowAI 0.83 (Table 3.2).

3.3 Impact on Gating Analysis

We reproduced the gating of TCM CD8 T cells, CD45RA-CDR7+ from the published paper [4] with and without using flowCut. The data that had

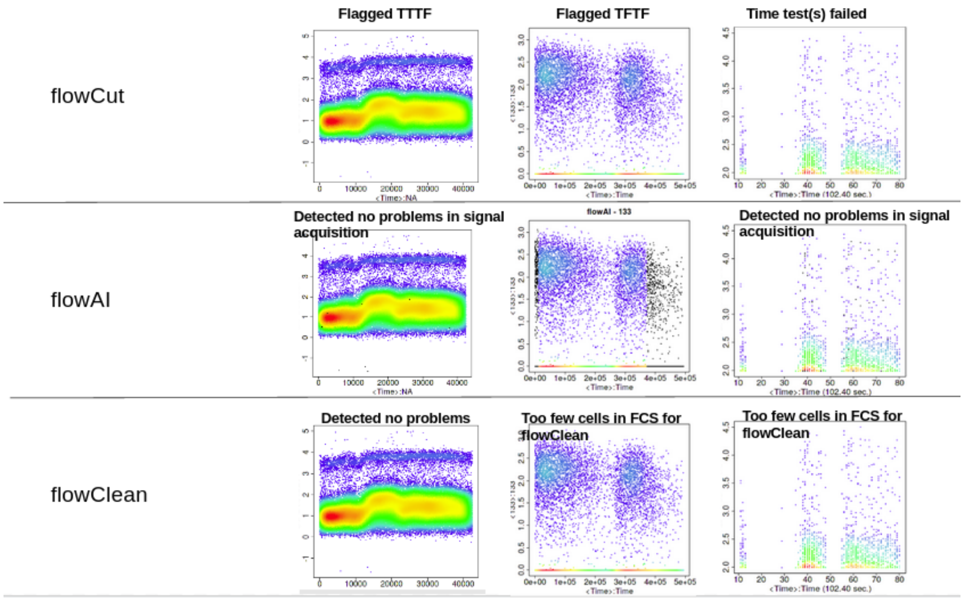
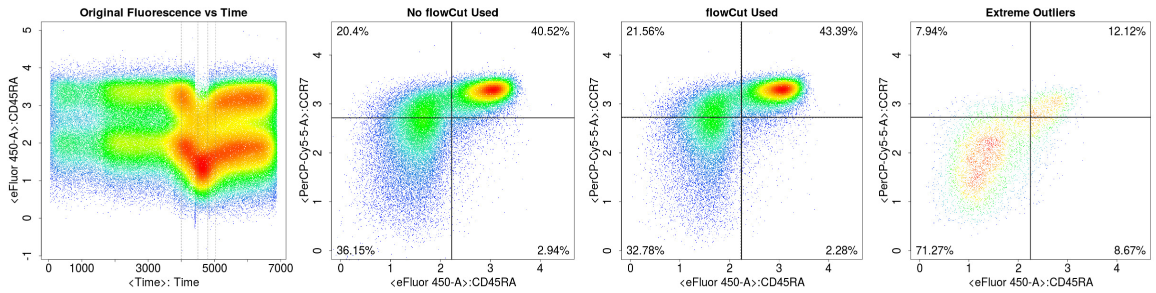


Figure 3.4: Documenting how each algorithm deals with category three files. flowCut flagged first two files to users, and did not process the third file due to time test issues. flowAI detected no problematic regions in the first and third. However, only flowAI detected some regions in the second file. flowClean detected no problematic regions in the first file, and could not process the 2nd and 3rd file.

	Percentage of events	flowCut	flowAI	flowClean
Shift	7.23%	0.67	0.40	0.20
Spike	2.60%	0.63	0.33	0.02
Low Density	1.72%	0.67	0.28	0.20
Discontinuity	0.98%	0.93	0.32	0.00
Clean	87.47%	0.96	0.93	0.93
Weighted F1	-	0.93	0.86	0.83

Table 3.2: Mean F1 scores of four types of problematic regions, mean percentage of events and the weighted F1 scores for all three algorithms

proper data quality control showed an increased proportion of the targeted cell population after gating (Fig 3.5 c). Gating on the outlier events (Fig



(a) (b) (c) (d)

Figure 3.5: (a) shows fluorescence drifting (Similarly for CCR7 - not shown). (a)-(d) show the difference between (b) not using and (c) using flowCut. (d) shows only the gated events between the middle two grey vertical lines in (a).

3.5 d) showed the outliers events lie mostly on the bottom left quadrant, indicating that they were biasing the gating to that region.

3.4 Algorithm Robustness

Out of 988 files from FlowRepository, a total of 114 files (11.5%) crashed flowClean, and 65 (6.6%) files crashed flowAI. Overall, a total of 145 files crashed either flowClean or flowAI. 0 files crashed flowCut. As of October 2019, I ran flowCut on all 117,115 FlowRepository files. It has 0% crash rate.

3.5 Conclusion

Data cleaned by flowCut improves the downstream gating. flowCut allows users to check the quality of the cleaning and to adjust the stringency of the algorithm if needed. Compared to existing methods, flowCut identified outlier events more accurately and did not fail to process any file.

Chapter 4

Biomarker Discovery in Minimal Residual Disease

4.1 Background

Patients with acute myeloid leukemia (AML) who underwent chemotherapy can sometimes relapse. The goal of this study is to examine the bone marrow regeneration pattern for characteristics that could associate with recurrence.

Immunophenotyping by flow cytometry is capable of detecting 1 leukemia cells in 10,000 normal cells [23], making it an ideal method for monitoring minimal residual disease. When choosing tools to study cell markers, I consulted FlowCAP studies [1, 2, 5] in which a list of currently available algorithms are evaluated against manual for their ability to find significant populations that correctly predict HIV patients' disease progression status. flowDensity[13] (both supervised and unsupervised), flowType and RchyOptimyx [18] pipeline stood out as the co-best method.

flowDensity can be used in both supervised and unsupervised ways. In a supervised fashion, it automates the manual gating process by using customized one-dimensional density thresholds for each cell population to mimic experts' hierarchical gating order. Unlike manual gating, where the placement of gate boundaries is inherently subjective, thresholds are adjusted in a data-dependent manner for each sample.

When used in the unsupervised method, gating thresholds are adjusted in a completely automated fashion per marker, removing customization. FlowReMI [24], the other best method identified in FlowCAP studies, used unsupervised flowDensity for marker partitioning. Since human intervention is removed from unsupervised gating analysis, experts might not find partitioning of some markers agreeable.

4.2 Study Design

4.2.1 Data

We obtained blood data of AML patients (both relapsed and non-relapsed) at three time points after chemotherapy for five tubes. Each tube has different set of markers. The three time points of interest are: day 22, last before 2nd induction, last before consolidation. To increase statistical power, I did random sampling so that each group at each time point contains 20 samples. There are an additional 20 normal samples for each tube. For supervised gating, the starting population for analysis is singlets. For the unsupervised method, the starting population is CD45+SSC-. The CD45+SSC- population was identified using k-means clustering by flowPeaks package [8].

	Markers
Tube1	CD56, CD13, CD34,CD117, CD33, CD11b, HLADR, CD45
Tube2	CD36,CD64,CD34,CD117,CD33,CD14,HLADR,CD45
Tube3	CD15,NG2,CD34,CD117,CD2,CD19,HLADR,CD45
Tube4	CD7,CD96,CD34,CD117,CD123,CD38,HLADR,CD45
Tube5	CD99,CD11a,CD34,CD117,CD133,CD4,HLADR,CD45

	Normal	Day22	Last before 2nd ind	Last before cons
Tube1	20 normal	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed
Tube2	20 normal	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed
Tube3	20 normal	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed
Tube4	20 normal	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed
Tube5	20 normal	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed	20 relapsed 20 non-relapsed

Table 4.1: Blood data for AML patients

4.2.2 Supervised Gating

Supervised gating requires users to have some prior experimental expectation, for example, if a user wants to replicate an existing manual process to target cell populations of interest in a specific way. In our case, we will require biologists to have some preexisting knowledge regarding the regeneration pattern of bone marrow and design a gating strategy to find populations that are likely to be interesting, i.e., differentiating between the relapsed and non-relapsed group. It requires expertise and efforts to come up with a gating strategy. We only obtained a gating strategy for tube 1.

The goal was analyzing the end populations to see if any are significantly different among the two groups of patients at each time point.

4.2.3 Unsupervised Gating and analysis pipeline

Unsupervised gating can substantially increase the scale of analysis. Combined with flowType and RchyOptimyx, we can examine all possible populations defined by the markers by removing the time-limiting step of customization for each tube. The supervised analysis was limited to one tube. However, unsupervised analysis can be applied to all tubes.

4.3 Results

4.3.1 Supervised Analysis Results

I wrote the automated gating pipeline, in Fig 4.1 according to the gating strategy provided. The gates were mostly determined by 1D density distribution of each marker. This was implemented by flowDensity[14] package. However, three gates required rotation of the axis to some degree to find proper separation. These gates are singlets gates, HLADR mast cells gates, CD13+B CD33- gates, singlets monocyte-derived cells high SSC, corresponding to gates 2,3,4,11 in Fig 4.1.

Subsequent comparisons of the cell proportions across three time points show no significant difference (significant when $p < 0.05$) among the two groups of patients.

4.3.2 Unsupervised Analysis Results

There were no significant populations between relapsed and non-relapsed groups based on t-tests analysis for all five tubes across all three time points.

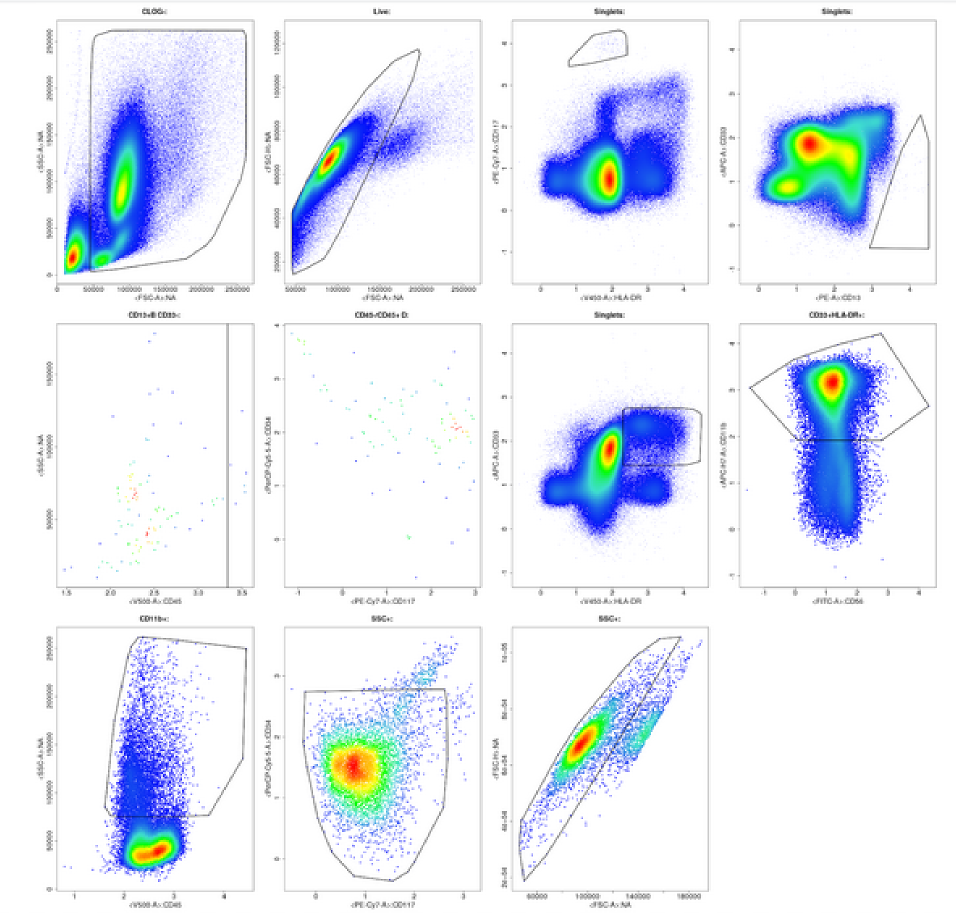
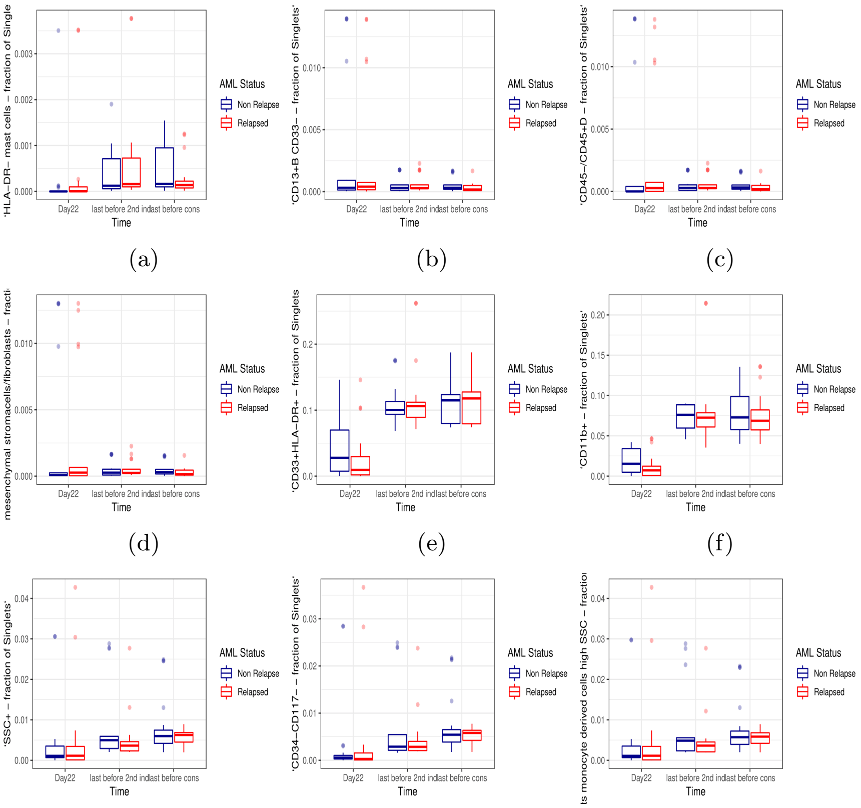


Figure 4.1: Automated gating pipeline according to tube one gating strategy

However, when comparing the two groups with healthy bone marrow individually, there were significantly different populations. A portion of these populations overlap, as illustrated in Fig4.3. I assumed that the overlapped regions are variations due to time differences, not patients' disease status. While we are interested in non-overlap cell populations in both relapsed versus normal and non-relapsed versus normal comparisons, I only reported



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.2: Cell counts across days for patients with two disease status

here the cell populations that are exclusively different between relapsed and normal.

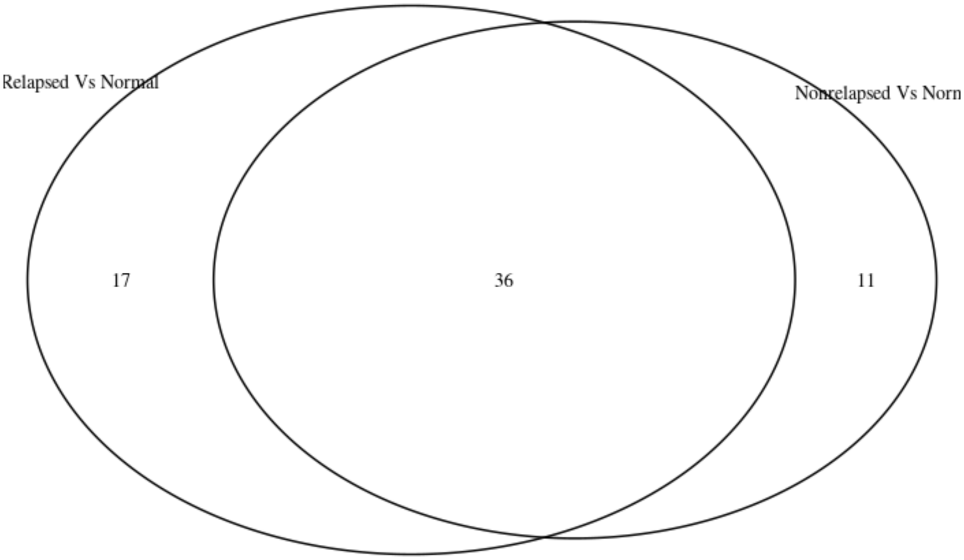


Figure 4.3: A Venn diagram generated for tube 2 last before 2nd induction illustrates the regions of interested cell populations, i.e. the non-overlapping regions of significant populations identified for relapsed vs normal (17) and non-relapsed vs normal (11).

flowDensity sets threshold for each marker into high, low expression based on a 1D density profile. flowType then reports cell counts for 3^Q phenotypes, where Q is the number of markers. Each marker has three possible outcomes: high, low expression or don't care. For reported cell populations, we only allow flowType to go down two levels in gating analysis, that is, representing each population with a maximum of four markers combination. Due to the limitation of not using a gating strategy, thresholds are set based on one single starting population. If unsupervised gating goes more than two levels, the density profile might be entirely different from the starting population, making it difficult to transfer the thresholds and verify the validity of the end population.

Optimization

The cell types returned by flowType can be redundantly represented, i.e., with uninformative markers. Uninformative markers are those that do not substantially increase the significance of the biomarkers associated with an external outcome. Marker significance can be best visualized on RchyOptimyx plot, as shown in Fig 4.4. CD64-HLADR- is the most significant biomarkers with the least number of markers used.

In tables 4.2, 4.3, 4.4, I summarized the resulting optimized biomarkers with associated p values and cell proportions across time points day22, last before 2nd induction and last before consolidation, respectively. Only a maximum of 10 populations for each day each tube are reported here.

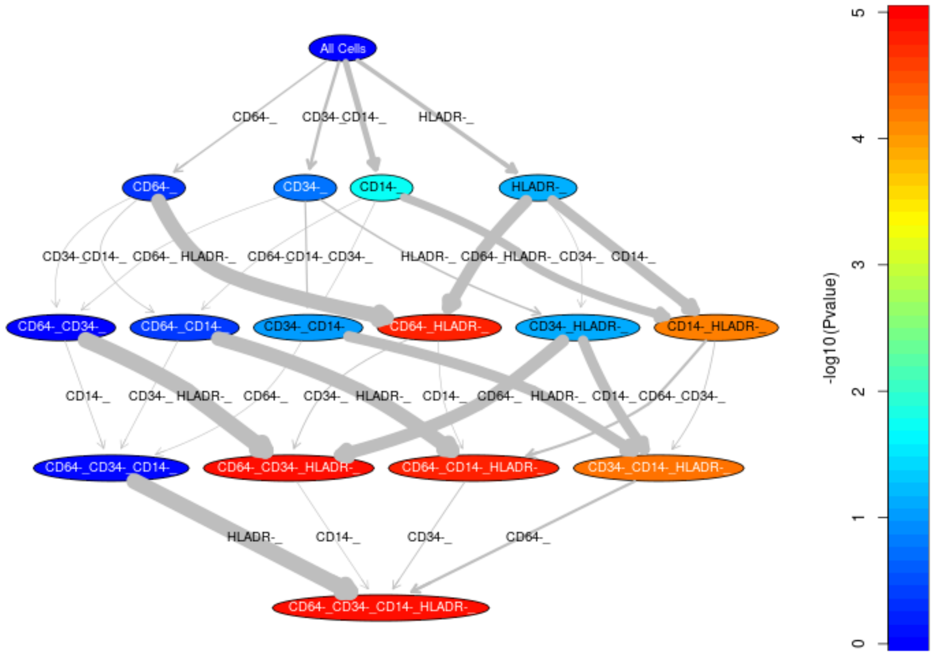


Figure 4.4: A RchyOptimyx plot lists all possible marker combinations for an end population. The biomarkers are colored by p-value. The most significant biomarkers are in red. The thickness of the arrow indicates the amount of increase of $-log10Pvalue$.

Table 4.2: Day 22 Biomarkers, their associated p-values, adjusted p-values, and the mean proportions in the relapsed, non-relapsed and normal cohorts

Phenotype	**P Value**	**P Value adjusted**	**Proportions relapsed**	**Proportions non-relapsed**	**Proportions normal**
Tube 1					
CD13-CD34-HLADR+	1.3e-05	0.021	1.70e-01	0.20	0.520
CD56+CD34-CD117+CD33+	1.4e-05	0.022	2.4e-03	0.0038	0.015
CD13-CD34-CD33+CD11B-	2.4e-05	0.040	5.7e-02	0.066	0.23
FSC-A+CD117-CD11B-HLADR-	1.5e-05	0.024	2.1e-01	0.19	0.062
CD13-CD117-CD11B-HLADR-	1.0e-05	0.017	2.1e-01	0.21	0.059
CD56-CD13-CD34-HLADR+	7.5e-06	0.012	1.6e-01	0.18	0.503
CD56+CD34-CD117+HLADR+	1.8e-05	0.030	2.4e-03	0.0031	0.014
CD13-CD117+CD33-HLADR+	1.5e-05	0.025	7.0e-03	0.017	0.037
Tube 3					
CD15+CD117+	2.6e-07	4.2e-04	0.047	0.065	0.16
CD34-CD117+	2.1e-05	3.4e-02	0.066	0.12	0.19

FSC-A+CD19-	8.3e-06	1.3e-02	0.45	0.52	0.93
CD15+CD19-	1.7e-08	2.8e-05	0.19	0.26	0.56
NG2-CD19-	3.5e-06	5.7e-03	0.41	0.55	0.90
CD117-CD19+	2.5e-05	3.9e-02	0.44	0.25	0.037
FSC-A+CD34-CD117+	1.9e-05	3.1e-02	0.061	0.11	0.18
FSC-A+CD34-CD2-	1.0e-05	1.6e-02	0.35	0.43	0.77
CD15+CD117+CD2-	3.0e-10	5.0e-07	0.029	0.050	0.16
CD34-CD117+CD2-	2.2e-05	3.4e-02	0.059	0.089	0.18
Tube 4					
FSC-A+CD7-	3.3e-06	5.1e-03	0.56	0.58	0.93
FSC-A+CD96-	3.1e-05	4.5e-02	0.66	0.70	0.92
CD117+CD123+	1.9e-05	2.8e-02	0.027	0.019	0.012
FSC-A+CD38-	2.2e-05	3.2e-02	0.37	0.53	0.93
CD96-CD38-	5.4e-07	8.4e-04	0.30	0.56	0.91
CD123-CD38-	3.6e-07	5.5e-04	0.25	0.52	0.88
CD96-CD38+	2.3e-05	3.4e-02	0.58	0.34	0.044

CD117-CD38+	2.9e-05	4.2e-02	0.52	0.31	0.033
CD123-CD38+	2.6e-05	3.9e-02	0.56	0.34	0.037
CD7-HLADR-	3.2e-05	4.7e-02	0.4	0.35	0.093

Table 4.3: Last before 2nd induction biomarkers, their associated p-values, adjusted p-values, and the mean proportions in the relapsed, non-relapsed and normal cohorts

Phenotype	**P Value**	**P Value adjusted**	**Proportions relapsed**	**Proportions non-relapsed**	**Proportions normal**
Tube 1					
CD34-CD117-	2.8e-06	0.0046	0.82	0.80	0.63
FSC-A+CD34-CD117-	3.5e-06	0.0057	0.81	0.79	0.62
CD56-CD34-CD117-	8.4e-06	0.013	0.78	0.76	0.61
CD56+CD13-CD117+	1.0e-06	0.0016	0.0031	0.0046	0.012
CD56-CD117-HLADR+	5.8e-06	0.0095	0.73	0.70	0.54
CD34-CD117-HLADR+	9.8e-06	0.015	0.73	0.70	0.54
CD56-CD117+HLADR+	1.2e-06	0.0020	0.12	0.14	0.27

CD56+CD13+CD117+CD11B-	2.7e-05	0.044	0.00063	0.0010	0.013
Tube 2					
CD36+CD64+	3.3e-06	0.0055	0.60	0.59	0.42
CD36+CD33+	4.1e-06	0.0068	0.69	0.68	0.50
CD36-CD117-CD33+	1.8e-05	0.029	0.045	0.049	0.10
CD36-CD117+CD33+	2.5e-05	0.040	0.11	0.11	0.21
CD64+CD117+CD14+	2.5e-05	0.041	0.012	0.012	0.0015
Tube 3					
CD117-	6.9e-06	1.1e-02	0.83	0.78	0.65
CD117+	6.9e-06	1.1e-02	0.17	0.22	0.35
CD117+CD2-	4.8e-06	7.8e-03	0.16	0.21	0.33
CD117+CD19-	7.6e-06	1.2e-02	0.16	0.21	0.33
Tube 4					
CD7+CD117+	1.5e-06	2.4e-03	0.0097	0.013	0.018
FSC-A+CD7+CD117-	9.6e-06	1.6e-02	0.039	0.042	0.027
CD7+CD96-CD117+	3.1e-08	5.3e-05	0.0043	0.0072	0.011

CD7+CD34-CD117+	4.4e-09	7.5e-06	0.0048	0.0066	0.012
CD7+CD117+CD123-	4.0e-07	6.6e-04	0.0086	0.011	0.017
CD7+CD117+CD38-	1.7e-06	2.8e-03	0.0075	0.0097	0.015
CD96-CD117+CD38+	1.0e-05	1.7e-02	0.0061	0.00915	0.013
CD7+CD96-CD34-CD117+	5.9e-10	9.9e-07	0.0022	0.0035	0.0071
CD7+CD96-CD117+CD123-	5.4e-09	9.1e-06	0.0035	0.0061	0.010
Tube 5					
CD34-CD4-HLADR+	9.5e-06	0.016	0.076	0.090	0.13

Table 4.4: Last before consolidation biomarkers, their associated p-values, adjusted p-values, and the mean proportions in the relapsed, non-relapsed and normal cohorts

Phenotype	**P Value**	**P Value adjusted**	**Proportions relapsed**	**Proportions non-relapsed**	**Proportions normal**
Tube 2					
CD64-HLADR-	1.6e-05	0.027	0.052	0.050	0.11
CD36+CD34-CD117+CD33+	8.5e-06	0.014	0.014	0.014	0.020

Tube 3					
CD15-	8.2e-06	0.013	0.67	0.68	0.42
CD15+	8.2e-06	0.013	0.32	0.31	0.57
FSC-A+CD15-	1.4e-05	0.022	0.65	0.67	0.41
FSC-A+CD15+	7.8e-06	0.013	0.32	0.31	0.56
CD15+CD34-	2.4e-05	0.037	0.31	0.30	0.53
CD15-CD2-	6.2e-06	0.010	0.62	0.63	0.38
CD15+CD2-	8.8e-06	0.014	0.32	0.31	0.57
CD15+CD19-	7.1e-06	0.011	0.31	0.30	0.56
Tube 4					
CD7-CD123+	1.0e-05	0.016	0.051	0.060	0.078
CD34-CD123+	1.7e-05	0.028	0.051	0.060	0.077
CD7-CD96+CD34-	3.0e-05	0.049	0.017	0.018	0.025
FSC-A+CD96-CD123-	2.4e-05	0.039	0.87	0.86	0.84
CD7-CD34-CD123+	4.5e-06	0.0075	0.041	0.051	0.070

CD96-CD34-CD123+	2.0e-05	0.033	0.050	0.060	0.076
Tube 5					
CD4-	8.8e-06	0.014	0.19	0.20	0.30
CD4+	8.8e-06	0.014	0.80	0.75	0.69
FSC-A+CD4-	1.8e-05	0.030	0.18	0.23	0.29
CD99-CD4-	1.2e-05	0.020	0.16	0.21	0.28
CD34-CD4-	3.5e-06	0.0058	0.13	0.18	0.22
CD133-CD4-	3.0e-06	0.0051	0.14	0.20	0.26
FSC-A+CD4+	6.2e-06	0.010	0.79	0.74	0.68
CD99-CD4+	5.1e-06	0.0085	0.79	0.73	0.67
CD117-CD4+	4.8e-06	0.0079	0.72	0.68	0.55
FSC-A+CD99-CD4+	3.7e-06	0.0062	0.78	0.72	0.66

We can validate the populations by plotting the gating with thresholds set by flowDensity. In Fig 4.5, I show the gating of one biomarker from each tube.

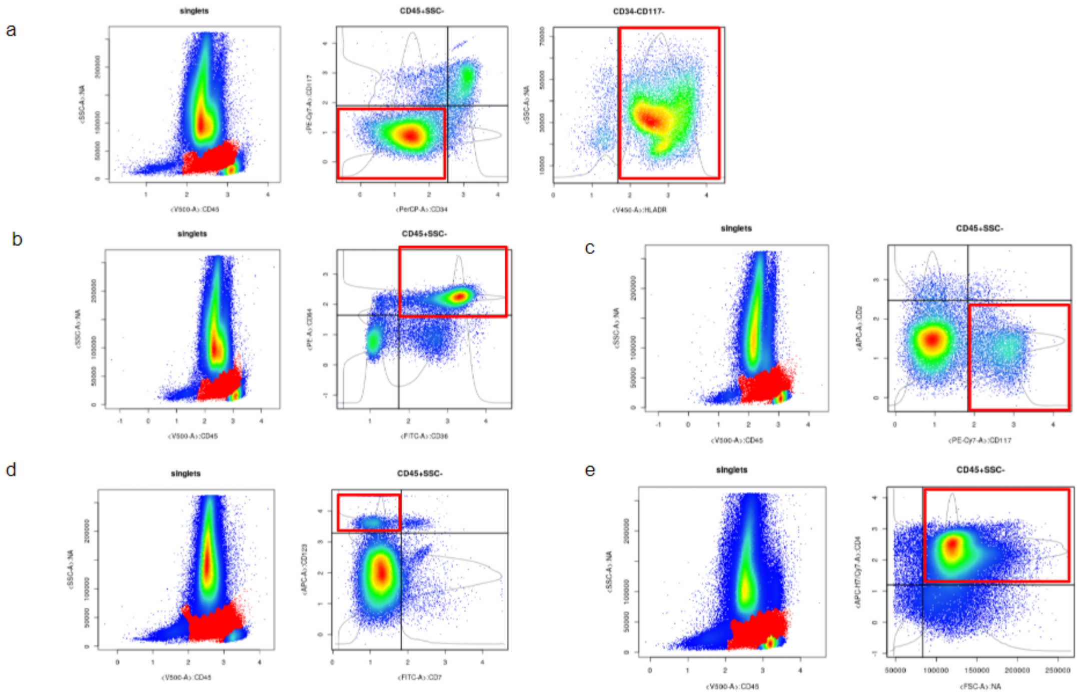


Figure 4.5: An example of gated population from tube 1-5, shown in a-e respectively, using unsupervised method.

4.3.3 Regeneration Dynamic

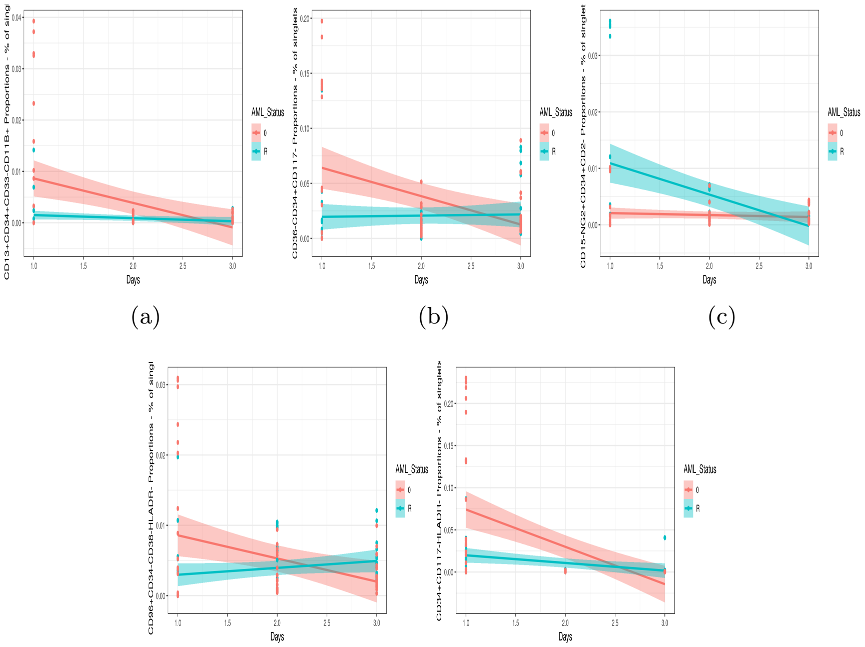
I compared the regeneration dynamics between relapsed and non-relapsed patients by fitting a linear regression model for each group. Slope comparison was done using the "contrast" function in "emmeans" package [21]. Here I show significantly different linear patterns in Fig 4.6.

I made slope comparisons for all $3^8 = 6561$ cell populations. For a large number of comparisons, there would be some significance due to chance. To reduce chance occurrence, I lowered the p-value to 0.02 and reported only the cell populations with the smallest p-value in each tube in Fig 4.6.

4.4 Conclusion

In this study, I discovered novel biomarkers through the unsupervised analysis. When time points were analyzed separately, these novel biomarkers have significantly (adjusted $pvalue < 0.05$) different proportions in the relapsed patients (n=20) compared to the healthy cohort (n=20). However, these populations were not significantly different between the relapsed and the non-relapsed patients.

When analyzing the changes of cell populations over time, I discovered populations with differential dynamics between the relapsed and non-relapsed patients.



(a)

(b)

(c)

(d)

(e)

Figure 4.6: One example of a linear model from each tube is shown. Red represents relapsed patients model. Blue represents non-relapsed patients model. a-c corresponds to tubes 1-5, respectively. Coefficients between two groups of patients have $pValue < 0.02$. Specifically, (a) CD13+CD34+CD33-CD11B+ has a p value of 0.0035,(b) CD36-CD34+CD117- 0.0024,(c) CD15-NG2+CD34+CD2- 0.00028, (d) CD96+CD34-CD38-HLADR- 0.017,(e) CD34+CD117-HLADR- 0.0018.

Chapter 5

Discussion

5.1 Data quality control study

In this thesis, I presented a new and effective approach to data quality control in the processing pipeline. I concluded that flowCut could successfully replace the current methods as a new state of the art algorithm for addressing data quality issues caused by technical variability. I provided strong evidence that flowCut improves the accuracy of the subsequent gating analysis. Overall, my research has contributed to the improvement of a crucial component in the automated analysis pipeline, which, in turn, helps to achieve the goal of resolving bottleneck issues in the field of flow cytometry.

FlowCAP studies had significant meanings in the field of flow cytometry. They not only identified the best algorithms for automated analysis but also established a method for evaluating and quantitatively comparing algorithms. This method influenced the study design of the comparative study included in this thesis. A key feature was using manual analysis as the truth, which could have both advantages and limitations. One advantage was that the manual analysis created a standard, without which the comparative study cannot occur. Another benefit was manual analysis is intuitive, with easy-to-follow procedures that do not require strong expertise. However, a major limitation was the subjectivity presented in the manual analysis. The subjectivity lied not only in the boundary placements for removal regions but also in defining the removal regions. For example, some users might not find some small spikes require removal, while others do. In addressing these limitations, I created categories based on the degree of subjectivity and evaluated each category separately. Algorithms had low F1 scores when subjectivity was high, as seen with category two files. This was intuitive, as the manual regions now contained some proportions of clean data. I chose the removal boundaries to be aggressive, i.e., removing as much bad data as necessary. However, it is worth evaluating the less aggressive cutting, that is, calculating F1 scores with narrower boundaries. This could potentially improve the overall F1 scores for category two files of all three algorithms, but the ranking of the algorithms might stay the same.

To add more confidence in the results of this study, I suggest replicating the current research by multiple people or labs and comparing the results. It would be valuable to replicate any stage of the entire protocol, from file selection to F1 score calculation.

5.2 Biomarker discovery study

In chapter 4, I documented the automated method for identifying biomarkers associated with an external clinical outcome.

Cell population identification Supervised analysis was most on par with experts' knowledge. However, it was limited by the time required to customize the pipeline. It also requires strong expertise for providing gating strategies. The customization process, including coding, verifying, and revising, lasted for 3 weeks for one tube. Once we verified the gating results, we can be confident that the subsequent analysis was accurate. We don't need to back gate to check the validity of the gates as the gates are already established during the customization process following experts' guidance. Although supervised analysis required significant upfront effort, it offered highly accurate analysis. In this study, the supervised analysis was constrained to one tube. It is worth attempting such analysis when gating strategies for other tubes are provided.

On the other hand, unsupervised cell population identification analysis was fast and scalable to all tubes. The marker thresholds were identified on 1D density distribution on CD45+SSC- cells. However, due to the elimination of human interventions in gating analysis, biologists might not find some gates agreeable, which can affect the legitimacy of the subsequent biomarkers discovered.

Biomarker discovery Biomarker discovery is done in an unsupervised manner by examining all possible marker combinations. In this study, the pipeline I used is the unsupervised cell population identification and unsupervised biomarker analysis.

For trade-off between accuracy and depth, I only let the gating analysis go down two levels as the density distribution of the starting population can be entirely different from the two or more levels down, making the gating thresholds not transferable. In other words, the thresholds found on density distribution on CD45+SSC- cells might not make sense if placed on a small subset of these cells, especially when evaluating against multiple markers. Each population is represented by a maximum of 4 markers. This can be limiting as we are examining only the tip of an iceberg. For example, tube 4 contains leukemia stem cell markers CD34, CD38 (CD34+CD38-). However, in the last two days, only CD34 is present in some populations. There was a potential of discovering more of these cell types if we increased the depth of analysis. One other limitation is the lengthy verification process to check the validity of the significant populations. In this study, I had examined the gating of around 14-30 biomarkers per tube, which was 39% (117/300) of all generated biomarkers after optimization. With increased depth of analysis, we can create an even more substantial amount of biomarkers to be examined.

For future direction, experts can set up a simple gating strategy where marker thresholds can be found easily without much customization. Relying on a gating strategy can effectively increase the depth and accuracy of subsequent analysis and reduce the lengthy validation effort.

We can investigate the unique roles of the biomarkers reported in AML patients, especially in those who relapsed after chemotherapy. The automated data analysis documented here can be one potential method for characterizing MRD regeneration patterns, which contributes to our understanding of the disease prognosis.

Bibliography

[1] N. Aghaeepour, P. Chattopadhyay, M. Chikina, T. Dhaene, S. Van Gassen, M. Kursa, B. N. Lambrecht, M. Malek, G. J. McLachlan, Y. Qian, P. Qiu, Y. Saeys, R. Stanton, D. Tong, C. Vens, S. Walkowiak, K. Wang, G. Finak, R. Gottardo, T. Mosmann, G. P. Nolan, R. H. Scheuermann, and R. R. Brinkman. A benchmark for evaluation of algorithms for identification of cellular correlates of clinical outcomes. *Cytometry A*, 89(1):16–21, Jan 2016.

[2] N. Aghaeepour, G. Finak, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, R. H. Scheuermann, D. Dougall, A. H. Khodabakhshi, P. Mah, G. Obermoser, J. Spidlen, I. Taylor, S. A. Wuensch, J. Bramson, C. Eaves, A. P. Weng, E. S. Fortuno, K. Ho, T. R. Kollmann, W. Rogers, S. De Rosa, B. Dalal, A. Azad, A. Pothen, A. Brandes, H. Bretschneider, R. Bruggner, R. Finck, R. Jia, N. Zimmerman, M. Linderman, D. Dill, G. Nolan, C. Chan, F. El Khettabi, K. O'Neill, M. Chikina, Y. Ge, S. Sealfon, I. Sugar, A. Gupta, P. Shooshtari, H. Zare, P. L. De Jager, M. Jiang, J. Keilwagen, J. M. Maisog, G. Luta, A. A. Barbo, P. Majek, J. Vil?ek, T. Manninen, H. Huttunen, P. Ruusuvuori, M. Nykter, G. J. McLachlan, K. Wang, I. Naim,

G. Sharma, R. Nikolic, S. Pyne, Y. Qian, P. Qiu, J. Quinn, A. Roth, P. Meyer, G. Stolovitzky, J. Saez-Rodriguez, R. Norel, M. Bhattacharjee, M. Biehl, P. Bucher, K. Bunte, B. Di Camillo, F. Sambo, T. Sanavia, E. Trifoglio, G. Toffolo, S. Dimitrieva, R. Dreos, G. Ambrosini, J. Grau, I. Grosse, S. Posch, N. Guex, J. Keilwagen, M. Kursa, W. Rudnicki, B. Liu, M. Maienschein-Cline, T. Manninen, H. Huttunen, P. Ruusuvuori, M. Nykter, P. Schneider, M. Seifert, and J. M. Vilar. Critical assessment of automated flow cytometry data analysis techniques. *Nat. Methods*, 10(3):228–238, Mar 2013.

[3] A. Bashashati and R. R. Brinkman. A survey of flow cytometry data analysis methods. *Adv Bioinformatics*, page 584603, 2009.

[4] Julie G. Burel, Yu Qian, Cecilia Lindestam Arlehamn, Daniela Weiskopf, Jose Zapardiel-Gonzalo, Randy Taplitz, Robert H. Gilman, Mayuko Saito, Aruna D. de Silva, Pandurangan Vijayanand, Richard H. Scheuermann, Alessandro Sette, and Bjoern Peters. An integrated workflow to assess technical and biological variability of cell population frequencies in human peripheral blood by flow cytometry. *The Journal of Immunology*, 198(4):1748–1758, 2017.

[5] G. Finak, M. Langweiler, M. Jaimes, M. Malek, J. Taghiyar, Y. Korin, K. Raddassi, L. Devine, G. Obermoser, M. L. Pekalski, N. Pontikos, A. Diaz, S. Heck, F. Villanova, N. Terrazzini, F. Kern, Y. Qian, R. Stanton, K. Wang, A. Brandes, J. Ramey, N. Aghaeepour, T. Mosmann, R. H. Scheuermann, E. Reed, K. Palucka, V. Pascual, B. B. Blomberg, F. Nestle, R. B. Nussenblatt, R. R. Brinkman, R. Gottardo, H. Maecker,

and J. P. McCoy. Standardizing Flow Cytometry Immunophenotyping Analysis from the Human ImmunoPhenotyping Consortium. *Sci Rep*, 6:20686, Feb 2016.

[6] Thomas A. Fleisher and Joao B. Oliveira. 92 - flow cytometry. In Robert R. Rich, Thomas A. Fleisher, William T. Shearer, Harry W. Schroeder, Anthony J. Frew, and Cornelia M. Weyand, editors, *Clinical Immunology (Fifth Edition)*, pages 1239 – 1251.e1. Content Repository Only!, London, fifth edition edition, 2019.

[7] K. Fletez-Brant, J. Spidlen, R. R. Brinkman, M. Roederer, and P. K. Chattopadhyay. flowClean: Automated identification and removal of fluorescence anomalies in flow cytometry data. *Cytometry A*, 89(5):461–471, 05 2016.

[8] Y. Ge and S. C. Sealfon. flowPeaks: a fast unsupervised clustering for flow cytometry data via K-means and density peak finding. *Bioinformatics*, 28(15):2052–2058, Aug 2012.

[9] F. Hahne, N. LeMeur, R. R. Brinkman, B. Ellis, P. Haaland, D. Sarkar, J. Spidlen, E. Strain, and R. Gentleman. flowCore: a Bioconductor package for high throughput flow cytometry. *BMC Bioinformatics*, 10:106, Apr 2009.

[10] Rebecca Killick and Idris Eckley. changepoint: An r package for changepoint analysis. *Journal of Statistical Software, Articles*, 58(3):1–19, 2014.

[11] J. A. Lee, J. Spidlen, K. Boyce, J. Cai, N. Crosbie, M. Dalphin, J. Furlong, M. Gasparetto, M. Goldberg, E. M. Goralczyk, B. Hyun, K. Jansen, T. Kollmann, M. Kong, R. Leif, S. McWeeney, T. D. Moloshok, W. Moore, G. Nolan, J. Nolan, J. Nikolich-Zugich, D. Parrish, B. Purcell, Y. Qian, B. Selvaraj, C. Smith, O. Tchuvatkina, A. Wertheimer, P. Wilkinson, C. Wilson, J. Wood, R. Zigon, R. H. Scheuermann, and R. R. Brinkman. MIFlowCyt: the minimum information about a Flow Cytometry Experiment. *Cytometry A*, 73(10):926–930, Oct 2008.

[12] Cossarizza A. Lugli E., Roederer M. Data analysis in flow cytometry: The future just started. *Cytometry A*, 77A:705–713, 2010.

[13] Mehrnoush Malek, Mohammad Jafar Taghiyar, Lauren Chong, Greg Finak, Raphael Gottardo, and Ryan R. Brinkman. flowDensity: reproducing manual gating of flow cytometry data by automated density-based cell population identification. *Bioinformatics*, 31(4):606–607, 10 2014.

[14] Mehrnoush Malek, Mohammad Jafar Taghiyar, Lauren Chong, Greg Finak, Raphael Gottardo, and Ryan R Brinkman. flowDensity: reproducing manual gating of flow cytometry data by automated density-based cell population identification. *Bioinformatics*, 31(4):606–607, 2015.

[15] G. Monaco, H. Chen, M. Poidinger, J. Chen, J. P. de Magalhaes, and A. Larbi. flowAI: automatic and interactive anomaly discerning tools for flow cytometry data. *Bioinformatics*, 32(16):2473–2480, 08 2016.

[16] Gianni Monaco, Hao Chen, Michael Poidinger, Jinmiao Chen, João Pedro de Magalhães, and Anis Larbi. flowAI: automatic and interactive anomaly discerning tools for flow cytometry data. *Bioinformatics*, 32(16):2473–2480, 04 2016.

[17] Sebastiano Montante and Ryan R. Brinkman. Flow cytometry data analysis: Recent tools and algorithms. *International Journal of Laboratory Hematology*, 41(S1):56–62, 2019.

[18] Kieran O'Neill, Adrin Jalali, Nima Aghaeepour, Holger Hoos, and Ryan R. Brinkman. Enhanced flowType/RchyOptimyx: a Bioconductor pipeline for discovery in high-dimensional cytometry data. *Bioinformatics*, 30(9):1329–1330, 01 2014.

[19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[20] M. Reimers and V. J. Carey. Bioconductor: an open source framework for bioinformatics and computational biology. *Meth. Enzymol.*, 411:119–134, 2006.

[21] Jonathon Love Paul Buerkner Maxime Herve Russell Lenth, Henrik Singmann. *emmeans: Estimated Marginal Means, aka Least-Squares Means*, 2020. R package version 1.4.5.

[22] J. Spidlen, K. Breuer, C. Rosenberg, N. Kotecha, and R. R. Brinkman. FlowRepository: a resource of annotated flow cytometry datasets associated with peer-reviewed publications. *Cytometry A*, 81(9):727–731, Sep 2012.

[23] P. Theunissen, E. Mejstrikova, L. Sedek, A. J. van der Sluijs-Gelling, G. Gaipa, M. Bartels, E. Sobral da Costa, M. Kotrov?, M. Novakova, E. Sonneveld, C. Buracchi, P. Bonaccorso, E. Oliveira, J. G. Te Marvelde, T. Szczepanski, L. Lhermitte, O. Hrusak, Q. Lecrevisse, G. E. Grigore, E. Fro?kov?, J. Trka, M. Br?ggemann, A. Orfao, J. J. van Dongen, and V. H. van der Velden. Standardized flow cytometry for highly sensitive MRD measurements in B-cell acute lymphoblastic leukemia. *Blood*, 129(3):347–357, 01 2017.

[24] Sofie Van Gassen, Celine Vens, Tom Dhaene, Bart N. Lambrecht, and Yvan Saeys. Floremi: Flow density survival regression using minimal feature redundancy. *Cytometry Part A*, 89(1):22–29, 2016.

Appendix A

Manual Gates for 52 files

Table A.1: Manual gates for 52 files. Every two numbers correspond to one removal region

Files	**Gates**			
Tphe0994300600_F7_R.fcs	0	2500	12000	200000
003.fcs	0	1000		
100_111 SEB.fcs	225	300		
100_111 vehicle.fcs	115	150		
125_114 Pre_6b.fcs	150	200		
15_24.fcs	3000	6000	8500	10000
2151_074_BA20120228_020.fcs	300	450		
2151_074_BA20120228_021.fcs	0	1000		
2nd grouphigh amylose maize_6_006.fcs	2500	4500		
7c_MA+.fcs	5325	6150		
9407_2_1_NKR.fcs	0	14800		
_BDL aLFA1.fcs copy	7600	10000		
TH004_TH004 mg.fcs	0	30000		
TS01 14 P7.fcs	7	18	21	23

TS01 25 P5.fcs	0	16						
TS16 406 P7.fcs	85	87						
TS21 802 P2.fcs	0	28	81	82	96	96.5	97.5	98.5
TS27 937 P4.fcs	78.5	79.5	83.7	84.3	96	97	99	110
TS27 952 P2.fcs	0	11	23	26	54	56	78.5	79.5
VD_TH003.fcs	0	1500						
Macrophages.fcs	10000	30000						
Macrophages + Leishmania + oATP.fcs	13800	30000						
Macrophages + oATP.fcs	11000	15000						
9399_2_1_NKR.fcs	2000	3000						
9399_2_4_NKR.fcs	2200	2400	3400	3600	9200	9400		
9606_3_9_NKR.fcs	16000	18000	30000	34000	55000	70000		
binding assay_dilution 6_007.fcs	0	1500						
Fig4_Algae_chlorination.fcs	0	6000						
PBL_7wk M5.fcs	15500	30000						
PBMC_HuKCD20014.fcs	0	250						

TS05 121 P6.fcs	0	20						
TS06 137 P3.fcs	0	20	82	87	95	100		
TS06 144 P5.fcs	0	3	7	11				
TS14 351 P6.fcs	0	22						
2 dias_Infectado 5.fcs	20000	30000						
13523 17012011_NS_F01.fcs	0	100						
50uM ALLNAsy48h.fcs	0	600	1100	1800	3200	3650	4500	4800
6A_liver_3d_WT.fcs	0	4000						
ah20130417 dilution_Tube_025 surface_Sup 2.fcs	0	325						
B3 C.reinhardtii H2O2 5mM_ stained.fcs	210	300						
D33 C.reinhardtii preloaded.fcs	190	260						
GDC0941CQ_D10_D10.fcs	0	20	760	1500				
IL220_pepstim1 1501.fcs(1)	0	800	28500	30000				
Macrophages + Leismania.fcs	12000	30000						
Paciente AH18_Isotipos.fcs	62.5	67.5	115	120	330	335		
PBMC_Tphe_PROP10813_P4_35_T48B_E08.fcs	0	800	1400	1500	1750	1850		

Phytoplankton_shallow sample.fcs	15000	20000				
Specimen_001_B6 LSK.fcs	1600	4000				
TS04 92 P6.fcs	3.5	5	10.5	11	97.5	98.5
TS21 813 P5.fcs	50	52	58.5	60	92.5	94.5
Unstained SPL_C3_C03.fcs	0	350	500	570	4750	4850
UR414 24_Direct ex vivo 1502.fcs	7250	7380	19150	19200	24000	26000