Dissertation

De novo binning strategy to analyze and visualize multi-dimensional cytometric data

Engineering of combinatorial variables for supervised learning approaches

zur Erlagung des akademischen Grades
“doctor rerum naturalium”
(Dr. rer. nat.)
in der Wissenschaftsdisziplin “Bioinformatik”

eingreicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
Institut für Biologie und Biochemie
AG Bioinformatik
der Universität Potsdam

und dem

außeruniversitären Institut
Deutsches Rheuma-Forschungszentrum (DRFZ) Berlin
Ein Institut der Leibniz-Gemeinschaft

vorgelegt von

Yen Hoang

Berlin, 2019

DRFZ BERLIN
Deutsches Rheuma-Forschungszentrum
Ein Institut der Leibniz-Gemeinschaft

Universität Potsdam

Matrikelnummer

800004

Zeitraum der Dissertation

01. 04. 2014 – 25. 09. 2019
AG Baumgrass, Deutsches Rheuma-Forschungszentrum (DRFZ) Berlin

Hauptbetreuer

Prof. Dr. rer. nat. Joachim Selbig

Zweitbetreuerin

Prof. Dr. rer. nat. Ria Baumgrass

Gutachterinnen und Gutachter

Prof. Dr. rer. nat. Joachim Selbig
Prof. Dr. rer. nat. Ria Baumgrass
Prof. Dr. rer. nat. Peter Beyerlein

“Simplicity is complex. It is never simple to keep things simple. Simple solutions require the most advanced thinking.” - *Richie Norton*

Dedicated to my parents, who have traded their licenses of an acknowledged engineer and physician for the pursuit of a civilized life for their child.

Acknowledgement

At this point I would like to thank all those who accompanied me during the preparation of this dissertation and contributed to the success of this dissertation with their support.

I like to start with my principal Prof. Dr. Ria Baumgrass of the German Rheumatism Research Center (DRFZ) Berlin. She initiated the research in the *PRI* studies. She has the talent of being a great supervisor and a continuous source of inspiration, support and encouragement.

I have to thank my supervisors Prof. Dr. Joachim Selbig of the Bioinformatics group at the University of Potsdam and Prof. Dr. Ria Baumgrass for offering me the opportunity to complete my dissertation at their institutions. Furthermore, I am very thankful for their professional advices and progress reviews.

In addition, I like to thank Dr. Detlef Groth for the initial support to get this project rolling and for further support in supervising the master students.

I like to acknowledge the profound biological expert knowledge of Prof. Dr. Ria Baumgrass and Dr. Stefanie Gryzik. The close teamwork while creating and evaluating the workflow and the implementation of the interactive tool of *PRI-ana* and *PRI-base* was very fruitful. Our sharing of ideas and the constructive critism took a great part in the progress of this study.

I am thankful for the former colleagues and students of the research group Baumgrass for their contributions and emotional support: Dr. Timo Lischke, Isabelle Kadner, Martin Schädlich, Ariana Mekonnen, Dr. Martin Karl and Dr. Katharina Hecklau. And a very special thank to Dr. Stefanie Gryzik who enlightened this tough path with her bright presence. Thank you for the countless lunch dates.

My special gratitude goes to Prof. Dr. Peter Beyerlein, who guided me through a big part of my studies and life decisions and who enabled the path for an exciting experience abroad and my doctorate.

Finally I would like to thank all my family and friends for their love and understanding. Without them I could not finish this dissertation.

Allgemeinverständliche Zusammenfassung

"De novo binning strategy to analyze and visualize multi-dimensional cytometric data"

Yen Hoang, Deutsches Rheuma-Forschungszentrum (DRFZ) Berlin

Massen- und Durchflusszytometrie-Messungen ermöglichen die detaillierte Einteilung von Zellgruppen nach Eigenschaften vor allem in der Diagnostik und in der Grundlagenforschung anhand der Erfassung von biologischen Informationen auf Einzelzellebene. Sie unterstützen die detaillierte Analyse von komplexen, zellulären Zusammenhängen, um physiologische und pathophysiologische Prozesse zu erkennen, und funktionelle oder krankheitsspezifische Characteristika und rare Zellgruppen genauer zu spezifizieren und zu extrahieren. In den letzten Jahren haben zytometrische Technologien einen enormen Innovationssprung erfahren, sodass heutzutage bis zu 50 Proteine pro Zelle parallel gemessen werden können. Und das mit einem Durchsatz von Hunderttausenden bis mehreren Millionen von Zellen aus einer Probe. Bei der Zunahme der Messparameter steigen jedoch die Dimensionen der kombinierten Parameter exponentiell, sodass eine komplexe Kombinatorik entsteht, die mit konventionellen, manuellen Untersuchungen von bi-axialen Diagrammen nicht mehr durchführbar sind. Derzeit gibt es schon viele neue Datenanalyse-Ansätze, die vorranging auf Cluster- bzw. Dimensionsreduktionstechniken basieren und meist mit einem vorgeschalteten Downsampling in Kombination eingesetzt werden. Diese Tools produzieren aber komplexe Ergebnisse, die größtenteils nicht reproduzierbar sind oder Proben- und Gruppenvergleiche erschweren.

Um dieses Problem anzugehen wurde in dieser Dissertation ein reproduzierbarer, halbautomatisierter Datenanalyse-Workflow namens *PRI* entwickelt, was für *pattern recognition of immune cells* (Mustererkennung von Immunzellen) steht. Dieser Workflow ist in drei Hauptteile untergliedert: die Datenvorbereitung und -Ablage; die Entwicklung innovativer, bin-basierter Merkmale von drei kombinierten Parametern namens TriploTs und dessen weiterführende Einteilung in vier gleich große TriploT-Areale; und die Anwendung von einem maschinellen Lernansatz basierend auf der Information von diesen Arealen. Als Ergebnis bekommt man eine Selektion der Areale, die am häufigsten von den überwachten Modellen ausgewählt wurden. Dies soll dem Wissenschaftler entscheidend dabei helfen, Zellpopulationen zu identifizieren, die am besten zwischen zwei Gruppen unterscheiden. Der vorgestellte Workflow *PRI* ist exemplarisch an einem kürzlich veröffentlichten Massenzytometrie-Datensatz validiert worden. Die von den Originalautoren hervorgehobene Zellpopulation konnte nicht nur identifiziert werden, sondern sogar wesentlich weiter spezifiziert werden. Außerdem wurden weitere Erkenntnisse von relevanten, kombinatorischen Proteinexpressionen festgestellt. Die Entwicklung der reproduzierbaren TriploTs führt dazu, dass sie als Basis für verständliche und leicht interpretierbare Visualisierungen, für eine strukturierte Erforschung der Daten mithilfe der Selektion der Areale, und für neuronale Netzwerkkonstrukte genutzt werden können.

PRI ermöglicht eine optimierte, semi-kontinuierliche Bestimmung der Expressionsstufen, die die Identifizierung von dominant vorherrschenden und diskriminierenden Proteinen in Zellsubpopulationen wesentlich erleichtert. Darüberhinaus erlaubt es die intuitive Erfassung von korrelierenden Mustern durch die innovative, reproduzierbare Darstellung der Proteinkombinationen und hilft bei der Erforschung von Zellsubpopulationen.

Abstract

“De novo binning strategy to analyze and visualize multi-dimensional cytometric data”
- Engineering of combinatorial variables for supervised learning approaches -

Yen Hoang, German Rheumatism Research Center (DRFZ) Berlin

Since half a century, cytometry has been a major scientific discipline in the field of cytomics - the study of system’s biology at single cell level. It enables the investigation of physiological processes, functional characteristics and rare events with proteins by analysing multiple parameters on an individual cell basis. In the last decade, mass cytometry has been established which increased the parallel measurement to up to 50 proteins. This has shifted the analysis strategy from conventional consecutive manual gates towards multi-dimensional data processing. Novel algorithms have been developed to tackle these high-dimensional protein combinations in the data. They are mainly based on clustering or non-linear dimension reduction techniques, or both, often combined with an upstream downsampling procedure. However, these tools have obstacles either in comprehensible interpretability, reproducibility, computational complexity or in comparability between samples and groups.

To address this bottleneck, a reproducible, semi-automated cytometric data mining workflow *PRI* (pattern recognition of immune cells) is proposed which combines three main steps: i) data preparation and storage; ii) bin-based combinatorial variable engineering of three protein markers, the so called triploTs, and subsequent sectioning of these triploTs in four parts; and iii) deployment of a data-driven supervised learning algorithm, the cross-validated elastic-net regularized logistic regression, with these triploT sections as input variables. As a result, the selected variables from the models are ranked by their prevalence, which potentially have discriminative value. The purpose is to significantly facilitate the identification of meaningful subpopulations, which are most distinguish between two groups. The proposed workflow *PRI* is exemplified by a recently published public mass cytometry data set. The authors found a T cell subpopulation which is discriminative between effective and ineffective treatment of breast carcinomas in mice. With *PRI*, that subpopulation was not only validated, but was further narrowed down as a particular Th1 cell population. Moreover, additional insights of combinatorial protein expressions are revealed in a traceable manner. An essential element in the workflow is the reproducible variable engineering. These variables serve as basis for a clearly interpretable visualization, for a structured variable exploration and as input layers in neural network constructs.

PRI facilitates the determination of marker levels in a semi-continuous manner. Jointly with the combinatorial display, it allows a straightforward observation of correlating patterns, and thus, the dominant expressed markers and cell hierarchies. Furthermore, it enables the identification and complex characterization of discriminating subpopulations due to its reproducible and pseudo-multi-parametric pattern presentation. This endorses its applicability as a tool for unbiased investigations on cell subsets within multi-dimensional cytometric data sets.

Contents

List of illustrations

Figures

Tables

List of abbreviations and notations

PRI	*pattern recognition of immune cells*
CLARA	clustering for large applications
CNN	convolutional neural network
CV	cross validation
eff	effective
erLR	elastic-net regularized logistic regression
FCS	flow cytometry standard
FMO	fluorescence minus one
GLM	generalized linear model
GUI	graphical user interface
ineff	ineffective
LM	linear model
LR	logistic regression
MDS	multi-dimensional scaling
MSI	mean signal intensity
MSI+	mean signal intensity of positive cells
NA	not available
OLS	ordinary least squares
PAM	partitioning around medioids
RMSE	root mean square error

RSEM	relative standard error of the mean
SD	standard deviation
SI	signal intensity
t-SNE	*t*-distributed stochastic neighbor embedding
VE	variable engineering
VR	variable ranking
VS	variable selection

N	number of samples
c	number of cells
m	number of markers
p	number of input variables
p'	subset of input variables
p''	final reduced subset of input variables
q	number of sections per triploT
B	number of triploT matrices
$\mathbf{Y}$	output vector
y_j	output element, $j = 1..N$
$\mathbf{X}'$	variable matrix, transposed
x_{ji}	variable element, $j = 1..N$ and $i = 1..p$
$arcsinh$	inverse hyperbolic sine
η_j	target function, $j = 1..N$
β_{ji}	regression weights vector, $j = 1..N$ and $i = 1..p$
β_{j0}	regression bias, $j = 1..N$
k	folds in CV
α	degree of elastic-net
λ	degree of shrinkage

1 Introduction

Studies of cells are important in the field of fundamental research, clinical diagnostics, water analytics, physiology and pathophysiology. There are two main fundamental strategies to be distinguished: bulk and single cell analyses. Bulk cell analyses can examine cell compounds as a whole and has provided key insights in, for example, cancer biology and micro-biomedical research. However, only the average genotype or expression signal for an ensemble of cells can be measured. To reveal the heterogeneity of cell populations and to discover the underlying combinatorial cell mechanisms, single cell analyses are necessary. Among these analyses, cytometry allows profound investigation of physiological processes, such as intracellular cytokine production and cellular proliferation, functional characteristics like cell viability and cell cycle, and rare events by analyzing multiple proteins on an individual cell basis, and by categorizing these cells with characteristics such as producing or non-producing for a certain protein [1, 2, 3].

In the field of cytometry, there are mass and flow cytometers, which measure the protein abundances by labelling the cells with metal isotopes and fluorescent markers, respectively. In the last decade, these techniques have witnessed a great increase in the number of measurable parameters. Present mass cytometers detect up to 50 channels in parallel, with the throughput of hundreds of thousands to millions of cells from an individual sample [4]. This leads to large data sets with a large number of protein combinations, which require efficient and comprehensive examination. A more extensive analysis would enable a better understanding in characterizing the properties of combined protein expression and in detecting intracellular interactions which coordinate activities of various cell types according to genetic and environmental contexts.

In cytometric data, each cell has its own unique expression of surface and intracellular proteins. Dealing with millions of cells with different characteristics, it is especially difficult to detect a cell subpopulation, which has a discriminative expression pattern between two or more groups. These subpopulations have the potential to be part of a disease mechanism or can be used as a biomarker, for example, to discriminate between responder and non-responder in cancer therapies. They are therefore the major target of many cytometric analyses. However, since the protein interactions are not fully understood, the expression of one protein in a cell can have a strong or weak activating or inhibitory effect, or no effect at all. These basic characteristics of proteins suggest a complexity that can be difficult to investigate, especially when trying to understand protein function in the proper biological context.

If $m = 50$ proteins are measured from each sample and each protein expression is considered only to be binary, as either positive or negative, the number of possible 'protein combinations' reaches over a billion ($2^{m=50}$). In addition, in many studies there is a limited amount of samples available to deeply explore biological interrelationships. When the ratio of protein combinations ($p = 2^{50}$) to the number of samples (N) is disproportionately high, one faces a $p >> N$ problem in classification or cluster analysis, and multidimensional scaling. The problem of estimating a predictive and accurate function becomes vastly harder as p, the dimension of the protein combinations as variables x, increases. This is called the curse of dimensionality [5]. In summary, with increasing number of dimensions p, more samples are required, storage and running time scale up exponentially and model results are prone to over-fit. Thus, small data sets with high-dimensional protein combinations underlie the curse of dimensionality. The more variables are present, the more data points are needed in order to fill the space and to avoid over-fitting. This is a common problem in the biotechnological and clinical area [6, 7].

1.1 Making sense of high-dimensional cytometric data

As the number of the measured proteins rises, conventional manual inspections on biaxial contour plots are time consuming and not viable, since the number of these plots increases exponentially with the number of measured protein markers. In other words, a sample in an experiment with $m = 50$ marker would require the investigation of $m \cdot (m - 1) = 2,450$ plots from every two marker combination. In the last decade, many sophisticated analysis techniques have been developed to examine cytometric data with an emphasis to overcome the curse of dimensionality in order to obtain information about the underlying characteristics of the cells. Some of them have been reviewed and benchmarked several times [3, 8, 9, 10, 11]. *viSNE* is one of the most popular dimensionality reduction techniques, which makes use of the algorithm *t*-distributed stochastic neighbor embedding (t-SNE) [12]. Another tool is *Scaffold Maps*, which uses the cluster algorithm partitioning around medioids (PAM) and combines the resulting clusters with manual landmarks to create force-directed graphs [13]. *Citrus* combines hierarchical clustering of cell events with machine learning approaches to identify statistically significant clusters between groups of samples, or to build a predictive model for a particular sample type [14]. However, these clustering algorithms need the amount of clusters or the minimum amount of cells to define a cluster to be set beforehand, and are often unable to detect low expression differences or rare cell subtypes due to the similar characteristics of the cells compared to other major populations. Matching these small cell populations across multiple samples is even more challenging. Manual analysis is often subjective and not reproducible, but prior biological knowledge provides guidance to reasonably identify these populations. Nevertheless, integrating this information into the exploratory clustering process has not often been deployed [15]. Furthermore, these methods are facing high computational efforts, which leads to integration of downsampling techniques in their approaches. Therefore, rare cell types are often not placed in any cluster since they are likely not contained in the remaining cells after the downsampling step. Hence, the pursuit of suitable methods is ongoing: a tool which is computationally inexpensive and is capable of generating reproducible, intuitively interpretable results with group comparison possibilities.

1.2 Author's approach

The author's approach is addressing cytometric data, in which information of a single experiment is represented by thousands to millions of rows. Each row represents a single cell and each column a protein marker with signal intensities (SIs) as values. This dissertation proposes a workflow named *PRI* for pattern recognition of immune cells, and is divided into three main steps: data preparation and storage, combinatorial variable engineering (VE), and subsequent machine learning on the basis of these innovative variables (Fig. 1.2). In particular, the workflow begins with processing the data directly after obtaining the raw files from the measurement device. It involves gating, data storage with the in-group implemented database management system *PRI-base*, data transformation, outlier removal and quality control. The second step is the VE method which is based on a binning strategy and combines information about two and three parameters. These engineered variables are herein called diploT and triploT, respectively. They serve for subsequent downstream analysis, and for visualization and manual inspection. Furthermore, they function as explanatory variables explored in the embedded variable ranking (VR) or, for example, as input layer in deep learning.

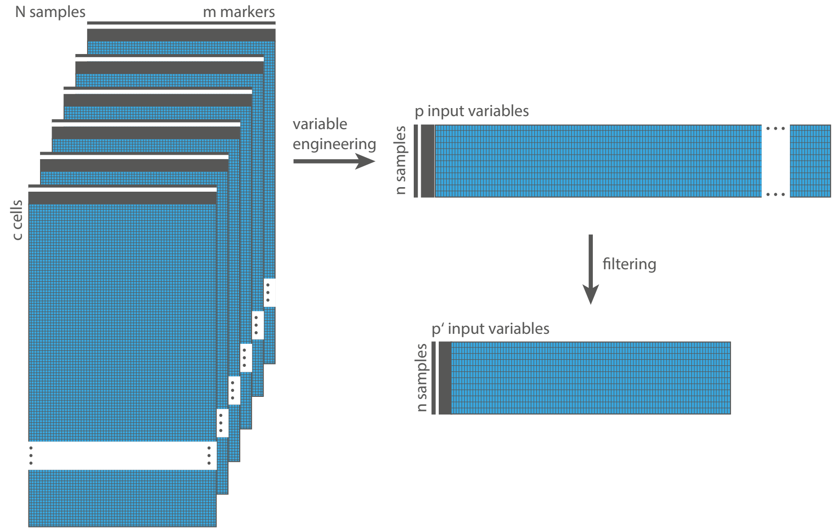


Figure 1.1: Scheme of the raw cytometric data structure and the dimension reduction due to variable engineering and domain knowledge filtering.

After the VE, the data structure is transformed from N number of sample tables which has different number of cells (c) multiplied by a fixed number of protein markers (m) into one table of N sample rows with p input variables as columns. Figure 1.1 illustrates the crucial data transformation and dimension reduction of the raw data, but the table incorporates the statistical measurements extracted from the original bi-axial plots. The elastic-net regularized logistic regression (erLR), a supervised machine learning algorithm, used in a nested cross validation (CV) proposes a ranking table of these engineered variables, which are significant and differentiate most between two groups.

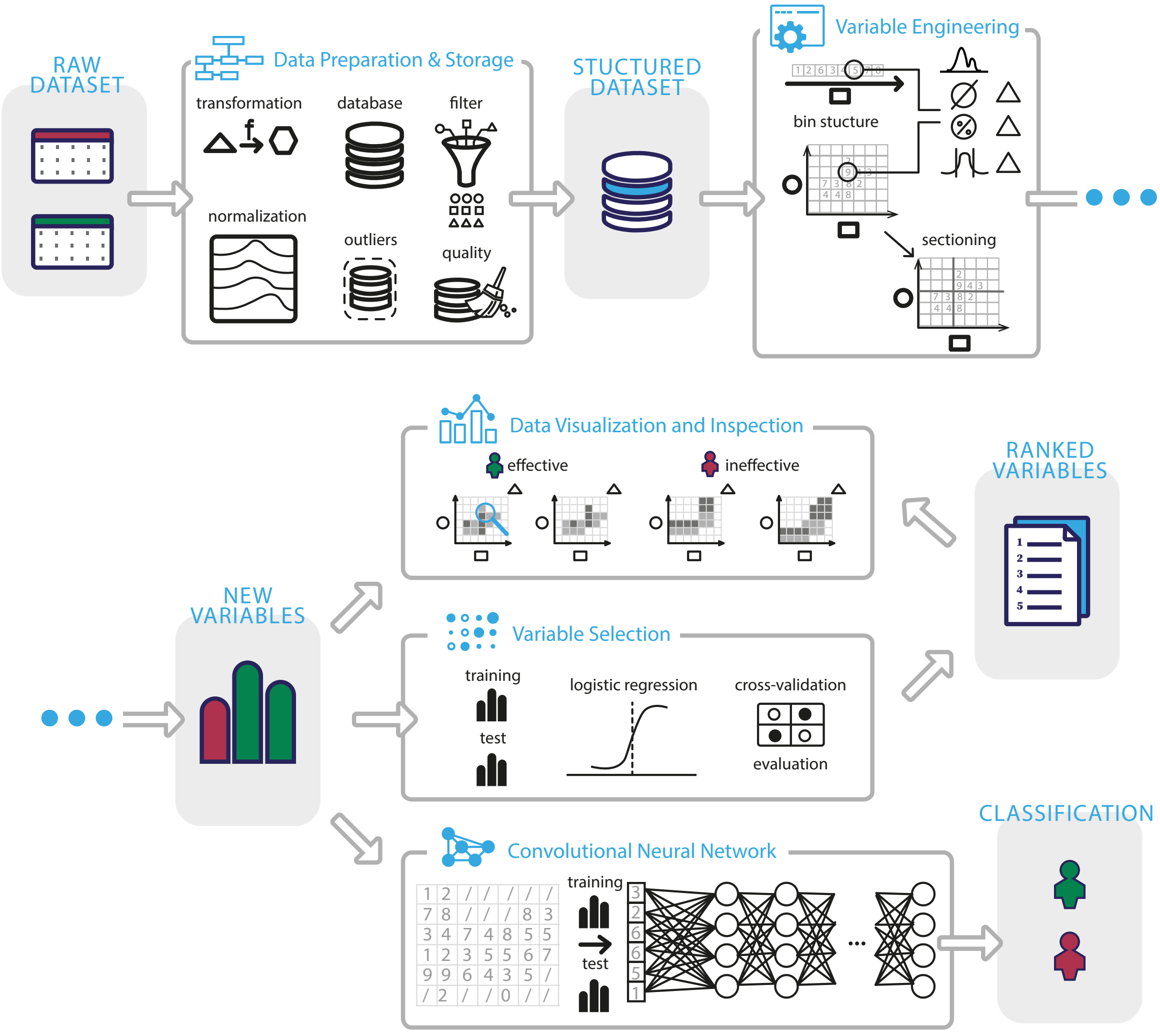


Figure 1.2: Proposed analysis workflow *pattern recognition of immune cells* (*PRI*) for cytometric data includes three main steps. In the data preparation step, raw files are gated and are stored within the in-group database management system tool *PRI-base*, signal intensity (SI) values are transformed, outliers are removed, and optional normalization and filters are applied. The data files are then combined to one structured data set. In the variable engineering (VE) step, cells in a uni-axial or bi-axial plot are grouped into bins and different statistical operations are applied discretely on these bin cells. Data visualization can hereby follow for manual inspection (top arrow). To obtain a list of interesting protein marker combinations, an additional refining step is deployed which outcomes then serve as basis for a supervised machine learning algorithm. Ranked variables are then suggested to the examiner to support the detection of subpopulations which can lead to biologically novel insights in cellular interactions (middle arrow). Another option arises for deep learning algorithms to create precise classification models (bottom arrow).

1.3 Objective

The goal of this study is to develop a comprehensive analysis workflow for cytometric data. To present the feasibility of the workflow, it is applied on the recently published public mass cytometric data set from *Spitzer et al.* [13]. The data is obtained from female mice with breast carcinomas, which were treated in four different ways. These treatments are categorized in two groups: *effective* and *ineffective* treatments. The authors identified a T cell subpopulation which discriminates between both treatments. Chapter 2 describes the example data set in more detail, as well as the basis of cytometric data along with conventional and recent approaches to examine these data. VE and variable selection (VS) are also introduced, with the focus on the embedded machine learning technique erLR and nested CV deployed in this dissertation. The demonstrated re-analysis in Chapter 3 strives, firstly, for manifesting the added value of the engineered variables. Subsequently, it is aimed to show the practicability of the VR approach which is designed to guide the investigator with specific discriminative three-protein marker combinations. To further facilitate the usage of this workflow, the implementation of a tool in *R* with a graphical user interface (GUI) is intended. Moreover, the combined VE and VR workflow is under review in Chapter 4. Both parts are subject to comparison with the previously introduced conventional and state-of-the-art analysis strategies. The purpose is to discover, if they enable a reproducible and interpretable examination at low computational cost which can cope with the high-dimensional marker combinations. At the end, Chapter 5 summarizes the overall findings in this dissertation and advises the scope and improvements that can be made in future work.

2 Background

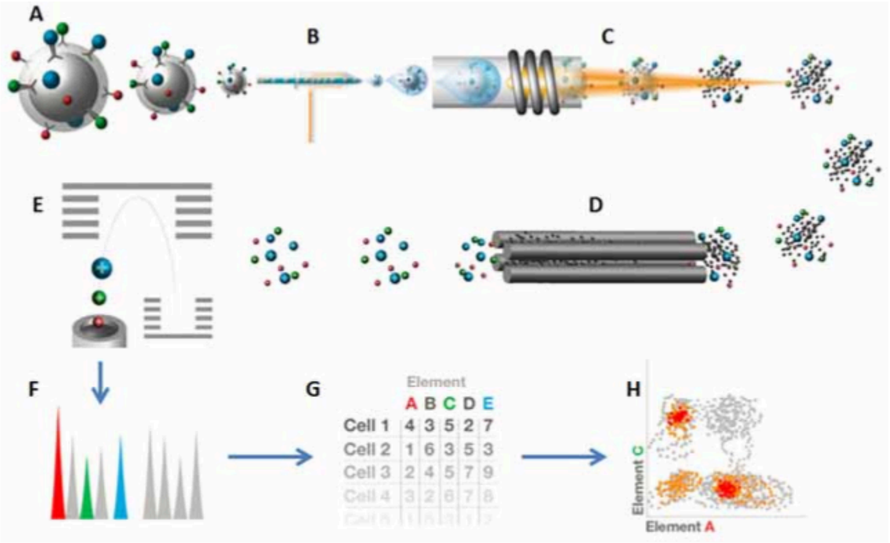
This chapter provides an outline of the topics involved in this dissertation, consisting of four sections. The first section briefly presents the technology of mass cytometry with its benefits and limitations compared to the similar technique flow cytometry. The second section introduces the conventional and state-of-the-art cytometric analysis tools and that used from *Spitzer et al* [13]. Section three gives an overview of variable selection (VS). It will depict the three general approaches, whereby the embedded method is used in this study. The supervised variable ranking (VR) algorithm with the applied regularization and statistics are then introduced. The last section contains description of the biological data, which was re-analyzed with the proposed analysis workflow, and the original authors' main results.

2.1 Mass cytometry

Cytometry involves measurement of single cells and analyses of quantitative single cells. It paves the way for research on cellular heterogeneity, characterization of rare cell subpopulations, discovery of biomarkers, understanding functionality, tracing lineages of cellular phenotypes, and comparing abundance of cell populations between different conditions, for example between patient groups [16]. Flow cytometry and the more recently introduced mass cytometry, also called cytometry by time-of-flight mass spectrometry, are high-throughput technologies that measure protein abundance on exterior surface or intracellular on a single-cell level. Due to the nature of the chosen example data set, the generation of cytometric marker intensities and the process of cytometric marker measurements in mass cytometry is described, and benefits and limitations compared with flow cytometry are presented in the following sections.

2.1.1 Technology

Mass cytometry utilizes antibodies tagged with metal isotopes from the lanthanide series [17]. In general, these labeled antibodies bind to specific proteins on exterior surface of the cell or intracellular. Up to 50 different labeled antibodies can then be detected per cell by a cytometer in parallel (Fig. 2.1). Thus, the signal intensity (SI) of each antibody is proportional to the abundance of the specifically bound protein. After measurement,the cytometer device will produce a table with m columns for m protein markers and c rows, each row corresponding to one cell, where the position of the cells are not relevant. The data is stored in flow cytometry standard (FCS) format, which is a combination of textual data with device specific information, followed by the intensity measurement as binary data [18].



	Element				
	A	B	C	D	E
Cell 1	4	3	5	2	7
Cell 2	1	6	3	5	3
Cell 3	2	4	5	7	9
Cell 4	3	2	6	7	8

Figure 2.1: Schematic workflow of generation of cytometric marker intensities and the process of cytometric marker measurements in mass cytometry. Cells labeled with metal-conjugated antibodies in solution (A) are injected into the nebulizer (B). They are reduced to single cell-containing droplets and are directed to the torch, where they are vaporized, atomized, and ionized in the plasma (C). The low-mass ions are removed (D), resulting in an ion cloud that enters the time-of-flight mass analyzer. The ions are separated based on their mass and are accelerated to the detector (E). The detector measures the quantity of each isotope for each individual cell in the sample. The data is generated in the FCS format (G) and analyzed e.g. in a conventional manner with bi-axial scatter plots (H) [17].

2.1.2 Advantages and disadvantages of mass cytometry compared to flow cytometry

Flow cytometry has been developed early in the 1950's and uses fluorophore labeled antibodies which attach to the protein. This has the advantage, that this method is non-destructive, thus, can be used to sort cells for further analysis. Common flow cytometric experiments measure 6–12 parameters, with modern systems measuring up to 20 channels [19], while new developments (e.g. BD FACSymphony [20]) promise to increase this capacity towards 50. Moreover, flow cytometry offers the highest throughput with tens of thousands of cells measured per second at relatively low operating costs per sample. The order of magnitude is about $10^5 - 10^7$ cells per sample. However, because of the spectral overlap between fluorophores, the number of parameters that can be reliably measured in parallel is still limited, and antibody panel design and correcting the so-called spillover of the data is a crucial part of flow cytometry [21]. On the contrary, by using rare metal isotopes, mass cytometry is not light- or time sensitive. Cell auto-fluorescence can be avoided, and the spectral overlap is drastically reduced, but is still present due to metal impurities and oxide formations, e.g. through coupling of antibodies to neighboring metals [22]. Yet, mass cytometry supports a higher dimension of parameters reliably measured per cell, with current panels using 50 parameters and the promise of up to 135, but the process throughput is slower (hundreds of cells per second) [17]. Combined with the study design, mass cytometric measurements usually result in a lower magnitude of cells per sample ($10^3 - 10^6$). Furthermore, the cells are destroyed during the ionization step (Fig. 2.1B).

Another aspect is normalization. In general, to assure a qualitative comparison of individual or groups of samples, it is necessary to normalize the data set. It is difficult to estimate if variations are background noise due to technical issues (e.g. instrument performance, sample storage and preparation) or real biological differences between samples [22, 23]. This is especially difficult with multi-center studies. For single-cell RNA sequencing there are house-keeping targets which can be used to remove technical variations from a sample and to determine data quality [15]. For flow cytometric data, examiners can use the modified measurements called fluorescence minus one (FMOs). An FMO contains all the fluorochromes in a panel, except for one. Hence, for m protein markers in a staining panel, one has m FMOs. As a consequence, SIs of a protein marker found in the FMOs are false signals due to background light contamination. Therefore, thresholds for true positive protein marker signals can be set. However, for mass cytometric data it is especially challenging, since there are no similar standard procedures.

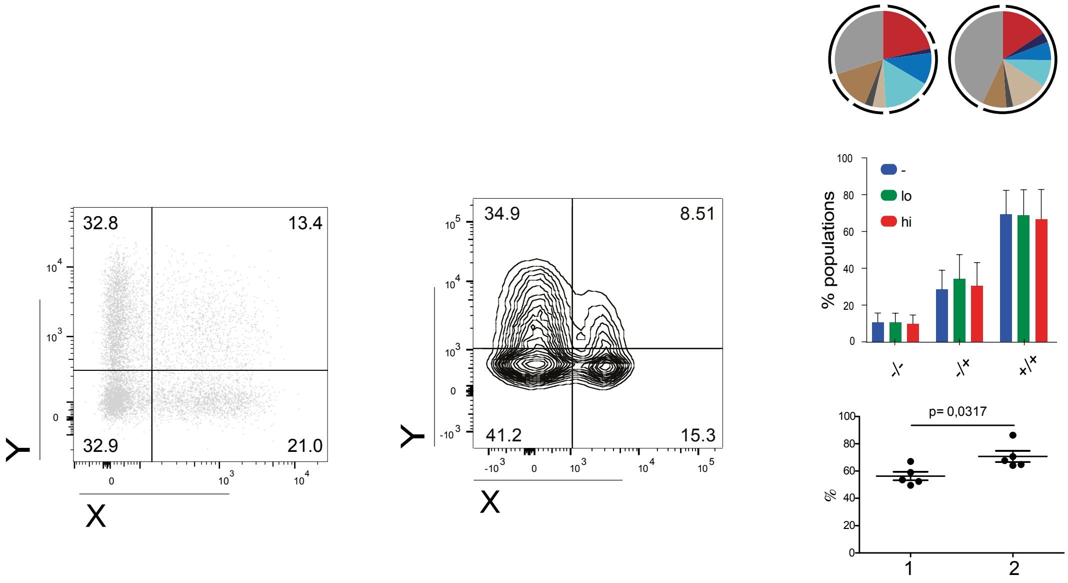
Due to the fast development of cytometers and thus the fast increase of measurable parameters, the demand for suitable exploratory tools to cope with high-dimensional data is present. In the following section, state-of-the-art analysis strategies are presented, which are used for comparison with author's approach.

2.2 Conventional and state-of-the-art analysis strategies

Due to the recent increase in the amount of simultaneous channel detection, many complex approaches for cytometric analyses have been developed to tackle the newly present curse of dimensionality in the marker combinations. These tools are mainly based on dimension reduction, clustering or graph theory [2, 3, 8, 11]. In this dissertation the common conventional and state-of-the-art techniques in single-cell analysis are introduced. For the latter, *viSNE* and *Citrus* are two of such widely used tools [12, 14]. *viSNE* refers to the visualization of the local data structure, and *Citrus* aims at identifying significant clusters in group comparisons. *Scaffold Maps* is a clustering-based technique as well and is presented in this dissertation, since this algorithm was used in the original authors' study from to eventually obtain their results [13]. Last but not least, *CellCnn* is introduced, which firstly uses a deep learning technique [24].

2.2.1 Conventional approaches

In the field of cytometry, the identification of cell population typically has been processed by manual gating, where a series of two-dimensional scatter plots are visually analyzed one after another. At each scatter plot, a subset of cells, either positive or negative for the two visualized markers, is selected and further filtered in the subsequent iterations until populations of interest across a range of marker combinations are captured. In the end, means and frequencies of subpopulations are opposed by bar, box or circle plots (Fig. 2.2). However, this so-called manual gating has drawbacks, such as subjectivity in setting the cutoff between positive and negative and bias toward well-known cell types. Furthermore, it is inefficient when analyzing large datasets, which also contributes to a lack of reproducibility. This inefficiency increases with the amount of measured parameters.



(a) scatter plot **(b)** contour plot **(c)** circle, bar and box plot

Figure 2.2: Conventional approaches to visualize and analysis cytometric data. Exemplified biaxial scatter **(a)** and contour plots **(b)** are serially gated until populations of interest across a range of marker combinations are captured for each sample. The means or frequencies of each subpopulation are captured and compared between samples with circle, bar or box plots with optional significance analysis **(c)**.

2.2.2 Recent approaches

viSNE

viSNE is a dimension reduction technique and makes use of the unsupervised *t*-distributed stochastic neighbor embedding (t-SNE) algorithm combined with a multi-core-processing option. The algorithm begins by calculating the pairwise similarity matrix in high-dimensional space and randomizing a starting position for each point in the low-dimensional space using the *Euclidean* metric (Fig. 2.3). It iteratively updates the position of points in low-dimensional space, resulting in the minimization of the relation between the similarities in high- and low-dimensional space using the *Kullback-Leibler* divergence (Eq. 2.1-2.6 [12]). There are four parameters to configure. Default options are: $iterations = 1,000$, $perplexity = 20$, $theta = 0.5$, and $eta = 200$. *iterations* denotes the number of iterations to calculate the distances between the cells, *perplexity* is the number of neighbors, while *eta* describes the learning rate. *theta* is a scale between 0 and 1, which indicates the trade-off between speed and accuracy. The higher the value of *theta*, the higher the approximation. The applied algorithm and the *Kullback-Leibler* divergence is shown in the following equations.

Let a_i be the i^{th} object in high dimensional space,

with σ_i as a_i's variance (∼perplexity).

Let b_i be the i^{th} object in low dimensional space.

Construct probabilities:

$$p_{j|i} = \frac{e^{(-||a_i - a_j||^2/2\sigma_i^2)}}{\sum_{k\neq i} e^{(-||a_i - a_k||^2/2\sigma_i^2)}} \tag{2.1}$$

$$q_{j|i} = \frac{e^{(-||b_i - b_j||^2)}}{\sum_{k\neq i} e^{(-||b_i - b_k||^2)}} \quad \text{, with } p_{i|i} = q_{i|i} = 0 \quad . \tag{2.2}$$

The joint similarity of a_i and a_j in high-dimensional space is described as

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{2.3}$$

Student t distribution with one degree of freedom is used to represent the low dimension:

$$q_{ij} = \frac{(1 + ||b_i - b_j||^2)^{-1}}{\sum_{k\neq l}(1 + ||b_k - b_l||^2)^{-1}} \tag{2.4}$$

Minimizing the sum of *Kullback-Leib*ler divergences between the joint probability distributions P and Q with

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} log(\frac{p_{ij}}{q_{ij}}) \tag{2.5}$$

by the gradient of the cost function

$$\frac{\delta C}{\delta b_i} = 4 \sum_j (p_{ij} - qij)(b_i - b_j)(1 + ||b_i - b_j||^2)^{-1} \quad . \tag{2.6}$$

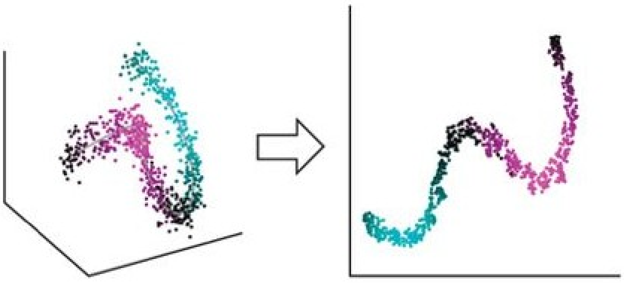


Figure 2.3: Scheme of the t-SNE algorithm from three dimensions to two dimensions. "1,000 points are randomly distributed with normally distributed noise around a polynomial of the third degree. *viSNE* projects a one-dimensional curve embedded in three dimensions (left) onto two dimensions (right). The color gradient shows that points that are in close proximity in three dimensions remain in close proximity in two dimensions" [12].

Citrus

Citrus is a mixture of unsupervised and supervised algorithms and comprises three main steps. The first step uses the conventional agglomerative hierarchical clustering to identify cell clusters within the dataset. The dissimilarity between any two cells is specified by *Ward's* linkage used as the agglomeration method and the *Euclidean* metrics between cluster markers (Eq. 2.7-2.8). The second step consists of the calculation of a statistical feature from these clusters, which is either the median expression level of the markers or the frequency of the cell clusters compared to the whole sample. The last step is the model construction with either supervised classification or survival regression, dependent on the number of groups. *Citrus* uses statistical features as input variables and group assignments as observations. The model determines the clusters which best predict the observation of the data set or an individual's survival risk. The classification uses $L1$-regularized logistic regression model (Sec. 2.4), and for the survival regression, many $L1$-regularized Cox proportional-hazards models [25] are constructed. The models are evaluated using k-fold cross validation, and the regularization thresholds minimizing the cross-validation error rate λ_{min} and within 1 standard error of the minimum model λ_{1se} is used to determine the subset of clusters for further prediction (Sec. 2.4.4). To use this tool, ten parameters requires configuration. Default options are $Compensation$ = File-internal; $Cluster\ characterization$ = abundance; $Event\ sampling = 5,000$; $Event\ sampling\ method$ = equal; $Minimum\ cluster\ size$ = 5%; $CV\ folds = 5$; $FDR = 1\%$; $Normalize\ scales$ = false; $Transform\ cofactor = 5$; $Association\ models$ = glmnet.

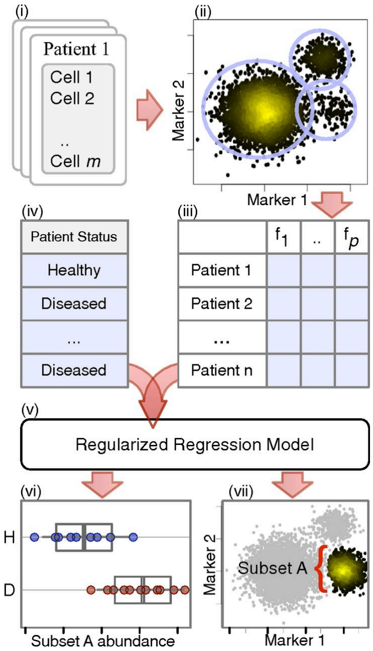


Figure 2.4: Schematic workflow of *Citrus*. "Cells from all samples (i) are combined and clustered by using hierarchical clustering (ii). Descriptive features of identified cell subsets are calculated on a per-sample basis (iii) and used in conjunction with additional experimental metadata (iv) to train a regularized regression model predictive of the experimental endpoint (v). Predictive subset features are plotted as a function of experimental endpoint (vi), along with scatter or density plots of the corresponding informative subset (vii). In this example, the abundance of cells in subset A was found to differ between healthy and diseased samples (vi; H, subset A abundance in healthy patients; D, subset A abundance in diseased patients). Scatter plots show that cells in subset A have high expression of marker 1 and low expression of marker 2 relative to all measured cells (shown in gray)" [14].

For clusters A,B and centroids $\bar{a}, \bar{b}$:

$$D_{Ward}(A,B) = \frac{d(\bar{a},\bar{b})^2}{1/|A| + 1/|B|} \tag{2.7}$$

with *Euclidean* metrics:

$$d(\bar{a},\bar{b})^2 = ||\bar{a} - \bar{b}||^2 \quad . \tag{2.8}$$

Scaffold Maps

The single-cell analysis by the fixed force- and landmark-directed maps (*Scaffold Maps*) algorithm consists of three main steps (Fig. 2.5) [26]. The first step is the creation of a cluster map which comprises cell clusters from manually identification (red nodes) and from conventional cell clustering by the clustering for large applications (CLARA) algorithm (blue nodes, App. A, Alg. A). CLARA makes use of the algorithm partitioning around mediods. Thus, it works similar to k-means clustering, but uses the medoids instead the means. A mediod in this context is the cell with the smallest dissimilarity to all others in the cluster. The second step generates a force-directed graph from both blue and red nodes in which similar nodes are located close together according to the similarity, and is used as the reference map with land mark populations. Each node is associated with a vector containing the median marker values of the cells in the cluster. The edge weights are defined as the *cosine* similarity between these vectors. Additional samples are also clustered by CLARA, and the resulting clusters are manually overlaid onto the red landmarks.

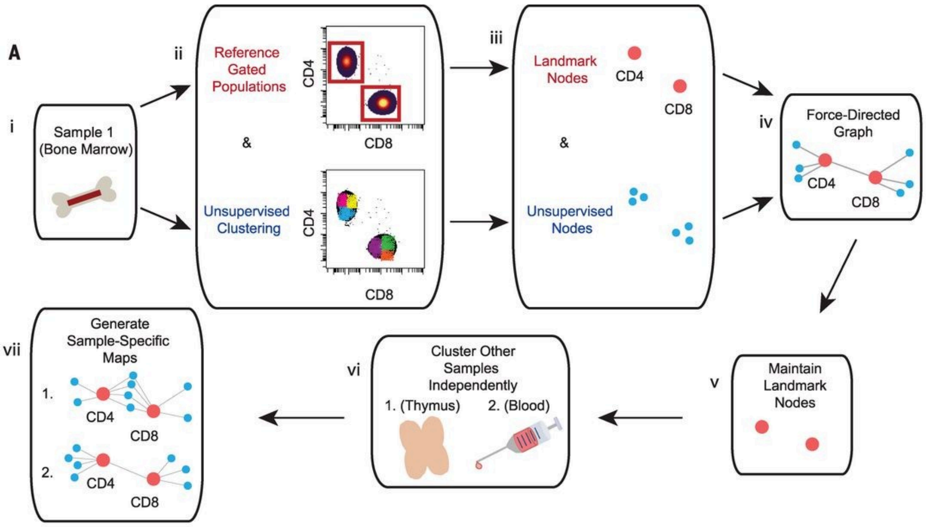


Figure 2.5: Schematic workflow of *Scaffold maps*. “(i) Bone marrow sample is the reference sample. (ii) Leukocytes are grouped according to prior knowledge to define landmark cell populations as reference points on the map. The same leukocytes are subjected to conventional clustering to provide an objective view of the tissue composition and organization. An illustration is provided with the two major lineages of mature T cells, which express either CD4 or CD8. (iii, iv) Both landmark populations (red nodes) and unsupervised clusters (blue nodes) are used to generate a force-directed graph in which similar nodes are located close together according to the similarity of their protein expression. Size of unsupervised clusters denotes the relative number of cells in that grouping. (v) Landmark populations from the bone marrow are fixed in place for subsequent maps to provide points of reference for rapid human interpretation. (vi) Additional samples are each subjected to conventional clustering via the same clustering algorithm. (vii) The resulting clusters for each sample are overlaid onto the original landmark nodes to generate tissue-specific *Scaffold maps*” [26].

CellCnn

CellCnn uses also a supervised learning algorithm, in which each observation corresponds to single-cell abundance profiles and each label is the corresponding phenotype. The values of each uni-variate marker is percentile normalized and the intensities are simplified to high and low. Original samples are distributed into training, validation and test set, which are used as input layer in convolutional neural network (CNN). The training set is randomly subsetted with replacement for multi-cell input training samples of the same size as the test set. The number of multi-cell inputs is chosen equally for each label. The distribution to training, validation and test set, the number of applied models and some of its corresponding parameters (filters, learning rate, dropout) and k-folds in cross validation (CV), and the choice of pooling measurement changes according to their best results and aim in the different public data sets. After training a model, the trained filter weights are used for variable selection (VS). The weights which correspond to the molecular profiles of relevant cell subgroups are extracted. These profiles can then be matched with the individual patch vector of the cells. Density-based clustering is then applied to detect filters with more than one cell type. The filtered cell types are eventually compared to the residual cells and are characterized in more detail with conventional approaches such as density and bar plots.

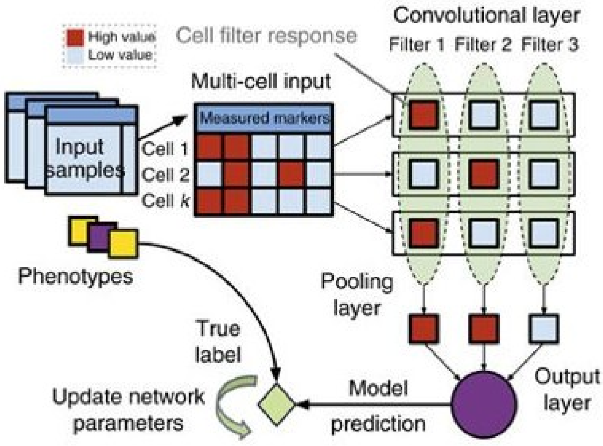


Figure 2.6: Schematic workflow of *CellCnn*. "*CellCnn* takes multi-cell inputs, where each input is annotated with a phenotype. Node activities in the convolutional layer are defined as weighted sums over single-cell molecular profiles. Nodes in the pooling layer evaluate the presence (max pooling) or frequency (mean pooling) of specific cell subsets. The output of the network estimates the sample-associated phenotype. Network training optimizes weights to match training data set phenotype. Trained filter weights correspond to molecular profiles of relevant cell subsets and allow for assignment of the cell subset membership of individual cells (cell-filter response)" [24].

2.2.3 Challenges

Common drawbacks of non-linear reduction algorithms as in *viSNE* is the difficult interpretation, the high complexity and the crowding problem. *Citrus* and *Scaffold Maps* are using clustering approaches on single cell information which results into intensive computation. It is therefore infeasible to apply these approaches to large data sets, which contain more than 50,000 cells, and prior downsampling procedures bears the risk of loosing rare subpopulations. *CellCnn* on the other hand does not work properly with small data sets, and needs more computational power compared to the other tools. There are also many other alternatives to this study's workflow, and several new algorithms are emerging or current ones are optimized. Many have also been reviewed [2, 3, 4]. However, an easily interpretable and conclusive visualization connected to a comprehensible, reproducible differential analysis is still lacking. Therefore, VS, in particular VR, and the machine learning algorithm and evaluation techniques deployed in the proposed workflow are introduced in the following sections.

2.3 Variables and their modifications

A variable in this dissertation describes an input variable or explanatory, predictive variable for a machine learning approach, and is distinct from a raw variable. A raw variable in this context is a raw measurement e.g. a SI from a protein marker in the published mass cytometric data set. In the following the two main categories are introduced which modify the variable space.

2.3.1 Variable engineering as a key part in data mining

Variable engineering (VE), also called feature engineering or variable construction, is an upstream domain and a key step in data mining. It supports the usage of domain knowledge of the data, while generating a new, relevant mathematical representation from the raw data. These engineered variables are used as input variables in machine learning approaches with the aim of helping these algorithms to build a robust model, and to improve the model performance on unseen data. Compared to the raw values, the modified variables can have the same, an enlarged or a reduced space dimensionality, or a combination of either direction [27]. The general question in VE is, if manipulating variables, such as removing or combining raw variables, is reasonable and whether it is more useful than the raw version.

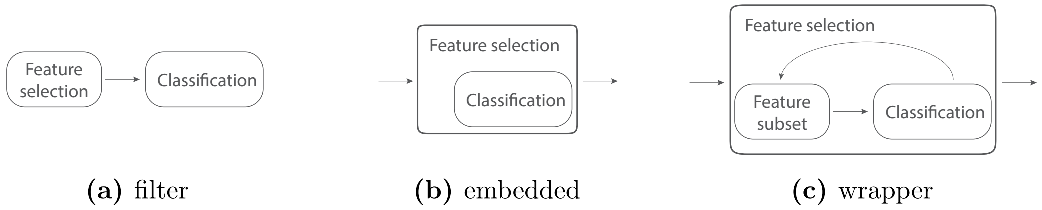
In this study, VE is applied on the *arcsinh*-transformed SIs of the measured protein markers from the example data set from *Spitzer et al.* The novel engineered variables are subject to visualization, manual inspection and to variable ranking, which latter is further described in the following section.

2.3.2 Variable selection methods

Only "a few percentage points of variation among cells can produce outcome differences of more than two orders of magnitude..." and "...can make all the difference between health and substantial autoimmune pathology" [28]. Variable selection (VS), also known as feature (subset) selection, could help to find these cells. It is a type of dimension reduction and is also a key step in statistics and the readability and interpretability of the machine learning outcomes. It allows for filtering out irrelevant variables which have no to little predictive value. Thus, it selects a subset from the predefined set of variables and does not create new ones (as in VE). There are three main categories of VS, which are shortly described in the following sections: filter, wrapper, and embedded methods (Fig. 2.7) [29]. Selecting the appropriate VS method can achieve the following [30]:

- interpretability of predictive models (simplification),
- reduction of data size,
- decrease of model training times,
- elimination of noisy variables,

- improvement of generalization (reducing over-fitting), and
- avoidance of the curse of dimensionality.



(a) filter **(b)** embedded **(c)** wrapper

Figure 2.7: Scheme of the three variable selection methods.

Filter methods

Filter methods evaluate the relevance of the variables prior to the model algorithm as a pre-processing step. They are also independent from the model algorithm and do not take into account any biases of the downstream algorithm, hence, making these methods comparably fast. Popular examples for filter methods are correlation based and variable weight based VS [31, 32]. In classification problems, each variable is individually evaluated to check if there is a plausible relationship between it and the observed classes. A common downside of filter methods is that they include redundant variables, and they do not capture inter-correlations between variables very well.

Wrapper methods

Wrapper methods evaluate variables indirectly by evaluating multiple models with different variable subsets. They add or remove variables to find the optimal combination that maximizes the model performance [27]. In classification problems, each variable is individually evaluated to check if there is a plausible relationship between the variable and the observed classes. Variables with important relationships are included in a classification model. Wrapper methods eliminate dependent and irrelevant variables very well. However, these methods are very intensive in computation, and even a small number of variables can lead to a combinatorial explosion. In addition, there is an increased risk of over-fitting with small sample sizes. Classical representations are deterministic greedy methods (forward selection and backward elimination), stochastic genetic algorithms, and support vector machines.

Embedded method

Embedded methods integrate VS or variable weighting as a part of model construction. The error function is optimized, and penalization of too many variables in the model is simultaneously applied. This leads to a quicker design than wrapper methods and less danger of over-fitting. Embedded methods for variable ranking (VR) assess individual variables by assigning weights to them according to their degrees of relevance in the process of training [29]. Some embedded strategies include decision tree learning, neural networks, and regularized models, the latter of which is deployed in this study and introduced in the following section.

2.4 Elastic-net regularized logistic models

In this section, the elements of the embedded VR is introduced, which is deployed in this dissertation. Fundamental elements of a supervised linear model (LM), its extension to generalized linear model (GLM), the elastic-net regularization to extend and optimize the model, cross validation (CV) and error measurements to evaluate the model performance are shown with the applied notations.

2.4.1 Linear regression models

A model is a simplification of a complex situation for better understanding of a phenomenon of interest. There are highly innovative tools in statistics available, but the main tool of the applied statistician remains the linear model (LM), which is a supervised learning approach. It is called supervised, because of the presence of the response variable which guides the learning process. It has the simplest and seemingly most restrictive statistical properties: independence, normality, constancy of variance, and linearity [33, 34, 35]. Linear models can be used for predictions, data description, parameter estimation, and variable selection, and can be applied to transformations of the original input variables.

In a linear regression setting, there are N samples where $\mathbf{Y} = \{y_1, \ldots, y_j, \ldots, y_N\}$ is the response variable vector and each explanatory variable $\mathbf{X'} = \{x_{j1}, \ldots, x_{ji}, \ldots, x_{jp}\}$ is a p-dimensional associated vector of variables. Dependant response variable Y can be predicted by a linear combination of explanatory variables. The term “linear model” usually encompasses both systematic and random components in a statistical model. It is assumed that the response variable has a normal distribution [33, 34]. The LM has the form

$$y_j = \eta_j + \epsilon_j = \beta_{j0} + \sum_{i=1}^{p} \beta_{ji} \cdot x_{ji} + \epsilon_j \quad , \tag{2.9}$$

where model error ϵ_j is the Gaussian noise to the predicted values, which cannot be explained by the combination of x_{ji} and β_{ji}. Thus, ϵ_j is independent of $x_{j1}, \ldots, x_{jp}$, the expected value from ϵ_j is $E[\epsilon_j] = 0$, and the variance is $Var[\epsilon_j] = \sigma^2$. β_{ji} are unknown coefficients (regression weights $\beta'_j = (\beta_{j1}, \ldots, \beta_{jp})$ and an intercept (or 'bias') term β_{j0} $\in \mathbb{R}$), which are needed to be estimated.

Generalized linear regression models

Generalized linear models (GLMs) are an extension of LMs, which allows the response variable to be non-gaussian [36]. With these models, it is possible to deal with problems by assuming that $\mathbf{Y}$ has an arbitrarily different distribution, e.g. if your response variable is binary. To model the distribution of $\mathbf{Y}$ conditional on a number of random variables $\mathbf{X}'$ to the type of conditional distribution $P(\mathbf{Y}|\mathbf{X}')$, there are four key assumptions:

1. The influences of the variables $\mathbf{X}'$ on $\mathbf{Y}$ can be summarized into an intermediate form, the *linear predictor* η_j;
2. η_j is a linear combination of $\mathbf{X}'$;
3. There is a smooth, invertible function l mapping η_j to the expected value $E[\mathbf{Y}] = \mu$;
4. The distribution $P(\mathbf{Y} = y_j; \mu)$ of $\mathbf{Y}$ around μ is a member of a certain class of noise function and is not otherwise sensitive to the variables $\mathbf{X}'$.

Assumptions 1 to 3 can be expressed by the following two equations. Assumption 4 implies conditional independence of $\mathbf{Y}$ from the variables $\mathbf{X}'$ given η_j.

$$\begin{aligned} \eta_j &= \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \epsilon_j & &\text{(linear predictor)} \quad (2.10) \\ \eta_j &= l(\mu) & &\text{(link function).} \quad (2.11) \end{aligned}$$

Summarized, a GLM has three parts: a structural component, a link function, and a response distribution. A link function is a function of the mean of the response variable $\mathbf{Y}$, which is used as the response instead of $\mathbf{Y}$ itself, which describes the relationship between the random and systematic components.

The logit link function

The *logit* transformation is a link function from the sigmoid function classes, which converts a real number from $(-\infty, +\infty)$ to a probability number $[0, 1]$ (Fig. 2.8). The function is the canonical link function for the *Bernoulli* distribution and is the natural log of the odds, that $\mathbf{Y}$ equals one of the categories. It is inserted into Equation 2.11 [33, Chap. 4.3].

$$\begin{aligned} logit(P) = ln\left(\frac{\mu}{1-\mu}\right) &= \beta_{j0} + \sum_{i=1}^{p} \beta_{ji} x_{ji} & &\text{, for } 0 < P < 1 \quad (2.12) \\ \frac{\mu}{1-\mu} &= exp\Big(\beta_{j0} + \sum_{i=1}^{p} \beta_{ji} x_{ji}\Big) & &\text{, with } x_{j0} = 1 \quad (2.13) \\ &= exp\big(\sum_{i=0}^{p} \beta_{ji} x_{ji}\big) & & \quad (2.14) \\ &= \prod_{i=0}^{p} exp(\beta_{ji} x_{ji}) \quad . & & \quad (2.15) \end{aligned}$$

With the *logit* link function described in Equation 2.12-2.15, coefficients and variables are multiplicative rather than additive as in a LM. Thus, the coefficients need to be interpreted exponentially. If $\beta_{ji} = 0.55$, then $exp(\beta_{ji}) = 1.73$ and the variable x_{ji} affects the odds ratio of the response to 1.73 of being true and has thus more influence than in a LM with the same value [37]. Combining GLM with a logit link function results in a generalized logistic regression model, short logistic regression (LR) model.

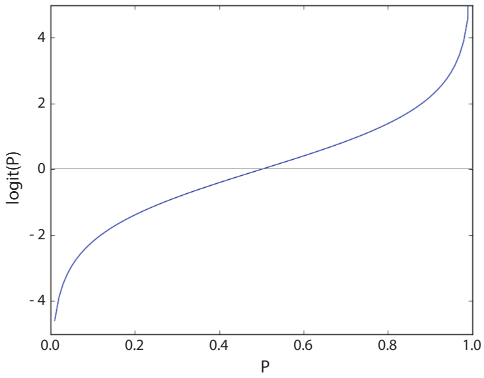


Figure 2.8: Logit link function.

Minimizing the loss function

The goal is to find the best vector x_{ji}. For this pupose, the loss function $(\mathbf{Y} - \eta_j(x))$ needs to be minimized. With the coefficients β_{ji}, it is possible to adjust the influence (relevance) of the variables x_{ji} in the regression. As the true β_{ji} are unknown, they have to be estimated from the data set. The common approach is the ordinary least squares (OLS), where $\hat{\beta_{ji}}$ are estimated by minimizing the squared-error loss function (Eq. 2.16). This leads to nonzero coefficients, which means that all variables are involved in the model.

$$\hat{\beta_{ji}} = \text{argmin}_{\beta}\{\sum_{j=1}^{N} \left(y_j - \beta_{j0} - \sum_{i=1}^{p} \beta_{ji}x_{ji}\right)^2\} \quad . \tag{2.16}$$

The classical OLS cannot distinguish variables with little or no influence. Thus, this model is most likely over-fitted in $p > N$ data sets, which results in a poor predictive power on unseen data. To solve that problem, regularizations on the estimation process with the principle of sparsity are applied. The principle and several regularizations are introduced in the following section.

2.4.2 Principle of sparsity

The principle of sparsity helps to reduce the number of variables p in classifications, e.g. in LR models. This is mainly applied on disproportional data sets. If the number of the variables p is much higher than the number of samples N, the models tend to over-fit, so that the model performs badly on unseen data. There are several different kinds of sparsity. The classical sparsity states that only a small number p' are relevant among the p explanatory variables. Another statement is, that although all of the p explanatory variables are important, one can find a small number of linear combinations of those variables that explain most of the variation in the response [38].

2.4.3 Regularizations

A regularization, or a constraint, in LR applies a penalty term to the loss function, so that the variance is reduced at the cost of introducing some bias [34]. There are many regularizations available, which improve the prediction accuracy by shrinking the values of the regression coefficients, or setting some coefficients to zero. The simplest penalty term takes the form of a sum of squares of all coefficients $||\beta_{ji}||^2$. This so-called $L2$ norm, a further popular regularization $L1$ norm, and a mixture of both $L1$ and $L2$ (elastic-net) are presented in the following sections.

$L2$-Regularization

If multi-collinearity is present in the data, there is no unique solution to estimate the coefficients. There are rather infinite possibilities, which would result to the same minimal loss. One solution to handle this problem is the deployment of the $L2$, or ridge, regularization. It shrinks the coefficients in a regression towards zero by imposing a penalty term on the sum of the squares of the coefficients (Eq. 2.17). This regularization does not shrink the variable count p, and operates in situations in which $p < N$.

$$\hat{\beta} = \text{argmin}_{\beta}\{\sum_{j=1}^{N}\big(y_j - \beta_{j0} - \sum_{i=1}^{p}\beta_{ji}x_{ji}\big)^2 + \underline{\lambda\sum_{i=1}^{p}{\beta_{ji}}^2\}} \quad , \tag{2.17}$$

where regularization parameter $\lambda \geq 0$ is a penalty parameter that controls the amount of shrinkage, which can be estimated by an external procedure such as cross-validation (Sec. 2.4.4). The bigger the λ, the more variables are filtered out.

$L1$-Regularization

The $L1$, or LASSO, regularization seeks to minimize the squared error loss (Eq. 2.16) under a norm analogous to the $L2$. Specifically, the loss function is subject to the sum of the absolute values of the coefficients β_{ji} (Eq. 2.18) [39]. This solution tries to solve the $p >> N$ problem by setting many β estimates to zero. If there are many correlating variables, this approach tends to highly increase the coefficient of one variable and set the coefficients of the other correlating variables to zero. Thus variable shrinkage is conducted, resulting in a very sparse set of variables $p' \leq N$.

$$\hat{\beta} = \text{argmin}_{\beta}\{\sum_{j=1}^{N}\big(y_j - \beta_{j0} - \sum_{i=1}^{p}\beta_{ji}x_{ji}\big)^2 + \underline{\lambda\sum_{i=1}^{p}|\beta_{ji}|\}} \quad . \tag{2.18}$$

Elastic-net regularization

Elastic-net regularization overcomes some of the limitations of the $L1$ by borrowing strength from the $L2$ [34]. Specifically, it

- allows to select $p > N$ amount of variables
- tends to jointly select or leave out groups of highly correlated variables, and
- improves the predictive performance with respect to $L1$.

The penalty function has both norms incorporated, resulting in a loss function of

$$\hat{\beta} = \text{argmin}_{\beta} \frac{1}{2N} \sum_{j=1}^{N} \Bigg\{ \Bigg(y_j - \beta_{j0} - \sum_{i=1}^{p} \beta_{ji} x_{ji} \Bigg)^2 + \lambda \sum_{i=1}^{p} \Big(\underline{\alpha|\beta_{ji}|} + \underline{(1-\alpha){\beta_{ji}}^2} \Big) \Bigg\} \Bigg] \quad .$$

The second term (similar to $L2$ norm) averages highly correlated variables, while the first term ($L1$ norm) encourages a sparse solution in the coefficients of these averaged variables. The elastic-net can yield more than N non-zero coefficients. This is especially advantageous for a data set with $p >> N$. Regularization parameter α constraints the scale of both terms.

The hyper parameters α and λ affect the sensitivity of the algorithms to detecting patterns, the bias-variance trade-off and the trade-off between model complexity and fitting of the data. To estimate these parameters, the model performance needs to be evaluated. Cross-validation is especially useful for limited data, and is therefore presented in the following section.

2.4.4 Cross-validation

After building a model, an evaluation of that model is important to assure a high accuracy of the prediction on unseen data. Thus, the data set needs to be divided in training and validation set. However, in order to build an accurate predictive model, many data is needed for training. With small data sets, the validation set is very small to profoundly estimate the performance of the model. One solution to solve this problem is to use the k-fold CV protocol, which is arguably the most common out-of-sample performance estimation protocol for relatively small sample sizes. In the following, CV and the deviance as error measurement in LRs is introduced.

The algorithm divides the samples into k folds and uses one fold for validation and the $k-1$ folds for training the model [33]. Specifically, k-fold CV iterates as follows:

- for $i = 1, \ldots, k$, hold out portion i and fit the model from the rest of the data;
- for $i = 1, \ldots, k$, use the fitted model to predict the hold-out samples;
- average the performance measurement over the k different fits.

This is repeated k times with another fold as validation set until all folds have been processed. Hence, the data set does not need to be divided into two sets initially. In this way, the combination of the regularization parameters α and λ can be determined to build the best model.

2.4.5 Error measurements

Deviance

The OLS method minimizes the sum of squared-error loss function (Eq. 2.16). Using the *logit* function in LRs, the coefficients and variables are factors rather than summands due to the log odds. Therefore, the error measurement deviance *dev* is used, which is a specific transformation of a likelihood ratio [40]. It is defined as minus twice the log-likelihood (Alg. 2.1). The model is estimated on the $k-1$ folds of the CV and applied to the remaining k^{th} fold. For the application on the remaining fold the log-likelihood-score is calculated. This is repeated k times and the mean of the k results for each λ of the above defined deviance measure is returned.

Algorithm 2.1: Mean deviance measurement

```
Input: Observations $\mathbf{Y}$; chosen number of variables by $\lambda$: $\hat{p}$; left_out samples
Output: mean(dev)
it = 0
foreach $j$ in samples do
    next if $j$ ==left_out
    it +1= it
    dev[it] = $-2 \cdot \left(y_j \cdot log(\hat{p}_j) + (1-y_j) \cdot log(1-\hat{p}_j)\right)$
end
```

Root mean square error

The root mean square error (RMSE) is the standard deviation of the prediction errors. It is the square root of the summed differences between predicted values and observed values. Thus, it is always non-negative, and a value of 0 would indicate a perfect fit to the data. The formula is

$$RMSE = \sqrt{\frac{\sum_j (\hat{y_j} - y_j)^2}{N}} \quad . \tag{2.20}$$

To demonstrate the proposed VR approach, a publicly mass cytometric data set was used, which is summarized in the following section.

2.5 Example data set

For an appropriate comparison of different cytometric analysis strategies, it is important that the selected data set has:

- the typical sample sizes in biological and clinical studies,
- a minimum amount of samples in each condition ($n = 3$),
- sufficient cell events,
- a good quality of the data,
- clear statements concerning a disease or other conditions, and
- an outlined subpopulation.

The data set from *Spitzer et al.* [13] fulfills all these requirements and are presented after a short introduction to the immune system, which serves for a better understanding of the nature of the data set and the re-analyses in Chapter 3.

2.5.1 Short introduction to the immune system

The immune system is a complex system which includes various organs, cell types and molecules. Its main functions are the protection against pathogens and the elimination of degenerated cells. In many species, two major subsystems have developed to mount an effective immune response: innate and adaptive immune system [41]. Macrophages, mast cells, dendritic cells, granulocytes and natural killer cells belong to the innate immune defense. Their response is fast, but non-specific for antigens. Instead, they recognize evolutionary conserved pathogen associated molecular patterns. Thus, the innate immune system is inflexible to detect novel pathogens. This is balanced with the antigen specificity of the adaptive immune system. B and T cells are part of that system and feature a high adaptive capacity towards unseen or changed germs. An important link for the communication between both systems are the T cells, which can be further categorized in $CD8^+$ and $CD4^+$ T cells. The former are considered as cytotoxic T cells and the latter have several functions and secrete certain small proteins, the so-called cytokines, which are regulating the immune response. To perform these functions the naïve $CD4^+$ T cells need to be activated and differentiated into regulatory (Treg) or T helper (Th) cells, among the latter are follicular T helper cells (Tfh), Th1, Th2, and Th17 cells [42, 43]. The differentiation into these T cell subpopulations are controlled by their master transcription factors (Fig. 2.9).

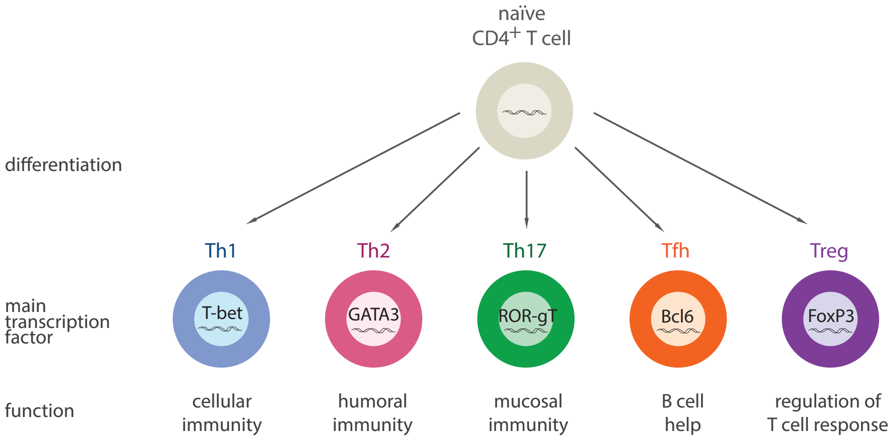


Figure 2.9: Scheme of $CD4^+$ T cell differentiation.

2.5.2 Structure of the data set from *Spitzer et al.*

Spitzer et al. used female mice from mouse model MMTV-PyMT (murine mammary tumor virus-polyoma middle T), which is a widely used spontaneous model of carcinoma [13]. These mice develop a very aggressive triple-negative breast cancer ($ER^-HER^-PgR^-$), which was treated when the primary tumor reaches a certain size. This day was referred to as day 0. They used four different treatment types in which two fall into ineffective (ineff) treatment: anti-PD-1 (aPD1) and no treatment (untr). The other two fall into effective (eff) treatment: treatment of an antibody-mix from two different mouse breeds (Tab. 2.1).

Table 2.1: Blood sample overview of *Spitzer et al.* [13].

Treatment	Samples	Treatment
untr	3	no treatment
aPD1	3	250 mg anti-PD-1
CD1	3+1	400 mg CD1 allo-IgG, 100 mg anti-CD40 and 100 mg IFNg
B6	3	400 mg B6 allo-IgG, 100 mg anti-CD40 and 100 mg IFNg

Anti-PD-1 serves here as a model for ineff treatment because its lack of efficacy in this tumor model. The antibody mix of the eff treatments consists of allogeneic immunoglobulin G (allo-IgG), anti-CD40 (aCD40) antibody and Interferon g (IFNg), where the two latter activate specific immune cells to battle against tumor cells. IgG identifies tumor cells by chance, and was extracted from an outbred CD-1 (CD1) and an inbred C57BL/6 (B6) mouse. They sacrificed mice from each treatment ($n = 3$ per treatment, with one extra replicate sample) and tissue from tumor, lymph node, spleen, bone marrow and blood were obtained. 41 protein markers were measured and the full parameter list consists of 59 parameters, which includes of protein markers and other device parameters (App. A, Tab. T1 and T2). Additional columns are listed such as *Time*, *Event_ length* and several numbers of bar codes.

2.5.3 Main results from *Spitzer et al.*

Spitzer et al. deployed unsupervised clustering on all cells and visualized similarities using force-directed graph (Sec. 2.2.2). Each tissue was colored uniquely, and cells from animals left untreated or treated with ineff therapy are indicated in black. They then ranked the populations by their connectivity in the network, and they saw, that CD4$^+$ T cells were most prominent in the top ranks. So they hypothesized that the CD4$^+$ T cells are more central to effective immune response than CD8$^+$ T cells, which is intriguing in the light of the dominant focus on CD8$^+$ T cell responses and targets for therapy [13].

They manually inspected these CD4$^+$ T cells with the *Scaffold Maps* and conventional univariate density plots. They found a subpopulation, which is most discriminative between eff and ineff treatment and they proposed this subpopulation as an activated, effector memory Th1 subset. It was specified by the protein characteristics CD44$^+$CD69$^+$CD62L$^-$ CD27$^{\text{low}}$CD90$^+$T-bet$^+$ in eff treatment (– as negative, not produced; low as low expression; and + as produced, either low or high). Furthermore, PD-L1 was identified to be upregulated in the T cells of the eff treatment group. PD-L1[1] is an immunsuppressor, which

[1] Not to confuse with PD-1.

inhibits the intended anti-tumor immune response. They tested another mouse group with a combined eff-anti-PD-L1 treatment, which resulted in an even more effective treatment.

In the end, the blood data set from *Spitzer et al.* [13] fulfills all aspects and is therefore chosen for this dissertation as an example data to proof the concept of this study's *PRI*. Certainly, the goal of a re-analysis is to first validate the main findings, second, to present the data information in the best possible way, and third, to extract more relevant information. Thus, the procedure and the results using the *PRI* approach are demonstrated in the following chapter.

3 Results

The acquisition of one cytometric sample generates a signal intensity (SI) matrix with the dimension of $c \times m$, according to c number of cells and m number of protein markers (Fig. 1.1). One marker SI is proportional to the abundance of the bound protein on or in the cell. The markers are considered dependent to each other to some extent, and the expression is individual for each cell. In the last decade, cytometric measurements has witnessed significant technical improvements, so that cytometers can detect up to 50 protein markers in parallel. Dividing the SIs of each marker in positive and negative (producing and non-producing protein) expression, the number of marker combinations comprises 2^{50} possibilities. Due to the high dimensionality of these combinations, the traditional analysis approach, to solely visually inspect a series of bi-axial plots, became increasingly laborious [4, 10, 44]. Although a variety of novel computational analysis approaches are available, none have fully fulfilled the requirements to create reproducible and interpretable results, and simultaneously support discriminant studies [1, 2].

This study proposes a cytometric data analysis workflow named *PRI* which combines three main steps: i) data preparation and storage, ii) bin-based variable engineering (VE) of several protein marker combinations, whose resulting variables serve for visualization in manual inspection, as well as input variables for the subsequent embedded variable ranking (VR) approach, and thus iii) the deployment of the supervised learning algorithm elastic-net regularized logistic regression (erLR) in a nested cross validation (CV) manner in order to rank these variables by their informative values for the purpose of substantially improving the identification of meaningful subpopulations associated to a specific phenotype. A proof of concept is presented herein by re-analyzing a published and publicly available mass cytometric data set from *Spitzer et al.*, as previously described in Section 2.5 [13]. They studied whether systemic immune activation can be detected in early phases of cancer therapies, and if there is a distinct subpopulation to classify an effective (eff) and ineffective (ineff) treatment. For this purpose, female mice with mammary tumors were treated in four different ways, in which two fall into ineff treatment, and the other two into eff treatment. They identified a promising subpopulation in blood samples which discriminates between both treatments. Herein, this subpopulation is preprocessed and then evaluated firstly by manual inspection with this study's novel bin-based visualization methods which are named 'diploT' and 'triploT'. Secondly, the proposed VRs workflow is applied on the triploT information to extract discriminating three-marker combinations. Additionally, results from the examination with the ‘gold standard’ visualization tool *viSNE*, and the classification tool *Citrus* is compared to the outcome of this work’s approach [12, 14].

3.1 Data preparation and storage

The example data set from *Spitzer et al.* is downloaded from the completely web-based cytometric data storage and analysis platform *Cytobank* [45]. The 13 blood samples at day 3 are collected ($n = 3$ per treatment, with one extra replicate sample), where 41 protein markers were measured (App. A, Tab. T1).

For mass cytometric measurements, the number of cells ranges commonly between 10^3 to 10^6 cells, dependent on the analysis strategy and project design. However, the measurement values can vary between different cytometers [4]. There can be small variations e.g. in detector gain or sensitivity. Storage, preparation and staining can also affect the results [22]. Additionally, cytometer settings can change with time and reagents used. Therefore, careful data preparation is important, particularly since uniformity is crucial for clear and accurate differentiation between groups. In the following, the preparation steps applied to the example data is displayed and constituted in chronological order. Step 1 explains the gating process. The steps ‘outlier removal’ (step 3), ‘data transformation’ (step 4) and ‘normalization’ (step 5) are essential for sample and group comparisons. The step ‘data curation and storage’ (step 2) facilitates data handling immensely, and the last step ‘quality control’ (step 6) shows initial quality issues by descriptive analyses.

Step 1: Gate on CD4$^+$ T cells

To have a meaningful comparison of the re-analysis in this work to the final outcome of *Spitzer et al.*, all 13 blood samples are gated firstly on living cells (CD45$^+$ and Cisplatin$^-$ [46]) and subsequently on the CD3$^+$CD4$^+$ cell subgroup. CD3 is a T cell co-receptor and is consequently a marker for T cells. CD4 is a marker for helper T cells. This subgroup, referred to as CD4$^+$ T cells, has helping and regulatory activities such as activation, proliferation, and differentiation to immune cells [47, 48]. The full gating strategy is displayed in Appendix A, Figure S1, and is similar to the gating strategy of *Spitzer et al.* . For this purpose, relevant bi-axial plots are inspected.

Step 2: Data curation and storage

In some data sets, curating names and metadata may be necessary, since manual entry and export errors can occur. To ensure proper downstream analysis, the first step is therefore to check for uniform parameter count and names. The original file names are shortened, systematically named as *[treatment]_[day#]_[tissue][ID]* and are subsequently used as the sample ID. Treatments are abbreviated as *untr* for untreated, *aPD1* for (classical) anti-PD1 treatment, *B6* for treatment with B6-antibody mix, and *CD1* for treatment with CD1-antibody mix.

Furthermore, the order of parameters is important for the VE step and needs to be consistent in all files. Consequently, all file parameters are harmonized using the structure and order of sample *CD1_ d3_ Bl1* as reference template. For the curation, marker names are transformed into low capital letters. Numbers are kept, but any other symbols and punctuation characters are removed. A short code is listed in Appendix D, Listing 2.

The web-based application *PRI-base* was implemented within the research group[2] to assure data storage in a standardized manner [49]. The gated and curated samples are interactively uploaded with *PRI-base*, which enables the replication of necessary information. It includes the request of meta information such as species, tissue type, date of experiment and name of experimenter [49, 50, 51]. Instrument details and annotation of experiment conditions are also stored as proposed from the International Society for Analytical Cytology Data Standards Task Force [18]. *PRI-base* has an organized infrastructure to easily access raw SI values of each sample in each project.

If the samples are stored in one file, future (re-)processing steps and accesses for reruns are accelerated. To have a consistent data structure, the samples with all 41 transformed protein marker SIs are concatenated into one single matrix file. To distinguish the cells, one column with their unique sample ID for identification is added.

Step 3: Outlier removal

Outlier cells are common in cytometric data, and it is a common procedure to remove these outliers from analysis [11, 52]. They affect the variables by increasing the marker range on the right end side and subsequently change the bin affiliation for each marker combination. Additionally, they increase the variation of the calculations in the bins. After gating (step 1), the majority of outliers are removed. However, only a few markers were inspected in that step and a systematical outlier removal is missing. Therefore, the residual, untouched markers by the gating process in each file are trimmed consecutively by cells which have the highest 0.05% SI, also called 0.05% quantile (App. D, Lst. 1). A total of 5,521 cells were trimmed from the sum of 347,788 CD4$^+$ T cells across all blood samples.

Step 4: Data transformation

A SI value of flow and mass cytometric data usually has a logarithmic relation to the biological concentration of the protein markers in and on the cell. Besides other transformation functions, the logarithmic *arcsinh* transformation is the simplest and most common way to display the cytometric data in an optimal data spread [14, 16, 53, 54]. With the *arcsinh* transformation, the linear scale at small values is preserved and the logarithmic scale at higher values is transformed to obtain an approximately linear relationship.

[2] Main development by Isabelle Kadner and Yen Hoang, further development by Alexander Rybak.

Other cytometric data studies suggest to use a co-factor for either skewing or widening the marker range. A co-factor of 5 is applied on several mass cytometric data [14, 55]. After exploring the densities of the protein marker SIs and some triploTs, widening (instead of compressing as with co-factor 5) the SI range of all markers in the example data set is deployed. Since grouping with other cells occurs to a lesser extent, cell subpopulations are then better separated and are more likely to be detected. However, additional artificial and thus falsely population separation can occur by widening the range too much. Therefore, the co-factor $cof = 0.1$ is used to facilitate visualization and interpretation. The following equations show the applied transformation and the resulting SIs are those used for subsequent analyses in the workflow.

$$arcsinh(x) = \ln\left(x + \sqrt{x^2+1}\right) \tag{3.1}$$

$$\text{SI} = arcsinh(\text{SI}/cof) \quad . \tag{3.2}$$

Step 5: Normalization

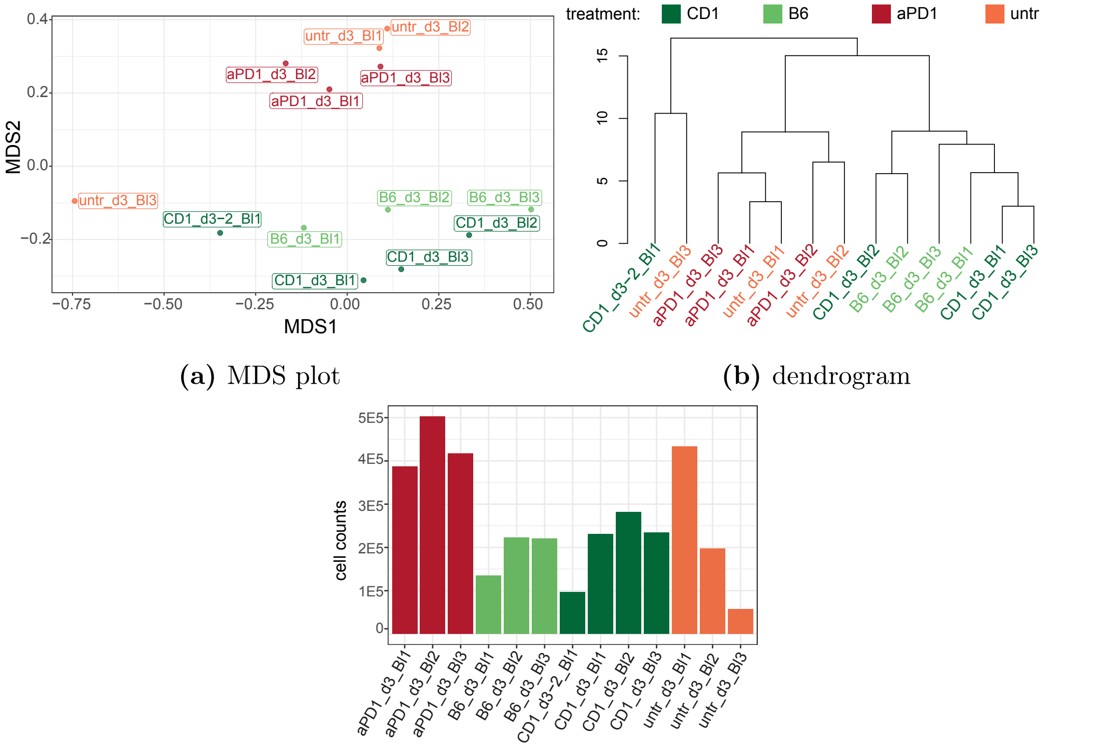
Spitzer et al. have already applied a bead normalization method to the example data set prior to storing. However, first descriptive explorations show that variations between samples still occur in terms of cell count, range and frequencies (Fig. 3.1 and App. B, Fig. S2). Although samples are heterogeneous throughout the groups, further normalization has not been deployed, since these sample variations are supposed to be true biological variations after bead standardization. Furthermore, in order to minimize technical variations, *Spitzer et al.* have simultaneously processed the samples and used the same antibody cocktails for all samples.

Step 6: Quality control

Particularly in high-dimensional data, it is common to use visualization methods to examine if technical errors occurred in individual samples while processing. To check for data quality, conventional metric multi-dimensional scaling (MDS) is herein deployed to visualize the distances between the samples. In general, if a sample from one group clusters to samples from another group, this sample is considered to be of low quality or too much noise, or both. It is then recommended to exclude this sample from further analyses.

Pairwise Euclidean distances are calculated on median expressions of the 41 biological markers in each sample. Figure 3.1a displays clearly that sample *untr_d3_Bl3* has different SI medians compared to other samples of the same ineff treatments (red), and is located closer to sample *CD1_d3-2_Bl1* from eff treatments (green). The other samples within ineff treatment are close together, and the samples within eff treatment are close only on the vertical axis (MDS2). An equivalent behaviour of the sample *untr_d3_Bl3* is seen with the dendrogram resulted from *Ward*'s hierarchical clustering (Fig. 3.1b). Both results suggest to exclude that sample from further examination with *PRI*. In addition, the cell count of that sample is the lowest compared to all other samples (Fig. 3.1c).

Further dendrograms in Appendix B, Figures S3a and S3b are shown with all samples except sample *untr_ d3_ Bl3* and *CD1_ d3-2_ Bl1*, respectively. The exclusion of sample *untr_ d3_ Bl3* results in correct clustering of sample *CD1_ d3-2_ Bl1* to its treatment group, but the other direction does not hold true. Therefore, the exclusion of sample *untr_ d3_ Bl3* seems reasonable and is also realised.



(a) MDS plot

(b) dendrogram

(c) cell count bar plot

Figure 3.1: Descriptive plots of blood samples color-coded by experimental condition: eff (green) and ineff (red) treatments. Calculations are based on the median of the *arcsinh*-transformed SIs of all 41 markers across all cells measured after outlier removal for each sample. Euclidean distances between samples' marker medians are applied in the MDS **(a)**, as well as in the dendrogram using the *Ward*'s hierarchical clustering method **(b)**. Colored bar plots of cell count of each sample is shown in **(c)**. Label names indicate sample IDs.

The prepared data set has 336,545 rows of data spread across 12 samples after outlier removal (and exclusion of sample *untr_ d3_ Bl3*). The example data set is ready for further processing, which is described in the following section.

3.2 Variable engineering with a binning strategy to analyze and visualize cytometric data

Variable engineering (VE) is the process of constructing explanatory variables from a given data set. It is a powerful step to extract significant variables. Additionally, expert knowledge can be brought in to enable successful model deployment. With this work's machine learning approach, a single table is required. Hence, VE in this context means consolidating all relevant information about each sample in one row with each column as an input variable.

Simple features in descriptive statistics such as arithmetic mean, median, minimum or maximum of a signal are commonly used in the field of cytometry [14, 54, 56]. This dissertation proposes de novo engineered variables which can be used as clear interpretable visualizations, for a structured variable exploration and as pattern layers in neural network constructs. This section shows different types of engineered variables, which are demonstrated on the example data set from *Spitzer et al.* The two-parameter combinatorial diploTs are the first effort of VE, and three parametric combinations are added with triploTs. Variables from both, diploTs and triploTs, are used for visualization and manual inspection. Furthermore, they are deployed for classification studies of cytometric data. The transformation of the triploTs for this usage is described in the last subsection.

3.2.1 DiploT - a two-parameter combinatorial binning scaffold

The first engineered variables are uni-axial representations which are called 'diploTs'. The diploT is a compact illustration of a conventional contour plot, common in manual cytometric data studies. For a diploT, the range of protein marker A is subdivided into bins of a specific size. Cells falling into a specific bin are captured if the minimum number of cells is reached in order to maintain stable information. Different bin sizes and minimum number of cells are shown and the applied bin size of $arcsinh(x) = 0.2$ is applied and the minimum number of cells= 20 are justified in Appendix B, Figure S5a and S5b. The cells in the bins can then be characterized by different features such as cell density, relative standard error of the mean (RSEM) and standard deviation (SD), respectively. These descriptive features capture uni-variate information, where the two latter features are mainly used for additional quality control.

For a bi-variate examination, information about another protein marker B is needed. Accordingly, the diploT bins display features such as mean signal intensity (MSI)(B), mean signal intensity of positive cells (MSI+)(B) or frequency of marker B^+ cells. The two latter features need a cutoff threshold of marker B for B^+ cells identification in order to create an additional insight in the analysis.

Transition of a conventional contour plot to a diploT

The transition from a conventional contour plot to a diploT is shown in Figure 3.2, using sample *B6_d3_Bl2* from the example data set. Protein markers CD44 and CD62L are subjects of interest, since they are part of the final cell subpopulation proposed by *Spitzer et al.* (Sec. 2.5). In addition, they are negatively associated, which can be visualized compactly by the diploTs. The top row displays the conventional contour plot of protein marker CD62L over CD44. The plot contains cell distribution corresponding to these markers. The information in the density plot (middle row) of marker CD44 is then reduced into a vertical bar and divided into small equally sized bins. The resulting combinatorial bins are named diploTs and are shown in the bottom row. Thus, the diploT with the similar information compared with the contour plot is shown in Figure 3.2c, line 5. A fixed bin size of $arcsinh(x) = 0.2$ and a minimum number of cells to 20 per bin is set to display a bin in pseudo colors: low values are represented by shades of blue, median values by green and high values by red. For better understanding and systematic denotation: protein marker plotted on x-axis is called 'basis marker' (CD44), protein marker displayed as bin information is called 'associated marker' (CD62L). The histogram above the diploTs shows that the peak in the histogram equals the maximum cell density in the diploT which is shown by red coloring (Fig. 3.2, line 6). The most dense regions in the contour plot are similarly positioned.

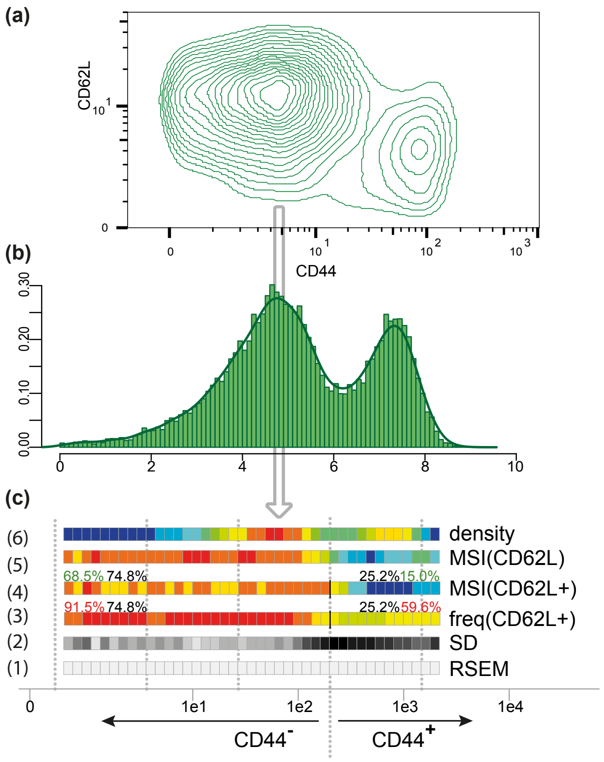
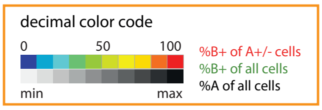


Figure 3.2: Transition of a conventional contour plot to diploTs.
(a) Bi-axial contour plot of cell distribution arranged by CD62L over CD44 created and exported with standard application *FlowJo* showing contour lines with each line containing 5% of cells. Range and *arcsinh* transformation could not be adjusted in concordance with the triploTs due to the nature of the configuration in the tool [57]. **(b)** Density plot of CD44 SIs with histogram of prevalences starting from $arcsinh(x) = 0.2$. **(c)** DiploTs displaying top down: cell density, MSI(CD62L), MSI(CD62L$^+$), frequency(CD62L$^+$), SD (min/max=0.65/1.81) and RSEM (min/max=0%/5%). The black lines indicate the cutoff for CD44$^+$ and percentage values in black are the cell rate to total cells left and right to the cutoff of CD44. Percentage values indicate the rate of producing cells of CD62L compared to total cells (green) and compared to total cells left and right to the cutoff (red), respectively. Bin sizes are set to $arcsinh(x) = 0.2$ and minimum number of cells= 20 per bin to color-code the bins. Dotted lines in the back of $arcsinh(x) = 2$ are plotted for better orientation.



The features SD and RSEM characterize the stability of the values in diploTs for MSI(B). SD shows the SI dispersion around the mean of the cells in the bins. The darkest bins are in the CD44$^+$ region in the top end of the diploT (Fig. 3.2, line 2), with a peak SD value of 1.81 regarding this sample. This indicates a high dispersion around the displayed bin means where the mean range is only 1.79. However, with RSEM in the bins, which quantifies the relative uncertainty in the estimate of the bin mean, the values stay below 10%, with the general estimate, that a RSEM of 25% or greater are subject to high sampling error and should be used with caution. Both values together indicate that the bin mean is acceptable, but with regards to the CD44$^+$ region there might be more than one cell subpopulation.

Additional information on the diploTs is provided by percentage values (Fig. 3.2, line 3 and 4). A cutoff for the basis marker displayed by a black vertical line indicates that the bins on the right hand side include cells which produce the basis marker (A$^+$), and bins on the left are considered negative for the basis marker (A$^-$). The values in different colors provide different indications. The black values indicate the cell rate of cells left and right of the cutoff of the basis marker to total cells. Moreover, red values show the cell rate of the associated marker B^+ in relation to the cells before and after the cutoff, and percentage values in green represent the cell rate of the associated marker B^+ left and right to the cutoff compared to total cell counts. Cutoffs can be set manually or by the approximation function presented in Appendix D, Listing 7.

Informative value with stacked diploTs

Figure 3.3 shows the MSIs of certain protein markers in relation to CD44 expression and their respective diploT illustration. CD44 is plotted as the basis marker and the residual markers are displayed using pseudo-colors, presenting the associated markers. CD44 is a memory T cell marker and Tbet is a marker of differentiated Th1 cells. CD90 and CD86 are T cell activation markers and CD27 is a co-stimulatory receptor. Foxp3 is a marker for regulatory T cells and CD69 is a marker for tissue residency. They are all associated with an active immune system. In contrast, CD62L is considered as a marker for naïve cells and, as a consequence, should be absent with increased SI of CD44. In the stacked diploTs a clear positive correlation between CD44 with Tbet, CD90, CD86, Foxp3 and CD69 (line 1-5) as well as a negative correlation with CD62L (line 7) can be seen at a glance. CD27 in line 6 is present throughout the bins but is also highest at the right end of the diploT. A clear cutoff for $CD44^+$ can be seen at $arcsinh(x) = 6$ (forth dashed line from the left).

These examples show the informative value and compact view of the stacked diploTs. In contrast to conventional approaches, no further gating is needed to extract these intensity distributions. CD44 expression is plotted along the horizontal axis. The complete continuous distribution of CD44 is displayed. Seven stacked diploTs display compactly the correlation of CD44 with seven different markers which play or do not play a role in T cell activation. To use the diploT information as variables, the bin color indices of the MSIs can be extracted to bin vectors in order to capture two-protein marker combinatorial interrelations. These vectors can then be used for further correlation studies between samples or groups. The approach based on the diploT intensities and its curve attributes are successfully applied in [58].

3.2.2 TriploT - a three-parameter combinatorial scaffold binning

In this dissertation, VE involves multiple transformations and filtering steps with the goal of identifying cell subpopulations which differentiate best between two groups. After the diploT development, one more parameter is desired for the combinatorial analysis to expand the dimension to improve the interpretative power. For this purpose, the innovative bin-based triploTs are engineered. The bin scaffold serves as an expanded visualization technique. In addition, several filtering steps and a transformation lead to variables which are used in the supervised learning algorithm in the subsequent section.

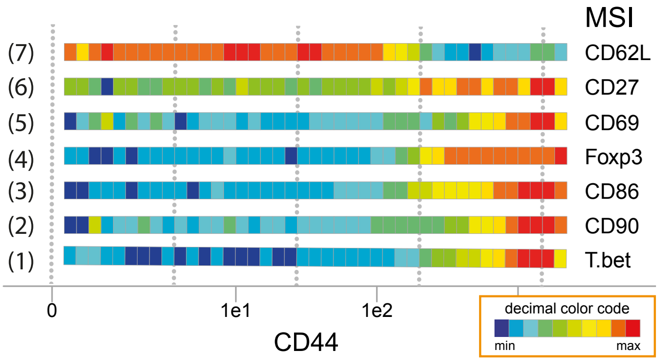


Figure 3.3: Stacked diploTs of CD44 as basis marker from sample *B6_d3_Bl2.* MSIs of Tbet, CD62L, CD86, CD69, CD90 and CD27 are shown as associated markers. Bin sizes are set to $arcsinh(x) = 0.2$ and minimum number of cells= 20 per bin to color-code the bins. Dotted lines in the back of $arcsinh(x) = 2$ are plotted for better orientation.

Transition of conventional contour plot to a triploT

The transition of a conventional contour plot to a triploT is shown in Figure 3.4a and 3.4b, and is explained as follows: First, the bi-axial contour plot area of basis markers A and B is distributed in quadrant bins of equal sizes of $arcsinh(x) = 0.2$, resulting in a grid of bins. Similar to the diploTs in Section 3.2.1, cells falling into a specific bin are captured and displayed if the minimum number of five cells is reached. The bins are displayed in pseudo colors: low values are represented by shades of blue, median values by green and high values by red. Subsequently, the grid of bins can be extended with a statistical information such as cell density, SD and RSEM, and other information about marker C for each bin by the calculation of, for example, MSI, MSI(+) and frequency of marker C^+. There are many statistical methods which can be applied to this bin construct. This bin scaffold is used for further investigation. For systematic denotation, a triploT is termed '[Marker A]-[Marker B]-statistical method([Marker C])'.

Additional information about cell rates is displayed with percentage values. These numbers are presented in diverse colors and indicate different pieces of information. The black values are only calculated if cutoffs for the basis markers are set. Then the bin area is partitioned by the cutoffs to four quadrants. The percentage numbers in each quadrant indicate the rate of cells to total cells in black. If the cutoff for the associated marker is set, the rate of producing cells of the associated marker compared to cells in the quadrant (red), compared to total cells (green) and compared to total producing cells (blue), are calculated and displayed. The cutoffs can be set manually or by a function as proposed in Appendix D, Listing 7.

Batch size calculation for triploT matrices

To create these triploT visualizations a matrix with the associated color values is calculated beforehand. There are three *for*-loops to create all combinatorial triploT matrices for one observation (sample). To reduce the running time, matrices are halved without duplicate axes marker combinations, meaning only one marker combination A over B or

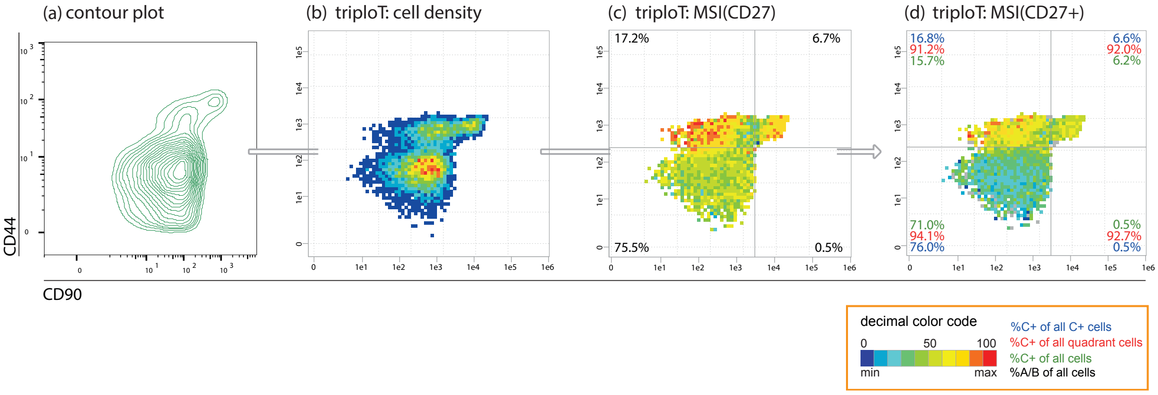


Figure 3.4: Transition from a conventional contour plot to triploTs demonstrated on sample *B6_ d3_ Bl2.* Bi-axial contour plot of cell distribution **(a)** arranged with CD90 on x-axis and CD44 on y-axis created and exported with standard application *FlowJo* showing contour lines with each line containing 5% of the cells (range and *arcsinh* transformation could not be adjusted in concordance with the triploTs due to the nature of the configuration in the tool). The bi-axial plot is divided into even sized square bins of size *arcsinh* = 0.2, x and y axes range: *arcsinh* = [0.2, 12). The colored bins consist of 5 cells at minimum, and bins with 4 cells or lower are not visible. Colors decode the cell density **(b)**, MSI **(c)** and MSI(+) **(d)** of the associated marker CD27, respectively. Grey lines indicate cutoffs for CD90$^{\text{high}}$ (vertical at 8.7) and CD44^{+} (horizontal at 6.0), respectively, and a cutoff for CD27^{+} is set at 3.6. Dotted grid lines of $arcsinh(x) = 2$ are plotted for better orientation. Percentage values indicate different rates in each quadrant as shown in the legend. Numbers in red, green and blue are calculated if cutoff of associated marker (here CD27 (d)) is set. Grey bins in **(d)** indicate bins with less than five CD27^{+} cells.

B over A on the axes is present, depending on the order of the markers listed in the data set (step 2 of Sec. 3.1). Equations 3.3-3.10 summarize the batch size for m given numbers of protein markers.

With a total number of 41 protein markers, the batch size of the example data set from *Spitzer et al.* is $B_{m=41} = \frac{1}{2} \times (41 - 2) \times (41 - 1) \times 41 = 31,980$ triploT matrices for each observation y_j. Listing 3 (App. D) manifests the simple frame of the batch size calculation as pseudo-code which also shows the algorithm complexity ($\mathcal{O}(m^3 \cdot N)$ for N observations). These matrices are used as visualization in the form of triploTs for manual inspection. The following section shows the intuitive and reproducible analysis with the semi-continuous visualization. Moreover, this batch of matrices for each sample can be used in machine learning, or even deep learning approaches. A detailed approach for machine learning is presented later in Section 3.2.3. Investigations with the latter approach have started, but preliminary results cannot be shown yet. However, the hypothetical concept is presented and discussed in Section 4.2.5.

$$o_1 = 1, 2, \ldots, (m-2), (m-1) \tag{3.3}$$

$$o_2 = (o_1 + 1), (o_1 + 2), \ldots, (m-1), m \tag{3.4}$$

$$o_3 = 1, 2, \ldots, m \qquad , \text{ with } o_3 \neq o_1, o_2 \tag{3.5}$$

summing up to

$$B_m = \sum_{o_1}^{m-1} \sum_{o_2=o_1+1}^{m} \sum_{o_3 \setminus (o_1,o_2)}^{m} 1 \tag{3.6}$$

transforming sums of $(o1, o2)$

$$\sum_{o_1}^{m-1} \sum_{o_2=o_1+1}^{m} = \sum_{o_1}^{m-1} \sum_{o_2=o_1}^{m-1} = \sum_{o_1}^{m-1} o_1 \tag{3.7}$$

$$\text{with } \hat{m}\text{th partial sum for } (o_1,o_2)\text{:} \quad \sum_{k=1}^{\hat{m}} k = \frac{\hat{m}\cdot(\hat{m}+1)}{2}$$

$$\text{and with } o_3 = (m-2)$$

$$\sum(o_3)\cdot\sum(o_1,o_2) = (m-2)\cdot \sum_{i=1}^{m-1} i \tag{3.8}$$

$$= (m-2)\cdot \frac{(m-1)\cdot(m-1+1)}{2} \tag{3.9}$$

$$B_m = \underline{\frac{1}{2}\cdot(m-2)\cdot(m-1)\cdot m.} \tag{3.10}$$

Representing the subpopulation from *Spitzer et al.* with triploTs

The example mass cytometric data set from *Spitzer et al.* is used to proof the concept of this study's data analysis workflow on cytometric data. *Spitzer et al.* found a cell subpopulation which discriminates between eff and ineff treatment after day 3 in blood. It is characterized by CD44$^+$CD69$^+$CD62L$^-$CD27$^{\text{low}}$CD90$^+$T-bet$^+$ in CD4$^+$ T cells (Fig. 3.5 and Sec. 2.5) and is firstly manually inspected with the triploTs in this work.

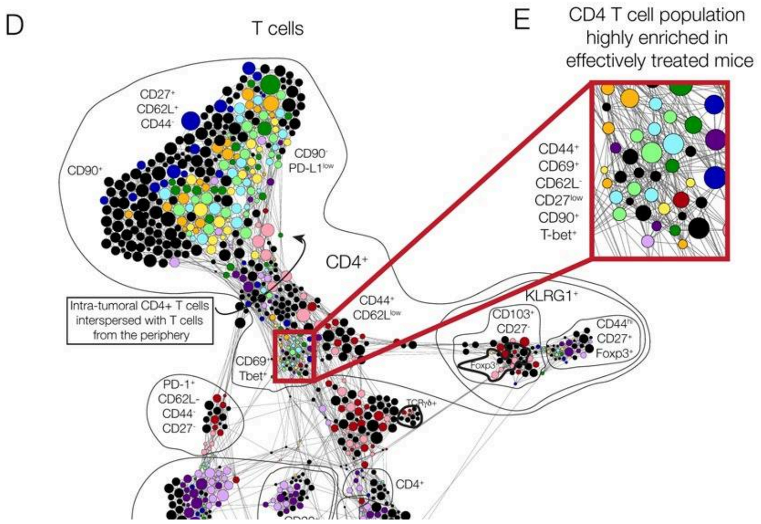


Figure 3.5: Final results from *Spitzer et al.* *Scaffold map* and identified subpopulation **(E)** in CD4$^+$ T cells is discriminative between eff and ineff treatment. The subpopulation is characterized by CD44$^+$CD69$^+$CD62L$^-$CD27$^{\text{low}}$CD90$^+$T-bet$^+$.

The comparing triploTs are shown in Figure 3.6 and the findings are stated as follows: The surface markers CD90, CD44 and CD27 are, combined, an optimal triploT (A-B-C) to show the difference between eff and ineff treatment. With biological expert knowledge, cutoffs for CD90$^{\text{high}}$ and CD44$^+$ are identified and indicated as grey lines. Due to the axes arrangement with CD90 on the x-axis and CD44 on the y-axis, neighboring regions are highlighted since they differ between both treatments. These bin regions are characterized by bin region I (CD90$^+$,CD44$^-$) and bin region III (CD90$^-$,CD44$^+$). Bin region I is characterized as CD44$^-$, CD62L$^{\text{high}}$ and CD27$^{\text{high}}$ which resembles naïve T cells (Fig. 2.9). In sample *B6_ d3_ Bl2* from eff treatment, that bin region indicates the absence of cells since there are no colored bins, but for sample *untr_ d3_ Bl1* from ineff treatment, these bins are present. This leads to the conclusion that these cells are activated in eff treatment. Furthermore, Foxp3$^{\text{high}}$ cells are only located in bin region II, thus the cells in this region represent memory Treg cells.

The triploTs enable the visual determination and differentiation of subpopulations within and between groups of samples. The scaled color-codes are helpful in terms of comparing the SIs. It is effortless to visually capture more than +/− populations, but in particular different levels of MSIs (–, low, med, high), since a semi-continuous display is achieved with the summarized and grouped cells in the bins. In addition, every subpopulation is characterized further with other markers due to the pseudo-multi-parametric view. With the additional information of the percentage values, an increase of the cell count to the factor of three in eff treatment is easily captured. This applies to the rate of general cell count (black) and the rate of producing cells of the associated marker (green) of bin region II and III compared to total cells, respectively (Fig. 3.6). These are meaningful pieces of information which are not easily tangible with conventional approaches, such as in bar or contour plots. With this study's approach it was also possible to further narrow down the treatment-induced T cell subpopulation which was introduced by *Spitzer et al.* The specified subpopulations could be characterized in more detail using its specific pattern.

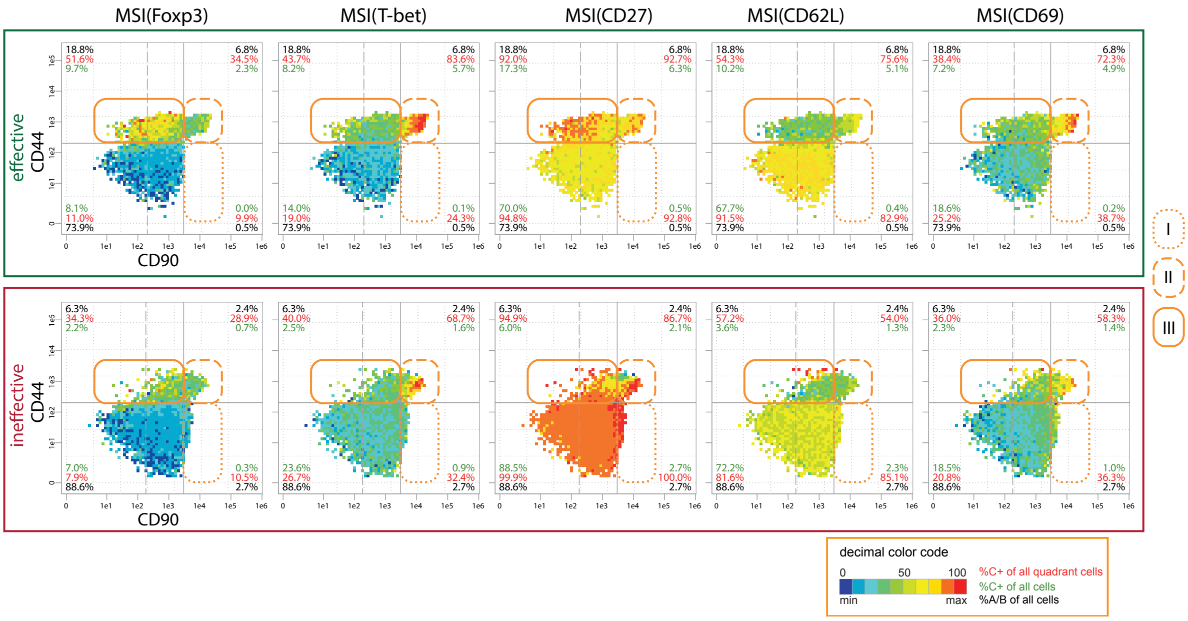


Figure 3.6: Pseudo-multi-parametric inspection of triploTs. Combinatorial triploTs are shown for CD90-CD44-MSI(Foxp3/T-bet/CD27/CD62L/CD69) for eff treatment (sample *B6_d3_Bl2*, top) and ineff treatment (*untr_d3_Bl1*, bottom) example with bin size of $arcsinh(x) = 0.2$ and minimum number of cells= 5, x and y axes range: $arcsinh(x) = [0.2, 12)$. Bin colors are scaled according to minimum and maximum of both samples. Legend shows the bin regions indicated with different line types in orange. Grey continuous lines indicate cutoffs for CD90$^{\text{high}}$ (vertical at 8.7) and CD44^{+} (horizontal at 6.0), respectively. The cutoffs for the associate markers C+ from left are set at 3.0, 3.0, 3.6, 4.0 and 3.0. Grey dashed line indicates cutoff of CD90^{+} (vertical at 6.0).

3.2.3 TriploT section values as input variables in the model

Despite of the implementation of the triploTs, which improves and facilitates the analysis of cytometric data, sole visual inspection is no longer feasible with the fast development of the cytometers. With every additional parameter, the amount of parameter combinations increases exponentially, and with this the amount of potential subpopulations. Manually exploring the data set with all 41 parameters by conventional gating strategies ($\frac{1}{2} \times 41 \times (41 - 1) = 820$ plots) or even with triploTs ($B_{41} = 31,980$ plots) become too much to handle and too time consuming. For example, when spending an average of 30 seconds on each plot, this takes roughly 7 hours and 11 days with contour plots and triploTs, respectively. Furthermore, manual inspection is hypothesis driven. Not all markers may be useful in the examination of e.g. T cells. Due to the biological meaning, some markers are considered to be not expressed in this context, but there are still too many triploTs to inspect even when only half of the total markers are of interest for T cells ($B_{20} = 3,420$ triploTs). To cope with the examination of the huge amount of triploTs, this dissertation's workflow provides an additional VE step which endorses the usage of a computational approach by filtering the triploTs to those whose three marker combinations are most significant and differentiating between two groups.

A first attempt to transform the triploT matrix information into informative variables as basis in a supervised machine learning approach for VR is illustrated in Figure 3.7. The range of the bins in a triploT is evenly divided into four rectangular sections. Depending on the selected calculation method, either mean, maximum, variance, relative or absolute range of the bin values in these triploT sections are extracted and added to the variable table. Only the absolute range as section calculation is demonstrated in this dissertation, since its results show the most differentiating variables after deploying the machine learning approach. The variable table construction is similar to the triploT matrix calculation but results in four values instead of a matrix in each *for*-loop (App. D, Lst. 4). Similar to the equation in Section 3.2.2, there are three *for*-loops for each observation. With $m = 41$ number of protein markers and sections of $q = 4$, the total variable count consists of $p_{m=41,q=4} = \frac{1}{2} \times (41-2) \times (41-1) \times 41 \times 4 = 109,668$ section values for each sample. The emphasis is on values and not matrices. The characteristics of the sections are labeled corresponding to the marker position, calculation method and section (e.g.˜'CD44_CD90_CD27_absRange_S4'). In the subsequent steps the section values are called explanatory variables p.

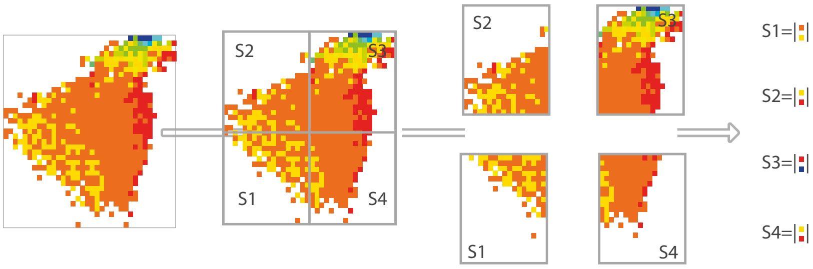


Figure 3.7: Scheme to calculate triploT bin sections S1-S4. The range of the bins in a triploT is evenly divided into four sections. The absolute range of the MSIs in the bin section is applied on each section and is extracted as four variables of each three-marker-combinatorial triploT. The section values are then used as input variables in the VR step. Illustration is shown on sample *untr_ d3_ Bl1* with CD44-CD90-MSI(CD27).

Filtering steps to reduce the curse of dimensionality

Apart from the enormous reduction of computational and storage costs, removing irrelevant and redundant variables without significant loss of information also reduces the curse of dimensionality in a sparse problem [37]. The ratio of variables p to samples N decreases, as well as the $p >> N$ problem in classification models. As a consequence, the models are less prone to overfitting and have better predictive power. Fur this purpose, the four following filtering steps are developed and applied to the example data set.

Filter I The following protein markers are not further investigated in the subsequent analysis: Ter119, CD19, CD8, IgD, IgM, B220, F4-80, PγMT, NK1.1, as well as FcER1a (Tab. T1). These protein markers are considered not to be expressed by CD4$^+$ cells. Hence, the protein marker count reduces from 41 to $m = 31$ which results in a lower number of $p' = 2 \times 31 \times 30 \times 29 = 53,940$ sections as input variable.

Filter II The density plot of each marker shows a different amount of cells with zero SI. This indicates no signal by the respective antibody (Fig. S2). However, these cells cannot be assigned correctly concerning their intensity values only for the respective marker. Therefore, the first bin rows and columns of $[0, 0.2)$ for each basis marker combination (A-B) are not considered for section calculations and the subsequent VR algorithm (Fig. S8). The cells falling into these bins are kept for other marker combinations, since they can still be correctly assigned in other intensity measurements. Thus, this filter does not reduce the dimension but increases the quality of the variables.

Filter III According to the expert knowledge, values of section S1 (Fig. 3.4) are much less informative in this study because these areas contain double negative cells (A$^-$B$^-$) and background signals from other markers, and are consequently filtered out. As a result, the variable space reduces to $p' = 53,940 \div 4 \times 3 = 40,455$ section variables.

Filter IV If a specific three-parametric combinatorial triploT matrix of a sample has a total displayed bin count of less than 400 bins, section values for this sample are set to not available (NA). Manual inspections showed that the X-Y-plane of a triploT with less than 400 bins is not scattered enough to extract meaningful section values compared to triploTs with other different basis marker combinations in this data set. An example is demonstrated in Figure S11b: sample *untr_ d3_ Bl3* (bottom right) has only 297 bins. The range of the bins (CD90: 3.4-10.6; PD-L1: 3.2-6.4) is different to the other samples in the ineff group (CD90: $\sim 0.4 - 10.6$; PD-L1: $\sim 0.6 - 6.4$). Furthermore, the bins with the low MSIs(CD86) is located in a different area compared to the other ineff samples. On the contrary, sample *CD1_ d3-2_ Bl1* from eff treatment (bottom left) has 412 bins and lacks of solely a few bins, but shows the same pattern compared to the other samples of the eff group. After finishing section calculations for the whole data set, each section value is reviewed. If there are 80% or more of the total number of samples (here 10 or more out of 12 samples) NA values, these section values are in turn not considered for further analysis.

After deploying the four filtering steps, the number of variables are reduced by a factor of 43. Table 3.1 shows the whole transformation from the raw data set over gating and filtering to the final engineering of the section values, of which gating and engineering constitute the majority part of the dimension reduction. The final dimension of the matrix comprises $p'' = 2,523$ triploT section values as explanatory variables and $N = 12$ samples as observations. This matrix is deployed for the further investigation in the embedded method in introduced in the following section, as a consequence to provide a variable table which ranks these variables in the order of the prevalence picked by the method.

Table 3.1: Data dimension reduction with the proposed VE workflow. The steps are shown where data dimension of example data set from *Spitzer et al.* are decreased. Final dimension on last row is then used for the subsequent VR step.

Workflow steps	Rows × columns (× samples)
Raw data set	1,726,796 × 41 (μ =132,830 × 41 × 13)
Gate CD4+	347,788 × 41 (μ =26,753 × 41 × 13)
Quality control	341,973 × 41 (μ =28,497 × 41 × 12)
Remove outlier	336,545 × 41 (μ =28,045 × 41 × 12)
Filter I: select biological marker	336,545 × 31 (μ =28,045 × 31 × 12)
Engineering variables: sections S1-S4	12 × 53,940
Filter III: select sections S2-S4	12 × 40,455
Filter IV: if 80% of samples have #bins≥ 400	**12 × 2,523**

3.3 Variable ranking based on regression analyses

Statistical models are used to learn specific patterns from a pool of training data. They are applied for instance in the fields of automated diagnostics, computer vision, speech recognition, credit card fraud detection and stock market screening. There are many categories of models available, such as models for unsupervised or supervised learning, models for classification or regression problems, and models based on linear regressions or decision trees. Herein, the embedded variable selection (VS) method elastic-net regularized logistic regression (erLR) is deployed for the labeled example data set. The model is a simple approach, thus has a low computational cost, and it allows to handle data sets with a $p >> N$ problem. The applied regularization deals with the multi-collinearity of the input variables and regulates the over-fitting of the model. The main goal is to provide significant and relevant triploT sections and protein markers, to guide the examiner, where to begin the inspection of the triploTs and subsequently uncovering interesting marker combinations to identify discriminating subpopulations. With 31 biological markers of interest out of 41 protein markers, there are $53,940$ and $109,668$ possible sections, respectively.

In this dissertation, data preparation and filtering steps are deployed, resulting in a drastic reduction of the amount of triploT section values, from possible $p = 109,668$ to $p'' = 2,523$ (Sec. 3.1 and 3.2.3). The resulting sections (S2-S4) for each three marker combination serve as input variables for erLR. With the assumption of sparsity most of the variables are not significant in multi-dimensional data. Consequently they will not have much additional information about the data and are therefore not predictive [37].

The erLR tries to select only the variables which explain a very large proportion of the variation in the data. Hence, these variables play an important role in distinguishing between different phenotypes. They are extracted from several iterations and are ranked in order of prevalence. The model is evaluated by cross validation (CV) in a nested manner which includes the tuning of two regularization parameters.

Originally, the example data set from *Spitzer et al.* has a multi-class problem which has been reduced to a binary problem. With the sparse amount of samples it is recommended to simplify the question. Therefore, this study uses the lowest reasonable number of groups. Additionally, since the treatments were categorized in two groups by the original author, it is reasonable to assemble them into two groups as well. Thus, treatments of antibody mix with CD1 or B6 antibodies are categorized into eff treatments. On the other hand, classical non-specific treatment of the antibody mix with anti-PD1 or no treatment (untreated) are summarized as ineff treatments.

3.3.1 Variable ranking comprises two cross-validation steps

The erLR with CV has two regularization parameters which need to be tuned: α and λ. α tunes the erLR towards either $L1$ (LASSO) or $L2$ (ridge) regularizations, which are introduced in Section 2.4.3. λ tunes the variable restriction of each CV. To estimate the performance of each configuration the error rates (deviances) of these models are calculated. The tuning of both parameters are processed in two consecutive CVs. Figure 3.8 shows an overview of the algorithm resulting in the final collected variables which are highly differentiating between both groups (eff and ineff treatment).

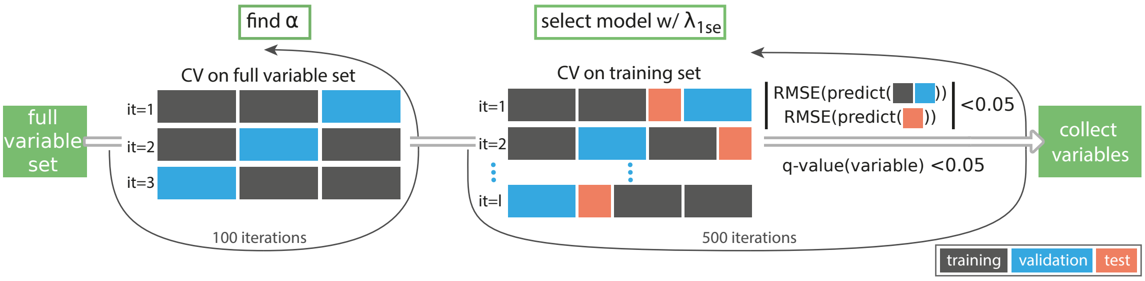


Figure 3.8: Schematic variable ranking workflow with triploT sections as explanatory variables.

The entire preprocessed example data set and the training subset are divided into $k = 3$ folds, with $k = 3$ meaning one third of the training set is left out for the internal validation (full set in 1st CV cycle=4/4/4, training set in 2nd CV cycle=3/3/4). The standard fold is set to $k = 10$. Due to the small amount of samples, a setting with $k > 3$ is not desired. However, a warning message using less than eight samples per fold appears after each CV run (App. D, Lst. 6). This warning is tolerated, as $k = 3$ is considered to be a good trade off between having at least one condition in one fold and still have some variance and stability with at least one more sample of a condition. To vary the partition and support the stability of the model, the folds are resampled in each CV run.

A generalized linear model as an embedded technique for variable ranking

With the classifier generalized linear model (GLM), there are $N = 12$ observations for the response variable $\mathbf{Y} = y_1, ..., y_j, ..., y_N$ and $p'' = 2,523$ associated explanatory variables presented as $x_{ji} = (x_{j1}, \ldots, x_{jp''})^T$ (Sec. 2.4). The response variable y_j can be described and predicted by a linear combination of the standardized explanatory variables. Binary response variables are created (1 = eff, 0 = ineff) and are added as a column to the data structure. Consequently, the relationship between probability and the response variables is not linear, but sigmoidal. Hence, a function of the probability is required which converts probability into a value that ranges from $-\infty$ to $+\infty$ and which has a linear relationship with the variables X. For this purpose, the logit of $\mathbf{Y}$ (Sec. 2.4.1) as the outcome in the regression equation is used.

Determination of the regularization in the first cross-validation

Regularization in regression models is a crucial technique to control the over-fitting, thus poor generalization, phenomenon. This involves the addition of a penalty term to the error function in order to restrict the other regularization parameter λ from reaching large values. Fewer variables would result in a less complex and more stable model that is less sensitive to statistical fluctuations in the input data. Regularization constrains the coefficient estimate λ towards zero. The elastic-net penalty is deployed in this analysis which combines both $L1$ and $L2$ regularizations mentioned in Section 2.4.3 [59]. The mixing parameter α adjusts the elastic-net to regularize more as a $L1$ or as a $L2$. As $\alpha \rightarrow 0$, the $L2$ regularization gains more weight than the $L1$ which results in an increase of explanatory variable numbers. The opposite happens for $\alpha \rightarrow 1$: the variable amount shrinks. As a consequence, erLR produces a regression model that is penalized with both the $L1$ and $L2$ regularizations resulting in an effective shrink of coefficients (like in $L2$), and some coefficients are set to zero (as in $L1$).

The Algorithm 3.1 finds the best setting between $L1$ and $L2$ regularizations for this data set. In the first CV, there is no need to partition the data set, since the aim of this CV is solely to find the suited configuration of α. A predefined but sampled *set.foldid* vector for k-folds is used with $k = 3$. In this way, each row is assigned to a random fold with both treatments integrated. α is iterated in the CV runs between 0 and 1 with step size= 0.1 and, importantly, with the same fold setting. For each α the deviances are collected and after one full iteration of α, the global minimal deviance determines the best α in this iteration, as illustrated in Appendix B, Figure S9. Then a new global iteration starts with a new random fold set and the iteration of α. This algorithm has been repeated ten times to see if 100 iterations are sufficient to get a stable α. In each repetition $\alpha = 0.9$ is picked more than 50 times. Associated R code for elastic-net is presented in Appendix D, Listing 5.

Algorithm 3.1: erLRM cross-validation to find suitable α.

```
Input: The variable matrix $\mathbf{X}' = (x_{ji})$; an outcome vector $\mathbf{Y} = y_{1,..,N}$; a vector
       $\alpha = 0, 0.1, ..., 1$; a randomized seed vector seeds;
Output: $\alpha'$ ($\alpha$ which is collected the most)
for $i \leftarrow 1$ to 100 do
    Step 1 set var seed = seeds[$i$]
    Step 2 set vector $\mathbf{fold_{id}} \in 0, 1, 2$ into 3 folds, every fold incorporates samples
      from both treatments
    foreach $\alpha$ do
        Step 3 do CV with cv.glmnet($\mathbf{X}'$,$\mathbf{Y}$,$\alpha$,$\mathbf{fold_{id}}$)
        Step 4 collect min(deviance)
    end
    Step 5 collect $\alpha$ with min(deviance)
end
```

Nested cross-validation to cope with the $p >> N$ problem

The true number of variables with informative and predictive value is not known beforehand. With too few variables the model cannot describe the data sufficiently and too many variables lead to over-fitting. In both scenarios, the predictive power is low. With logistic regression (LR) and the existent $p >> N$ problem in the example data set, it will eventually lead to over-fitting. To overcome this problem, this work's algorithm uses the CV technique, which aids in estimating the error over the data set, and in deciding what parameters work best for the model. In fact, a nested CV is deployed, which combines an inner CV which is equivalent to the training partition within an outer CV which is equivalent to the test partition. Hence, the data set is divided into training and test set with the ratio of 4:1 (80%/20%). To balance the need to use data to select a model and the need to use data to asses prediction, 3-fold cross-validation is used. Purely random partitioning can result in partitions containing only one condition, especially with such a small amount of samples. To assure proper classification in every run of the inner and outer CV, each condition is represented in both the training and test set as well as in the 3-fold partitions in the CV.

Algorithm 3.2 shows the second CV cycle to tune λ. This time, the CV uses the estimated α from the first CV cycle to extract predictive section values and their coefficients. To select the best configuration of these parameters to employ on the data set, only the training samples are used for this CV. In the final step of the performance estimation the separate test set is used to further evaluate the model. The section values as variables are collected for every iteration if the performance of the model is sufficient.

Algorithm 3.2: erLRM in a nested CV to extract differentiating triploT sections.

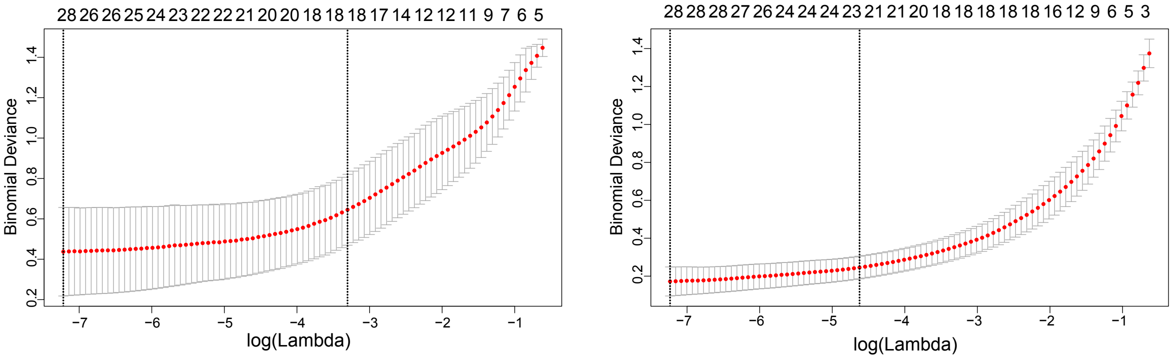
```
Input: A matrix $\mathbf{X}' = (x_{ji})$, divided into training and test set (ratio 4:1); outcome
       vector $\mathbf{Y} = y_{1,...,n}$; a seed vector seeds; regularization parameter $\alpha'$
Output: v;coeff
Step 0 Initialize $\mathbf{v}, \mathbf{coeff}, success.it = 0$
for $i \leftarrow 1$ to 500 do
    Step 1 set var $seed = \mathbf{seeds}[i]$
    Step 2 set vector $\mathbf{fold_{id}} \in 0, 1, 2$ training set into 3 folds, every fold incorporates
      samples from both treatments
    Step 3 do CV with cv.glmnet(training,$\mathbf{Y}, \alpha', \mathbf{fold_{id}}$)
    Step 4 select $\lambda_{1se}$ from CV
    Step 5 calculate RMSE with prediction model and $\lambda_{1se}$
              - on training test
              - on test set
    if | RMSE(test)-RMSE(training) | < 0.05 then
        $success\_it + 1 = success\_it$
        extract variables and associated coefficients from prediction model
        if $q$-value(variables)< 0.05 then
            collect variables to $\mathbf{v}[success.it]$ and associated coefficients to
              $\mathbf{coeff}[success.it]$ from prediction model
        end
    end
end
```

3.3.2 Error measurements for evaluation

Deviance as error measurement in cross-validation

To measure the goodness-of-fit of models an evaluation method is used. The deviance is commonly applied and is defined as minus two multiplied by the log-likelihood on the left-out test fold in a CV (Sec. 2.4.5). The deviance function is very useful for comparing two models when one model has parameters that are a subset of the second model [34, 36]. It is therefore the difference of their individual residual deviances. Figure 3.9 shows two scenarios of the CV. They are exemplary for all other CV runs. The penalization is in terms of shrinkage of the model variables. On the left is the deviance for the full, λ-unpenalized model and on the right, there is the heavily shrunk fit with large penalties. The mean deviances with different $log(\lambda)$ are displayed as red dots. The bars below and above the dots indicate the ± standard error of the mean deviance from the three replications. The two dashed lines show the locations of two different λ values. $log(\lambda_{min})$ (left line) is the $log(\lambda)$ with the lowest deviance error, and $log(\lambda_{1se})$ (right line) is the largest $log(\lambda)$ value within 1 standard error of $log(\lambda_{min})$. The numbers across the top are the amount of variables. Usually, the model with $log(\lambda_{1se})$ is chosen for further steps to reduce the effect of over-fitting [37].

The deviance is steadily increasing with larger values of $log(\lambda)$ on the rightmost edge of the plot. However, in the beginning the deviance is relatively flat over a large range of values of $log(\lambda)$ as the number of variables decreases (on top of the plot). This indicates little to no change in deviance while the number of variables in the model declines. Hence, removing some variables does not severely affect the model fit. These results lead to the conclusion that variable selection does work while handling correlated covariates with the elastic-net regularization. Furthermore, the results agree with the assumption of sparsity mentioned at the beginning of this chapter. However, standard errors do vary a lot between different CV runs. scenario 1 (Fig. 3.9a) shows high variation between the 3-folds. On the other hand, scenario 2 (Fig. 3.9b) is more precise with less standard error, thus restricting shrinkage of the variables with the usage of $log(\lambda_{1se})$. This implies that the sparse amount of samples still affects the outcome of the model. To further restrict the models, additional error measurements were deployed in this work to assure qualitative extraction of the most differentiating variables.



(a) CV scenario 1

(b) CV scenario 2

Figure 3.9: Two deviance measurement representatives of different $log(\lambda)$, with $\alpha = 0.9$ and section calculation = absolute range. Confidence intervals represent error estimates for the loss metric (red dots). They are computed using 3-fold CV. The two dashed lines show the locations of $log(\lambda)$. $log(\lambda_{min})$ on the left and $log(\lambda_{1se})$ on the right. The numbers across the top are the numbers of variables.

Two additional measurements to evaluate the resulting model and variables

The test set consists of only two samples, one representative from each condition. Hence, the test set is very small compared to the likewise small training set of ten samples. Root mean square error (RMSE) is used for this purpose (Sec. 6). The error of the training set is almost zero in any of the second CV runs (RMSE(training) $< 1e^{-7}$). The difference between the RMSE of training and test set lies between $(0.0002, 0.1661)$, where 70% quantile is < 0.05 (357 out of 500 runs). Algorithm 3.2 shows that variables from each CV run are only collected if the difference from RMSE of the test and training set is lower than 0.05. This restricts to usage of the variables of the model to only those that do not lead to over-fitting, and the CV run is then termed successful.

A final restriction is not applied to the model but to the selected variables. The variables from each successful CV run are individually tested if the means of the variables between both treatment groups are significantly different. Given that these samples are independent from one another, unpaired two-sided t-tests are deployed with adjustments on the *p*-values using the *Benjamini and Hochberg* [60] correction method. 97 different variables are selected from the models, which are significant between both treatments and are examined in the following section.

3.4 Variable ranking table as examination guidance

Building a highly predictive classification model with small amount of samples is cumbersome and not the main goal of this study. This workflow uses erLRM-CV as an embedded VR tool. The outcome of this workflow is a VR table which provides the most differentiating section variables from three marker combinations and a ranked marker table by prevalence. These variables can then be used as guidance to look for interesting subpopulations in triploTs with the three-parametric combinations extracted from the variables.

3.4.1 General description of the variable ranking table

Ranking of three combinatorial marker sections

The collected section variables are ranked in order of prevalence as predictive variable outcome of each erLRM-CV run in each iteration. These variables comprise the absolute range of the MSIs in the sections and are not based on biological importance. The original section count could be further drastically reduced by the erLR from 2,523 to 97, resulting in a reduction factor of 26 fold. The total rank table of the section variables is shown in Table T5, where the three markers on position A, B and C and the section number, which discriminates between eff and ineff treatment, are extracted from these variables and displayed on the right. In addition, prevalence count, frequency of these counts, and mean coefficient values are listed. The count values can be used for cutoff setting, and the latter value is the mean of all regression coefficients and shows the direction of the variable. If the coefficient is positive, the section value is higher in eff treatment than in ineff treatment. Most variables have positive coefficients (83 out of 97). The ranked variables are subsequently ranked again corresponding to their positions in a triploT. The 10 highest positions are evaluated by manual inspection in the following subsections.

Ranking of markers relating to their position in a triploT

Apart from ranking the three combinatorial variables to assist identifying differentiating subpopulations, two additional tables are provided. The tables present the ranked markers in the order of prevalence as axes marker on x- or y-axis (A+B) and in the order of prevalence as associated marker (C), respectively. Table 3.2a and 3.2b show the ranked marker counts from the total rank table. It is evident that the number of ranked markers of combined axes positions is smaller than the number of associated markers in position C in a triploT. This is due to the restrain by the deployed Filter IV from Section 3.2.3, which filters the axes combination where the ranges from both markers are wide enough, so that the triploT bin count is 400 or higher. Furthermore, it is apparent that the prevalence in each table falls rapidly in the first three positions. Position 5 (CD86 in Tab. 3.2a and KLRG1 in Table 3.2b) appears less than 8% in both tables, suggesting that the biggest differences between eff and ineff treatment occur mainly in a few marker combinations.

Table 3.2: Full VR table (Tab. T5) grouped by markers in x and y axes (a) and in position C as associated marker (b), ordered by prevalence. The ranking is the outcome of a nested erLRM-CV with $\alpha = 0.9$ and section calculation = absolute range.

(a) markers in axes positions x and y (A,B)

Pos	Rank	Marker A+B	Counts	% Counts
1	1	CD90	48	0.247
2	2	CD44	27	0.139
3	3	Ki67	21	0.108
4	4	CD138	18	0.093
5	5	CD86	15	0.077
6	6	KLRG1	14	0.072
7	7	PDCA.1	10	0.052
8	8	PD.L1	9	0.046
9	8	CD62L	9	0.046
10	10	T.bet	8	0.041
11	10	CD103	8	0.041
12	12	CD16.32	3	0.015
13	12	CD69	3	0.015
14	14	Ly6G	1	0.005

(b) markers in position C

Pos	Rank	Marker C	Counts	% Counts
1	1	CD86	25	0.258
2	2	Ly6C	10	0.103
3	3	PD.L1	9	0.093
4	3	CD27	9	0.093
5	5	KLRG1	6	0.062
6	5	T.bet	6	0.062
7	7	CD90	5	0.052
8	7	MHC-II	5	0.052
9	9	CD44	4	0.041
10	10	CD45	3	0.031
11	10	Foxp3	3	0.031
12	12	CD11c	2	0.021
13	13	Ki67	1	0.01
14	13	CD138	1	0.01
15	13	PDCA.1	1	0.01
16	13	CD62L	1	0.01
17	13	CD103	1	0.01
18	13	CD16.32	1	0.01
19	13	CD11b	1	0.01
20	13	CD64	1	0.01
21	13	RORgt	1	0.01
22	13	SiglecF	1	0.01

3.4.2 Evaluation of top ranked variables

The highest ten ranked variables are visually inspected to evaluate this study's approach. It is evident that these variables have positive coefficients, meaning all section values, which decode for the absolute range of the MSIs in the sections, are increased in eff treatments. The box plots of these variables also show clear separation of both treatments (App. B, Fig. S10).

The top five positions (pos. 1-5) of the VR table are selected in every model of a successful CV run and consequently have the same prevalence count of 357 (Tab. 3.4), but there is no sharp decline in counts with the following five positions. CD90 and CD86 are dominant in this table: in seven positions, both markers appear together in combination, and in two positions either CD90 or CD86 is placed. Only one position does not include any of them (pos. 6). This can also be captured in the VR table of markers grouped by axes and associated marker C from the top ten ranked only (App. B, Tab. T3). In addition, the section S2 is only selected in one position (pos. 9). All other positions are determined in section S3 (6x) and S4 (3x).

Table 3.4: VR table from 10 most selected section values. From left: ranking position, prevalence count and frequency, mean model coefficient and split variable information (marker combination for x and y axes (A,B) and associated marker (C)) and section selection (S). The ranking table is part of the outcome of erLRM-CV with $\alpha = 0.9$ and section calculation = absolute range (absRange).

Pos	Rank	Variables	Counts	% Counts	Coeff	A	B	C	S
1	1	CD86.CD138.CD90.absRange.S3	357	0.046	0.3686	CD86	CD138	CD90	S3
2	1	CD90.CD103.CD86.absRange.S4	357	0.046	0.3892	CD90	CD103	CD86	S4
3	1	CD90.CD138.CD86.absRange.S3	357	0.046	0.4979	CD90	CD138	CD86	S3
4	1	CD90.KLRG1.CD86.absRange.S4	357	0.046	0.3696	CD90	KLRG1	CD86	S4
5	1	CD90.PD.L1.CD86.absRange.S3	357	0.046	0.6033	CD90	PD.L1	CD86	S3
6	6	CD138.Ki67.CD27.absRange.S4	355	0.046	0.9503	CD138	Ki67	CD27	S4
7	7	Ly6G.CD90.CD86.absRange.S3	347	0.045	0.1607	Ly6G	CD90	CD86	S3
8	8	CD90.PDCA.1.CD86.absRange.S3	345	0.045	0.1442	CD90	PDCA.1	CD86	S3
9	9	CD62L.Ki67.CD86.absRange.S2	337	0.044	0.3056	CD62L	Ki67	CD86	S2
10	10	CD90.CD138.Ly6C.absRange.S3	333	0.043	0.2321	CD90	CD138	Ly6C	S3

Manual inspection of the top 10 with triploTs from all samples shows that almost all positions have noticeable, small dense subpopulations which are high in eff treatment. Figure 3.10 illustrates the triploTs from the previously used samples *Bl6_ d3_ Bl2* from eff and *untr_ d3_ Bl1* from ineff treatment. Only position 10 shows fewer differences between both treatments. In positions 1-5 it is evident firstly that CD90 and CD86 are positively correlated and that the CD90$^{\text{high}}$ bin area is CD86^{+} and highly discriminative between both treatments. This can be observed with CD90 as basis marker and CD86 as associated marker (pos. 2-5 and 8) and vice versa (pos. 1). The triploTs in position 4 indicate a bin area at KLRG1$^{\text{high}}$ in eff treatment which is not present in the other group. KLRG1 is a differentiation marker which should be increased in a well-functioning treatment and therefore this triploT information agrees with this VR outcome. The subpopulation can also be characterized as regulatory T cells, and is CD86^{+} (Fig. 3.10). Position 6 is the only one in which the MSI values are higher in ineff treatment. Furthermore, the marker combination in this position does not include CD90 and CD86. CD27 appears to be higher in Ki67^{+} in eff treatment, but acts inversely with ineff treatment.

3.4.3 Additional characteristics of the subpopulation with guidance of the variable ranking table

TriploTs from positions 1 to 5, 7 and 8 in Figure 3.10 give a clear indication of a subpopulation in eff treatment which is characterized by CD86^{+} and CD90$^{\text{high}}$, and absent in ineff treatment. Both markers are higher expressed in eff treatment. To find a better separation of the subpopulations, triploTs with basis markers CD90 and CD86 are subsequently inspected.

A strong correlation between marker CD44 with both basis markers CD90 and CD86 is observed in eff treatment (Fig. 3.11). The subpopulation CD86^{+}CD90$^{\text{high}}$ is hence not only CD44^{+}, but CD44$^{\text{high}}$ (orange rectangle). Furthermore, the cells of this subpopulation are positive for Tbet, CD69, and Ki67, which emphasizes that these are Th1 cells (Tbet$^{\text{high}}$), are activated (CD69$^{\text{high}}$), and are proliferating (Ki67$^{\text{high}}$). Interestingly, due to this axes arrangement with CD90 in x-axis and CD86 in y-axis, the cells in this area are missing in

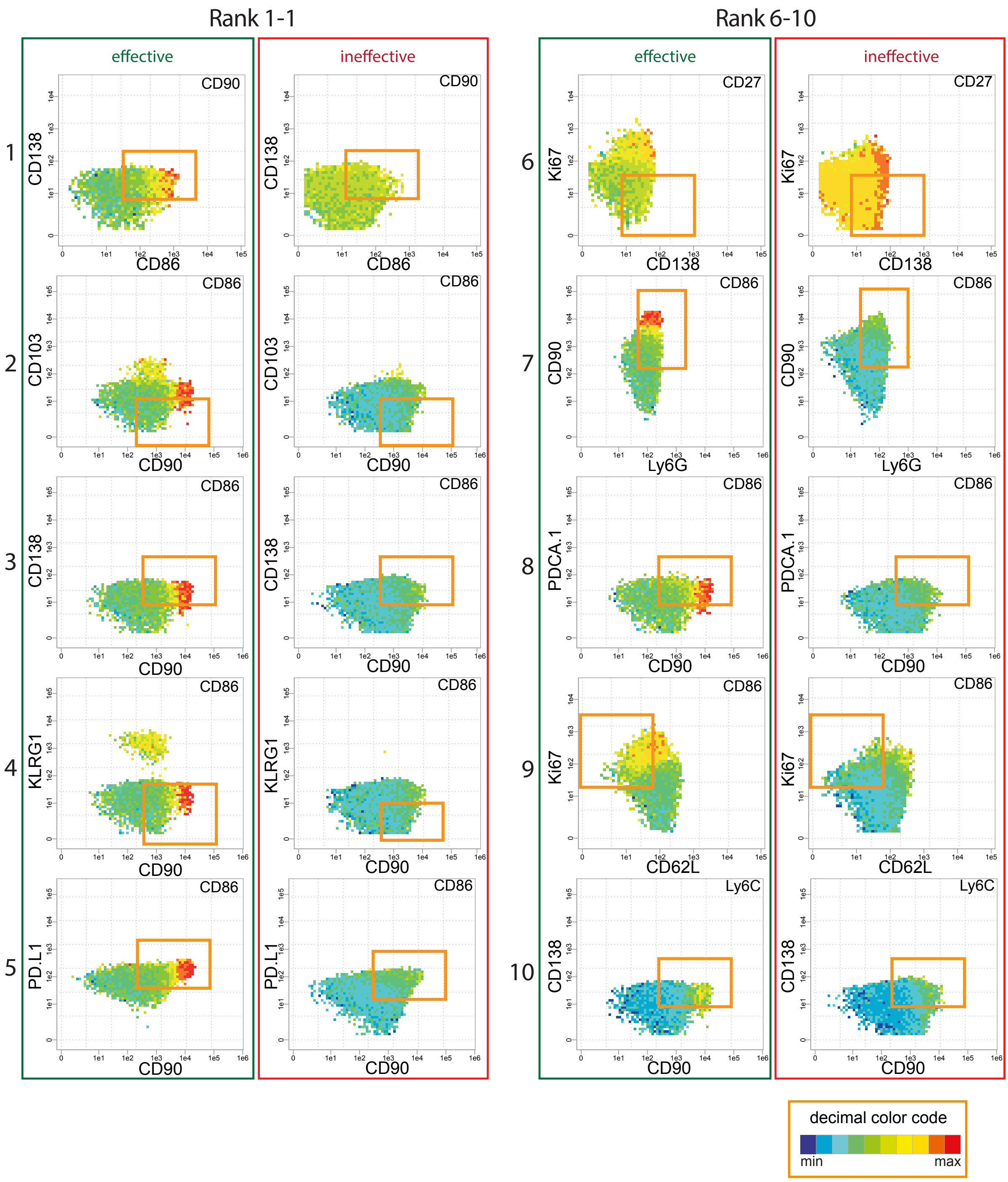


Figure 3.10: TriploTs of samples *B6_d3_Bl2* in eff treatment and *untr_d3_Bl1* in ineff treatment with top 10 ranked three marker combinations. Numbers on the left representing the positions in the table. The bin sections selected from the embedded VR are approximately marked with orange rectangles. The bin colors in both samples are scaled according to the global minimum and maximum of MSI(C).

ineff treatment which confirms the statement made with the triploTs from position 1 to 5, 7 and 8 in Figure 3.10. Another interesting point is, that the immunosuppressor PD-L1 is also highly expressed in eff treatment (App. B, Fig. S11 and S12). This was also reported from *Spitzer et al.* (Sec. 2.5). Albeit technically, PD-L1 could be a discriminative marker between eff and ineff treatment, in the biological meaning, this marker is not part of a functional subpopulation of activation markers. It is therefore not listed together in one subpopulation. Foxp3 is notable as well, which marks for Treg cells. It is not distinct in this plotting frame (App. B, Fig. S12), and is found by sole visual inspection with triploTs.

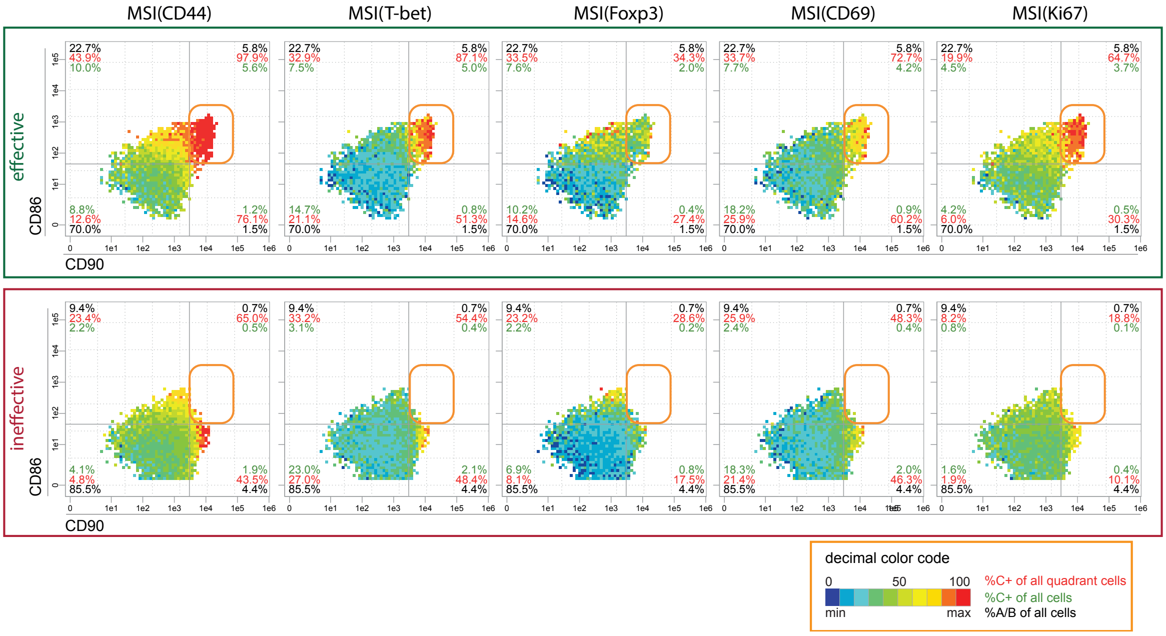


Figure 3.11: Pseudo-multi-parametric inspection of triploTs. Combinatorial triploTs of CD90-CD86-MSI(CD44/Tbet/Foxp3/CD69/Ki67) in eff treatment (*B6_d3_Bl2*, top) and ineff treatment (*untr_d3_Bl1*, bottom). Bin colors of both samples are scaled according to the global minimum and maximum of both samples. Orange rectangles indicate subpopulation found with guidance of VR algorithm. Grey continuous lines indicate cutoffs of CD90$^{\text{high}}$ (vertical at 8.7) and CD86^{+} (horizontal at 4.5), respectively. The cutoffs for the associate markers are set at 4.0, 3.0, 3.0, 3.0 and 5.5 (from the left). TriploTs with basis markers CD90 and CD86 and all residual associate markers from both samples are displayed in Appendix B, Figure S12 and S13.

In conclusion, after the exclusively manual inspection of the subpopulation from *Spitzer et al.* in Section 3.2.2, the identified subpopulation could be narrowed down to three subregions, namely naïve, Th1 and Treg cells. With guidance of both top 10 VR, the total marker table and their respective triploTs, one clearly observes a concentration of cells. The Th1 cells are further characterized by CD90$^{\text{high}}$CD86^{+}CD44$^{\text{high}}$Tbet$^{\text{high}}$CD69$^{\text{high}}$Ki67$^{\text{high}}$.

A proof-of-concept of this study's proposed workflow is demonstrated on the example data from *Spitzer et al.* With the consolidation of data preparation, engineering triploTs and their respective section values, the cross-validated erLR, and the biological knowledge, an interesting subpopulation is extracted which is highly significant between eff and ineff treatment. Now one is curious to see if these results are comparable to the results from the current gold standard visualization tool *viSNE* and the classifying tool *Citrus*.

3.4.4 State-of-the-art tools as validation

A well-established visualization technique is based on a dimension reduction algorithm *t*-distributed stochastic neighbor embedding called *viSNE* (Sec. 2.2.2) [12, 61]. As a result, artificial axes emerge from the calculation similar to the MDS, against which the cells are plotted. The considerable difference between *viSNE* and MDS is that *viSNE* is a non-linear approach. Ideally, there would be separable groups of cells on these axes which represent subpopulations with differentiating characteristics. The state-of-the-art classifying tool *Citrus* uses a similar cluster approach as in other popular tools, but additionally is comprised of a classification algorithm (Sec. 2.2.2) [14]. It has been successfully applied to mass cytometry based cancer diagnosis. Both tools are deployed with default options using the same raw data set after gating on CD4$^+$ T cells (after data preparation step 1) with the 31 biological relevant markers and with exclusion of sample *untr_ d3_ Bl3*, as filtered from this study to have a comparable starting point.

***viSNE* map shows same subpopulation found with *PRI*'s ranking table**

Despite the use of the provided markers from *Spitzer et al.* (Sec. 3.2.2), it is not straightforward to identify their subpopulation with the *viSNE* map. With the suggestions from *PRI*'s VR table, the Th1 subpopulation could be further narrowed down, and thus more markers could be used to find the subpopulation in the *viSNE* map. Figure 3.12 presents one version of this map with marks of a cell cluster which has (partly) similar characteristics to the Th1 subpopulation (Sec. 3.4.3). Several runs of *viSNE* could not reproduce the same *viSNE* map, but the same subpopulation could be recognized again on another location on the map (App. C, Fig. S19). This validates the further characterized subpopulation. Interestingly, many cells of eff treatment are located on the map where the cells of ineff treatment are not positioned and *vice versa*, leading to the conclusion that the characteristics of the cells are to some extent different between the treatments. It should be noted, that the cells from the marked cluster is uniformly CD90$^{\text{high}}$ and CD44$^{\text{high}}$, but show a mixed composition of low to high expressed CD86, CD69, Tbet and Ki67. The subpopulation Treg from sole manual inspection (Sec. 3.2.2, bin region II) could not be captured even after several *viSNE* runs with random seeding.

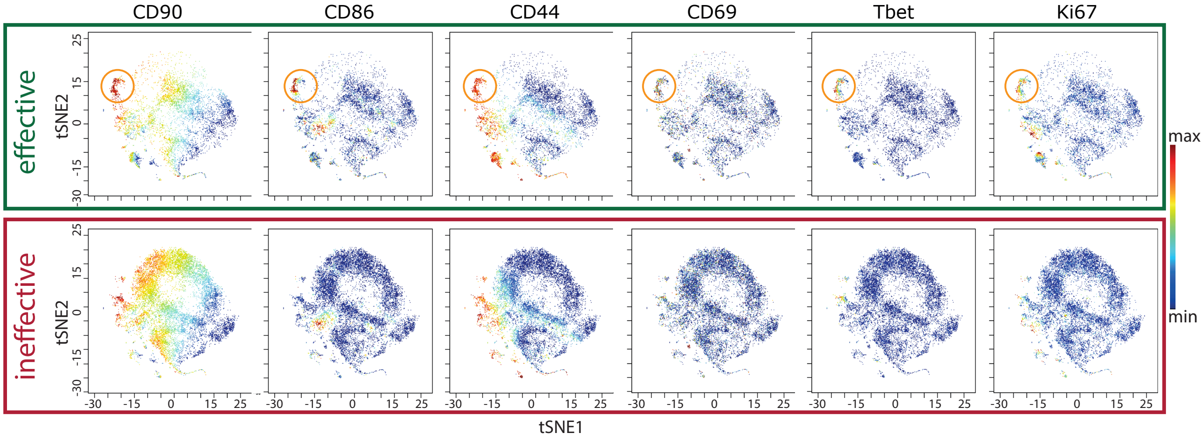


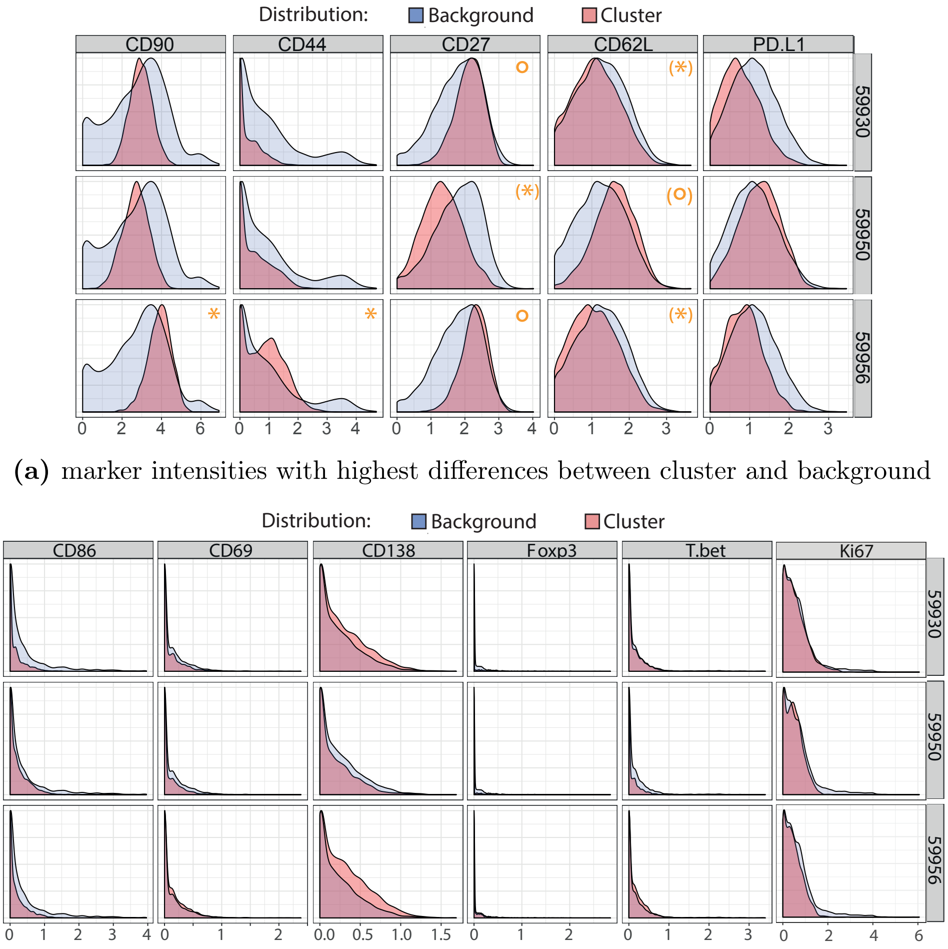
Figure 3.12: *viSNE* map using all samples except of sample *untr_ d3_ Bl3*, showing the SIs of CD90, CD86, CD44, CD69, Tbet and Ki67 with global scale between samples of eff and ineff treatment. Orange circles indicate the location of the subpopulation Th1 found with guidance of *pattern recognition of immune cells* (*PRI*)'s VR table (CD90$^{\text{high}}$CD86$^+$CD44$^{\text{high}}$CD69$^{\text{high}}$Tbet$^{\text{high}}$Ki67$^{\text{high}}$ in CD4$^+$ T cells). Default settings: total events=100,000 with proportional subsampling; iterations=1,000; perplexity=30; theta=0.5; random seed.

Citrus' clusters have different characteristics

Citrus begins with hierarchical clustering on the down-sampled data set. The median expressions of the resulting cell clusters are then used as input in the supervised learning algorithm to find the best predictor clusters. The classifier is similar to this study's approach: $L1$-regularized generalized logistic regression (Sec. 2.2.2). The resulting 33 clusters of cellular populations are visualized by a tree and labeled by numbers as identifiers (App. C, Fig. S20). These clusters contain redundant events so that each branch is a subgroup of the mother cluster. Compared to the whole down-sampled data set, the clusters are selected by the model, and are colored in dark red using CV with $log(\lambda_{1se})$. In this scenario, the selected clusters from CV with $log(\lambda_{1se})$ and with $log(\lambda_{min})$ are the same. Five out of 33 clusters are identified to be most predictive between both treatments, out of which the main clusters with ID 59930, 59950 and 59956 are picked subsequently for comparison with this study's outcome, because the two other significant clusters show similar behaviour to their respective mother cluster.

Next to the feature tree, overlapped densities are provided from *Citrus*, which show the intensity distribution from 11 out of 31 protein markers of clusters with ID 59930, 59950 and 59956, respectively, to the total residual down-sampled data set. Herein, these 11 markers are distributed in markers with highest visible variation to background (Fig. 3.13a), and in markers, which have been reported with discriminative character from *Spitzer et al.* and *PRI* (Fig. 3.13b). If the density distribution of a marker in a cluster is shifted to the right compared with the background, it is presumed that this marker is produced in the cells ($^{+}$), and a shift to the left results in a non-production of the respective marker ($^{-}$).

In general, there are many density distributions of the clusters, which have almost zero to only small variation in peak location and distribution, or both, compared to the density distribution of the background. Only five out of 31 markers have in some extent discernible variations, which are markers CD90, CD44, CD27, CD62L and PD-L1 (Fig. 3.13a). These markers also appear in the top eight ranks of this study's VR table ordered by axes positions (Tab. 3.2a), but CD27 only appears at rank 3 in the marker table of position C as associated marker (Tab. 3.2b). Looking at the VR table, all markers are present in the top 10 ranks except for CD44. CD44 appears first at position 13, which is still highly ranked (App. B, Tab. T5). The characteristics with CD90^{+}CD44$^{\mathrm{low}}$CD62L$^{\mathrm{low}}$ in the cluster with ID 59956 show the most similarities to the subpopulation from *Spitzer et al.* (CD44^{+}CD69^{+}CD62L^{-}CD27$^{\mathrm{low}}$CD90^{+}T-bet^{+}), but has contrasting specifics of CD27 (Fig. 3.13a). For *PRI*'s Th1 subpopulation on the contrary (CD90$^{\mathrm{high}}$CD86^{+}CD44$^{\mathrm{high}}$CD69$^{\mathrm{high}}$Tbet$^{\mathrm{high}}$Ki67$^{\mathrm{high}}$), only the characteristics of marker CD27 are similar in the clusters with ID 59930 and 59956, and the opposite characteristics of CD62 can be seen in cluster ID 59950. Looking at PD-L1, only cluster ID 59550 has similar specifications ($^{+}$) to the reports from *Spitzer et al.* and *PRI* (PD-L1$^{\mathrm{high}}$). Peculiarly, markers CD69 and Tbet, which are part of the subpopulation from *Spitzer et al.*,



(a) marker intensities with highest differences between cluster and background

(b) marker intensities for comparison to *Spitzer et al.* and *PRI*

Figure 3.13: *Citrus*' overlapped density distribution output from 11 out of 31 marker intensities from cells in clusters with ID 59930, 59950 and 59956, respectively, to the residual down-sampled cells after running *Citrus* with default options. Density plots are distributed in markers with highest discernible variation between cluster and background **(a)** and in markers of interest in *Spitzer et al.* and *PRI* **(b)**. Orange label marks common characteristics to the subpopulation of *Spitzer et al.* * and *PRI* °, respectively, and brackets show some approximated characteristics. Complete overlapped density distribution from cluster ID 59956 is displayed in Appendix, Figure S21.

do not appear differently compared with the background. The same statement holds true for *PRI* with all markers shown in Figure 3.13b. Thus, one could conclude, that the subpopulation of neither *Spitzer et al.* nor the narrow downed one of *PRI* can be identified by one of the selected clusters from *Citrus*.

Currently, investigators without programming expertise in the biological field are common [11]. To support them and to facilitate prospective analyses and comparisons, a user-friendly graphical user interface (GUI) is developed. This tool connects to the databases created by *PRI-base*, displays and saves diploTs and triploTs, with selected markers and selected plot and bin parameters. This is processed intuitively without any coding or query preparation, which is preferred by many biologists.

3.5 User-friendly GUI to facilitate the use of *PRI*

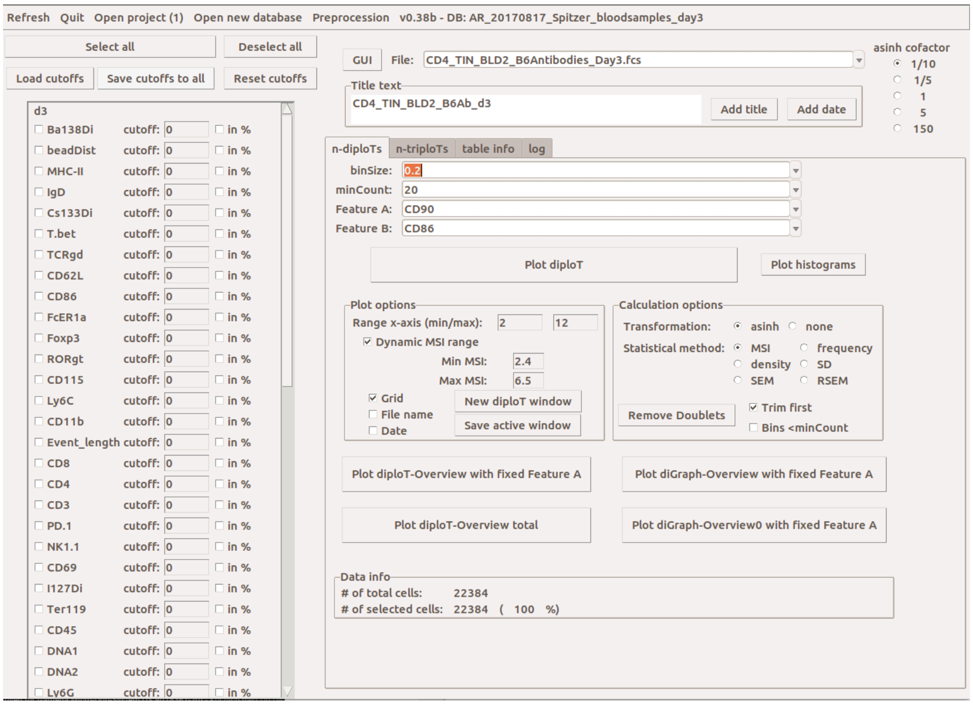
To develop a user-friendly tool, a GUI is deployed to interactively control and execute data preparation, to facilitate the diploTs and triploTs generation for manual inspection, and to apply VR on a selected data set. The tool is named *PRI-ana* as in 'analyzer'. By sourcing the script, *PRI-ana* begins directly with a pop-up window which asks for the selection of a database to start the analysis with. These databases are created by *PRI-base*. *PRI-ana* is thus aligned with *PRI-base*'s database structure.

3.5.1 Main window

The GUI's main window consists of three frames: the sample selection on top, the marker selection and cutoff setting on the left, and - the biggest part - the tab frame on the right with diploTs and triploTs functions, respectively, the *table info* and the *log* tab. The manual cutoffs can be set as *arcsinh*-values or in percentage values of the top highest values when the box on the right is checked. Furthermore, there are two drop-down menus at the top of the main window. In the first menu it is possible to chose a different project within the database, a different database entirely or an ordinary closing. The second menu consists of preparation steps including interactive or automated cutoff setting. When choosing this function, another window pops up. For an interactive cutoff setting, histograms of selected markers are displayed. When clicking on a position in the graph, the associated x-value will be saved for the cutoff between +/− population of the respective marker. The histogram of another selected marker displays directly after the click. For the automated cutoff setting, the algorithm determines the slope characteristics. It approximates the cutoff at a valley if there is a bi-modal density function. If no valley is detected, a shoulder with minimal slope or a cutoff at 20% highest intensity is set (App. D, Lst. 7).

There are many extensions implemented for ease of use and plotting. For one, if-queries are introduced and a pop-up window appears to notify the user if the cases occur. In an instance where the plot range excludes more than 5% of the cells, the whole bin structure will most likely not be displayed. In another event, a pop-up window appears when cutoffs for the selected markers are not set but is necessary for the selected statistical method such as frequency or MSI(+). Another extension is the scaling option of the MSI for the associated markers. Due to this, differences in tendencies or highlights in diploTs and triploTs can be easily captured in comparison between samples or groups. Furthermore, tooltips have been deployed, which appear when the mouse cursor hovers the radio and check buttons for a short description of the functions.

The two tabs with functions of plotting diploTs and triploTs, respectively, are described in the following subsections. The window frame in the tab *table info* shows the meta data listed in the database such as device information, user information, and file and column identifier, respectively. The window frame in the tab *log* lists the bug fixes and add-ons implemented in each published version of *PRI-ana* in reverse chronological order.



v0.38b - DB: AR_20170817_Spitzer_bloodsamples_day3

File: CD4_TIN_BLD2_B6Antibodies_Day3.fcs

Title text: CD4_TIN_BLD2_B6Ab_d3

asinh cofactor: ● 1/10 ○ 1/5 ○ 1 ○ 5 ○ 150

d3
□ Ba138Di cutoff: 0 □ in %
□ beadDist cutoff: 0 □ in %
□ MHC-II cutoff: 0 □ in %
□ IgD cutoff: 0 □ in %
□ Cs133Di cutoff: 0 □ in %
□ T.bet cutoff: 0 □ in %
□ TCRgd cutoff: 0 □ in %
□ CD62L cutoff: 0 □ in %
□ CD86 cutoff: 0 □ in %
□ FcER1a cutoff: 0 □ in %
□ Foxp3 cutoff: 0 □ in %
□ RORgt cutoff: 0 □ in %
□ CD115 cutoff: 0 □ in %
□ Ly6C cutoff: 0 □ in %
□ CD11b cutoff: 0 □ in %
□ Event_length cutoff: 0 □ in %
□ CD8 cutoff: 0 □ in %
□ CD4 cutoff: 0 □ in %
□ CD3 cutoff: 0 □ in %
□ PD.1 cutoff: 0 □ in %
□ NK1.1 cutoff: 0 □ in %
□ CD69 cutoff: 0 □ in %
□ I127Di cutoff: 0 □ in %
□ Ter119 cutoff: 0 □ in %
□ CD45 cutoff: 0 □ in %
□ DNA1 cutoff: 0 □ in %
□ DNA2 cutoff: 0 □ in %
□ Ly6G cutoff: 0 □ in %

binSize: 0.2
minCount: 20
Feature A: CD90
Feature B: CD86

Plot options
Range x-axis (min/max): 2 12
☑ Dynamic MSI range
Min MSI: 2.4
Max MSI: 6.5
☑ Grid
□ File name
□ Date

Calculation options
Transformation: ● asinh ○ none
Statistical method: ● MSI ○ frequency ○ density ○ SD ○ SEM ○ RSEM
☑ Trim first
□ Bins <minCount

Data info
# of total cells: 22384
# of selected cells: 22384 (100 %)

Figure 3.14: GUI of *PRI-ana* with frame *n-diploTs* in focus.

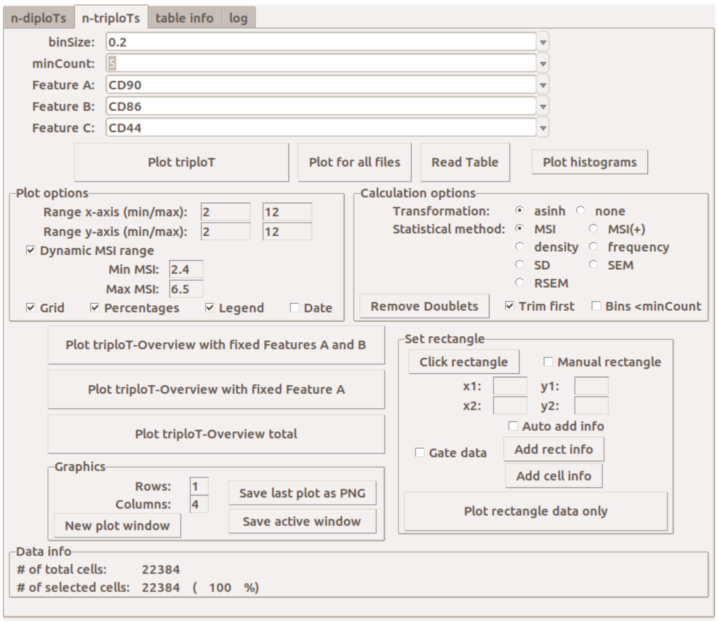
3.5.2 The window frame in the tab *n-diploTs*

The window frame in the tab *n-diploTs* starts with the selection area with options to choose different bin sizes, minimum number of cells (minCount) and basis markers and associated markers labeled as ‘Feature A’ and ‘Feature B’, respectively (Fig. 3.14). The subsequent buttons create a diploT with the set configurations and plots histograms of the selected markers from above, respectively. In the subframe ‘Plot options’, plot areas and bin color range can be set. Additional information such as grids for coordination guidance or file name and date stamps can be plotted for correct assignments when saving these plots for subsequent analyses. The subframe ‘Options’ supports different transformation methods and statistical methods applied on the bins. In addition, outlier removal or doublets removal (with flow cytometric data) can be applied before plotting. Also, bins can be displayed in pale colors if there are cells but are not sufficient to be displayed in full colors due to the minimum number of cells setting.

To speed up the plotting process, additional buttons are added which create multiple diploTs of the selected markers. Either diploTs with fixed basis marker A and all other selected associated markers are plotted in a window (Button ‘Plot diploT-Overview with fixed feature A’), or a file in PDF-format will be created with each selected marker as basis marker and all others as associated markers (Button ‘Plot diploT-Overview total’). In addition, diGraph-Overviews can be created which show the chosen statistical method as overlapped line graphs (App. B, Fig. S6).

3.5.3 The window frame in the tab *n-triploTs*

The window frame in the tab *n-triploTs* has a similar structure to the one in the tab *n-diploTs* in order to get the user familiarized with this tool (Fig. 3.15). First, drop-down menus are arranged one by one for bin sizes, minimum number of cells (minCount) and basis markers A and B and associated marker C. Large buttons ‘Plot triploT’ and ‘Plot histograms’ follow which create a triploT and histograms of the set markers, respectively. The two subframes labeled with ‘Plot options’ and ‘Options’ also have similar checking choices.



binSize: 0.2
minCount: 5
Feature A: CD90
Feature B: CD86
Feature C: CD44

Plot options
Range x-axis (min/max): 2 12
Range y-axis (min/max): 2 12
☑ Dynamic MSI range
Min MSI: 2.4
Max MSI: 6.5
☑ Grid ☑ Percentages ☑ Legend ☐ Date

Calculation options
Transformation: ● asinh ○ none
Statistical method: ● MSI ○ MSI(+)
○ density ○ frequency
○ SD ○ SEM
○ RSEM
☑ Trim first ☐ Bins <minCount

Set rectangle
☐ Manual rectangle
x1: y1:
x2: y2:
☐ Auto add info
☐ Gate data

Graphics
Rows: 1
Columns: 4

Data info
# of total cells: 22384
# of selected cells: 22384 (100 %)

Figure 3.15: GUI of the window frame *n-triploTs*.

This tab also provides several multi-plotting options. One function takes both basis markers A and B from the section above and puts the selected markers from the left frame as associated marker C. Another one takes only the basis marker A from the section above and puts the selected markers from the left frame as basis marker B and associated marker C, respectively. Button ‘Plot triploT-Overview’ takes all selected markers and creates triploTs with any possible combination of three markers. With these plot options, a PDF file is created in the desired directory path. Furthermore, if the cutoffs are appropriately set and additional percentage information is provided on the triploTs, a table in csv-format is created, which include one row for each plotted triploT. The row comprises the information of the marker combination, the cutoffs of the markers and the percentage values indicated in black, red, green or blue.

Subframe ‘Graphics’ helps with the size of the plotting frame. A window with the size of the chosen number of rows and columns is invoked with several slots of empty plots, and triploTs called with the function ‘Plot triploT’ will be plotted in these slots. The active plotting frame can then be saved in a PDF format. The subframe ‘Set rectangle’ on the right provides the manual gating option. It can be used as a detailed triploT where only cells inside the rectangle are calculated and plotted. It can also be used as a further gating strategy. For this purpose, the option ‘Gate data’ needs to be checked.

PRI-ana is developed by continuous exchange with several group members of AG Baumgrass who are mainly not familiarized with programming and who will very likely use this tool the most. The GUI is designed for an intuitive and day-to-day usage. Several error pop-up windows are implemented to notice the users if parameters are falsely set. To use this tool, three *R* packages need to be installed, while one of them is already installed when using *PRI-base*. This tool can be used on different operating systems (Unix, Windows, Mac OS). *PRI-ana* is the wrapped and user-friendly implementation of the majority of this study’s presented workflow.

4 Discussion

A raw mass cytometric sample has c millions of cells as rows and up to $m = 50$ protein markers as columns, with signal intensities (SIs) as values. The dimension of this type of data is long and skinny, and one could argue, that this data is simple to analyze, but there are several aspects to consider. Firstly, every cell needs to be regarded individually, and some cells have similar SIs of the same markers. Furthermore, categorizing the SIs in at least positive or negative would lead to $\geq 2^m$ marker combinations in each cell. In the end, a central component in biology is the few percentages of cells which make the difference between two groups, for example, healthy and diseased, but not all measured proteins are obliged to be relevant. With these points, the data is highly complex and suffers from the curse of dimensionality, thus it is difficult to inspect. Many tools have been developed to discover the discriminative group of cells, the so-called subpopulation, but they are facing the problems of either reproducibility, comprehensible interpretation, high computational complexity or suitability for group comparisons.

The goal of this dissertation is firstly to demonstrate that the innovative process of obtaining engineered variables from measured cytometric data contain more concise information, and its visualization results in a more continuous and combinatorial view compared to conventional and state-of-the-art data analysis strategies; and secondly, to show that this study's variable ranking (VR) table reduces the examination time drastically by serving as guidance to identify meaningful subpopulations which are discriminative between two groups. The herein proposed workflow *pattern recognition of immune cells* (*PRI*) is applied on an already published mass cytometric data set from *Spitzer et al.* They used female mice with breast cancer and treated them in four different ways, in which two fall into effective (eff) and the other two in ineffective (ineff) treatment. The results of the re-analysis with *PRI* were presented and evaluated, and subsequently compared to the outcome of *Spitzer et al.*, the 'gold-standard' visualization tool *viSNE* and classification tool *Citrus* to show the added value of the new approach.

In the following, all three main steps from this study's workflow are addressed. Particularly, steps 4, 5 and 6 of the data preparation and storage part are subject for discussion. Furthermore, the engineered variables, diploTs and triploTs, and the attributes of the latter are intensively discussed and compared to conventional and state-of-the-art approaches. Next, the elastic-net regularized logistic regression (erLR) used as an embedded

VR tool is addressed. Finally, the examination with the VR table and the resulting subpopulations are reviewed in terms of biological knowledge and in comparison to the approaches from *Spitzer et al.* and *Citrus*.

4.1 Data preparation and quality control

Due to the novel invention of mass cytometry and the fast development of flow cytometers in the last decade, parallel measurements of protein markers has risen from a few markers to up to 50 protein markers. Thus, manual inspection of a series of conventional contour plots has become unfeasible. However, computational cytometric data analysis is still in the early stages of development. Many approaches and bioinformatics tools are introduced to identify subpopulations, but data preparation is often of secondary importance. This study's workflow includes six steps for necessary data preparation and manipulation. In the following, the steps are addressed, whose configurations have a potentially big impact on the outcome of this approach.

4.1.1 Effects of data alteration

Normalization

Recent mass cytometric measurements have included beads with specific metals [23]. These beads should function similarly as house-keeping targets, analogous to sequencing studies. *Spitzer et al.* incorporated this technique and applied a bead normalization. However, the data set still shows noticeable variation. Thus, several single channel normalization methods were herein tested. Method 'range' normalizes the ranges and method 'warp' normalizes by the peaks of the densities of the samples in each group. Both methods work well on many channels, where variations in the groups have been diminished. However, a few channels show distortions after normalization (App. B, Fig. S7). These distortions show different behaviour and can lead to artificially skewed results. As a consequence, no further normalization is herein deployed on this example data set.

Co-factor in *arcsinh* transformation

The bigger the co-factor the more skewed the density curve, and the smaller the range of the values. The opposite effect occurs with co-factors $cof < 1$: the curve and the range widens. A common co-factor $cof = 5$ is applied to several mass cytometric studies [10, 14, 54, 62], but after manual inspection of the density plots, another co-factor is herein chosen. The triploTs and the corresponding density plots in Figure S17 (App. B) show the effect of the usage of different co-factors. With the co-factor $cof = 0.1$, the range has almost doubled and the bi-modal distribution is enhanced. The cell populations are better separated, as seen with CD90 and CD44. A shoulder at CD27 is also better captured with the smaller co-factor. However, caution is needed, because applying a co-factor which is too small can lead to formation of an artificial cell population. Among others, CD86 shows a bi-modal curve development with the applied co-factor, which has been not identified before. The examination with the guidance of biological expert knowledge leads to the assumption that this population can be acknowledged as biologically meaningful, and would get lost with the common co-factor setting. This emphasizes the fact, that biological and computational experts need to work hand in hand to develop an optimized and meaningful analysis workflow.

4.1.2 Introduction to compensation on mass cytometric data

Flow cytometry is a well-established and older technique than mass cytometry. The main difference is the protein labeling with fluorochromes instead of metal isotopes. This results to a smaller amount of measured protein marker, because of its well known problem of numerous interference and overlaps in the light spectrum of different fluorescent dyes [63]. This means that some SIs of the protein marker can be artificial signals, or signals from other fluorescent dyes. There are already procedures that were introduced to overcome this problem. For one, it is compensation: the process of correcting the spillover from one primary signal in each secondary channel it is measured in. Another step is to use fluorescence minus one (FMO) controls to assure the proper cutoff setting from a positive population. These procedures are common practice and are also implemented in several computational approaches [57, 64, 65].

With the newer technology mass cytometry, signal overlap is supposed to be minimal, since the cytometer detects discrete isotope peaks without relevant overlap (Sec. 2.1) [66]. However, minor spillover effects from other metal channels are still captured in this example mass cytometric data set. This can occur due to isotopic impurities or unspecific binding in the antibody panel. In addition, the panel design for mass cytometry is dependent upon the choice of the metal tag, as there are less sensitive channels at the extreme ends of the mass range than in the central range [67]. Recently, authors have suggested the deployment of panel optimization procedures and single stain experiments with each antibody used in the experiment, and apply tools for systematic correction of these spillovers, analoguously to FMOs in flow cytometric investigations [67, 68].

The varying peak sizes found in the lower range of each channel within the re-analyzed samples from *Spitzer et al.* can be explained by these spillover effects (App. B, Fig. S8). Since no single stain experiments have been realized, a systematic correction is not possible. One approach to reduce the impact of these effects is deployed in Filter II (Sec. 3.2.3). There, the first bin rows and columns [0,0.2) of each basis marker are not regarded for visualization, nor for the VR algorithm. For prospective re-analysis of mass cytometric data sets, it is advisable to first do descriptive explorations with, e.g. overlapped density and multi-dimensional scaling (MDS) plots as in Section 3.1, to capture the quality of the data set and eventually apply specific preparation steps to reduce noise effects, for example, removing outliers or Filter II.

4.1.3 Descriptive statistics to identify outlier samples

The medians from all markers from each sample are used as input in MDS and hierarchical clustering. Since the sample size is so small, every sample is of high importance. Figures 3.1a and 3.1b show that sample *untr_ d3_ Bl1* clusters within the other treatment group, but tendencies and similar patterns in the triploTs might still be visible. It is possible that some subpopulations are missing in sample *untr_ d3_ Bl1* but one could argue that the inclusion of this sample could also emphasize some other triploT sections as variables listed further down in the VR table. However, sample *untr_ d3_ Bl1* has the smallest cell count of the data set, and has different signal medians compared to the other samples of the same treatment. These properties, coupled with the setting of the minimum number of cells within the plots, can lead to an artificially smaller or shifted range of the (colored) bins. Since the variables for the VR are based on equal distributions into four triploT sections, the sections of that sample would include different information than the other samples of the same treatment (App. B, Fig. S11). Furthermore, it is possible that the subpopulation of interest is not present, or differently expressed, in a sample of low quality, resulting in an artificial outlier variable. Therefore, samples of similarly low quality, and also having a low amount of cells, should not be included in further studies with the proposed workflow.

4.2 Novel engineered variables based on a bin scaffold

The proposed workflow of this study involves variable engineering (VE), whose resulting variables serve both as visualization in manual inspection as well as input variables in VR. This investigation of cytometric data is based on subgrouping (binning) of the cells of similar characteristics. The similarity of the cells is specified by the SIs of one or two specific protein markers. In the following, both *PRI* variables are discussed in detail.

4.2.1 Added values compared to conventional approaches

Commonly, a sample undergoes a series of manual gates in bi-axial contour plots, and the amount of cells or the mean signal intensities (MSIs) of protein markers of interest are collected at the end. Then, bar or pie charts of either frequencies or MSIs of these cells in different combinations are compared to another subgroup, or to the total cells (App. B, Fig. S15). This conventional process is highly subjective, hypothesis-driven, time-consuming, and gets more complex with the increasing amount of measured proteins. Last but not least, other potentially relevant subpopulations are likely to be overlooked.

Stacked diploTs as a compact visualization technique

A diploT is a novel illustration method for cytometric data. It depicts statistical measurements of a protein marker as an associated marker compared to the whole range of one basis marker which is distributed in bins. Displaying several protein markers compared to one basis marker on stacked diploTs results in a compact and easily explorable visualization (Fig. 3.3). Appendix B, Figure S6 shows another visualization example similar to diploTs and are named diGraphs, which have the same information as in the stacked diploTs, but require much more space without a clear overview. The expression correlations between markers are better captured on the basis of the color gradient in the diploTs rather than on the curve shape in the diGraphs. Thus, assumptions can be made more rapidly with the compact stacked diploTs. Furthermore, the semi-continuous and reproducible display supports not only the comparison between groups, but also the comparison of progression studies. Shifts of SIs and frequencies can be recognized without difficulties. However, if the tendencies of some markers do not differ strongly among them, underlying small subpopulations cannot be seen clearly. This compactness and simplicity can lead to a loss of information, particularly if heterogeneous populations are examined or unsuitable markers were chosen. Nonetheless, the insightful illustration is likely also accessible to a broader community, compared with conventional plots. Due to the semi-continuous visualization of at least one associate marker, existing correlative patterns are observed in a straightforward manner. This is accomplished without any further time-consuming and biased gating strategies, and results in a fast general overview of the associated markers' inter-relationships with respect to the basis marker.

TriploTs for intuitive capturing of inter-correlations

With triploTs, one additional dimension is added in the vertical direction. The similarity of the cells is therefore specified by the SIs of two basis markers and is plotted as a bi-axial plot. This plot is partitioned in quadrant bins of equal sizes resulting in a bin scaffold. The cells in each bin are then aggregated. This aggregation and the setting of the minimum number of cells allows for numerous stable calculations of statistical methods such as the MSI of a third parameter within each bin. The range of the statistical attributes are then displayed in pseudo-colors.

In fact, the visualization with these triploTs provides many advantages compared to conventional diagrams. The binning of conventional bi-axial plots supports the plotting of many statistical methods of (at least) one associated marker in the same plot area (Fig. 3.4). Several statistical methods as bin properties have been successfully applied to identify and characterize novel subpopulations [69]. Furthermore, each bin takes a group of cells into consideration, rather than the information of an individual cell. Along with the setting of the minimum number of cells, the bin information is more reliable and inter-correlations between the basis markers (A,B) and the associated marker (C) can be captured without much effort. The triploTs applied on the example cytometry data set *Spitzer et al.* showed, that a clear highlighting was rapidly recognized in the upper left and upper right bin region (Fig. 3.4c), and the cutoffs for CD90$^{\text{high}}$ and CD44$^{\text{high}}$ could be set in a straightforward manner which resulted in a clear division of the bin scaffold into regions that differ in CD27 expression. This visualization strategy is particularly beneficial, for example, for capturing clinical relevant changes in compositions of population.

Further auxiliary information is displayed with several percentage calculations. If the cutoffs for both basis markers (A,B) are set, the cell rate in the quadrants in relation to the total amount of the cells is presented in black. In addition, if the cutoff for the associated marker (C) is set, the cell rate of the marker producing cells in the quadrants can also be displayed without any further gating strategies for each quadrant. The cell rate of C^+ in each quadrant is presented either in relation to the amount of cells in its respective quadrant (red) or in relation to the total amount of cells (green). Consequently, the percentages in red and green show the cell rates of the three marker combination (A-B-C), which have not yet been applied in any plots for cytometric investigations. The added value of this information is demonstrated with the triploTs of CD90-CD44-MSI(Foxp3/Tbet/CD27/CD62L/CD69) in Figure 3.6. It is apparent in bin region I ($CD90^+$,$CD44^-$), that in eff treatment there are no colored bins, but in ineff treatment there are. The corresponding percentages in black (eff=0.5% and ineff=2.7%) confirm this. However, an increase in bin count of high MSIs, as highlighted in bin region II ($CD90^+$,$CD44^+$) for Tbet and CD69, and in bin region III ($CD90^-$,$CD44^+$) for Foxp3 in eff treatment, also presumes a higher amount of producing cells of the associated markers, but does not show how much it is increased. With the information of the percentages in red and green, the increase is numerical. E.g. an increase of the presented associated markers compared to the total amount of cells (green) by a factor of roughly three can be recognized in bin region II (difference: 3.44 ± 0.48 s.e.m.) and III (difference: 7.20 ± 1.11 s.e.m.) in eff treatment. To obtain that information with conventional contour plots, an additional gating step for each percentage number in each quadrant would be necessary.

4.2.2 TriploT functionalities for visual pattern perception

This section aims at exploring the possibilities and possible limitations of the triploTs in greater detail. It is investigated whether patterns can also be derived in terms of visual examination and the performance and interpretability of such patterns will be compared to those based on state-of-the-art visualization techniques.

Fixed width bins are better interpretable than varying bin widths

The engineered *PRI* variables have fixed-width bins with a varying number of cells, rather than varying bin widths containing a fixed number of cells, throughout the re-analysis. This visualization was chosen, because the cells tend to accumulate near the center of the data but flatten out fast near the boundary of the data (contour plots in Figure 3.2 and 3.4). Moreover, it is more difficult to interpret the data with varying bin widths, for example, with percentile-based bins which are deployed in [70, 71]. The outcome of percentile-based bins is highly affected by the pre-chosen number of bins, thus the number of percentiles. In contrast, the plots with fixed bin widths can have different bin sizes, but the pattern is similar and can still be recognized, which allows an enhanced comparability between samples and groups (App. B, Fig. S16). In addition, the comparability is further supported by standardization in the sample acquisition, so that sample outcomes are more robust and variations are minimal. Appendix B, Figure S11 shows that even the omitted sample *untr_ d3_ BL3* has a similar triploT pattern compared to the other samples in its group.

TriploTs require many cells

The omitted sample *untr_ d3_ BL3* may resemble the other samples in the eff treatment. However, that sample has less than one tenth of the biggest sample. A cell count which is too low can lead to an artificial shift of the range and the mean values, and therefore also a shift of the corresponding cell rates which are displayed in different colors representing percentage values (Sec. 3.1).

The bin calculation methods standard deviation (SD) and relative standard error of the mean (RSEM) are two adequate options for display. They aid in the configuration of bin size and minimum number of cells, and in examination of the variation in the bins. As a consequence, they present the stability of the bin information in, for example, the MSI, as similarly shown with the diploTs (Sec. 3.2.1). Furthermore, these displays are also suitable to check if the cells in some bins are more heterogeneous than the cells in other bins. If there is a bin region with high variation of the associated marker, this can be a hint, that there is either no specific or more than one subpopulation.

TriploTs support pseudo-multi-parametric viewing

With the triploT's bin scaffold, it is possible to study statistical properties of small populations per bin. These properties are visualized in pseudo-colors, and as a result, correlating patterns can be observed. If different associated markers are plotted next to each other, a pseudo-multi-parametric view is achieved (Fig. 3.6), which can be analogously considered to the stacked diploTs. Due to the easily interpretable approach, a lot of meaningful information could already be accessed through visual examination.

The heterogeneity of the samples within the groups can be seen in the uni-variate display (App B, Fig. S2 and S11a). Nevertheless, *PRI* approaches the data in the combinatorial matter as triploTs (App B, Fig. S11b). The varying percentages of the upper right bin region in black lead to non-significant information (of two-marker combinations A-B) as seen in the box plots (App B, Fig. S11c). Similar conclusions can be made with conventional manual gatings and bar plotting (App B, Fig. S15). However, the triploT patterns are different between eff and ineff treatment. In the eff group, a concentrated highlighting can be recognized. To obtain the percentage values displayed in green and red with conventional approaches, several further manual gating steps would be necessary. The significance of the colored percentages in the box plots confirms that, even without additional gating steps, the pattern of the triploTs can already indicate interesting subpopulations which are discriminative between two or more groups. Another advantage is the semi-continuous visualization. Some cells can develop in a hierarchical structure, for example, naïve CD4$^+$ T cells to Th1, Th2 and Treg cells and further (Fig. 2.9). As a consequence, the transition between the cells results in a continuum rather than in distinct clusters [4]. With *PRI*, these transition states can be visually captured.

Further narrow down of the subpopulation from *Spitzer et al.* with triploTs

The aforementioned benefits of the examination with triploTs are demonstrated by re-analyzing an example biological data set (Sec. 3.2.2). *Spitzer et al.* identified a T cell subpopulation which is described by CD44$^+$CD69$^+$CD62L$^-$CD27^{low}CD90$^+$T-bet$^+$ (Fig. 3.5). This phenotype is most likely comparable to the bin region II, which is characterized as Th1 cells with T-bet^{high} (Fig. 3.6). Nonetheless, the characteristics of the highlighted subpopulation are refined with triploTs. CD69 is not positive but has the highest MSI present, and CD62L is positive in this region, and not negative as proposed. Table 4.1 shows the summarized comparison of the residual markers' characteristics. Another finding is the two to three fold increase of cell rates in the bin regions II and III from eff treatment compared to *ineff* treatment. This is seen in greater abundance of percentages in black and respective to the associated markers in green. It is also evident that CD27^{high} cells are only located in these regions. Thus CD27^{low}, which is proposed by *Spitzer et al.*, is not discriminative, but CD27^{high} is seen in the CD44$^+$ area. Furthermore, CD69$^+$ and CD62L$^-$ are confirmed in the bin region III for Treg cells, but solely CD62L$^-$ is discriminative between eff and ineff treatment.

Table 4.1: Summary of statements from *Spitzer et al.* compared to this study's manual inspection with triploTs. TriploT's bin regions II and III of CD90-CD44-MSI regarding markers CD62L, CD69 and CD27. Discordant statements are colored in red.

	CD62L	CD69	CD27
Spitzer et al.	-	+	low
triploT - bin region II (Th1)	med	high	+
triploT - bin region III (Treg)	-	+	high

In conclusion, the manual inspection with triploTs confirms the result of *Spitzer et al.* that CD90 and CD44 are discriminative markers in general. The amount of T-bet$^+$ cells is also increased in eff treatment. However, with a trained eye, it is evident that CD90$^+$ (grey dashed lines) is not discriminative, but CD90^{high} (grey continuous line) is. In the end, with CD90 and CD44 as basis markers, three bin regions are separately characterized. These bin regions refer to naïve (region I), Th1 (region II), and Treg cells (region III) (Fig. 2.9).

4.2.3 TriploTs compared to state-of-the-art visualization techniques

Color Maps has a similar approach

The providers of the broadly used cytometric analysis tool *FlowJo* also realized that the color-coding of a third parameter can significantly facilitate the investigation of cytometric data [57]. They have simultaneously developed a similar graphical display which is called *Color Maps* (App. C, Fig. S18). It also uses fixed-width bins (of unknown size) to display a third marker in a color-coded manner. Next to MSIs of neighboring groups, *Color Maps* also displays SIs from scattered cells. This means that all bins are shown with at least one cell involved. This preserves some single cell information, but actually distorts the pattern, since information of possible outlier cells is presented in combination with information about groups of cells. As a result, the similar color-coding does not explain the subpopulations very well. Furthermore, there is no setting of bin size and minimum number of cells involved. Thus they display only the statistical attributes in the bins.

The triploTs go beyond all this. Firstly, they allow to manually or automatically set cutoffs for SIs for positive and negative populations without further gating. As a consequence, frequencies and mean signal intensity of positive cells (MSI+) and other statistical information of at least one associated marker can be visualized and auxiliary information in percentages are additionally provided. Furthermore, the optional configuration of the minimum number of cells has the advantage of displaying more robust and reliable statistics on the bins, and in addition, it supports diminishing the effect of outliers. This results in a clearer illustration, and subpopulations are captured more easily (Fig. 3.6). However, increasing the minimum number of cells or decreasing the bin sizes leads to the requirement that many cells need to be measured. In practice, at least 10,000 cells are necessary for the display with triploTs and 20,000 or more cells are needed to comfortably work with percentage values of samples within same groups in general. As also seen with sample *untr_ d3_ Bl1*, 5,000 cells are generally insufficient (App. B, Fig. S11).

TriploTs compared to the dimension reduction method *viSNE*

In general, the interpretation of the *viSNE* maps is cumbersome, since the algorithm ignores the global structure (Fig. 3.12 and App. C, S19). Cells with similar characteristics are displayed closely together, but cells from another random location can also have these characteristics. In turn, neighboring cells do not have to be similar. As a consequence, this visualization technique bears the risk of over-interpretation in structures and distances. Moreover, the transformation of the data using suitable coordinate axes is convenient for cluster detection, but this makes it more difficult to reach biological inferences from these plots compared to bi-axial marker plots [72]. The cell populations may be properly clustered, but a *viSNE* map contains no information on the SI interdependence of the markers. As a result, these clusters need to be subsequently visualized in another manner. Conventional overlapped density plots, bar plots and heat maps are commonly used subsequently to see significant differences in groups [12, 24, 73]. The examination with *PRI*, on the other hand, has SIs from two basis markers (A,B) as axes and is bin-based. This allows for an intuitive and reproducible pseudo-multi-parametric display based on bin patterns, where changes in expression of the associated marker (C) are tangible and interpretable.

The subregions found with manual inspection could not be directly identified with *viSNE*. However, the further narrowed down characteristics of the Th1 subpopulation, which resulted by the guidance of the VR table, enabled its recognition within the *viSNE* map (Sec. 3.4.4). One prominent reason why only the Th1 subpopulation among the three bin regions was found is due to the subsampling step in *viSNE*, which randomly selects a certain amount of cells (default=5,000) within each sample. A stochastic exclusion of some or many cells of this subgroup within the samples can occur, which can result in a cell count which is too low to be visually identifiable on the *viSNE* map.

Another limitation of *viSNE* is that the complexity of this algorithm is high ($\mathcal{O}(c^2 \cdot m)$) with c number of cells and m number of markers. Due to the quadratic relation to the number of cells it is not feasible to apply *viSNE* to data sets that contain more than 100,000 cells[4]. To overcome this issue, subsampling methods are commonly deployed beforehand and also advised from the authors [12]. This in turn is also problematic since many, or often the majority, of cells are not considered which can lead to a loss of rare and small but important cell subgroups. And due to the arbitrary choice of the subsampling and starting point, the algorithm creates different maps with different subsamples (App. C, Fig. S19). Hence, the reproducibility is also poor while using *viSNE*. In particular, if samples are added or removed, the maps might drastically change, because the subsampled data contains another mixture of different cells from each sample.

4.2.4 TriploT section values as basis in variable ranking

This study demonstrates the first attempt to use the triploT information as basis for a regression algorithm. The approach uses properties of equally sized triploT sections mimicking the manual cutoff setting without any bias. As a result, four conjunctive section values from every three-parametric combinatorial triploT without duplicate axes marker combinations are collected. With 31 protein markers which are potentially important in differentiating the treatments, the amount of sections is at $p' = 53,940$. After two further filtering steps (Filter III and IV in Sec. 3.2.3), the triploT section count is drastically reduced to $p'' = 2,523$.

After fruitless deployment of several common descriptive statistic features (mean, median, standard deviation and variance) as section properties in the VR step, the relative and absolute range of the MSIs were applied, from which the latter resulted in the most differentiating triploT sections in the top 10 ranking list after using these properties as variables in this work's embedded function for VR. In fact, this resulted in many useful

[4] A *viSNE* run with 50,000 cells took more than 18 hours.

combinations of markers which has a bin area with localized cell subgroups (Fig. 3.10). Hence, the results depend on the way the sections are defined.

In the used mass cytometric data set, there are no similar controls such as FMOs for flow cytometric data sets. Due to these FMO controls, compensation issues can be resolved as briefly mentioned in Section 4.1.2 and the question can be clarified as to which position to set the threshold between positive and negative population. The option to deploy manual cutoffs or automatic cutoffs (e.g. with the implemented function in App. D, Lst. 7) for triploT sectioning is therefore strongly advised for flow cytometric data. Herein, some cutoffs are easier to set than others. CD45^{+} or CD19^{-}CD3^{+} and CD4^{+}CD8^{-}, as seen in the last set gates of Appendix A, Figure S1, are clearly separated subgroups which can be gated in a few seconds. Other cutoffs, however, are more difficult to define. This is especially true when there are activation markers within a continuum, or low levels of positivity, as with the protein marker Foxp3 in regulatory T cells. Nevertheless, setting manual cutoffs is subjective and time-consuming. For this reason, the use of equal division of the sections is preferred in this study.

Equal sectioning is fast and, most importantly, the output of the VR workflow is then easier to comprehend. Moreover, it is expandable to a matrix of 3 × 3, 4 × 4 or more sections. The area of each section will be reduced, which potentially leads to more concentrated subpopulations. Nevertheless, a larger sectioning will increase the section count as well. The resulting amount of variables will also face the curse of dimensionality. This is counteractive to the current $p >> N$ problem with the example data set. To overcome this issue, another idea has been developed using matrices of higher section counts in artificial neural networks, which in fact comprises multiple layers of logistic regression (LR) models and is described in the following section.

4.2.5 TriploT matrices as input in deep learning

Deep learning is a subdomain of machine learning which uses artificial neural networks with numerous hidden layers between the input and the output layer. Convolutional neural network (CNN) is a class of deep neural networks [74]. It is originally designed to process two-dimensional structures and is consequently one of the most popular neural network architectures which are broadly used for image processing. This approach is famous for the high performance of the supervised learning on the public image data sets MNIST, CIFAR-10 and CIFAR-100 [75, 76]. In particular, it is useful for large data sets, if (non-linear) inter-correlations are unknown. That is where the triploT variables come into play. The idea is actually not to section the bin area in a matrix of 2 × 2 or higher dimensions. In fact, the already created (triploT's) bin matrices, which are used for inspection and have the bin properties bin size of $arcsinh(x) = 0.2$ and minimum number of cells= 5, can be used as input analogous to image studies using pixel intensities. Instead of using the pseudo-color-visualization, the bin information matrices with the respective phenotype as label are directly used as input layer for CNN, hence skipping the converting step from image to matrix. Furthermore, by including the upstream data preparation, filtering and VR step, only the most informative and discriminative triploTs are selected to construct the model. This will certainly increase the prediction power of the model.

CNN was first used in the field of cytometry by the tool *CellCnn* (Sec. 2.2.2) [24]. Each line is one cell and its patch vector of high and low intensities is individually assigned. These vectors are used as input in a CNN with three layers, and are labeled with the associated disease status or survival information. After training the model, the trained filter weights are used for variable selection (VS). The weights which correspond to the molecular profiles of relevant cell subgroups are then matched with the individual patch vector of the cells. The filtered cells are then compared to the residual cells and are characterized in more detail with conventional approaches such as density and bar plots.

The CNN approach herein, however, uses bin information of grouped cells instead of using single cell information. An advantage is, that the input size is not dependent on the amount of cells (in the samples) but the amount of protein markers and samples. That is to say, that the cell count can grow to hundreds of millions but still does not affect the input size. This is especially useful with big data sets, but the limitation here is again the minimum cell count in each sample as mentioned in Section 4.2.3. Another advantage is, that *PRI* scales the bin properties from 0 to 9. This scale keeps tendencies tangible instead of the reduction to low and high. In the end, the main difference to *CellCnn* is, that this study's CNN approach is to build a classification model rather than to use CNN for VS.

In conclusion, the engineered *PRI* features paves the way for reproducible and automatable cytometric analyses. In addition, they provide a statistically robust pattern map for visual inspections and serve as input for pattern recognition in deep learning approach.

4.3 Regularized logistic regression as an embedded variable ranking tool

Due to the deployment of the filters in Section 3.2.3 the total amount of triploT section values is reduced from $p = 109,668$ to $p'' = 2,523$, but with $N = 12$ samples, one still faces the curse of dimensionality. However, practical experience suggests that, in some cases, it is still possible to make good statistical inferences and predictions. The intuition behind these approaches is a form of simplicity, namely the sparsity principle (Sec. 2.4.2) [34, 37]. Only a small fraction of the total variables explains a very large proportion of the variation in the data and therefore, this fraction plays an important role in discriminating between groups. This is where VS comes into play. It facilitates data understanding and opposes the curse of dimensionality by improving the performance of models.

There are three techniques of VS: filters, wrappers and embedded methods (Sec. 2.3). Filters are a variable subset selection alone without choosing a classifier and are used in the preprocessing step. It is argued that filters are the fastest technique, because they are not based on any learning algorithm. However, they do not capture the combinatorics of the variables [27, 77]. Wrappers typically use a predefined model's learning performance to evaluate variable relevance. They focus on finding a subset that is useful to build a good model [37, p.658]. Traditional approaches use exhaustive search which repeatedly chooses a subset of variables and then evaluate the performance, but this is computationally intensive [29, 78]. Furthermore, variables for building a model are not necessarily relevant variables in the biological context. And contrarily, correlative variables are mainly excluded due to their redundant information [29], but these correlations might be important functions in biology. Correlative variables are also necessary for the highlighting of the triploT bin areas in *PRI*. Embedded methods, on the other hand, deploy VS in the process of training. They couple the classification algorithm with the parameter estimation and are usually optimized with a single objective function like error rates [29, 79]. They include the benefits of both wrapper and filter methods: an embedding VS with model learning are more efficient compared to wrappers [78]. An embedded method is therefore chosen for this study's VR workflow.

4.3.1 Logistic regression fits to the data complexity

The underlying function in the data set *Spitzer et al.* is unknown and possibly not linear. It could be almost linear, and require some minor transformation of the input data to work correctly. It could be also non-linear in which case the assumption to use a LR is wrong and the approach will produce poor results. Simple models might not be able to capture the relevant inputs which are driving the variable of interest. Using a LR to describe the data set means, that this model is highly constrained by that form, which may turn out to be unsuitable for a particular application [33, p.68]. Instead of using this model type, which is a parametric distributed, generalized linear function, the non-parametric methods k-nearest neighbors or decision trees could be applied for example [80]. Non-parametric methods, also referred to as distribution-free methods, work particularly well when there is no prior knowledge available, and if there is no time or intention to adjust or optimize the model with, for example, VS. However, these complex algorithms need a large volume of data, because they are prone to over-fitting with small data sets. Moreover, many complex algorithms cannot be applied as an embedded VS tool.

There is not a so-called 'best method' for a specific problem setting. In this study, using the engineered triploT section values as input variables, the dimensionality hence the complexity has been reduced. The results from this approach demonstrated that this model complexity suffices to extract meaningful variables from the biological mass cytometric data set obtained by *Spitzer et al.* . Summarized, there are several advantages using this type of model. Firstly, these models are very fast to learn from data. Hence, they do not require as much training data and can work well even if the fit to the data is not perfect [37]. Furthermore, LR models are easy to understand, and results have a simple and intuitive meaning represented by coefficients that are either negative for a decrease, or positive for an increase within the variables according to the category. One only needs to keep in mind the exponential attribute of the coefficients (Sec. 2.4.1). And last but not least, this model form enables its usage as an embedded method for the deployed VR workflow, to support and speed up the inspection of the novel engineered variables by providing a list of most differentiating three parametric marker combinations.

4.3.2 Elastic-net regularization deals with multi-collinearity

With the assumption of sparsity, the total engineered variable set from example data set *Spitzer et al.* has many non-informative sections. In general, using erLR has a good trade-off between the goodness-of-fit and the model complexity. The former is aimed to be maximized and for the latter, the number of explanatory variables is desired to be minimized [40]. The model form is rather simple. LR creates a generalized linear model (GLM) with all variables supplied to the regression. A regularization to the model deploys penalties to the function and allows the favored shrinkage of the explanatory variables.

The characteristics of an individual cell, hence the inter-relationship between the protein expressions of the cells are in some extent collinear. *Citrus* is considered to be the state-of-the-art classification tool. It uses $L1$ regularization on SI medians in the cell clusters (Sec. 2.2.2). The advantage of the $L1$ regularization, is that it shrinks the coefficients to be exactly zero if the variables have minor contribution to the model. It produces a more modest model that incorporates only a reduced set of the variables as predictors. However, using this regularization, the variables are restricted to the number of samples N. This might work on cluster level, but is not suited for *PRI* variables. Another issue with applying the $L1$-norm is that they tend to pick one variable among the correlated ones and put all the (coefficient) weight on it [37, 40]. This is problematic, when it comes to the biological meaning. If there are multiple proteins highly expressed but only one is chosen by the $L1$-regularization, the resulting variables in the model do not reflect the importance of the proteins in reality. Therefore, the biological interpretation is cumbersome.

The properties of the three-parametric combinatorial triploT sections aggregate information of the bins. This means, the values do not include the multi-collinearity on the single cell level. However, with inclusion of information of the three-parametric marker combinations and the large amount of variables, the variables are possibly still correlated. $L2$ regularization selects all of the correlated variables, and shrink their coefficients towards each other. The tendency is for all of those coefficients to be equal. Nevertheless, it does not shrink the amount of variables. This is where elastic-net comes to practise [59]. Due to the combination of both $L1$ and $L2$ penalties, elastic-net is powerful when there are correlations among the explanatory variables, and especially useful when a sparse solution is either necessary or desirable [34, 40]. To identify the influence degree of the penalties, cross validation (CV) is deployed with different $L1/L2$ influences, as conveyed by the constraint α. That is why α is an important factor, which is adjustable according to the nature of the data to improve the model fitting.

4.3.3 Cross-validation to assess the model fitness

In this study, the 3-fold CV is applied in two separate cycles (Fig. 3.8). The first cycle is aimed at identifying the degree to which the two techniques $L1$ ($\alpha = 0$) or $L2$ ($\alpha = 1$) are working best on this particular example data set. Appendix B, Figure S9 illustrates an example dot plot to obtain the best α for each iteration of a CV run. Other iterations have similar outcomes, only varying in the maximum value of min(deviance). Interestingly, either $\alpha = 0$, $\alpha = 1$, or both perform worst in these iterations. This supports the hypothesis that the elastic-net regularization is a better choice for this VR approach and this data set than the classical $L1$ and $L2$ regularizations. The second cycle involves determining λ_{1se} only and is used for variable collection after certain criteria are met. Using this construct, there are actually three hyper-parameters to adjust: k, α and λ_{1se}, however the first parameter is dependent on the sample size N and the latter is usually defined by the error measurement in the CV.

Root mean square error (RMSE)[5] is deployed as the performance metric of the resulted LR model, because of the following three reasons: i) the sample size is low, ii) the sample distribution into training and test set is subsequently unbalanced, and iii) the test set consists of only one sample from each treatment. Therefore, any metric from the conventional confusion matrix is not suitable in this scenario, since the test set consists of two samples and some metrics would be just zero or not available (NA), depending on the constellation in the matrix. The accuracy, for instance, of the resulting model in all 500 CV achieved 100%, meaning both samples in the test set were predicted correctly. However, this metric is not conclusive with only two samples in the test set. Furthermore, models with 100% performance tend to over-fit. Thus, the variable set might not be optimal or the model performance measurement is not suited. RMSE is a less biased estimate of error variance, since the division by n removes the effect for different sample sizes. Additionally, the square root ensures the same scaling and unit as the prediction variable. The amount of models which achieved a difference of RMSE< 0.5 was only $\sim 70\%$ (357 out of 500 runs). It is therefore a good evaluation metric for this purpose.

The particular strength of this VR tool is that it takes the advantages of: i) LR as an effective and less computationally intensive model structure for less data and for variable shrinkage in combination with ii) elastic-net regularization for handling with collinear variables, and iii) nested CV and error measurements for model optimization. After applying this VR approach, the resulting ranking list is discussed in comparison to cluster approaches in the following section.

[5] Not to confuse with RSEM, the relative standard error of the mean.

4.4 Variable ranking table provides discriminant three marker combinations

The focus of this study is to develop an upstream filter, firstly, to guide the inspection of most relevant markers with triploTs, and secondly, to drastically decrease manual examination and computation time. To implement a workflow, properties of combinatorial marker intensities are used and a specific machine learning approach is deployed to rank the marker combinations which discriminate best between two groups of samples, and with optimised visualization configurations.

4.4.1 Ranking table facilitates the identification of subpopulations

Looking at the top 10 of the VR table (Tab. 3.4) from the erLR-CV cycle, the absolute range from the triploT section was obtained (Fig. 3.10). KLRG1 appears in row 4 (rank 1) in combination with CD90 and CD86. It is of special biological interest in this context, because KLRG1 is known to be expressed by highly differentiated T cells. Furthermore, PD-L1 in row 5 (rank 1) is very interesting as well. It is also placed on rank 8 as axes positions (basis markers) and even on rank 3 as position C (associated marker) from the total VR table (Tab. 3.2). This marker could be part of a counter regulation of the immune system activation, thus was not considered in this context so far [81, 82]. However, a counter regulation implies the intended immune system activation by the eff treatment. Moreover, PD-L1 was also found highly expressed by *Spitzer et al.* and should be consequently further inspected in this context. Recapitulated, these marker proposals and the further narrowed down Th1 subpopulation mentioned in Section 3.4.3 with the characteristics of CD90$^{\mathrm{high}}$CD86^{+}CD44$^{\mathrm{high}}$Tbet$^{\mathrm{high}}$CD69$^{\mathrm{high}}$Ki67$^{\mathrm{high}}$ in CD4^{+} T cells shows that this VR table helps biological experts to analyze high-dimensional cytometric data by guiding which parameter to set as basis or associated marker in *PRI*.

CD90 and CD44 in combination as basis markers are selected by manual inspection to identify the different bin regions I-III (Fig. 3.6), but this combination is not directly selected by the top 10 ranks of the VR table. Actually, CD90 and CD44 in combination as variables (CD44.CD90.XX in the VR table) are selected twice at rank 39 and 41 respectively, with CD27 and PD-L1 as associated markers, but this accounts for less than 0.5% of the total set of the variables (App. B, Tab. T5b). A reason could be the too rough distribution of the bin range into (only) four triploT sections. Looking at Figure 3.6 more precisely, one can see that a division into 3 × 3 sections would separate these bin regions more accurately. On top of that, another reason could be the underlying heterogeneous expression within the four sections of the samples. Looking at the section values with other combinations as basis markers, such as CD138 (rank 24 with CD44.CD138.CD27.absRange.S2, rank 25 with CD90.CD138.PD.L1.absRange.S2), it is noticeable that these values are significantly more different compared to the CD44-CD90 basis marker combination (App. B, Fig. S14). It is evident, that the distance of the median values are higher and the p-values are lower between eff and ineff treatment for CD138-X basis markers than CD44-CD90 basis markers. This leads to the conclusion, that the marker CD138[6] is a good choice for a basis marker (as also seen in Table 3.2a), since it is clearly a good differentiator between groups. That aside, the marker combination CD44-CD90-CD27 might be not listed in the top ranks, but CD44 and CD90 are the top selected basis markers (Tab. 3.2a) and CD27 is at rank 3 for the associated marker (Tab. 3.2b). The examination with guidance of both tables would potentially lead to the identification of this marker combination and subsequently the characterization of the bin regions I-III.

Due to the reproducible display and the fixed choices of the x- and y-axes, a further benefit with triploTs is the comparable expression pattern of a marker combination in different samples. An example is shown in Appendix B, Figure S11. CD90-PD.L1-MSI(CD86) is displayed for all samples with cutoff settings for CD90$^{\text{high}}$, PD-L1^{+} and CD86$^{\text{high}}$. Looking at the frequencies of CD90$^{\text{high}}$PD-L1^{+}, which are placed as percentage numbers in the upper right quadrant, the cell count of the quadrant (black, eff: 3,6-7.2%, ineff: 4.2-7.7%) and of marker CD86$^{\text{high}}$ (green, eff: 1.1-4.9%, ineff: 0.1-0.2%) compared to total cell count, respectively, the numbers fluctuate in different scales. Despite the varying frequencies, concentrated areas, which indicate subpopulations, can still be seen as a consequence of the continuous display of the expression distribution from associated marker C. It promotes an intuitive capture and comprehensible interpretation of these patterns and facilitate the characterization of cell subpopulations within and between groups.

4.4.2 *PRI* results differ from *Citrus* and *Scaffold Maps*

The comparison of the three studied approaches *PRI*, *Citrus* and *Scaffold Maps*, has heterogeneous results, and are summarized in Table 4.2. Firstly, despite their differences in the algorithms, five markers were found consistently to be discriminative: CD90, CD44, PD-L1, CD62L and CD27. Furthermore, the outcome of *PRI* has the same amount of protein markers in common with *Citrus* (7) and with *Scaffold Maps* (7) in regard to their discriminative power. However, looking deeper into the characteristics (– or + for low/high) of the markers, they are mainly not concordant and depict, in part, contradictory populations (Sec. 3.2.2 and 3.4.4). Thus, the identified subpopulations of each approach have manifestly different characteristics. One reason for the different outcomes are the different approaches. *Spitzer et al.* uses *Scaffold Maps* which is based on a clustering algorithm, and so does *Citrus*, whereas *PRI* uses a collection of three-parametric-combinatorial bin properties which are aligned to different combinations of two basis markers. In principle, *Citrus* uses a similar classification algorithm to *PRI*, but deploys cluster information as input variables in its algorithm. The results need to be validated *in vivo* or *in vitro* to objectively benchmark the quality of the outcomes. Recent authors also suggest to apply multiple tools in order to view data in different ways and fully extract biological meaning [9, 86, 87].

[6] CD138 plays an important role in, among others, wound healing and translocation in endothelial cells and fibroblasts [83], and regulates homeostasis in innate-like T cells [84, 85]. The biological meaning of CD138 in the context of tumor-specific CD4^{+} T cells needs to be further investigated.

Table 4.2: Summary of the outcomes of *PRI*, *Citrus* and *Scaffold Maps*. Top five highest marker counts from VR table (sum of markers in Tab. 3.2a and 3.2b), markers from manual inspection with triploTs, *Citrus'* noticeably different markers from discriminating clusters, and the identified subpopulation of *Spitzer et al.*, ordered by total prevalence count of the full VR in Appendix B, Table T5. Check marks indicate the discriminating proteins, disregarded if either positive or negative population. Underlined markers are common in the three approaches.

Marker	Table 3a and 3b	*PRI*	*Citrus*	*Spitzer et al.*
CD90	48 + 5 = 53	✓	✓	✓
CD86	15 + 25 = 40	✓	✓	-
CD44	27 + 4 = 32	✓	✓	✓
Ki67	21 + 1 = 22	✓	-	-
KLRG1	14 + 6 = 20	✓	-	-
PD-L1	9 + 9 = 18	✓	✓	✓
CD138	18 + 0 = 18	✓	✓	-
Tbet	8 + 6 = 14	✓	-	✓
CD62L	9 + 1 = 10	✓	✓	✓
Ly6C	0 + 10 = 10	✓	-	-
CD27	0 + 9 = 9	✓	✓	✓
CD69	3 + 0 = 3	-	-	✓
Foxp3	0 + 3 = 3	-	-	-

Notably, Ki67, KLRG1 and Ly6C are reported discriminative in the top ranked table of *PRI*'s VR workflow (Tab. 3.2a+b), but this have not been seen with *Citrus* or *Scaffold Maps*, even though Ki67, at least, as a proliferation marker might be of interest. The protein marker Foxp3, which marks for Treg cells and is found to be an important marker in the manual inspection with the triploTs (Fig. 3.6), but is not picked by *PRI*'s VR (only appears three times in the full VR table, but could have been observed in the CD90-CD44-plane) or *Citrus* or *Scaffold Maps*. This indicates that it is difficult to find populations which are either low in quantity or low in changes. For *PRI*, the triploT sectioning into 2 × 2 is, again, possibly too broadly ranged. One could increase the sectioning to 3 × 3 or 4 × 4 to better capture the bin properties in the region where Foxp3 is highly concentrated. Another possible reason is the choice of the cofactor in the *arcsinh* transformation of the data which can result in an alteration of the differential power of a protein marker. However, running *Citrus* with $cof = 0.1$ resulted in an even worse consensus to the results with *PRI* and *Scaffold Maps*.

The quality in characterizing the markers are very different between these approaches. *Scaffold Maps* provides a very complex map, but subpopulations need to be selected and characterized manually. *Spitzer et al.* subsequently used density plots for this purpose. Even though *Citrus* provides the density plots of the discriminative clusters, but it also does not support further characterization. Thus, both approaches lead to a rough characterization of the markers in mainly positive (+) and negative (-). *PRI* however, uses the triploTs as visualization and as basis in the classification algorithm. This has the advantage, that discriminative three-combinatorial triploT sections are chosen, which can be in turn easily validated with the triploT visualization itself. A further characterization, such as negative, low or high SI of a marker in a discriminating subpopulation can be visually perceived, because of the full SI range of the markers and information provided in the triploTs (Fig. 3.6 and 3.11).

4.5 Bin-based *PRI* approach vs. clustering techniques

Clustering techniques are commonly used in the field of cytometry, for example, the gold standard classification tool *Citrus*, and *Scaffold Maps*. Among others, *SPADE* is also a frequently used approach, which is based on hierarchical clustering [54, 61]. This approach is particularly advantageous for rare cell detection due to its upstream density-based down-sampling procedure, but provides a highly simplified overview of the cellular phenotype. *Scaffold Maps* also uses a clustering algorithm (k-mediods, Tab. 4.3), but does not work fully automated. In fact, it is to its advantage, that manually gated landmark populations can be included in their workflow. They are connected with the computationally defined clusters. This process is performed with biological expertise. However, the interpretation of the behaviors or phenotypes of the identified cell clusters are still challenging [72, 88]. Both tools, *SPADE* and *Scaffold Maps*, lack in providing a method to highlight differentially expressed cell clusters between the groups, and a visualization method to characterize the phenotypes of these clusters. Furthermore, the underlying algorithm k-medoids is prone to background noise and outliers, since all cells are regarded in the calculation of the cluster centers (App. A, Alg. A). *Citrus* on the other hand, has only few parameters to adjust and runs without manual interaction, and uses $L1$-regularized LR to solve the differentiation. However, *Citrus* only provides overlapped density distributions of the clusters compared to the residual cells as visualization. In addition, in clustering one does not know the subpopulation size, hence the cluster size. Manual merging of clusters is cumbersome, not only because of the necessity of the biological expert knowledge but also the speculation of many unclear definitions of the clusters. And in the end, results of cluster algorithms are not reproducible when including, excluding or concatenating different samples. This is a crucial point in clinical diagnostics.

Another concern with clustering techniques in general, is that cell transitions, as they occur in a hematopoietic system, are continuous rather than discrete as mentioned in Section 4.2.2. This makes the evaluation and characterization of clusters additionally challenging, because the clustering resolution is considerably low for homogeneous populations. They are often unable to cluster cell populations with relatively similar phenotypes, such as activated and non-activated T cells. Then again, *PRI* provides a semi-continuous visualization where also small inter-correlations are easily identifiable.

With the VR table as guidance with where to look for subpopulations, the top 10 highest rankings are manually inspected and confirmed with triploTs. They all show differentiating characters in the absolute ranges of the triploT sections between both treatments. After consultation with biological experts, these areas are partly not biologically relevant or already known. Since this algorithm exclusively uses the bin properties, the biological relevance cannot be evaluated by this tool. However, the narrowed down Th1 subpopulation (CD90$^{\text{high}}$CD86^{+}CD44^{+}Tbet^{+}CD69^{+}Ki67^{+}) is identified with the aid of this table, which could not be validated by the clustering approach *Citrus*, but could be validated with the dimension reduction technique *viSNE* (Sec. 3.4.4). This subpopulation is enriched of highly proliferating cells.

Furthermore, the examination with *PRI* includes a novel visualization technique which is insightful and easily interpretable. Due to the bin structure, other statistical information than the cell density (as in contour plots) can be displayed such as MSI, frequency and MSI+ of at least one associated marker. In fact, even several associated markers can be visualized, which leads to a comprehensive pseudo-multi-parametric visualization. The first attempt to plot the frequencies of two associated markers (resulting to a combinatorial display of four markers) shows the additional benefit of these *PRI* variables [69]. Of course, this combinatorial display is limited to three or four markers, but it is difficult for the human eye to understand and visualize more inter-correlations at once. Therefore, the information of these insightful visualizations are used as input variables in the erLR to extract the discriminative three-parametric marker combinations, from which the examiner can investigate further with the triploTs.

4.6 *PRI* as a complement to current analysis strategies

Several section properties have been tested in which the results with absolute range were fruitful. There are endless options to define the section properties which, in combination with an increased sectioning in a 3 × 3 or 4 × 4 grid, could lead to the identification of smaller significant subpopulations. The aforementioned deployment of the +/− cutoff setting function would be another option for sectioning which would also increase the automation of this workflow. In addition, including these cutoffs would enable the display of, among others, MSI(C+) and frequency(C+), which could lead to an optimization of this workflow. The vast array of options can confuse and lead to the question of whether or not this approach is valuable. In fact, the combination of intuitive and reproducible visualization linked with the guided analysis strategy actually makes this approach already very valuable. Since the results from this workflow can be validated easily through visual perceptions, this strategy is more accessible to the community in the field of cytometry than other approaches. Furthermore, the *PRI* workflow, especially the VE part, is designed, but not restricted, to mass and flow cytometric data. Any numerical three-combinatorial parameter can be visualized if row count and bin range are suitable.

Another advantage of *PRI* is elucidated in Table 4.3. The table lists the time complexity of dimension reduction and clustering techniques addressed in this study. It manifests again, that *PRI* is independent from the cell count c, which is the biggest factor in this scenario. With greater numbers of samples, the cell count increases rapidly into hundreds of millions. Therefore, other algorithms need to use upstream downsampling approaches, which possibly leads to a loss of information as practised with *viSNE* (Section 4.2.3). However, *PRI* groups the cells into bins and extracts the bin information for downstream analysis. Hence, *PRI* workflow is especially suitable for large cytometry data studies.

In this study, with the additional interactive tools *PRI-base* and *PRI-ana*, this straightforward interpretable workflow is possibly more accessible to the biological community of cytometry than comparable interactive cytometric workflow tools such as *cytofkit*, *cytofast* and the commercial *Cytobank* and *FlowJo* [87, 89, 45, 57]. *Cytofast* provides an interactive platform for clustering and visualization, mainly based on *t-SNE*. The other tools also provide a preparation step, and a variety of clustering and visualization techniques. Defining what constitutes a successful analytical approach is difficult in general. This study's workflow bridges the gap between conventional gating strategies and semi-automated analyses to extract meaningful information, and aids in discovering interesting subpopulations on a basis of comprehensive three-parametric combinatorial bi-axial plots. The focus of this study is to create variables and to build an embedded VR workflow whose outcomes are more tangible for biologists, rather than cell clusters or arbitrary axes which are difficult to comprehend and interpret. Thus, *PRI*'s approach facilitates examinations with intuitively interpretable results but is limited in a minimal cell count, and its focus is not in single cell resolution, whereas many cluster and dimension reduction techniques show single cell information, but are limited in cell count and need to use upstream downsampling procedures. Hence, *PRI* is an excellent complement to already established tools, and combining *PRI* with several other algorithms is preferred to cover the full scope of analysis approaches. A migration of *PRI* into the interactive cytometric workflow applications would reduce the examiner's familiarization phase with the GUI, and a comprehensive investigation with several approaches would be facilitated. An expansion of *cytofkit* with *PRI* would be especially preferred, since *cytofkit* is freely available, it is also written in *R*, and it already contains several approaches. *PRI* would substantially extend their major components: pre-processing, cell subset detection, and cell subset visualization and interpretation [87].

Table 4.3: Algorithms and their time complexity with c cells, k clusters, m markers, N samples, p variables (derived from m) and it iterations. Tools for cytometric data are listed, which includes these algorithms (partly).

Method	Example tools	Time complexity
hierarchical clustering	*Citrus*, *SPADE*	$\mathcal{O}(c^3)$
k-means	*flowMeans* [90]	$\mathcal{O}(it \cdot k \cdot c)$
k-mediods	*Scaffold Maps*	$\mathcal{O}(c^3 + k(c-k)^2 \cdot it)$ [91]
t-SNE	*viSNE*	$\mathcal{O}(c^2 \cdot m)$
GLM	*Citrus*, *PRI*	$\mathcal{O}((N^2 m + m^3) \cdot it)$
triploT sectioning	*PRI*	$\mathcal{O}(m^3 \cdot N)$

5 Conclusion and outlook

Manual gating strategies combined with conventional plots are time-consuming and cannot fully capture the combinatorics of high-dimensional mass and flow cytometric data. State-of-the-art analysis strategies are mainly based on clustering algorithms and the respective frequencies of the clusters. The central problems with these methods are the scope of interpretation, reproducibility and computational complexity. To address these problems, the cytometric analysis and visualization workflow *pattern recognition of immune cells* (*PRI*) is proposed in this dissertation. The proof of concept of this workflow is demonstrated by re-analyzing the already published mass cytometric data set from *Spitzer et al.* . With discriminating protein marker combinations discovery in focus, the workflow starts from scratch with data preparation and storage. Subsequently, it goes through bin-based variable engineering (VE) and dimension reduction to variable ranking (VR) with characteristics of the engineered variables by elastic-net regularized logistic regression (erLR). Moreover, the implementation of the interactive and standalone database management system *PRI-base* and the visualization and analysis tool *PRI-ana* have greatly facilitated the examination of in-group created flow cytometric data and re-analyses of mass cytometry data.

PRI, with its various bin-based visualizations and calculation methods, have been invented to tackle the difficulties to generate reproducible, comprehensible and comparable results between groups. It allows to identify and characterize cell subpopulations and correlating patterns in a homogeneous system, which are not easily tangible with conventional and state-of-the-art approaches. The use of markers on the axes combined with the bins in triploTs enables a reproducible, pseudo-continuous display and a comparable scaffold for pseudo-multi-parametric viewing. Similar and state-of-the-art visualization alternatives *Color Maps* and *viSNE* are provided and shown on the data set to perform less well and less insightful. Due to the absence of artificial axes, the interpretation of the depicted subpopulation from *Spitzer et al.* is intuitive and has resulted in a characterization of three different subregions. Furthermore, this study's workflow is valuable to guide and design prospective experiments, e.g. the choice of the protein markers in flow and mass cytometric studies.

These triploTs were further engineered to serve as variables in a machine learning approach for VR. The usage of filters and logistic regression (LR) coupled with two optimized regularization parameters α and λ resulted in a drastic reduction of the variable count from 109,668 to 97. The deployed embedded VR pipeline provided a shorten variable table with sections of three-marker-combinatorial triploTs which are most discriminative between the effective (eff) and ineffective (ineff) treatments.

With the help of this study's triploTs, the subpopulation from *Spitzer et al.* could be further narrowed down into three subpopulations which are referred to as naïve, Th1 and Treg cells. With neither *Scaffold Maps* nor *Citrus* and *viSNE*, one has captured the characterization of these subpopulations. Furthermore, the inspection of the VR table resulted in a further specification of the depicted Th1 subpopulation which is characterized by CD90$^{\text{high}}$CD86^{+}CD44$^{\text{high}}$Tbet$^{\text{high}}$CD69$^{\text{high}}$Ki67$^{\text{high}}$. With that specification, this subpopulation could be validated by the established approach *viSNE*. *PRI* is therefore an essential complement for the current analysis strategies.

5.1 Key features of the proposed approach

Three particular strengths of the proposed *PRI* approach are i) the innovative VE which enables conclusive visualization techniques, ii) the inclusion of expert-guided upstream filters, and iii) the combination of erLR with the information of the engineered variables as basis. The first two points lead to a drastic dimension reduction, and a conclusive representation of the raw data, which is straightforward to follow and justify. The latter point has the advantage, that the model has a low complexity and handles multi-collinearity, and the results in the VR table are comprehensive and its variables are, as a matter of fact, evaluable by their own visualization in *PRI*. The advantages and limitations of *PRI* are recapitulated in Table 5.1.

5.2 Outlook

The study presented in this dissertation has opened a number of research lines that should be explored in the future. Firstly, to validate the proposed workflow, one option is to confirm the resulting three discriminative subpopulations. This could be processed statistically by another (larger) project with the same setting of model, disease and treatment. A biological validation could be the inspection of the functionalities of the found subpopulation in knock-out mice, and if successful, it is necessary to examine if this mouse model is transferable to humans. Therefore, one should investigate if this subpopulation can be found in blood tissue of breast cancer patients, which would hopefully indicate a successful treatment. Another option would be, in general, a proof of concept of *PRI* with a biologically different mass or flow cytometric data set, which has sufficient cell count per sample. It would be of particular interest, if this workflow is scalable to bigger data sets and multiple groups. And of course, one could also benchmark this workflow on a data set provided from the *FlowCAP* challenge [92]. Last but not least, another milestone is the evaluation of the expert-guided filters and, particularly, their possible generalized application, since they are critical for the outcome of this approach. The role of the error measurements in the embedded VR should be also addressed.

Table 5.1: Benefits and limitations of *PRI*.

Benefits	Limitations
+ reproducible and conclusive visualization	– many cells are necessary ($>\sim 10,000$)
+ setting of bin size and minimum number of cells support statistical robustness	– broad bin range for subpopulation identification
+ intuitive capturing of inter-correlations and characteristics of a subpopulation of combinatorial markers	– no single cell resolution (but single cell information remains stored)
+ same patterns are easily obtained, including those between different samples	– the right marker combination for A-B-C is necessary to obtain meaningful information
+ a lot of useful statistical information displayable	– data-driven (VR does not include biological meaning)
+ particularly suitable for homogeneous populations	
+ insightful visualization as a basis for VR workflow	
+ VR reduces the marker combinations only to those significant between groups	
+ VR table as guidance similar to RNAseq analyses	
+ automatable	
+ less complex computation	
+ easy handling with GUI	

In this study, the deployed embedded variable selection (VS) method resulted in a VR table. An important goal is to set a frequency cutoff on that table to turn a ranking of variables into a smaller relevant variable subset. Applying this workflow to several data sets, a trend can be presumably determined. A more simplistic approach is a cutoff score such as only the top 20 or top 25 variables, or the point at which the frequency of the summed variables crosses a threshold of 80% or 90% of the total VR set. A more sophisticated option is the downstream deployment of a filter method such as the minimum redundancy maximum relevance selection. The main benefit of this method is, that by reducing mutual redundancy within the ranked variable set, these variables capture the class characteristics in a broader scope. Further, it is independent of class prediction methods, and thus does not directly aim at producing the best results for any prediction method. Last but not least, another option is to subsequently deploy a wrapper method to find the best subset of features, which should not be computational demanding, since there are only 97 selected triploT sections in the VR table.

In general, choosing the appropriate VS method for a given scenario is not an easy-to-solve question. To optimize the results of this study's VR workflow in another way, an aggregated VR could be deployed. It may be that a combination with another VR method is even more robust and reliable. For instance, random forest is also a robust, multi-variate and embedded, but non-linear VR method. This method has also the advantage as in *PRI*, that the complexity is independent from the number of cells ($t\sqrt{N} \cdot p \cdot log(p)$, with t number of trees) [27]. The variable sets can be aggregated by intersection, and the selected variables could be then ranked with a combined importance score. The first application of random forest on cytometric data was established in [93], which uses the information of this method to improve the distances of cell clusters in the *viSNE* map. Nonetheless, this method might not work well with such a small sample size.

Last but not least, *PRI*'s workflow could be migrated into already published interactive cytometric analyses applications, such as *cytofkit* [87] and *cytofast* [89], and the commercial *FlowJo* [57] and *Cytobank* [45]. A collaboration could be requested, although the authors from the first two mentioned tools would be preferred, since these applications are written in *R*, and are non-commercial.

A Appendices: *Spitzer et al.*

Panel overview of *Spitzer et al.*

Table T1: Panel overview of example data provided by [13], with two additional columns to the right. In column '*Bio': 1 indicates the relevant marked proteins and 0 indicates the proteins which are irrelevant for this study since these proteins should be negative when gated for CD4$^+$ T cells, also called T helper cells. Column '*Location' shows the location of the protein expression extracted from [94, 95]. Ter119 (listed as #1) is a marker for erythrocytes which are red blood cells and are therefore not part of the immune system; immunglobulins IgG and IgM and the subunit FcER1a of IgE (listed as #4,30,38) are (parts of) antibodies which are produced in B and plasma cells; PγMT (listed as #8) marks viral T antigens; CD8 is a marker for cytotoxic T cells, which express F4-80 (listed as #19,33), and was filtered in the last gating step (Fig. S1); and B220 and CD19 (listed as #23,37) are expressed by the B cell lineage. These ten proteins have therefore no biological meaning in the context of CD4$^+$ T cells.

#	Channel	Metal	Protein	Clone	*Bio	*Location
1	113	In	Ter119	TER119	0	Monoclonal antibody
2	115	In	CD45	30-F11	1	hematopoietic cells
3	139	La	Ly6G	1A8	1	neutrophils
4	140	Ce	IgD	11-26c.2a	0	mature naive B cells
5	141	Pr	CD16/32	2.4G2	1	NK cells, neutrophils and macrophages
6	142	Nd	CD49b	HM2	1	B-cells, monocytes, activated T-cells,...
7	143	Nd	CD11c	HL3	1	T- and B-cell subsets, monocytes,,...
8	144	Nd	PyMT	PyMT	0	polyoma middle tumor
9	145	Nd	CD27	LG.3A10	1	T-cells
10	146	Nd	CD138	281-2	1	pre-B-cells, breast cancer cells,...
11	147	Sm	PD-L1	10F.9G2	1	activated T- and B-cells,...
12	148	Nd	CD103	20000000	1	lymphocytes
13	149	Sm	SiglecF	E50-2440	1	neutrophils
14	150	Nd	PDCA-1	120g8	1	T-cells, monocytes, NK cells,...
15	151	Eu	Ly6C	HK1.4	1	monocytes, macrophagesm,...
16	152	Sm	Ki67	SolA15	1	various stages in the cell cycle
17	153	Eu	CD11b	M1/70	1	monocytes, NK cells, T- and B-cells,...
18	154	Sm	cKit	2B8	1	hematopoietic stem cells and progenitors
19	155	Gd	CD8	53-6.7	0	thymocyte subsets, cytotoxic T cells,...
20	156	Gd	CD4	RM4-5	1	thymocyte subsets, Th cells, Treg cells,...
21	157	Gd	CD3	17A2	1	mature T-cells and thymocytes
22	158	Gd	PD-1	29F.1A12	1	activated T- and B-cells
23	159	Tb	B220	RA3-6B2	0	hematopoietic cells in B cell lineage
24	160	Gd	NK1.1	PK136	1	NK cells
25	161	Dy	T-bet	04-46	1	Th1 cells
26	162	Dy	TCRgd	GL3	1	T subset
27	163	Dy	CD62L-FITC	MEL-14	1	B- and T-cell subsets,monocytes,...
28	164	Dy	CD86	GL-1	1	activated B- and T-cells,macrophages,...
29	165	Ho	CD69	H1.2F3	1	activated leukocytes and macrophages,...
30	166	Er	FcER1a	MAR-1	0	Subunit of receptor of IgE
31	167	Er	Foxp3	NRRF-30	1	T subsets
32	168	Er	RORgt	B2D	1	lymphoid compartments
33	169	Tm	F4/80	BM8	0	wide range of mature tissue macrophages
34	170	Er	CD115	AFS98	1	monocytes, macrophages,...
35	171	Yb	CD64	X54-5/7.1	1	monocytes and macrophages
36	172	Yb	KLRG1	2F1	1	mast cells
37	173	Yb	CD19	6D5	0	B cells and follicular dendritic cells
38	174	Yb	IgM	RMM-1	0	B cells
39	175	Lu	CD44	IM7	1	most lymphohematopoietic cells
40	176	Yb	CD90	G7	1	hematopoeitic cells
41	209	Bi	MHC	M5/114.15.2	1	macrophages

Full column table list in the fcs-files

Table T2: The column table list in the fcs-files has 59 entries in total, including protein markers, bar codes and other device specific measurements.

1	Ba138Di	11	Foxp3	21	NK1.1	31	CD49b	41	BC4	51	Time
2	beadDist	12	RORgt	22	CD69	32	CD11c	42	BC5	52	Tl203Di
3	MHC-II	13	CD115	23	I127Di	33	PyMT	43	BC6	53	Tl205Di
4	IgD	14	Ly6C	24	Ter119	34	CD27	44	PD.L1	54	F4-80
5	Cs133Di	15	CD11b	25	CD45	35	CD138	45	CD16.32	55	Xe131Di
6	T.bet	16	Event_length	26	DNA1	36	CD103	46	Cisplatin	56	CD64
7	TCRgd	17	CD8	27	DNA2	37	PDCA.1	47	SiglecF	57	KLRG1
8	CD62L	18	CD4	28	Ly6G	38	BC1	48	Ki67	58	CD19
9	CD86	19	CD3	29	CD44	39	BC2	49	cKit	59	IgM
10	FcER1a	20	PD.1	30	CD90	40	BC3	50	B220		

Gating scheme applied on example data set *Spitzer et al.*

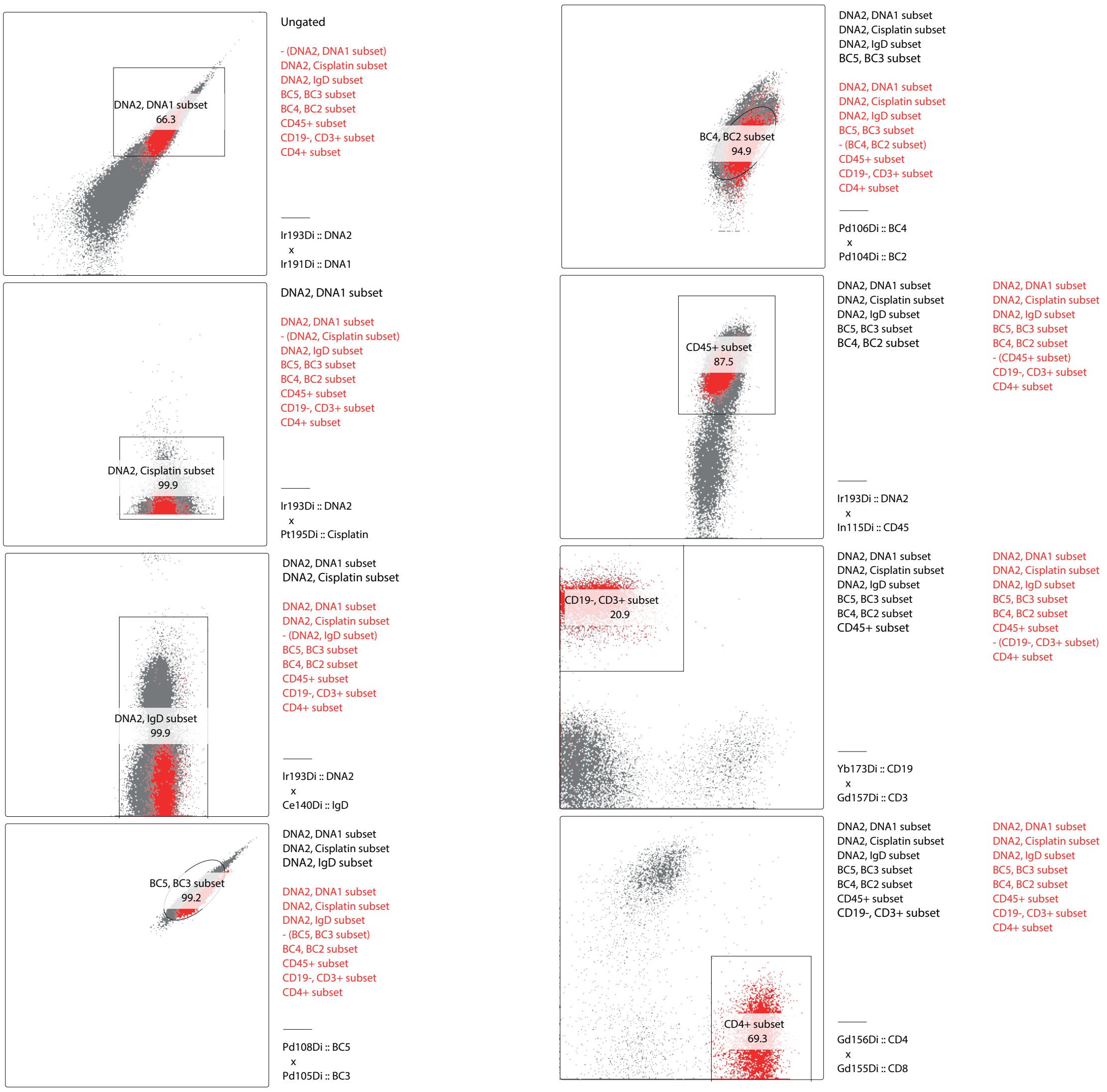


Figure S1: The gating process is rapidly performed by the software *FlowJo* (v9.9.6) [57], guided by a mass cytometry specialist.

Algorithm clustering for large applications (CLARA)

Algorithm .0: CLARA applied by *Spitzer et al.*

```
Input: A data frame $\mathbf{Df}$ with $c$ cells and $m$ markers; $\mathbf{K}$ medoids, with $0 < K < c$; $\mathbf{j}$
       subsample size, with $K < j \leq c$
Output: A data frame $\hat{\mathbf{Df}}$ with cluster assignment for each cell
Step 1: Partition cells into $j$ subsets
foreach partition $C_j$ do
    Step 2: Initialize cluster centers ($K$ medoids)
    Step 3: Assign each cell to the closest medoid:
    foreach cluster $C_k$ do
        Find cell $w$ in the cluster minimizing total distance to other cells in that
          cluster:
            $$i_k^* = \text{argmin}_{\{i:C(i)=k\}} \sum_{C(i')=k} D(w_i, w_{i'}) \qquad (.1)$$
        Then $m_k = w_{i_k^*, k=1,2,...,K}$ are the current estimates of the cluster centers.
    end
    Minimize total error by assigning each cell to the closest cluster center:
            $$C(i) =_{1 \leq k \leq K} D(w_i, m_k) \qquad (.2)$$
     Step 4: Repeat Step 3 until assignment does not change.
end
```

B Appendices: *PRI* analysis

Overlapped density plots of all biological relevant markers

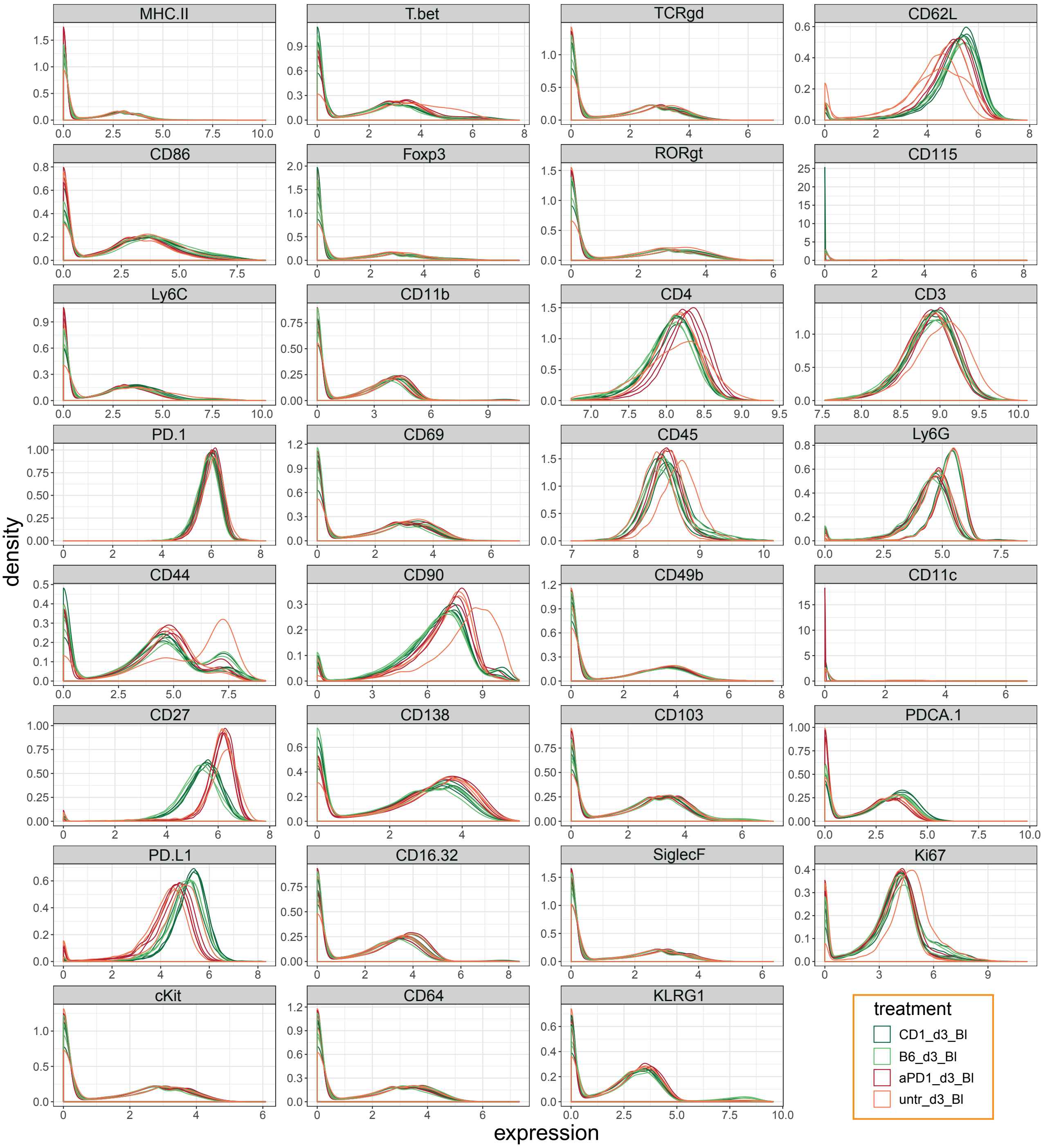
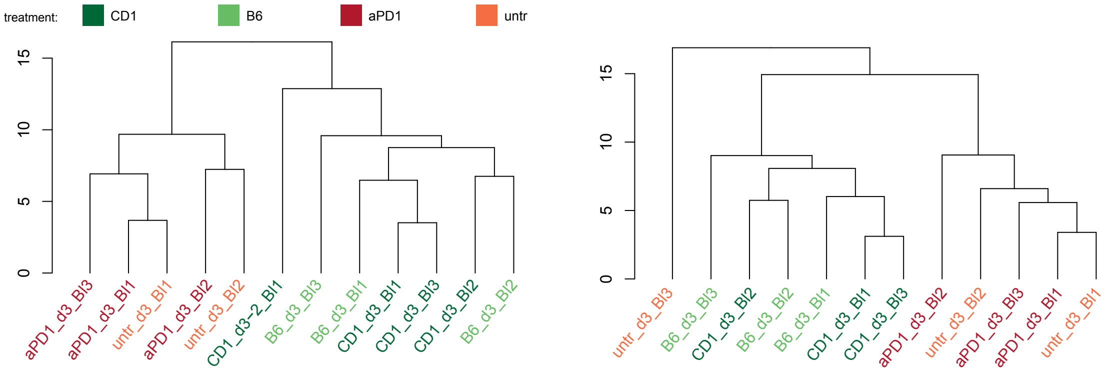


Figure S2: Overlapped density plots of all samples showing the 31 biological relevant markers after *arcsinh*-transformation with $cof = 0.1$.

Dendrograms of all samples except one



(a) dendrogram without sample *untr_ d3_ Bl3* **(b)** dendrogram without sample *CD1_ d3_ Bl1-2*

Figure S3: Dendrograms of all blood samples except of sample *untr_ d3_ Bl3* (a) and sample *CD1_ d3_ Bl1-2* (b), respectively, color-coded by experimental group: eff (green) and ineff (red) treatments. Hierarchical clustering method *ward.D2* was applied. Label names indicate sample IDs.

Density plots with different normalizations

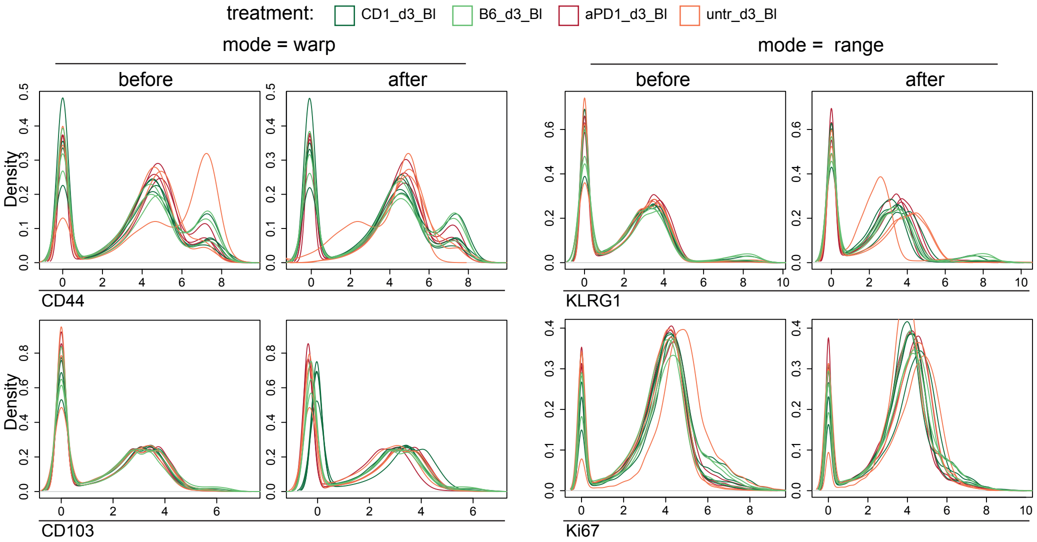
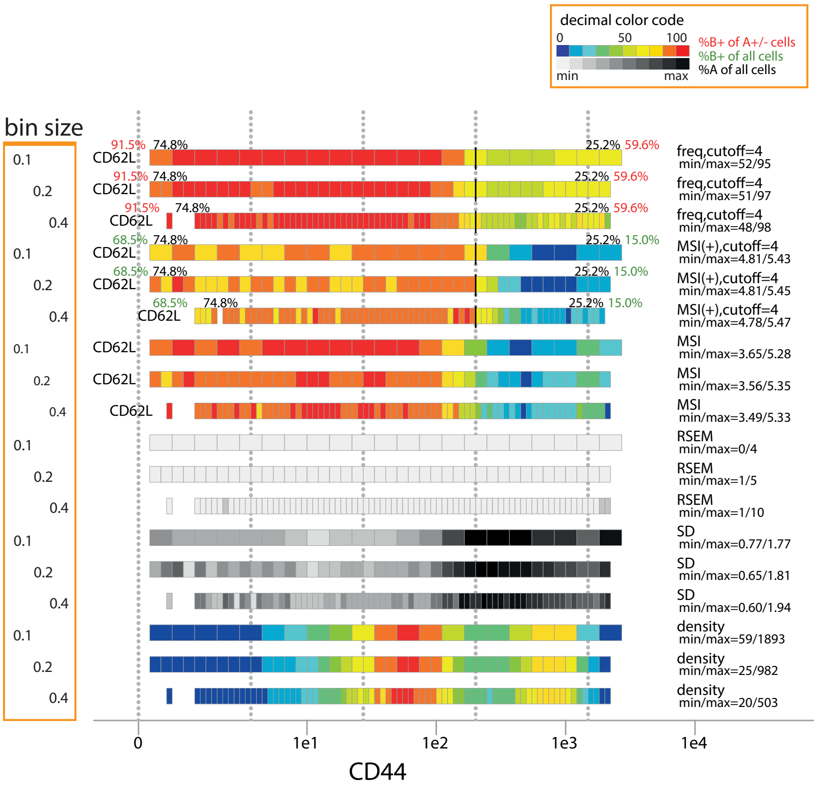


Figure S4: Density plots before and after the applied normalization *warp* (left) and *range* (right) for two example markers in which curve properties have changed a lot.

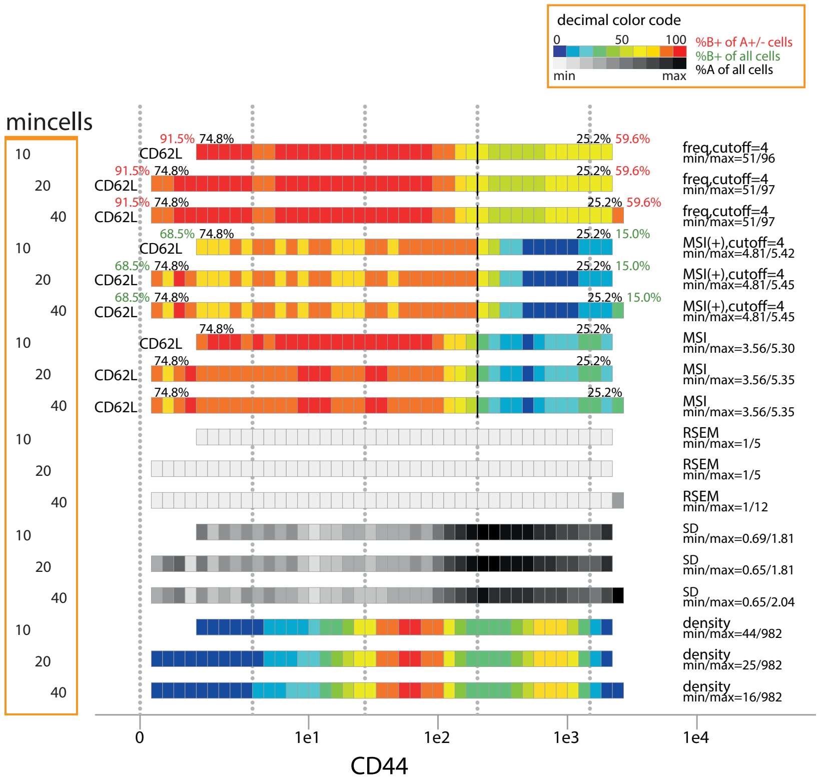
DiploTs bin size and minimum count of cells setting

Both configurations, bin size and the minimum number of cells, are adjustable. To elucidate the current setting, diploTs are created with different settings. Bin sizes of 0.1, 0.2, and 0.4 and the minimum number of cells of 10, 20 and 40, respectively are shown in Appendice B. Looking at Figure S5a at MSI(CD44), MSI(CD62L$^+$) and freq(CD62L$^+$), respectively, the diploTs with bins of size $arcsinh(x) = 0.2$ have a more stable resolution and preserves the continuity of the bins compared to $arcsinh(x) = 0.1$. The diploTs with bins of size $arcsinh(x) = 0.2$ and 0.4 have similar tendency information, but for smaller subpopulations the setting with smaller bin sizes is preferred. Looking at the configuration of the minimum number of cells in Figure S5b, the diploTs with different minimum number of cells are very similar. Only diploTs with the minimum number of cells= 10 shows differences at the right end. The diploTs have one additional bin which are highest for SD and RSEM. This effect can be seen at MSI(CD62L$^+$) and freq(CD62L$^+$). The diploTs with the minimum number of cells= 40 have fewer bins on the left in all diploTs, which could contain interesting information. Therefore, the setting with bin size $arcsinh(x) = 0.2$ and minimum number of cells= 20 is chosen for the subsequent analysis.



(**a**) bin size settings

Figure S5: DiploTs of CD44 as basis marker and CD62L as associate marker from sample *B6_ d3_ Bl2*, with different minimum number of cells **(a)** and bin size (**b**, next page) settings. The statistical method and its range are listed on the right of each diploT.



(b) the minimum number of cells (mincells) settings

DiGraphs

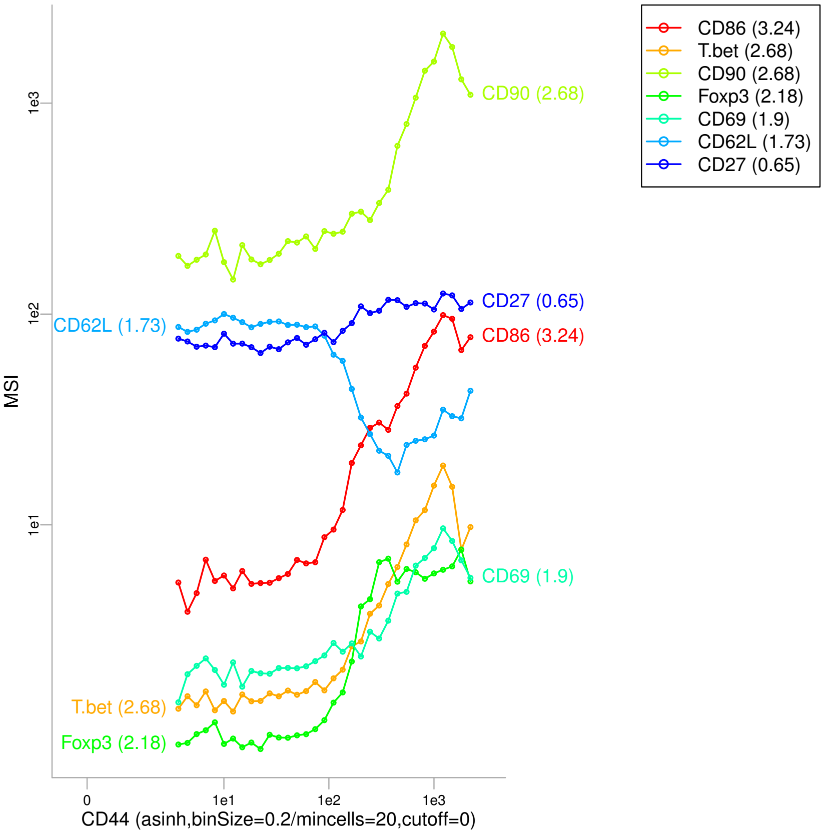


Figure S6: Digraphs of CD44 as basis marker and mean signal intensities (MSIs) of Tbet, CD62L, CD86, CD69, CD90 and CD27 are shown as associated markers, analogous to the diploTs in Fig. 3.3. Bin sizes are set to $arcsinh(x) = 0.2$ and minimum number of cells= 20. The MSIs in the bins are shown on the y-axis.

Density plots with different normalizations

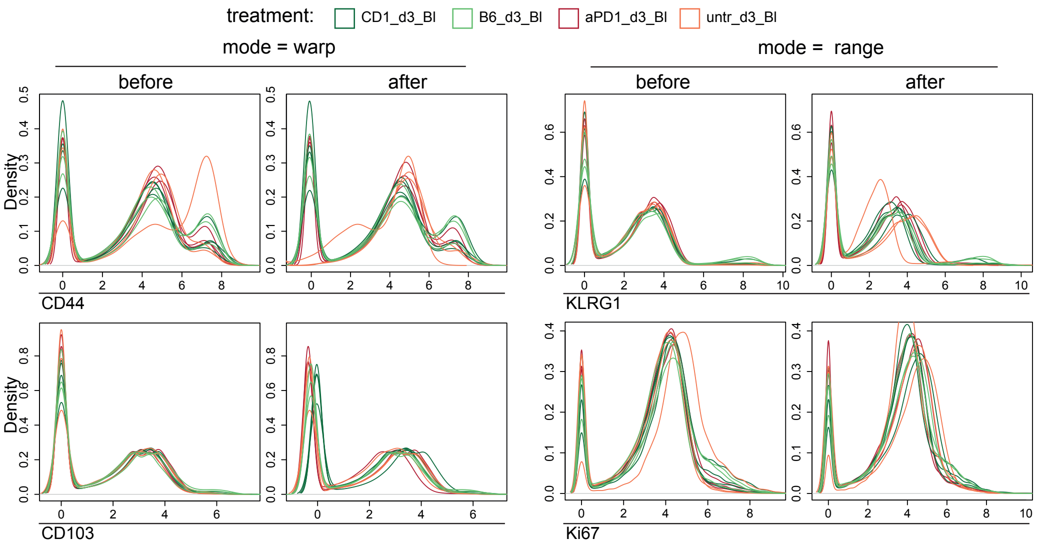


Figure S7: Density plots before and after the applied normalization *warp* (left) and *range* (right) for two example markers in which curve properties have changed a lot.

TriploT example with density plot

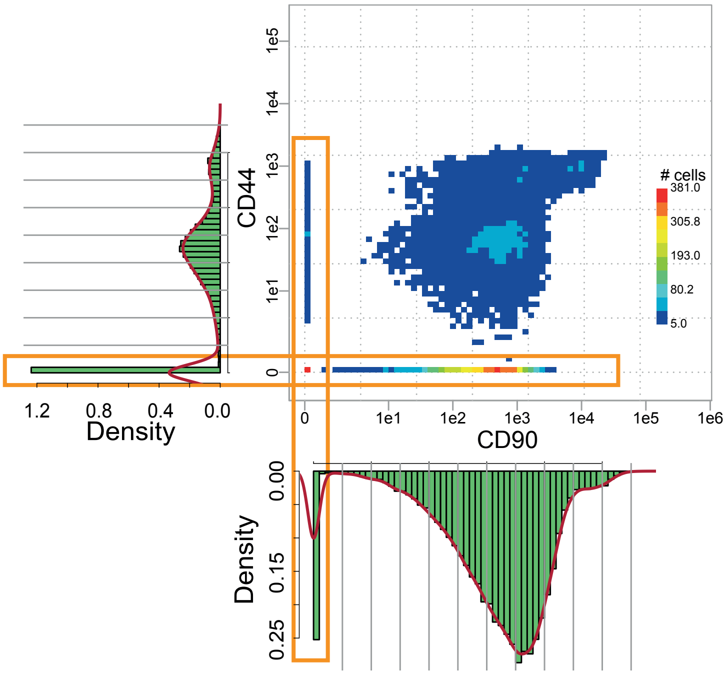


Figure S8: TriploT CD90-CD44-density and density plots from exemplary sample *B6_d3_Bl2*. Orange rectangle indicate the bins of the first row and column in the CD90-CD44-plane, respectively, and the respective peak at the density plots.

Cross-validation to find α

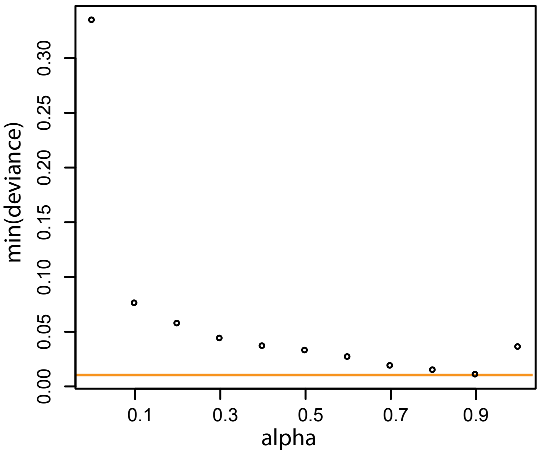


Figure S9: Example dot plot of minimal deviance (error rate) of cross validation (CV) runs with different α. Orange line indicate the α with the lowest error rate in one CV cycle. Therefore $\alpha = 0.9$ is depicted in this iteration. When the absolute range of the MSIs in the bin sections were deployed as input variables, $\alpha = 0.9$ was chosen the most.

Variable ranking table of top 10 variables grouped into axes positions and associated marker

Table T3: Variable ranking table of top 10 variables only, grouped into x and y axes (a), and associated marker (b), ordered by prevalence in Table 3.4. The ranking table is the outcome of erLR in a nested CV, and with *syma* = 0.9 and the absolute range of the MSIs in the bin sections (absRange) deployed as input variables.

(a) marker list of combined axes positions

Pos	Rank	Marker A+B	Counts	% Counts
1	1	CD90	7	0.35
2	2	CD138	4	0.2
3	3	Ki67	2	0.1
4	4	CD86	1	0.05
5	4	CD103	1	0.05
6	4	KLRG1	1	0.05
7	4	PDCA.1	1	0.05
8	4	PD.L1	1	0.05
9	4	Ly6G	1	0.05
10	4	CD62L	1	0.05

(b) marker list of position C

Rank	Marker C	Counts	% Counts
1	CD86	7	0.7
2	CD90	1	0.1
3	Ly6C	1	0.1
4	CD27	1	0.1

Box plots of top 10 ranked variables

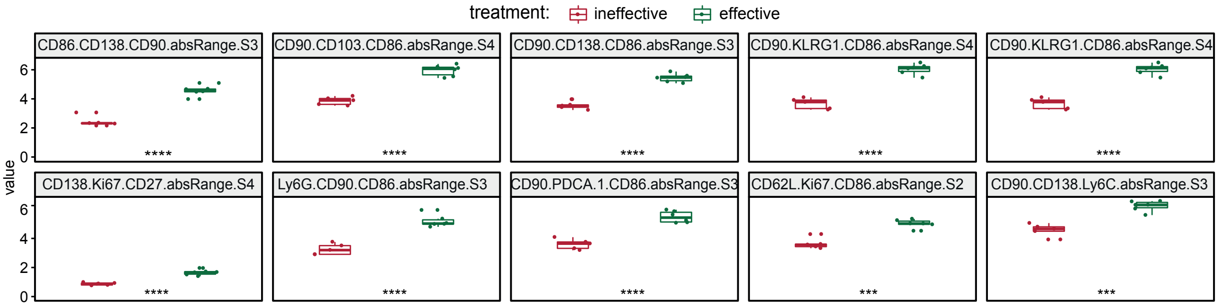


Figure S10: Dot plots of top 10 ranked variables with boxplots. From CV runs with settings $\alpha = 0.9$ and calc.meth = absolute range (absRange). Significance test with unpaired two-sided t-test on section values, labeled with asterisks: p-values ≤ 0.0001(****) and $p \leq 0.001$ (***).

Full variable ranking table

Table T5: Variable ranking table from selected section values (part1). From left: ranking position, prevalence count and frequency, mean model coefficient and split variable information (marker combination for x and y axes (A,B) and associated marker (C)) and section selection (S). The ranking table is the outcome of erLR in a nested CV with $\alpha = 0.9$ and the absolute range of the MSIs in the bin sections (absRange) deployed as input variables.

Pos	Rank	Variables	Counts	% Counts	Coefficient	A	B	C	S
1	1	CD86.CD138.CD90.absRange.S3	357	0.046	0.3686	CD86	CD138	CD90	S3
2	1	CD90.CD103.CD86.absRange.S4	357	0.046	0.3892	CD90	CD103	CD86	S4
3	1	CD90.CD138.CD86.absRange.S3	357	0.046	0.4979	CD90	CD138	CD86	S3
4	1	CD90.KLRG1.CD86.absRange.S4	357	0.046	0.3696	CD90	KLRG1	CD86	S4
5	1	CD90.PD.L1.CD86.absRange.S3	357	0.046	0.6033	CD90	PD.L1	CD86	S3
6	6	CD138.Ki67.CD27.absRange.S4	355	0.046	0.9503	CD138	Ki67	CD27	S4
7	7	Ly6G.CD90.CD86.absRange.S3	347	0.045	0.1607	Ly6G	CD90	CD86	S3
8	8	CD90.PDCA.1.CD86.absRange.S3	345	0.045	0.1442	CD90	PDCA.1	CD86	S3
9	9	CD62L.Ki67.CD86.absRange.S2	337	0.044	0.3056	CD62L	Ki67	CD86	S2
10	10	CD90.CD138.Ly6C.absRange.S3	333	0.043	0.2321	CD90	CD138	Ly6C	S3
11	11	CD90.CD103.Ly6C.absRange.S3	332	0.043	0.2337	CD90	CD103	Ly6C	S3
12	12	CD86.Ki67.CD90.absRange.S3	313	0.041	0.1405	CD86	Ki67	CD90	S3
13	13	CD62L.CD44.CD86.absRange.S2	284	0.037	0.3523	CD62L	CD44	CD86	S2
14	14	Ki67.KLRG1.Ly6C.absRange.S4	268	0.035	0.1179	Ki67	KLRG1	Ly6C	S4
15	15	CD44.Ki67.CD86.absRange.S3	258	0.034	0.316	CD44	Ki67	CD86	S3
16	16	CD62L.CD86.CD90.absRange.S3	195	0.025	0.1055	CD62L	CD86	CD90	S3
17	17	CD44.CD138.T.bet.absRange.S2	160	0.021	-0.1815	CD44	CD138	T.bet	S2
18	18	CD62L.CD44.CD27.absRange.S4	155	0.02	0.1977	CD62L	CD44	CD27	S4
19	19	CD62L.CD90.CD86.absRange.S2	145	0.019	0.1402	CD62L	CD90	CD86	S2
20	20	T.bet.CD44.MHC-II.absRange.S3	119	0.015	0.2286	T.bet	CD44	MHC-II	S3
21	21	CD44.PD.L1.CD86.absRange.S3	98	0.013	0.2201	CD44	PD.L1	CD86	S3
22	22	CD44.PDCA.1.PD.L1.absRange.S2	93	0.012	-0.349	CD44	PDCA.1	PD.L1	S2
23	23	CD62L.CD90.CD86.absRange.S3	92	0.012	0.1385	CD62L	CD90	CD86	S3
24	24	CD44.CD138.CD27.absRange.S2	89	0.012	0.3081	CD44	CD138	CD27	S2
25	25	CD90.CD138.PD.L1.absRange.S2	80	0.01	-0.8485	CD90	CD138	PD.L1	S2
26	26	CD90.CD138.CD45.absRange.S3	73	0.009	0.6699	CD90	CD138	CD45	S3
27	27	CD86.CD44.PD.L1.absRange.S2	70	0.009	-0.2526	CD86	CD44	PD.L1	S2
28	28	CD44.CD103.Ly6C.absRange.S4	63	0.008	0.2523	CD44	CD103	Ly6C	S4
29	29	CD90.CD16.32.CD86.absRange.S4	59	0.008	0.05	CD90	CD16.32	CD86	S4
30	30	Ki67.KLRG1.CD86.absRange.S4	57	0.007	0.5668	Ki67	KLRG1	CD86	S4
31	31	CD90.CD138.CD86.absRange.S4	56	0.007	0.1564	CD90	CD138	CD86	S4
32	32	CD69.CD90.T.bet.absRange.S4	55	0.007	0.2697	CD69	CD90	T.bet	S4
33	32	CD90.CD103.CD86.absRange.S3	55	0.007	0.335	CD90	CD103	CD86	S3
34	34	CD90.PDCA.1.Foxp3.absRange.S2	52	0.007	0.0755	CD90	PDCA.1	Foxp3	S2
35	35	CD90.KLRG1.CD62L.absRange.S4	48	0.006	0.3057	CD90	KLRG1	CD62L	S4
36	36	CD44.PDCA.1.CD86.absRange.S4	39	0.005	0.1064	CD44	PDCA.1	CD86	S4
37	36	CD44.PD.L1.CD90.absRange.S3	39	0.005	0.1133	CD44	PD.L1	CD90	S3
38	38	CD44.PDCA.1.CD27.absRange.S2	35	0.005	0.1872	CD44	PDCA.1	CD27	S2
39	39	CD44.CD90.CD27.absRange.S3	34	0.004	0.09	CD44	CD90	CD27	S3
40	39	T.bet.CD90.CD86.absRange.S3	34	0.004	0.0637	T.bet	CD90	CD86	S3
41	41	CD44.CD90.PD.L1.absRange.S2	33	0.004	-0.239	CD44	CD90	PD.L1	S2
42	42	CD90.PD.L1.CD11b.absRange.S4	31	0.004	-0.0316	CD90	PD.L1	CD11b	S4
43	43	T.bet.CD90.CD86.absRange.S4	30	0.004	0.2273	T.bet	CD90	CD86	S4
44	44	CD90.PDCA.1.CD44.absRange.S2	28	0.004	0.09	CD90	PDCA.1	CD44	S2
45	45	Ki67.KLRG1.T.bet.absRange.S2	26	0.003	-0.0461	Ki67	KLRG1	T.bet	S2
46	46	CD44.Ki67.Foxp3.absRange.S4	25	0.003	0.0885	CD44	Ki67	Foxp3	S4
47	47	CD86.CD138.CD44.absRange.S3	23	0.003	0.2187	CD86	CD138	CD44	S3
48	47	CD90.PD.L1.Ki67.absRange.S3	23	0.003	0.1801	CD90	PD.L1	Ki67	S3
49	49	CD90.CD103.KLRG1.absRange.S2	22	0.003	0.0159	CD90	CD103	KLRG1	S2
50	50	CD90.PDCA.1.Ly6C.absRange.S3	21	0.003	0.0812	CD90	PDCA.1	Ly6C	S3

Table T5: Variable ranking table from selected section values (part2) showing the positions 51-97.

Pos	Rank	Variables	Counts	% Counts	Coefficient	A	B	C	S
51	50	CD90.PD.L1.Ly6C.absRange.S3	21	0.003	0.0815	CD90	PD.L1	Ly6C	S3
52	52	CD90.CD16.32.CD86.absRange.S3	20	0.003	0.1001	CD90	CD16.32	CD86	S3
53	53	CD86.CD44.CD45.absRange.S3	19	0.002	1.0978	CD86	CD44	CD45	S3
54	53	T.bet.CD44.CD86.absRange.S3	19	0.002	0.1895	T.bet	CD44	CD86	S3
55	55	CD44.CD103.CD86.absRange.S4	18	0.002	0.2616	CD44	CD103	CD86	S4
56	55	Ki67.KLRG1.CD44.absRange.S4	18	0.002	0.0321	Ki67	KLRG1	CD44	S4
57	57	CD138.Ki67.KLRG1.absRange.S3	17	0.002	0.1441	CD138	Ki67	KLRG1	S3
58	57	CD62L.CD86.Ly6C.absRange.S2	17	0.002	0.1395	CD62L	CD86	Ly6C	S2
59	57	CD86.Ki67.MHC-II.absRange.S3	17	0.002	0.026	CD86	Ki67	MHC-II	S3
60	57	CD90.CD138.CD27.absRange.S2	17	0.002	0.9323	CD90	CD138	CD27	S2
61	57	PD.L1.Ki67.CD86.absRange.S3	17	0.002	0.0469	PD.L1	Ki67	CD86	S3
62	57	T.bet.CD44.MHC-II.absRange.S2	17	0.002	0.5219	T.bet	CD44	MHC-II	S2
63	63	CD44.PD.L1.KLRG1.absRange.S4	16	0.002	0.0185	CD44	PD.L1	KLRG1	S4
64	63	CD62L.CD86.CD44.absRange.S3	16	0.002	0.127	CD62L	CD86	CD44	S3
65	65	CD86.CD138.MHC-II.absRange.S3	15	0.002	0.5879	CD86	CD138	MHC-II	S3
66	65	CD86.CD44.KLRG1.absRange.S3	15	0.002	0.2922	CD86	CD44	KLRG1	S3
67	65	Ki67.KLRG1.T.bet.absRange.S4	15	0.002	0.0169	Ki67	KLRG1	T.bet	S4
68	68	CD44.PDCA.1.CD45.absRange.S3	14	0.002	1.5084	CD44	PDCA.1	CD45	S3
69	68	T.bet.CD90.Ly6C.absRange.S3	14	0.002	0.3324	T.bet	CD90	Ly6C	S3
70	70	CD90.CD103.RORgt.absRange.S3	13	0.002	0.1861	CD90	CD103	RORgt	S3
71	71	CD44.CD138.KLRG1.absRange.S3	12	0.002	0.0185	CD44	CD138	KLRG1	S3
72	71	CD69.CD90.CD86.absRange.S3	12	0.002	0.0175	CD69	CD90	CD86	S3
73	71	T.bet.CD90.PD.L1.absRange.S4	12	0.002	-0.1449	T.bet	CD90	PD.L1	S4
74	74	CD90.KLRG1.CD138.absRange.S4	11	0.001	0.1075	CD90	KLRG1	CD138	S4
75	74	CD90.KLRG1.PD.L1.absRange.S2	11	0.001	-0.3587	CD90	KLRG1	PD.L1	S2
76	76	CD86.CD138.CD27.absRange.S4	10	0.001	0.1397	CD86	CD138	CD27	S4
77	76	CD86.CD90.T.bet.absRange.S2	10	0.001	-0.0139	CD86	CD90	T.bet	S2
78	78	CD90.KLRG1.SiglecF.absRange.S4	9	0.001	0.0238	CD90	KLRG1	SiglecF	S4
79	79	CD44.CD138.CD86.absRange.S4	8	0.001	0.029	CD44	CD138	CD86	S4
80	79	CD44.Ki67.CD27.absRange.S2	8	0.001	0.5716	CD44	Ki67	CD27	S2
81	79	T.bet.CD90.CD16.32.absRange.S3	8	0.001	0.2764	T.bet	CD90	CD16.32	S3
82	82	CD90.KLRG1.Ly6C.absRange.S4	7	0.001	0.047	CD90	KLRG1	Ly6C	S4
83	83	CD44.CD103.CD90.absRange.S3	6	0.001	0.0466	CD44	CD103	CD90	S3
84	83	CD86.CD138.CD64.absRange.S4	6	0.001	0.0383	CD86	CD138	CD64	S4
85	85	CD69.CD90.MHC-II.absRange.S2	3	<0.001	0.046	CD69	CD90	MHC-II	S2
86	85	CD86.CD90.Ly6C.absRange.S3	3	<0.001	0.0156	CD86	CD90	Ly6C	S3
87	87	CD90.Ki67.CD103.absRange.S2	2	<0.001	0.0793	CD90	Ki67	CD103	S2
88	87	CD90.Ki67.PD.L1.absRange.S4	2	<0.001	-0.0018	CD90	Ki67	PD.L1	S4
89	87	Ki67.KLRG1.CD11c.absRange.S2	2	<0.001	-0.0486	Ki67	KLRG1	CD11c	S2
90	87	Ki67.KLRG1.PDCA.1.absRange.S3	2	<0.001	0.1476	Ki67	KLRG1	PDCA.1	S3
91	91	CD138.Ki67.PD.L1.absRange.S4	1	<0.001	-0.0101	CD138	Ki67	PD.L1	S4
92	91	CD62L.Ki67.PD.L1.absRange.S4	1	<0.001	-0.081	CD62L	Ki67	PD.L1	S4
93	91	CD90.CD16.32.KLRG1.absRange.S3	1	<0.001	0.017	CD90	CD16.32	KLRG1	S3
94	91	CD90.KLRG1.CD11c.absRange.S4	1	<0.001	0.0085	CD90	KLRG1	CD11c	S4
95	91	CD90.PDCA.1.CD27.absRange.S2	1	<0.001	0.0082	CD90	PDCA.1	CD27	S2
96	91	CD90.PDCA.1.T.bet.absRange.S2	1	<0.001	0.0181	CD90	PDCA.1	T.bet	S2
97	91	PD.L1.Ki67.Foxp3.absRange.S2	1	<0.001	0.1973	PD.L1	Ki67	Foxp3	S2

Overview of CD90-PD.L1-MSI(CD86)

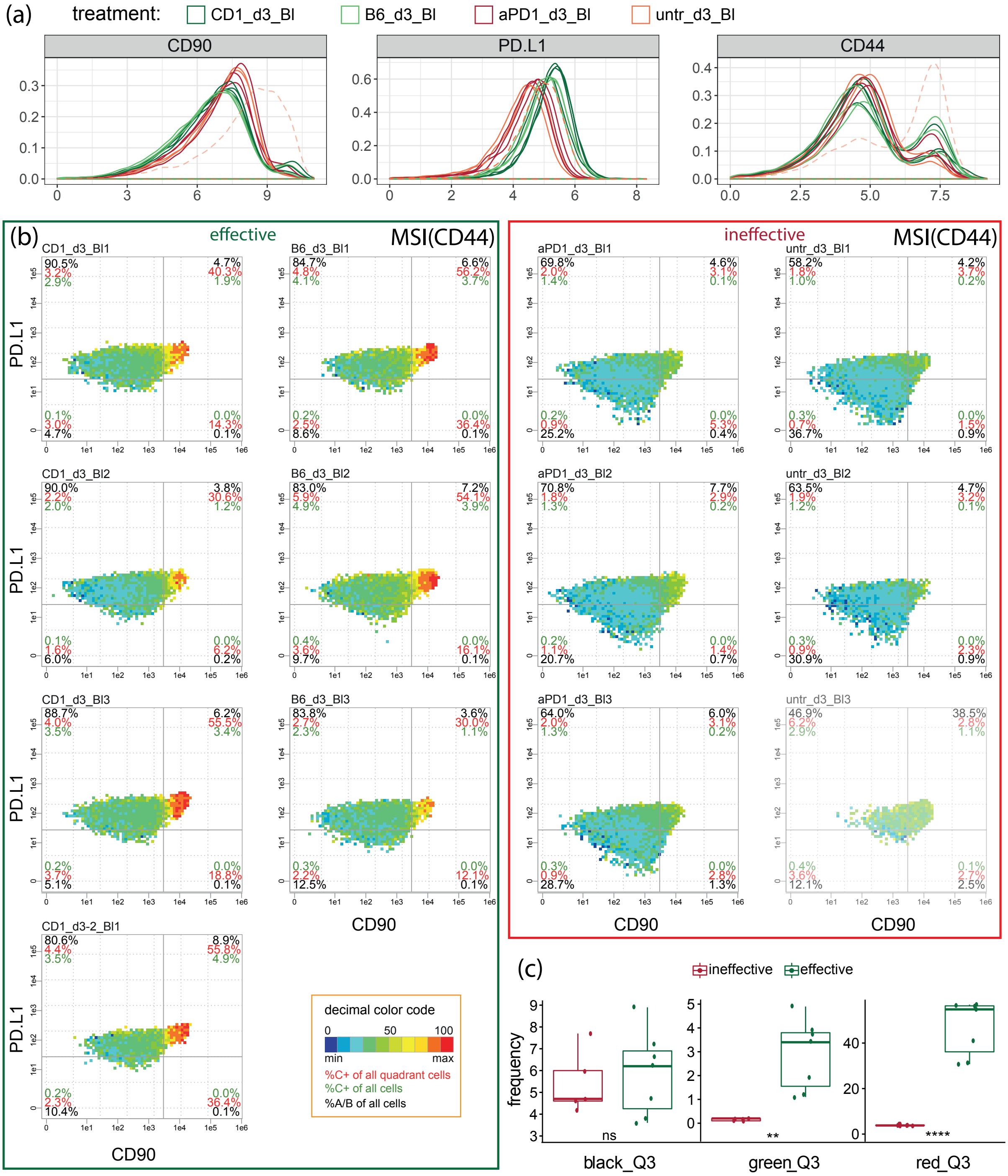


Figure S11: Overview of CD90-PD.L1-MSI(CD86) from all samples. High variation in density and frequencies, but a concentrated area in eff treatment due to the expression pattern in triploTs is still tangible. **(a)** Density plots from all blood samples without zero SI entries, dashed line indicate sample *untr_ d3_ Bl3*. **(b)** TriploTs of CD90-PD.L1-MSI(CD86) from all samples, the sample *untr_ d3_ Bl3* is paled, since it was filtered out after data preparation. **(c)** Box plots of percentages in top right quadrant of all samples except of sample *untr_ d3_ Bl3* with unpaired two-sided *t*-test, labeled with asterisks: *p*-values ≤ 0.0001(****), $p \leq 0.01$ (**) and $p > 0.05$ (ns).

TriploT Overview of CD90-CD86-MSI from sample *B6_ d3_ Bl2*

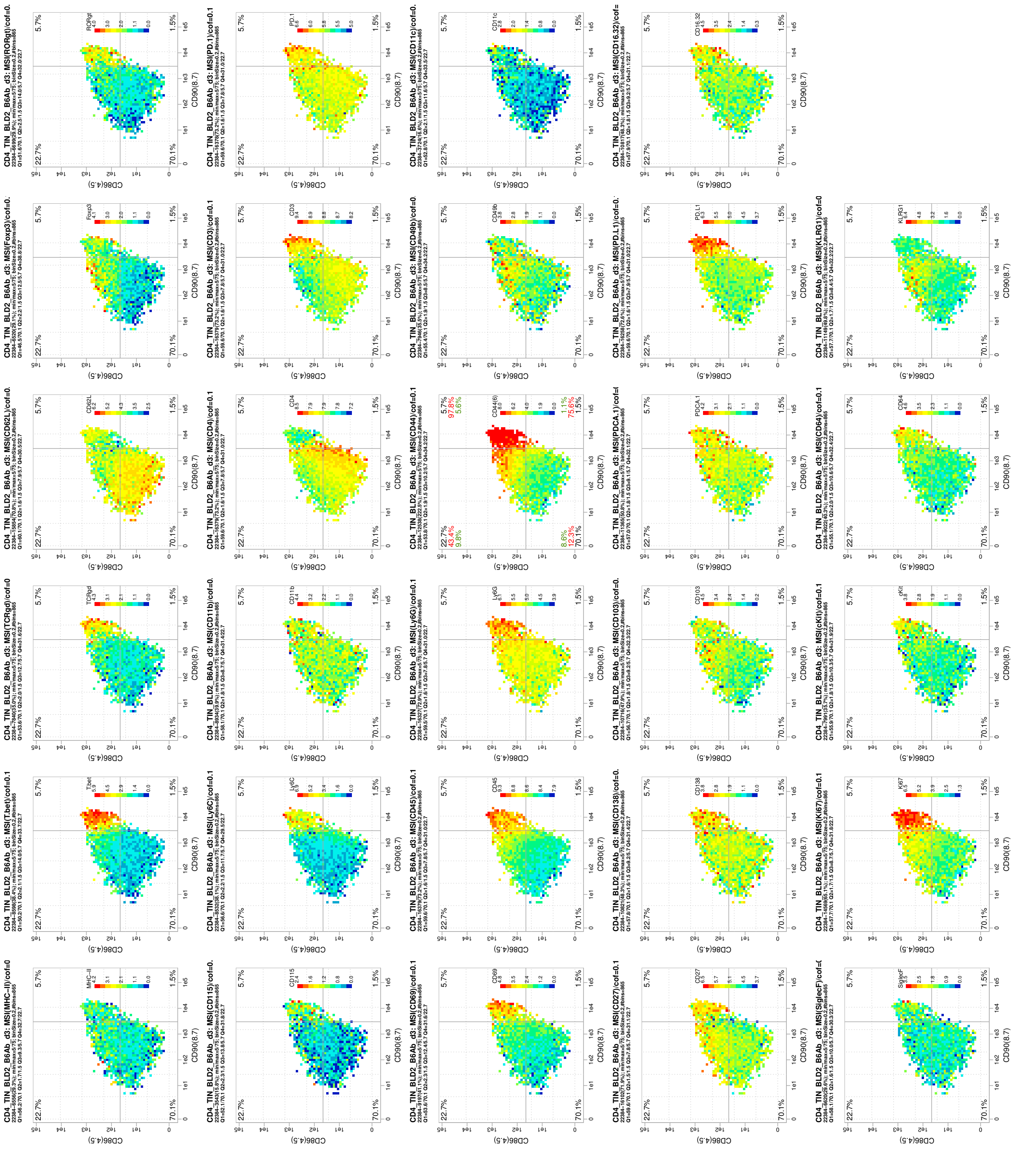


Figure S12: TriploT Overview of CD90-CD86-MSI(all biological markers) from sample *B6_ d3_ Bl2* with dynamic ranges showing highest MSIs of the respective marker in red in lowest in blue. Grey continuous lines indicate cutoffs of CD90$^{\text{high}}$ (vertical at 8.7) and CD86^{+} (horizontal at 4.5), respectively. Created with *PRI-ana*.

TriploT Overview of CD90-CD86-MSI from sample *untr_d3_Bl1*

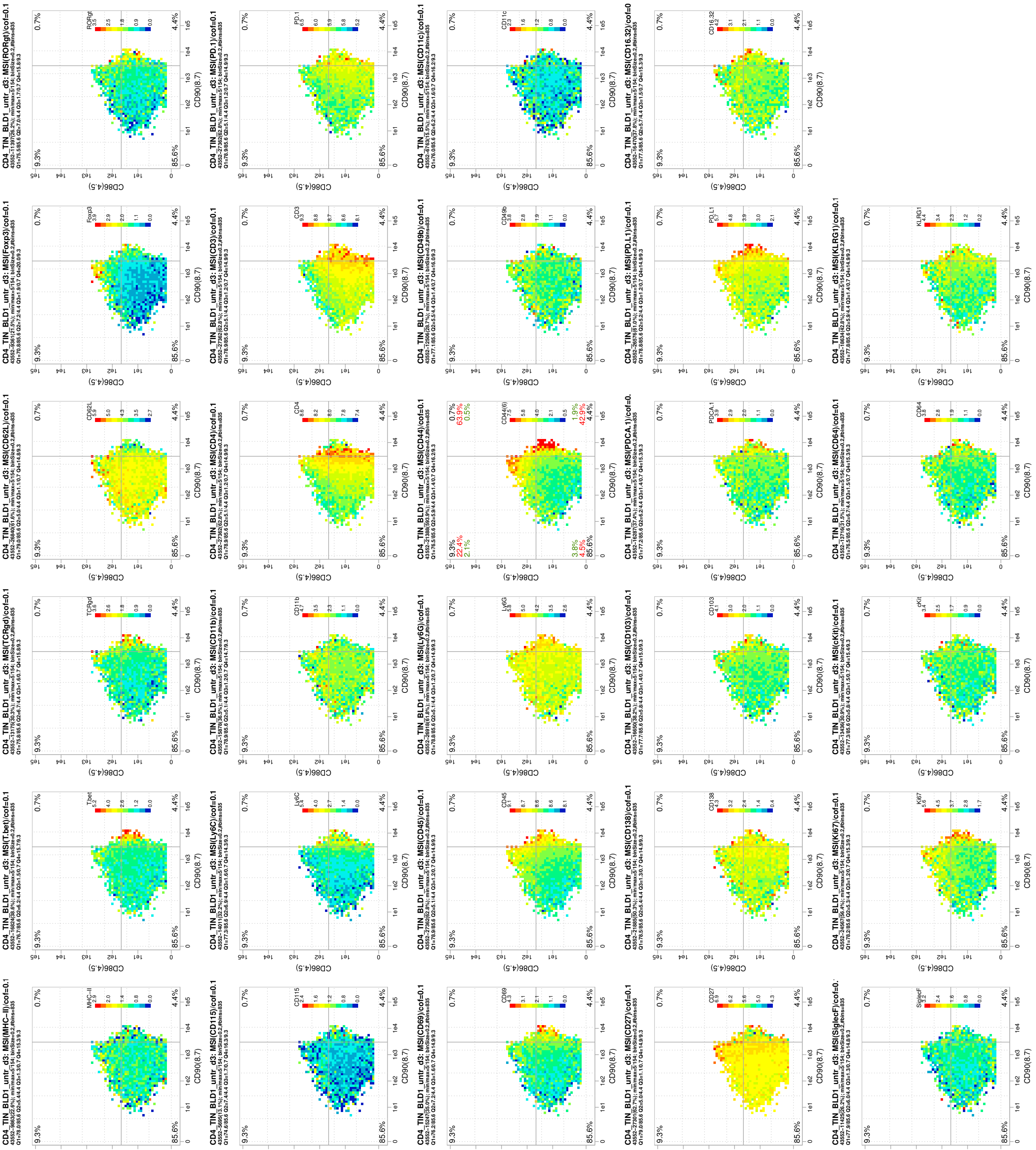


Figure S13: TriploT Overview of CD90-CD86-MSI(all biological markers) from sample *untr_d3_Bl1* with dynamic ranges showing highest MSIs of the respective marker in red in lowest in blue. Grey continuous lines indicate cutoffs of CD90$^{\mathrm{high}}$ (vertical at 8.7) and CD86^{+} (horizontal at 4.5), respectively. Created with *PRI-ana*.

Box plots of triploT section values

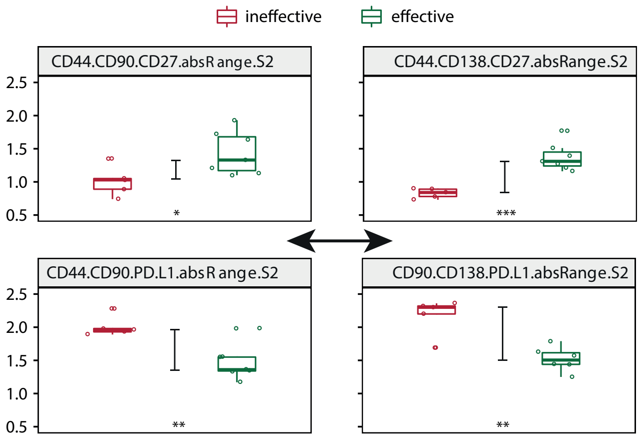


Figure S14: Comparison of triploT section values from different basis marker combinations. Box plots of CD44-CD90 (left), and CD44-CD138 and CD90-CD138 (right) with unpaired two-sided t-test (labeled with asterisks: p-values≤0.0001(****),$p \leq 0.01$(**)), and with the range distance of the medians between eff and ineff treatment.

Bar plot of frequencies of CD90$^+$PD-L1$^+$

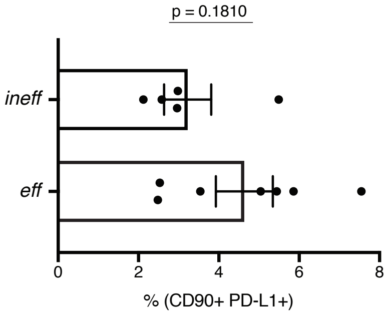


Figure S15: Bar plot of frequencies after manual gating to CD90$^+$PD-L1$^+$. Significance test with unpaired two-sided t-test on frequencies, and data is presented as the mean+s.e.m.

TriploTs with different bin sizes

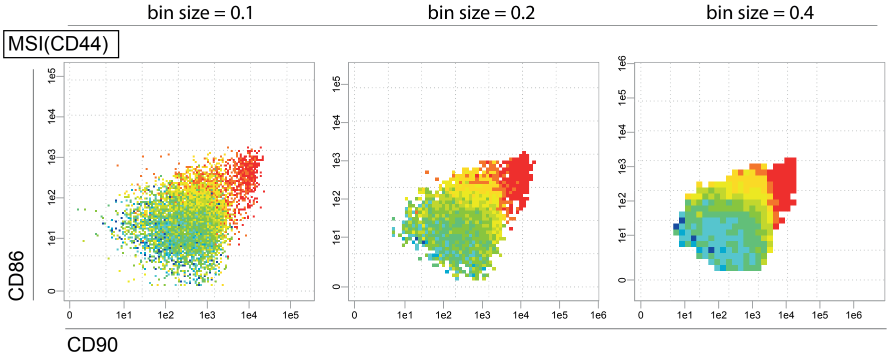
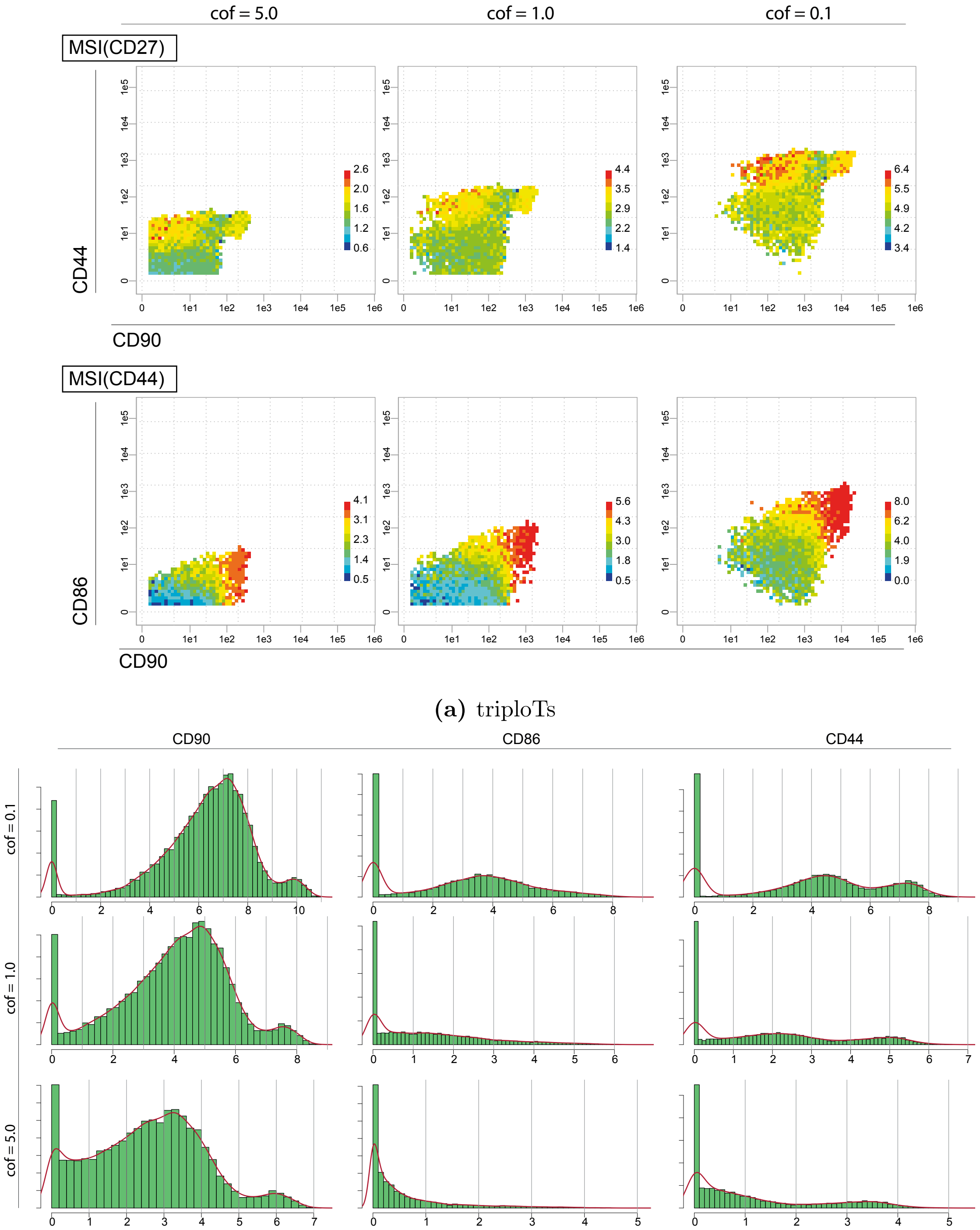


Figure S16: TriploTs with different bin sizes show that the pattern does not change. Exemplified on sample *B6_ d3_ Bl2* with CD90-CD86-MSI(CD44). Minimum number of cells= 2, 5, 10 are set relative to the bin size.

Effects of different *arcsinh* cofactors



(a) triploTs

(b) density plots

Figure S17: Effects of different *arcsinh* cofactors. **(a)** TriploTs of CD90-CD44-MSI(CD27) and CD90-CD86-MSI(CD44) are displayed with *arcsinh* co-factor 5.0 (commonly used [10, 14, 54, 62]), 1.0 (no cofactor) and 0.1 (herein applied). **(b)** Density plots with different co-factors are displayed for CD90, CD86 and CD44 are displayed with co-factor 0.1 (top), 1.0 (middle), and 5.0 (bottom). Sample *B6_ d3_ Bl2* is presented.

C Appendices: Results from state-of-the-art analysis tools

FlowJo's *Color Maps*

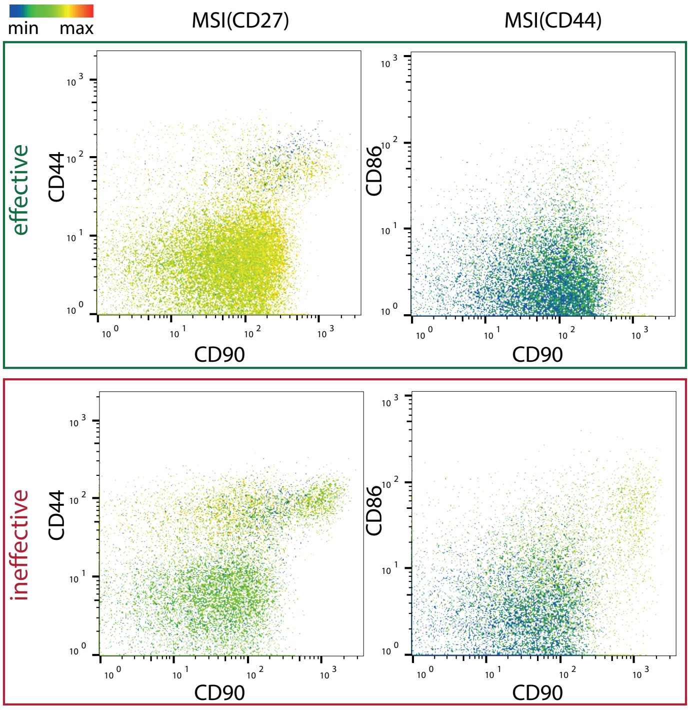


Figure S18: *FlowJo*'s *Color Maps* of CD90-CD44-MSI(CD27) (left) and of CD90-CD86-MSI(CD44) (right) from sample *B6_d3_Bl2* in eff treatment and sample *untr_d3_Bl1* in ineff treatment, respectively. Colormaps uses also fixed-width bins (of unknown size) to display a third marker in a color-coded manner with global scale between both treatments.

viSNE maps from two different runs

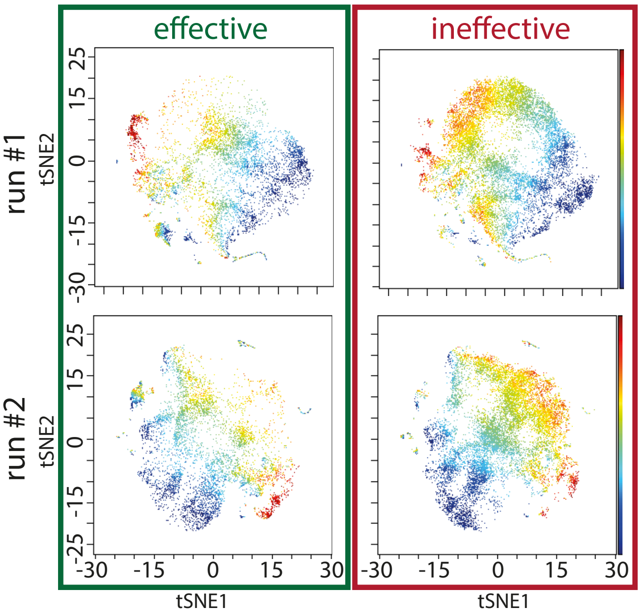


Figure S19: *viSNE* maps from two different runs using all blood samples except of sample *untr_d3_Bl3*, showing the SIs of marker CD90 with global scale between the samples of eff and ineff treatment. Default settings: total events=100,000, proportional sampling; iterations=1,000; perplexity=30; theta=0.5; random seed.

Citrus' feature tree

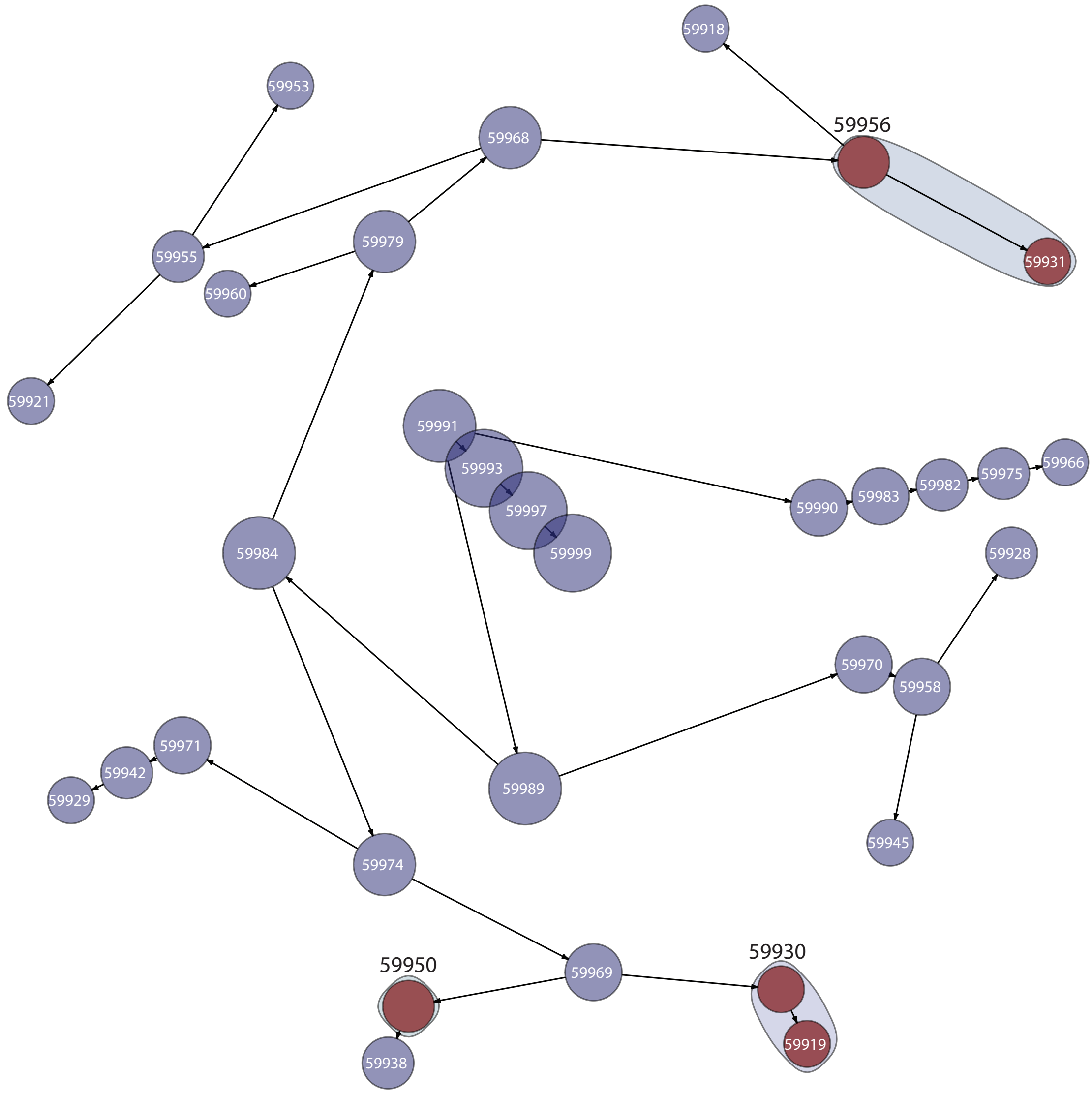


Figure S20: Cluster tree. Cell clusters colored in dark red are selected from *Citrus'* algorithm that explain differences between eff and ineff treatment, using cross-validation error rate within 1 standard error of the minimum model. Default options: Compensation=File-internal; Cluster characterization= abundance; Event sampling=5,000; Event sampling method=equal; Minimum cluster size=5%; CV folds=5; FDR=1%; Normalize scales=false; Transform cofactor=0.1; Association models=glmnet.

This run took 1,406 seconds on 1 cluster. With increase on cluster size, the processing time could be reduced by half (with 3 nodes) or by five (with 5 nodes). A comparable example run with configuration 'Event sampling=50,000' (almost all cells in the samples) results to a process time of 23,210 seconds with 5 nodes. Different runs (with different nodes) result in different outcomes in cluster count, size and characteristics of the clusters. The first run is used for further inspecting.

Overlapped densities of *Citrus'* cluster ID 59965

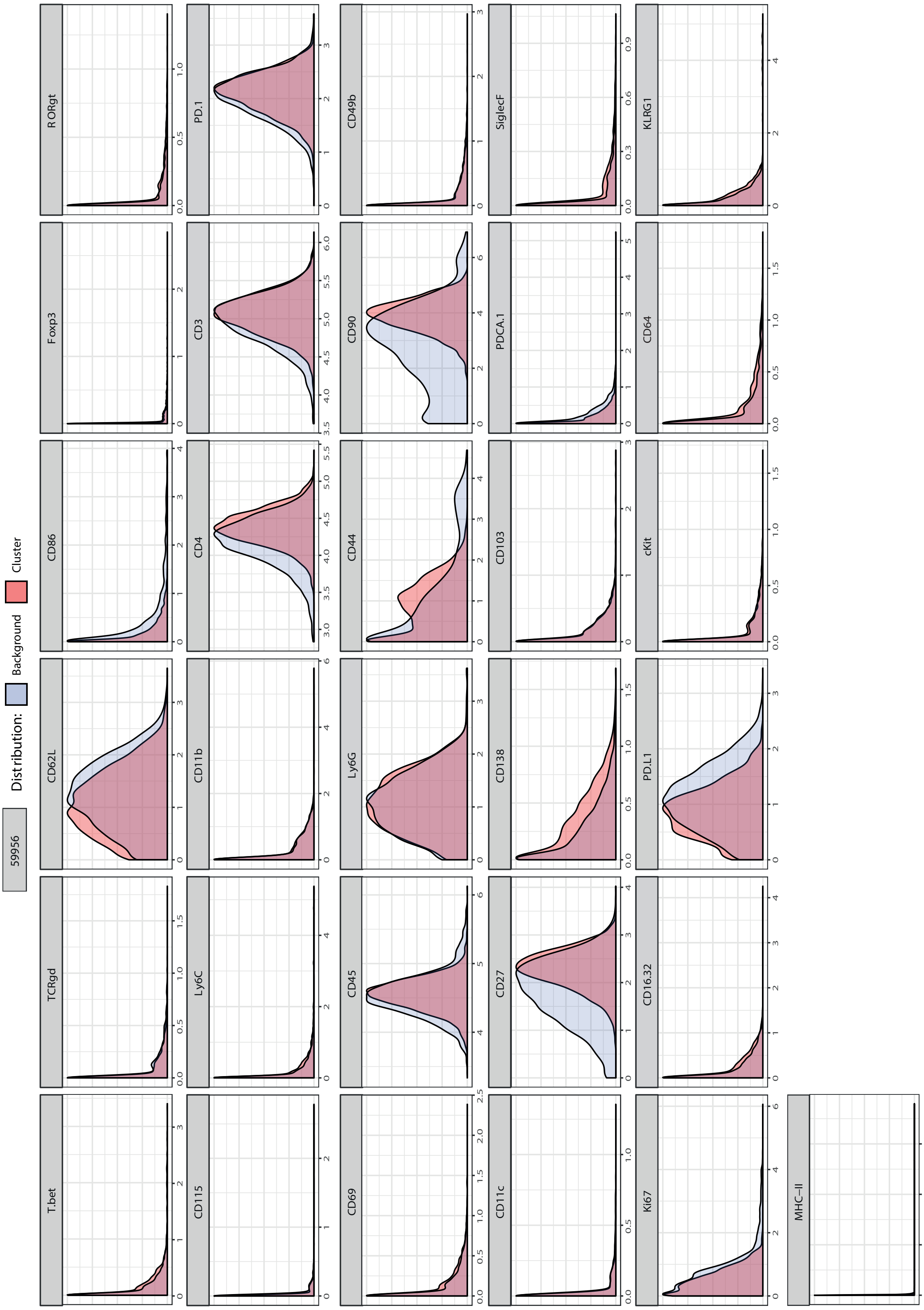


Figure S21: Overlapped densities of *Citrus'* cluster ID 59965 and residual down-sampled data.

D Appendices: *R* code and pseudo code

Code availability

The *R* code is available on `https://github.com/yhoang/dissertation`.

R packages

MDS plot was created with the function *plotMDS* from the *R* package *limma* (v3.38.3) (Fig. 3.1a) [96]. Bar and box plots are created with *R* package *ggplot2* (v3.1.0) (Fig. 3.1b+c) [97]. *R* package *ggpubr* (v0.2.999) was used for plotting the density overview (Fig. S2) [98]. Several single channel normalization methods were herein tested from the *R* package *cydar* (v1.6.1) [99].

The *R* packages *RSQlite* (v2.2.1), *tcltk2* (v1.2.11) and *R.devices* (v2.16.0) are used within *PRI-ana* [100, 101, 102]. The first one is to connect and interact with the databases and the second one enables *tcl* commands and *Tk* widgets to work in *R*. *tcl* is a dynamic programming language and *Tk* is a cross-platform graphical user interface (GUI) for *tcl*. The last package facilitates the creation of PDF files.

Running time of implemented *R* scripts on the example data set

A powerful computer with Intel Xeon(R) CPU E5-2667 v3 @3.20GHz x 32 and 16GB RAM was used for this study, on operation system Ubuntu 64-Bit 16.04.5 LTS.

Calculation of triploT matrices

The sample *CD1_d3_Bl1* has 23,081 cells and 41 marker proteins. To creates 31,980 matrices is processed in 4,202 seconds ($\sim$ 70 minutes) with *lapply()*. The outcoming object list has the size of 1.1 GB. Saving this object as an RDS file[7] leads to a drastic size reduction to 17.3 MB. The respective meta data is 17,141,328 bytes (RDS.file $\sim$ 137 KB) big. To speed up this process, *R* packages *doParallel* (v1.0.14) and *foreach* (v1.4.4) are used in combination with *R* lists [103, 104]. The script is modified in compliance with these packages, resulting to a decrease to 3,165 seconds and 2,482 seconds, when using 3 and 5 cores, respectively. R packages *reshape2* (v1.4.3) and *dplyr* (v0.7.8) are applied for data manipulation [105, 106].

Calculation of triploT sections

434 seconds are required to create the section values for sample *B6_d3_Bl2*. The implementation allows for using multiple CPUs with *R* packages *doParallel* (v1.0.14) and *foreach* (v1.4.4). Processing time is reduced by half when using 3 cores.

[7] saveRDS() serializes the object and then saves it with gzip compression.

Cross-validation

The first CV is processed in approximately 130 seconds. The second process needs roughly 100 seconds. No parallelization is deployed. *R* package *glmnet* (v2.0-16) is used for using erLR and for CV [107].

Top 0.05% quantile outlier removal

```r
trim.size <- 0.0005
ncells.trimmed <- 0
for (t in ungated.columns) {
  trim.idx <- which(data[,t] > quantile(data[,t],c(1 - trim.size)))
  data <- data[-trim.idx,]
  ncells.trimmed <- ncells.trimmed + length(trim.idx)
}
```

Listing 1: OutlierRemoval.R

Check for marker names and order

```r
for ( i in 1:length(metadata.d3$original_file_name) ) {
  db_index = which(this$current.filenames==metadata.d3$original_file_name[i])

  ### get marker names in database
  this$selected.vars = this$getVariables(index=db_index)

  if ( i==1 ) {
    vars.order = this$selected.vars
  } else {
    if ( !all(vars.order==this$selected.vars) ) {
      print("Wrong order. Check vars list!")
      print(vars.order)
      print(this$selected.vars)
    }
  }
}
```

Listing 2: CheckMarkerOrders.R

Calculating triploT matrices in one sample

```
find: global(min(x),min(y),max(x),max(y))
find: marker, len.marker

### parallelize:
for sample in Samples:
  for m1 in 1 to (len.marker -1):
    for m2 in (m1+1) to len.marker:
      it = it +1

      tmp.marker = marker[-c(m1,m2)]

      for each m3 in tmp.marker do:
        sampl.data = data[sample, c(m1, m2, m3)]
        meta = c(sample, condition, calc_meth, markers, range)
        sampl_mat = calc_triplot_matrix(sampl.data)

        triplot_mat = append(triplot_mat,meta,sampl_mat)
      }
    }
  }
}
```

Listing 3: Pseudocode for calculating triplot matrices.

Calculation of triploT sections

```
find(marker); len.marker = length(marker)

### look at batch size calculation
init_df(quadrant.dataframe, columns =(len.marker-2)(len.marker-3)(len.marker-4)*2,
     rows = length(Samples) )

### parallelize:
for ( sample in Samples) {
  init_vector(triplot_quads)

  for (m1 in 1:(len.marker -1) ) {
    for ( m2 in (m1+1):len.marker ) {

      tmp.marker = marker[-c(m1,m2)]
      for each ( m3 in tmp.marker ) do {
        sampl.data = data[sample, c(m1, m2, m3)]
        ### calculate triplot quadrants
        triplot_quads = calc_triplot_quadrants(sampl.data)
      }
      combine_columns(triplot_quads,triplot_quads)
    }
    combine_columns(triplot_quads,triplot_quads)
  }
  combine_rows(quadrant.dataframe,triplot_quads)
}
save(quadrant.dataframe)
```

Listing 4: Pseudocode for calculating triplot sections (modified for parallele use).

Code chunk of erGLM-CV to find α

```r
alphalist <- seq(0,1,by=0.1)
it.total = it.run = 0

while (it.run < 100) {
    # use a certain seed in whole run for resampling
    set.seed(seed.vec[it.total]);
    set.foldid = sample(rep(seq((1/3)*nrow(data)),length=nrow(data)))

    if (
      all(df.total[which(set.foldid==1),typeColNum]==0) |
      all(df.total[which(set.foldid==2),typeColNum]==0) |
      all(df.total[which(set.foldid==3),typeColNum]==0) |
      all(df.total[which(set.foldid==1),typeColNum]==1) |
      all(df.total[which(set.foldid==2),typeColNum]==1) |
      all(df.total[which(set.foldid==3),typeColNum]==1)
    ) next;
    it.run = it.run + 1

    elasticnet <- lapply(alphalist, function(a) {
        cv.glmnet( x=as.matrix(data[,-condition]), y=data$condition,
            alpha=a, family="binomial",lambda.min.ratio=.0005,
            type.measure="deviance",foldid = set.foldid
        )
    })

    min.err = vector()
    for (i in 1:11) {
      min.err = c(min.err,min(elasticnet[[i]]$cvm))
    }
    min.err.idx = which(min.err==min(min.err)) # index for best performing alpha
    alpha.collect = c(alpha.collect, alphalist[min.err.idx])
}
```

Listing 5: erGLM-CV.R

Warning message using cv.glmnet()

```
In lognet(x, is.sparse, ix, jx, y, weights, offset, alpha, nobs,  :
  one multinomial or binomial class has fewer than 8  observations; dangerous ground
```

Listing 6: Warning message using cv.glmnet()

Approximation function to set cutoffs automatically

```r
### call density and get x and y values
d=density(tdata)
x=d$x
y=d$y

### initialize
stepsize = 3
incline.x = incline.diff = vector()
idx.minima = idx.maxima = idx.shoulder = idx.remove = vector()

### first: save ranges from each position
for (range in (1+stepsize):(length(y)-stepsize)) {
  # save x-value and  difference in range (x[range-stepsize],x[range+step])
  incline.x[range] = x[range]
  incline.diff[range] = diff(c(y[range-stepsize],y[range+stepsize]))
}

### second: check for minima, maxima and shoulders
for (inc in 1:(length(incline_list)-1)) {

  ### check if there is at least one minima: -/+ change
  if (incline.diff[inc]<0 & incline.diff[inc+1]>0) {
    idx.minima=c(idx.minima,inc)
  }

  ### check if there is at least one maxima: +/- change
  if (incline.diff[inc]>0 & incline.diff[inc+1]<0) {
    idx.maxima=c(idx.maxima,inc)
  }

  ### check if there is a shoulders left and right of a peack
  if (abs(incline.diff[inc])<0.01) {
    idx.shoulder=c(idx.shoulder,inc)
  }
}

### third: substract maxima indices from minima and shoulder indices
# remove minima indices near maxmia
minus.minima=intersect(idx.maxima,idx.minima)
if (length(minus.minima)>0) {
  for (j in (minus.minima-3):(minus.minima+3)) {
    idx.remove = c(idx.remove,which(idx.minima==j))
  }
  idx.minima = idx.minima[-idx.remove]
}
# remove shoulder indices near minima x+-0.3
minus.shoulder=intersect(idx.maxima,idx.shoulder)
if (length(minus.shoulder) > 0) {
  x.shoulder=incline.x[minus.shoulder]
  for (k in idx.shoulder) {
    if (abs(x.shoulder-incline.x[k])<=0.3) idx.remove=c(idx.remove,which(idx.shoulder==k))
  }
  idx.shoulder = idx.shoulder[-idx.remove]
}

if (length(idx.minima)>0) { # if there is a minima
  cutoff = incline.x[idx.minima[1]]
} else if (length(idx.shoulder)>0 ) { # if there is a shoulder
  cutoff = incline.x[idx.shoulder[1]]
} else { # if there is no minima nor shoulder found, than set cutoff at 20% quantile
  cutoff = quantile(tdata,0.8)
}
```

Listing 7: AutomatedCutoff.R

R sessionInfo()

```
> sessionInfo()
R version 3.5.1 (2018-07-02)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.5 LTS

Matrix products: default
BLAS: /usr/lib/libblas/libblas.so.3.6.0
LAPACK: /usr/lib/lapack/liblapack.so.3.6.0

locale:
 [1] LC_CTYPE=de_DE.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=de_DE.UTF-8
 [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=de_DE.UTF-8
 [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C

attached base packages:
 [1] stats4    parallel  tcltk     stats     graphics  grDevices datasets
 [8] utils     methods   base

other attached packages:
 [1] glmnet_2.0-16               Matrix_1.2-15
 [3] limma_3.38.3                cydar_1.6.1
 [5] SingleCellExperiment_1.4.1  SummarizedExperiment_1.12.0
 [7] DelayedArray_0.8.0          matrixStats_0.54.0
 [9] Biobase_2.42.0              GenomicRanges_1.34.0
[11] GenomeInfoDb_1.18.1         IRanges_2.16.0
[13] S4Vectors_0.20.1            BiocGenerics_0.28.0
[15] BiocParallel_1.16.6         doParallel_1.0.14
[17] iterators_1.0.10            foreach_1.4.4
[19] R.devices_2.16.0            tcltk2_1.2-11
[21] RSQLite_2.1.1               ggpubr_0.2.999
[23] reshape2_1.4.3              limma_3.38.3

loaded via a namespace (and not attached):
 [1] viridis_0.5.1          viridisLite_0.3.0      bit64_0.9-7
 [4] R.utils_2.7.0          shiny_1.2.0            assertthat_0.2.0
 [7] blob_1.1.1             GenomeInfoDbData_1.2.0 robustbase_0.93-3
[10] pillar_1.3.1           lattice_0.20-38        glue_1.3.0
[13] digest_0.6.18          promises_1.0.1         XVector_0.22.0
[16] colorspace_1.4-0       htmltools_0.3.6        httpuv_1.4.5.1
[19] R.oo_1.22.0            plyr_1.8.4             pcaPP_1.9-73
[22] pkgconfig_2.0.2        zlibbioc_1.28.0        purrr_0.3.0
[25] flowCore_1.48.1        xtable_1.8-3           corpcor_1.6.9
[28] mvtnorm_1.0-8          scales_1.0.0           later_0.8.0
[31] tibble_2.0.1           ggplot2_3.1.0          lazyeval_0.2.1
[34] magrittr_1.5           crayon_1.3.4           mime_0.6
[37] memoise_1.1.0          R.methodsS3_1.7.1      MASS_7.3-51.1
[40] graph_1.60.0           tools_3.5.1            munsell_0.5.0
[43] cluster_2.0.7-1        bindrcpp_0.2.2         compiler_3.5.1
[46] rlang_0.3.1            grid_3.5.1             RCurl_1.95-4.11
[49] BiocNeighbors_1.0.0    bitops_1.0-6           base64enc_0.1-3
[52] gtable_0.2.0           codetools_0.2-16       DBI_1.0.0
[55] rrcov_1.4-7            R6_2.3.0               gridExtra_2.3
[58] dplyr_0.7.8            bit_1.1-14             bindr_0.1.1
[61] Rcpp_1.0.0             DEoptimR_1.0-8         tidyselect_0.2.5
```

Listing 8: sessionInfo()

Bibliography

[1] C. Chester and H. T. Maecker, "Algorithmic Tools for Mining High-Dimensional Cytometry Data.," *Journal of Immunology*, vol. 195, pp. 773–9, Aug 2015.

[2] Y. Saeys, S. V. Gassen, and B. N. Lambrecht, "Computational flow cytometry: helping to make sense of high-dimensional immunology data," *Nature Reviews Immunology*, vol. 16, pp. 449–62, Jun 2016.

[3] E. W. Newell and Y. Cheng, "Mass cytometry: Blessed with the curse of dimensionality," vol. 17, pp. 890–5, Jul 2016.

[4] L. Olsen, M. Leipold, C. Pedersen, and H. Maecker, "The anatomy of single cell mass cytometry data," *Cytometry Part A*, vol. 95, pp. 156–72, Oct 2018.

[5] Bellman, R., Bellman, R.E., and K. M. R. Collection, *Adaptive Control Processes: A Guided Tour.* Princeton legacy library, Princeton University Press, 1961.

[6] L. Zhang, M. Kuhn, I. Peers, and S. Altan, *Nonclinical Statistics for Pharmaceutical and Biotechnology Industries (Statistics for Biology and Health).* Springer, 1 ed., 2016.

[7] P. Bacchetti, S. G. Deeks, and J. M. McCune, "Breaking free of sample size dogma to perform innovative translational research," *Science Translational Medicine*, vol. 3, p. 87ps24, Jun 2011.

[8] R. Melchiotti, F. Gracio, S. Kordasti, A. K. Todd, *et al.*, "Cluster Stability in the Analysis of Mass Cytometry Data," vol. 91, pp. 73–84, Jan 2017.

[9] M. H. Spitzer and G. P. Nolan, "Mass Cytometry: Single Cells, Many Features," *Cell*, vol. 165, pp. 780–91, May 2016.

[10] L. M. Weber and M. D. Robinson, "Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data," *Cytometry Part A*, vol. 89, pp. 1084–96, Dec 2016.

[11] P. Kvistborg, C. Gouttefangeas, N. Aghaeepour, A. Cazaly, *et al.*, "Thinking Outside the Gate: Single-Cell Assessments in Multiple Dimensions," vol. 42, pp. 591–2, Apr 2015.

[12] E.-a. D. Amir, K. L. Davis, M. D. Tadmor, E. F. Simonds, *et al.*, "viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia," *Nature Biotechnology*, vol. 31, pp. 545–52, Apr 2013.

[13] M. H. Spitzer, Y. Carmi, N. E. Reticker-Flynn, S. S. Kwek, *et al.*, "Systemic Immunity Is Required for Effective Cancer Immunotherapy," *Cell*, vol. 168, pp. 487–502, Jan 2017.

[14] R. V. Bruggner, B. Bodenmiller, D. L. Dill, R. J. Tibshirani, *et al.*, "Automated identification of stratifying signatures in cellular subpopulations," *Proceedings of the National Academy of Sciences*, vol. 111, pp. E2770–7, May 2014.

[15] A. Cossarizza, H.-D. Chang, A. Radbruch, I. Andrä, *et al.*, "Guidelines for the use

of flow cytometry and cell sorting in immunological studies," *European Journal of Immunology*, vol. 47, pp. 1584–797, Oct 2017.

[16] V. van Unen, T. Höllt, N. Pezzotti, N. Li, *et al.*, "Visual analysis of mass cytometry data by hierarchical stochastic neighbour embedding reveals rare cell types," *Nature Communications*, p. 1740, Nov 2017.

[17] S. Quake and G. Worthington, "Fluidigm corporation." `https://www.fluidigm.com/`, 1999. Accessed 2019-07-07, 09:17.

[18] J. A. Lee, J. Spidlen, K. Boyce, J. Cai, *et al.*, "MIFlowCyt: The Minimum Information about a Flow Cytometry Experiment from the International Society for Analytical Cytology Data Standards Task Force," *Cytometry A*, vol. 73, pp. 926–30, Aug 2008.

[19] Y. D. Mahnke and M. Roederer, "Optimizing a Multicolor Immunophenotyping Assay," *Clinics in Laboratory Medicine*, vol. 27, pp. 469–85, Sep 2007.

[20] B. Biosciences, "Bd facsymphony." `http://www.bdbiosciences.com/en-us/instruments/research-instruments/research-cell-analyzers/facsymphony`, 1897. Accessed 2019-07-08, 18:11.

[21] M. Roederer, "Compensation in flow cytometry.," *Curr Protoc Cytom*, pp. 1.14.1–20, Dec 2002.

[22] M. D. Leipold, G. Obermoser, C. Fenwick, K. Kleinstuber, *et al.*, "Comparison of CyTOF assays across sites: Results of a six-center pilot study," *Journal of Immunological Methods*, vol. 453, pp. 37–43, Feb 2018.

[23] R. Finck, E. F. Simonds, A. Jager, S. Krishnaswamy, *et al.*, "Normalization of mass cytometry data with bead standards," *Cytometry Part A*, vol. 83, pp. 483–94, Mar 2013.

[24] E. Arvaniti and M. Claassen, "Sensitive detection of rare disease-associated cell subsets via representation learning," *Nature Communications*, p. 14825, Apr 2017.

[25] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent," *Journal of Statistical Software*, vol. 39, pp. 59–70, Mar 2011.

[26] M. H. Spitzer, P. F. Gherardini, G. K. Fragiadakis, N. Bhattacharya, and others, "An interactive reference framework for modeling a dynamic immune system," *Science*, vol. 349, Jul 2015.

[27] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature Extraction: Foundations and Applications*. Springer, 2006.

[28] R. N. Germain, "An innately interesting decade of research in immunology," *Nature Medicine*, vol. 10, pp. 1307–20, dec 2004.

[29] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–82, Mar 2003.

[30] Y. Hoang, J. Pfeil, M. Zagorščak, A. Thieffry, *et al.*, "Report on the "advanced big data training school for life sciences", barcelona 3th-7th september 2018," *EMBnet.journal*, vol. 24, Oct 2019.

[31] M. Andrew. Hall, *Correlation-Based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, Department of Computer Science, Apr 1999.

[32] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 129–34, AAAI Press, Jul 1992.

[33] Christopher Bishop, *Pattern Recognition and Machine Learning*. Springer, 1 ed., 2006.

[34] J. Trevor Hastie, Robert Tibshirani, *Statistical Learning with Sparsity - The Lasso and Generalizations*. Monographs on Statistics & Applied Probability (143), Chapman & Hall/CRC, 1 ed., 2015.

[35] A. Rencher and G. Schaalje, *Linear Models in Statistics*. Wiley, 2008.

[36] P. McCullagh and J. Nelder, *Generalized Linear Models*. Monographs on Statistics & Applied Probability (37), Chapman & Hall/CRC, 2 ed., 1989.

[37] J. Trevor Hastie, Robert Tibshirani, *The Elements of Statistical Learning*. Springer, 2 ed., 2008.

[38] D. Banks. `http://www2.stat.duke.edu/~banks/218-lectures.dir/dmlect9.pdf`. Accessed: 2019-04-03, 12:48.

[39] R. Tibshirani, "Regression shrinkage selection via the lasso," *Journal of the Royal Statistical Society Series B*, vol. 58, pp. 267–88, 1996.

[40] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, vol. 33, Aug 2010.

[41] C. A. Janeway and R. Medzhitov, "Innate Immune Recognition," *Annual Review of Immunology*, vol. 20, pp. 197–216, Apr 2002.

[42] S. L. Swain, K. K. McKinstry, and T. M. Strutt, "Expanding roles for CD4+ T cells in immunity to viruses," *Nature Reviews Immunology*, vol. 12, pp. 136–48, Feb 2012.

[43] J. M. Schenkel, K. A. Fraser, L. K. Beura, K. E. Pauken, V. Vezys, and D. Masopust, "Resident memory CD8 T cells trigger protective innate and adaptive immune responses," *Science*, vol. 346, pp. 98–101, Oct 2014.

[44] G. Finak, J. Frelinger, W. Jiang, E. W. Newell, J. Ramey, M. M. Davis, S. A. Kalams, S. C. De Rosa, and R. Gottardo, "OpenCyto: An Open Source Infrastructure for Scalable, Robust, Reproducible, and Automated, End-to-End Flow Cytometry Data Analysis," *PLoS Computational Biology*, vol. 10, pp. 1–12, Aug 2014.

[45] N. Kotecha, P. O. Krutzik, and J. M. Irish, "Web-based analysis and publication of flow cytometry experiments," *Current Protocols in Cytometry*, vol. 53, pp. 10.17.1–24, Jul 2010.

[46] H. G. Fienberg, E. F. Simonds, W. J. Fantl, G. P. Nolan, and B. Bodenmiller, "A platinum-based covalent viability reagent for single-cell mass cytometry," *Cytometry Part A*, vol. 81A, pp. 467–75, Jun 2012.

[47] T. R. Mosmann and R. L. Coffman, "Heterogeneity of cytokine secretion patterns

and functions of helper t cells," vol. 46 of *Advances in Immunology*, pp. 111–47, Academic Press, Jan 1989.

[48] A. K. Abbas, K. M. Murphy, and A. Sher, "Functional diversity of helper T lymphocytes," *Nature*, vol. 383, pp. 787–93, Oct 1996.

[49] I. Kadner, "Analysis of autoimmune disease-dependent protein expression using a web based platform for flow cytometry," Master's thesis, May 2017.

[50] W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson, *shiny: Web Application Framework for R*, 2018. R package version 1.2.0.

[51] B. Ellis, P. Haaland, F. Hahne, N. Le Meur, N. Gopalakrishnan, J. Spidlen, M. Jiang, and G. Finak, *flowCore: flowCore: Basic structures for flow cytometry data*, 2019. R package version 1.48.1.

[52] A. Bashashati and R. R. Brinkman, "A Survey of Flow Cytometry Data Analysis Methods," *Advances in Bioinformatics*, vol. 2009, pp. 1–19, Aug 2009.

[53] J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, *et al.*, "Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis," *Cell*, Jun 2015.

[54] P. Qiu, E. F. Simonds, S. C. Bendall, K. D. Gibbs, *et al.*, "Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE.," *Nature biotechnology*, vol. 29, pp. 886–91, Oct 2011.

[55] S. C. Bendall, E. F. Simonds, P. Qiu, E.-a. D. Amir, *et al.*, "Single-Cell Mass Cytometry of Differential Immune and Drug Responses Across a Human Hematopoietic Continuum," *Science*, vol. 332, pp. 687–96, May 2012.

[56] E. W. Newell, Y. Cheng, M. T. Wong, and L. Van Der Maaten, "Embedding Soli-Expression by Nonlinear Stochastic Heterogeneity with One-Dimensional Categorical Analysis of Human T Cell Categorical Analysis of Human T Cell Heterogeneity with One-Dimensional Soli-Expression by Nonlinear Stochastic Embedding," *The Journal of Immunology*, vol. 14, pp. 924–32, Jan 2017.

[57] TreeStar and Software, "Flowjo." `https://www.flowjo.com/`, 2013. Accessed 2018-03-07, 16:44.

[58] T. Janetzek, "Semi-automated high-dimensional mass cytometric data analysis using a two dimensional binning approach," Master's thesis, Jun 2019.

[59] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 67, pp. 301–20, Jan 2005.

[60] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate - a practical and powerful approach to multiple testing," *J. Royal Statist. Soc., Series B*, vol. 57, pp. 289 – 300, Mar 1995.

[61] A. K. Kimball, L. M. Oko, B. L. Bullock, R. A. Nemenoff, L. F. van Dyk, and E. T. Clambey, "A Beginner's Guide To Analyzing and Visualizing Mass Cytometry Data," vol. 200, pp. 3–22, Jan 2018.

[62] C. Krieg, M. Nowicka, S. Guglietta, S. Schindler, *et al.*, "High-dimensional single-cell

analysis predicts response to anti-PD-1 immunotherapy," *Nature Medicine*, vol. 24, pp. 144–53, 2018.

[63] K. O'Neill, N. Aghaeepour, J. Špidlen, and R. Brinkman, "Flow cytometry bioinformatics," *PLOS Computational Biology*, vol. 9, pp. 1–10, Dec 2013.

[64] K. Feher, J. Kirsch, A. Radbruch, H. D. Chang, and T. Kaiser, "Cell population identification using fluorescence-minus-one controls with a one-class classifying algorithm," *Bioinformatics*, vol. 30, no. 23, pp. 3372–78, 2014.

[65] M. Malek, M. J. Taghiyar, L. Chong, G. Finak, *et al.*, "FlowDensity: Reproducing manual gating of flow cytometry data by automated density-based cell population identification," *Bioinformatics*, vol. 31, pp. 606–7, Feb 2015.

[66] S. C. Bendall, G. P. Nolan, M. Roederer, and P. K. Chattopadhyay, "A deep profiler's guide to cytometry," *Trends in Immunology*, vol. 33, pp. 323–32, Apr 2012.

[67] S. Tricot, M. Meyrand, C. Sammicheli, and J. Elhmouzi-younes, "Evaluating the Efficiency of Isotope Transmission for Improved Panel Design and a Comparison of the Detection Sensitivities of Mass Cytometer Instruments," *Cytometry Part A*, vol. 87, pp. 357–68, Apr 2015.

[68] S. Chevrier, H. L. Crowell, V. R. Zanotelli, S. Engler, *et al.*, "Compensation of Signal Spillover in Suspension and Imaging Mass Cytometry," *Cell Systems*, vol. 6, pp. 612–20, Mar 2018.

[69] S. Gryzik, Y. Hoang, T. Lischke, and R. Baumgrass, "Pattern perception enabled the identification of a super-functional Tfh-like cell subpopulation in murine lupus-nephritis," *Nature Immunology*, (submitted Aug 2019).

[70] M. Roederer, a. Treister, W. Moore, and L. a. Herzenberg, "Probability binning comparison: A metric for quantitating univariate distribution differences," *Cytometry*, vol. 45, pp. 37–46, Aug 2001.

[71] Y. Shen, B. Chaigne-Delalande, R. W. Lee, and W. Losert, "CytoBinning: Immunological insights from multi-dimensional data," *PLoS ONE*, vol. 13, pp. 1–19, Oct 2018.

[72] X. Yang and P. Qiu, "Automatically generate two-dimensional gating hierarchy from clustered cytometry data," *Cytometry Part A*, vol. 93, pp. 1039–50, Sep 2018.

[73] Y. Lavin, S. Kobayashi, A. Leader, E.-a. D. Amir, *et al.*, "Innate Immune Landscape in Early Lung Adenocarcinoma by Paired Single-Cell Analyses," *Cell*, vol. 169, pp. 750–65, May 2017.

[74] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, pp. 541–51, Dec 1989.

[75] Y. LeCun, "The mnist database of handwritten digits." `http://yann.lecun.com/exdb/mnist/`. Accessed: 2019-08-03, 19:48.

[76] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, May 2012.

[77] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowledge and Information Systems*, vol. 34, Mar 2012.

[78] J. Li and H. Liu, "Challenges of Feature Selection for Big Data Analytics," *IEEE*, vol. 32, pp. 9–15, Mar 2017.

[79] M. Kuhn, "Building predictive models in r using the caret package," *Journal of Statistical Software, Articles*, vol. 28, no. 5, pp. 1–26, 2008.

[80] O. Veksler, "Machine Learning in Computer Vision, Chapter 2." `http://www.csd.uwo.ca/courses/CS9840a/Lecture2_knn.pdf`. Accessed: 2019-04-01, 13:15.

[81] S. Munir, G. H. Andersen, I. M. Svane, and M. H. Andersen, "The immune checkpoint regulator PD-L1 is a specific target for naturally occurring CD4+ T cells," *OncoImmunology*, vol. 2, pp. 1–9, Apr 2013.

[82] S. J. Schachtele, S. Hu, W. S. Sheng, M. B. Mutnal, and J. R. Lokensgard, "Glial cells suppress postencephalitic CD8+ T lymphocytes through PD-L1," *Glia*, vol. 62, pp. 1582–94, Jun 2014.

[83] M. J. McCarron, P. W. Park, and D. R. Fooksman, "CD138 mediates selection of mature plasma cells by regulating their survival," *Blood*, vol. 129, pp. 2749–59, Apr 2017.

[84] H. Dai, A. Rahman, A. Saxena, A. Jaiswal, R. Majithia, A. Mohamood, L. Ramizre, S. Noel, H. Rabb, C. Jie, and A. R. A. Hamad, "Syndecan-1 identifies and controls the frequency of IL-17- producing naïve natural killer T (NKT17) cells," *European Journal of Immunology*, vol. 45, pp. 3045–51, Sep 2015.

[85] A. K. Jaiswal, M. Sadasivam, and A. R. A. Hamad, "Unexpected alliance between syndecan-1 and innate-like T cells to protect host from autoimmune effects of interleukin-17," *World Journal of Diabetes*, vol. 9, no. 12, pp. 220–5, 2018.

[86] K. E. Diggins, P. B. Ferrell, and J. M. Irish, "Methods for discovery and characterization of cell subsets in high dimensional mass cytometry data," *Methods*, vol. 82, pp. 55–63, Jul 2015.

[87] H. Chen, M. C. Lau, M. T. Wong, E. W. Newell, *et al.*, "Cytofkit: A Bioconductor Package for an Integrated Mass Cytometry Data Analysis Pipeline," *PLOS Computational Biology*, vol. 12, p. e1005112, Sep 2016.

[88] G. Gautreau, D. Pejoski, R. Le Grand, A. Cosma, *et al.*, "SPADEVizR: An R package for visualization, analysis and integration of SPADE results," *Bioinformatics*, vol. 33, pp. 779–81, Mar 2017.

[89] G. Beyrend, K. Stam, T. Höllt, F. Ossendorp, *et al.*, "Cytofast: A workflow for visual and quantitative analysis of flow and mass cytometry data to discover immune signatures and correlations," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 435–442, Oct 2018.

[90] N. Aghaeepour, R. Nikolic, H. H. Hoos, and R. R. Brinkman, "Rapid Cell Population Identification in Flow Cytometry Data," *Cytometry A*, vol. 79, pp. 6–13, Jan 2011.

[91] J. Aßfalg, C. Böhm, K. Borgwardt, M. Ester, *et al.*, "Knowledge Discovery in Databases I." `http://www.dbs.ifi.lmu.de/Lehre/KDD`. Accessed: 2019-06-03, 12:48.

[92] N. Aghaeepour, G. Finak, D. Dougall, A. Khodabakhshi, *et al.*, "Critical assessment

of automated flow cytometry data analysis techniques," *Nature methods*, vol. 10, pp. 228–38, Feb 2013.

[93] M. B. Pouyan, J. Birjandtalab, and M. Nourani, "Distance metric learning using random forest for cytometry data," in *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, p. 2590, IEEE, Aug 2016.

[94] ThermoFisher Scientific, "Human CD and Other Cellular Antigens." `https://www.thermofisher.com/de/de/home/life-science/cell-analysis/cell-analysis-learning-center/cell-analysis-resource-library/ebioscience-resources/human-cd-other-cellular-antigens.html`, 2006. Accessed 2019-09-07, 16:41.

[95] Abcam, "Human CD antigen chart." `https://www.abcam.com/primary-antibodies/human-cd-antigen-guide`, 1998. Accessed 2019-09-07, 16:42.

[96] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, *et al.*, "*limma* powers differential expression analyses for RNA-sequencing and microarray studies," *Nucleic Acids Research*, vol. 43, p. e47, Jan 2015.

[97] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, Jun 2016.

[98] A. Kassambara, *ggpubr: 'ggplot2' Based Publication Ready Plots*, 2018. R package version 0.2.999.

[99] A. T. L. Lun, A. C. Richard, and J. C. Marioni, "Testing for differential abundance in mass cytometry data," *Nat. Methods*, vol. 14, pp. 707–9, May 2017.

[100] K. Müller, H. Wickham, D. A. James, and S. Falcon, *RSQLite: 'SQLite' Interface for R*, 2018. R package version 2.1.1.

[101] P. Grosjean, *SciViews-R: A GUI API for R*. UMONS, MONS, Belgium, 2018.

[102] H. Bengtsson, *R.devices: Unified Handling of Graphics Devices*, 2018. R package version 2.16.0.

[103] M. Corporation and S. Weston, *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*, 2018. R package version 1.0.14.

[104] Microsoft and S. Weston, *foreach: Provides Foreach Looping Construct for R*, 2017. R package version 1.4.4.

[105] H. Wickham, "Reshaping data with the reshape package," *Journal of Statistical Software*, vol. 21, no. 12, pp. 1–20, 2007.

[106] H. Wickham, R. François, L. Henry, and K. Müller, *dplyr: A Grammar of Data Manipulation*, 2018. R package version 0.7.8.

[107] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, pp. 1–22, Aug 2010.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und unter keinen anderen als den von mir angegebenen Quellen und Hilfsmitteln verfasst habe. Wörtlich übernommene Sätze und Satzteile sind als Zitate belegt, andere Anlehnungen hinsichtlich Aussage und Umfang unter Quellenangabe kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen und ist auch noch nicht veröffentlicht.

Berlin, den 23.09.2019

Yen Hoang