Automated analysis of flow cytometry using deep learning for the detection of B-cell neoplasms

Doctoral thesis

to obtain a doctorate (PhD)

from the Faculty of Medicine

of the University of Bonn

Nanditha Mallesh

from Doddaballapur, India

2023

Written with authorization of

the Faculty of Medicine of the University of Bonn

First reviewer: Prof. Dr. med. Peter Krawitz

Second reviewer: Prof. Dr. med. Peter Brossart

Day of oral examination: 13.07.2023

From the Institute for Genomic Statistics and Bioinformatics

Director: Prof. Dr. med. Peter Krawitz

Dedication

This thesis work is dedicated to my mother, *Jyothi Mallesh*, who has been a constant source of support and encouragement during graduate school and life challenges. Thank you for the unconditional love and the myriad of ways in which, throughout my life, you have actively supported me in my determination to find and realize my potential and dreams.

Table of Contents

List of abbreviations

FCS Flow cytometry

CD Cluster of differentiation - CD antigen

NHL Non-Hodgkin's lymphoma

AI Artificial intelligence

CNN Convolutional neural network

SOM Self-organizing map

TL Transfer learning

QC Quality control

ROC Receiver operating characteristic

AUC Area under the curve

1. Introduction

B-cell neoplasms are the clonal expansion of the various stages of B lymphocytes in bone marrow, blood, or other tissues. The term mature B-cell neoplasm is used to describe biologically and clinically heterogeneous diseases of the B-lymphatic system. B-cell neoplasms account for over 85% of non-Hodgkin lymphomas (Armitage and Weisenburger, 1998; Perry et al., 2015), making it the most common type of lymphoma. B-cell neoplasms are not only the most common type but also a very heterogenous group of lymphoproliferative malignancy with different behavior patterns and treatment responses. The WHO classification describes 34 different entities (Swerdlow et al., 2017) based on histology and immunophenotype.

Immunophenotyping is a process used to identify cells based on the antigens or markers present on the cell's surface. These markers are generally cell surface proteins involved in cell functions such as adhesion, signaling, and cell-cell communication. The analysis process involves using antibodies directed against the surface markers to differentiate cells of interest. Antibody array (Belov et al., 2001) or flow cytometry (Gedye et al., 2014) can be used for immunophenotyping. Flow cytometry (FCS) is a high-throughput technique with a well-established role in diagnosing mature B-cell neoplasms.

Flow cytometry analyzes single cells or particles as they pass single or multiple lasers while suspended in a buffered solution (Figure 1). Each particle is analyzed for visible light scatter and one or multiple fluorescence parameters. A detector in front of the light beam measures forward scatter (FS), and several detectors to the side measure side scatter (SS). Forward scatter can indicate the relative size of the cell, while side scatter indicates the internal complexity or granularity of the cell. In addition to the light scatter, fluorescence measurements are recorded by staining the samples with fluorescently conjugated antibodies. Multi-parameter flow cytometers can simultaneously and rapidly quantify numerous cell surface markers using a multi-laser system (Shapiro, 2005). Today, it is a critical step in both research and clinical decision-making for leukemia (Henel and

Schmitz, 2007; Craig and Foon, 2008) and other hematological diseases.

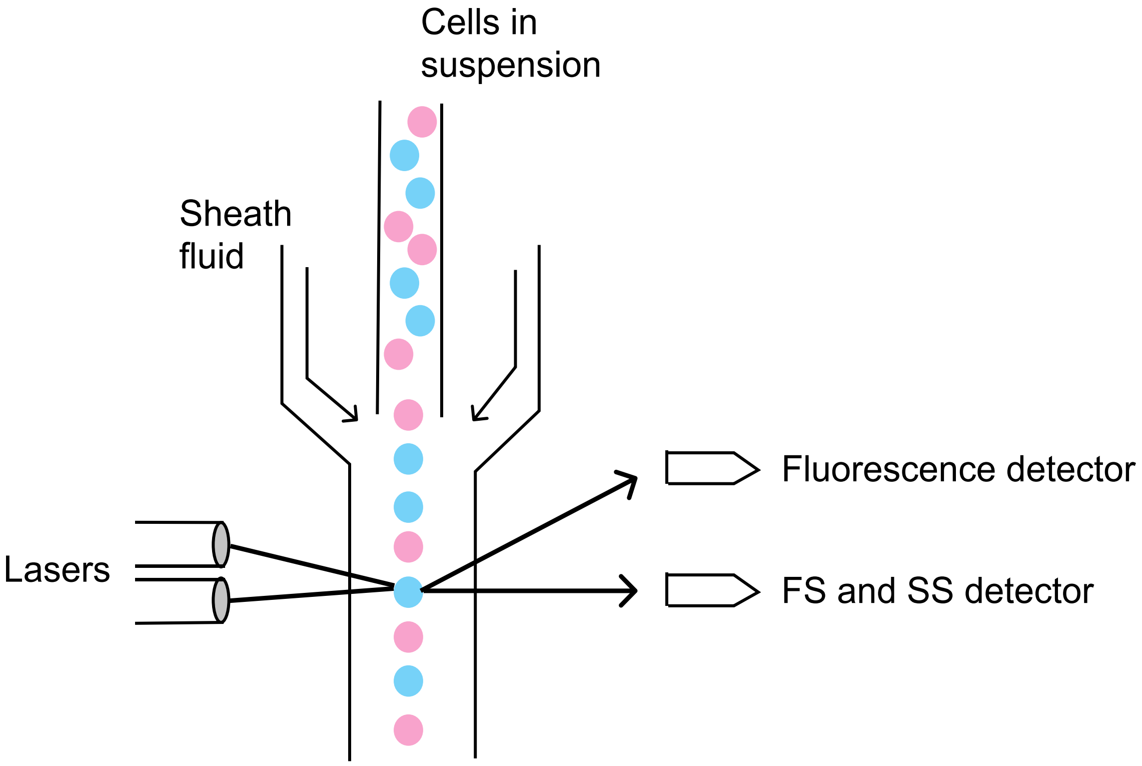


Figure 1: Flow cytometer. A schematic representation of flow cytometry is shown here. The stained cells in suspension are passed through the instrument and are focused into a single file using sheath fluid. The lasers illuminate the cell surface, causing emissions from the fluorescent dyes that are then captured by specific wavelength detectors.

When more than one fluorochrome is used to stain cells, as in multi-parameter flow cytometry, one fluorochrome may add brightness to the others, creating significant background noise and affecting the accuracy of the signal. This phenomenon is called spillover (Figure 2). Spillover occurs due to the physical overlap among the emission spectra of fluorochromes, which can activate different detectors other than the ones intended for the given wavelength. This background noise needs to be corrected before the data can be analyzed.

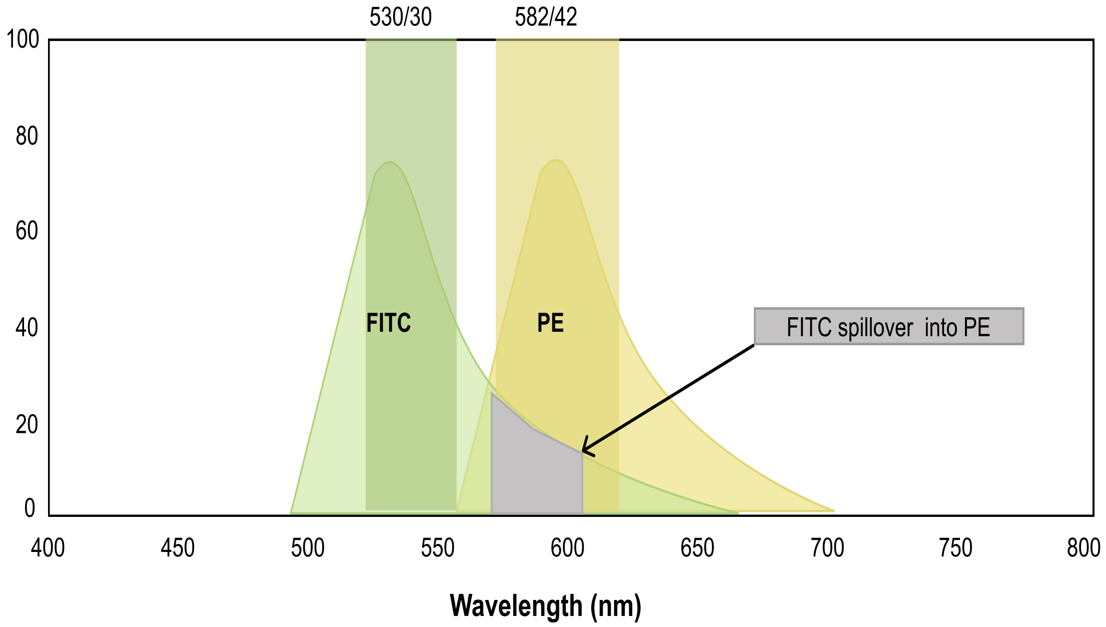


Figure 2: Spillover. Example of FITC spillover into the PE channel.

The process of correcting for fluorescence spillover is called compensation. Compensation removes the signal from a given fluorochrome from all adjacent channels where it is also detected. Compensation is usually performed as a control step once the FCS panel has been designed. First, spectral overlap values are measured using single fluorochrome beads or single-color controls and stored as a matrix. Next, the spectral overlap values are used to calculate the compensation matrix that is applied to correct each color's spectral overlaps into every detector.

Once the compensation matrix is computed and set up, the cytometer uses the calculated compensation matrix to correct spillovers in subsequent measurements.

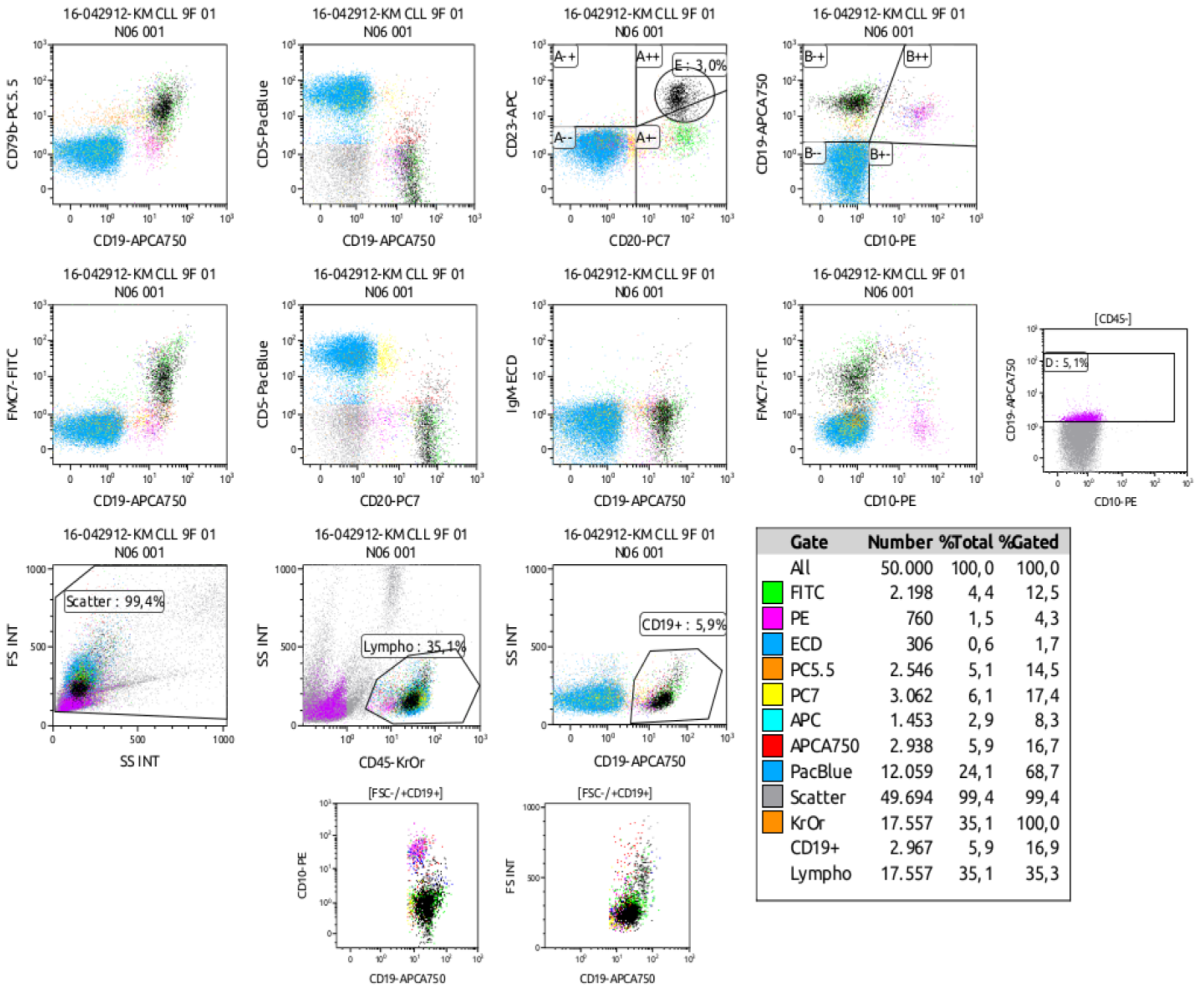
1.1 Role of flow cytometry in the diagnosis of B-cell neoplasms

The fundamental role of flow cytometry in diagnosing B-cell neoplasms is lineage assignment and the distinction between neoplastic and non-neoplastic B-cells. While specific cell-lineage markers such as CD19, CD20, CD22, and CD79 (Craig and Foon, 2008) are used to identify B-cells, the maturation state can be distinguished using surface immunoglobulin light chains that are expressed in most of the mature B-cells. Once the light chain markers are evaluated along with the necessary additional markers to diagnose mature B-cell neoplasm, FCS can further be used to sub-classify and identify the correct subtype of the disease by evaluating additional markers such as CD5, CD10, and others. Since FCS allows for rapid quantification of cell surface markers, it can speed up the diagnosis, especially for aggressive subtypes such as mantle cell lymphoma (MCL) that may benefit from immediate treatment.

1.1.1 FCS Data analysis

The recorded scatter, and fluorescence intensities in a cytometer are stored as a data matrix using the flow cytometry standard (FCS, 1990). The analysis of the stored intensity values to arrive at an accurate diagnosis involves identifying and quantifying cell populations of interest. The cell populations are identified in terms of expression profiles of specific markers that are characteristic of a given disease subtype.

Identifying cell populations of interest is mainly done manually through sequential gating in 2D scatter plots (Figure 3). A typical sequence of steps to identify B-cell neoplasm would include excluding debris and doublets (aggregates of cells) using the scatter values. Next, the side scatter, along with a human leukocyte marker such as CD45, is used to identify leukocytes that are side scatter negative and CD45 positive (SS-/CD45+). The gated leukocytes are further classified as B-cells using a typical B-cell marker such as CD19. Further, sub-classification is done by gating specific B-cell clones such as CD5+ cells or evaluating Kappa and Lambda light chains. Figure 3 shows a manual gating scheme with the various gating plots.



	Gate	Number	%Total	%Gated
	All	50.000	100,0	100,0
🟩	FITC	2.198	4,4	12,5
🟪	PE	760	1,5	4,3
🟦	ECD	306	0,6	1,7
🟧	PC5.5	2.546	5,1	14,5
🟨	PC7	3.062	6,1	17,4
🟦	APC	1.453	2,9	8,3
🟥	APCA750	2.938	5,9	16,7
🟦	PacBlue	12.059	24,1	68,7
⬜	Scatter	49.694	99,4	99,4
🟧	KrOr	17.557	35,1	100,0
	CD19+	2.967	5,9	16,9
	Lympho	17.557	35,1	35,3

Figure 3: Manual gating. A series of 2-dimensional scatter plots show the gating scheme for a B-cell neoplasm sample. Typical gate definitions manually defined with the various quantified populations can be seen in these plots.

While manual gating is the gold standard for diagnosis, it is a time-consuming and subjective process (Bendall and Nolan, 2012; O'Neill et al., 2013). The process of drawing gates using the threshold values for each pair of markers is based on experience and can vary drastically. Furthermore, as the number of markers measured increases, the number of 2-dimensional scatter plots to be gated and analyzed also increase rapidly, making manual gating impractical.

Several computational methods for gating have been developed that are able to reach expert accuracies (Weber and Robinson, 2016; Aghaeepour et al., 2013). However, these approaches still require expert supervision to adjust the automatic gate definitions between multiple files, and they do not scale well computationally with the number of samples. Furthermore, while these methods can generate simple gate definitions, they cannot be used for differentiating multiple subtypes of hematological disorders (Aghaeepour et al., 2013). The gated cells must still be manually analyzed and quantified to arrive at the diagnosis. Thus, there is a need for completely automated methods that can analyze large amounts of data generated without any expert supervision and classify the data into multiple subtypes with high accuracy.

1.2 Deep learning

Deep learning is a subclass of machine learning that can understand and manipulate large amounts of data. Deep learning allows models composed of multiple processing layers to learn data representations with multiple levels of abstraction (LeCun et al., 2015). These methods are good at discovering intricate structures in high-dimensional data necessary for classification. In recent years, deep learning architectures such as deep convolutional neural networks (CNN) have been used successfully for different classification tasks on medical imaging data (Greenspan et al., 2016; Shen et al., 2017). The CNN is inspired by the organization of the animal visual cortex and is designed to learn spatial relationships in the data. It is designed to process data containing grid patterns, such as images.

A schematic representation of a CNN is shown in Figure 4. The building block of a CNN architecture is the convolution layer which performs the feature extraction from input images using linear (convolution) and non-linear (activation) operations. The convolution operation involves applying multiple “kernels,” which are small pre-defined matrices, across the input to generate a feature map. The feature maps represent different characteristics or patterns of the input image; different kernels can, thus, be considered as different feature extractors. The convolution layers are followed by a pooling layer that allows the downsampling of the feature maps. The pooling layers provide dimensionality reduction of the feature maps and thus reduce the number of parameters the network needs to learn. The final convolution or pooling layer's output feature maps are typically flattened and processed by one or more fully connected layers, also known as dense layers. The dense (fully connected) layers map the features extracted by the convolution layers to the final outputs of the network, such as the probabilities for each class in classification tasks.

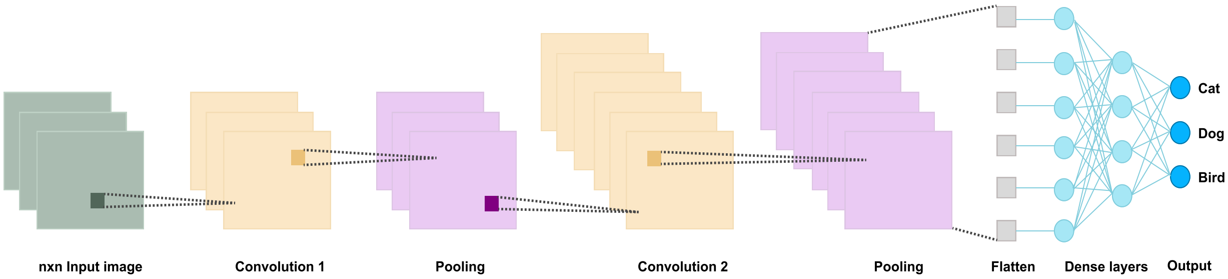


Figure 4: CNN schematic. A schematic representation of a convolutional neural network is shown here. The input layer is processed by the hidden layers consisting of convolution, pooling, and fully connected or dense layers. The output layer generates softmax probabilities for a given number of classes.

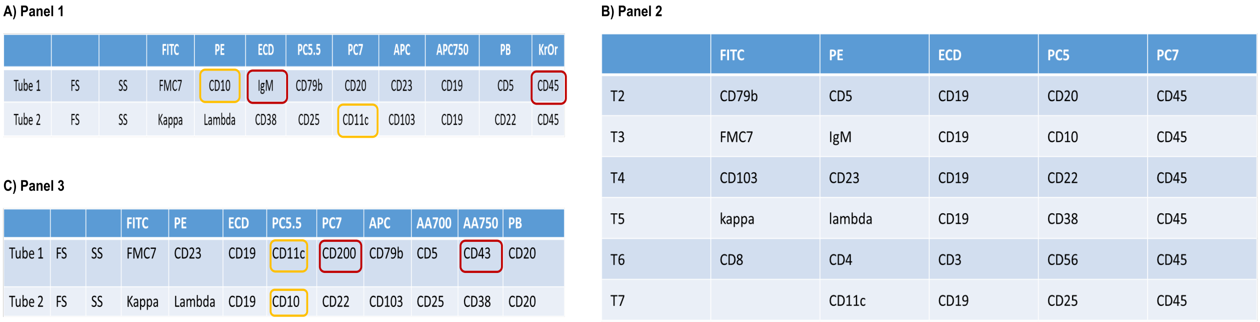
The feature extraction capabilities of the CNN make it well-suited for recognizing patterns in images such as portrait photos of dysmorphic patients (Gurovich et al., 2019), MRI (Abdelaziz Ismael et al., 2020; Taheri Gorji and Kaabouch, 2019), histology (Matek et al., 2019) and others.

Identifying cell populations and classifying B-cell neoplasms from FCS data can be defined as a pattern recognition problem for a CNN. While the FCS data is not an image, the fluorescent intensities can be represented as an image using algorithms based on unsupervised learning techniques such as self-organizing maps (Kohonen, 1990). By generating an image representation of the FCS data, the pattern recognition capabilities of the CNN can be harvested to analyze and classify FCS data with high accuracy.

1.3 Challenges

While the ever-increasing number of parameters that can be measured with modern devices, a widely adopted flow cytometry standard for data by all manufacturers, and the possibility of data anonymization along with the need for fully automated data analysis make FCS ideal for deep learning, other aspects of flow cytometry and the diagnostic process creates unique challenges that need to be addressed.

The flow cytometry panel design across various laboratories varies depending on the markers to be analyzed and the cytometer available. In many cases, the number of markers needed to be analyzed exceeds the number the cytometer can measure in a single run. Standard practice is to aliquot a sample into multiple tubes, which often includes a set of shared or backbone markers. (Van Dongen et al., 2012) This process is standard for modern clinical diagnosis of FCS data, especially when immunophenotyping leukemia and lymphoma. Furthermore, the choice of markers depends on the diagnostic workflow and is not standardized. These differences result in different antibody panels (FCS panels) being used in different laboratories. Figure 5 shows three different FCS panels to diagnose the same B-cell neoplasm subtypes. While all three panels have similar markers that are measured, each panel has a different number of aliquots per sample, markers associated with different fluorochromes, and, more significantly, markers that are only measured in one panel and not the other.



A) Panel 1

			FITC	PE	ECD	PC5.5	PC7	APC	APC750	PB	KrOr
Tube 1	FS	SS	FMC7	CD10	IgM	CD79b	CD20	CD23	CD19	CD5	CD45
Tube 2	FS	SS	Kappa	Lambda	CD38	CD25	CD11c	CD103	CD19	CD22	CD45

B) Panel 2

	FITC	PE	ECD	PC5	PC7
T2	CD79b	CD5	CD19	CD20	CD45
T3	FMC7	IgM	CD19	CD10	CD45
T4	CD103	CD23	CD19	CD22	CD45
T5	kappa	lambda	CD19	CD38	CD45
T6	CD8	CD4	CD3	CD56	CD45
T7		CD11c	CD19	CD25	CD45

C) Panel 3

			FITC	PE	ECD	PC5.5	PC7	APC	AA700	AA750	PB
Tube 1	FS	SS	FMC7	CD23	CD19	CD11c	CD200	CD79b	CD5	CD43	CD20
Tube 2	FS	SS	Kappa	Lambda	CD19	CD10	CD22	CD103	CD25	CD38	CD20

Figure 5: FCS panels. Three different FCS panels used to evaluate B-cell neoplasms are shown here. Panels 1 and 2, shown in A) and B), illustrate changes in panels used in the same laboratory over time. Both panels measure the same markers; however, two different cytometers were used, resulting in a different number of tubes per sample. Furthermore, the markers are not in the same order between the two panels and are associated with different fluorochromes (e.g., CD79b is associated with PC5.5 in panel 1, whereas it is associated with FITC in panel 2). C) shows a third FCS Panel from a different laboratory. Although the same number of aliquots are used per sample, the markers are associated with different fluorochromes and are in a different order compared to panel 1. Additionally, panel 3 has new markers (CD43 and CD200) and is missing markers (IgM and CD45) compared to panel 1 (shown in red). Further, markers like CD10 and CD11c (shown in yellow) are measured in different tubes compared to panel 1.

Such changes in the underlying FCS panel result in datasets with different dimensions (panels 1 and 2) or datasets with marker discrepancies (panels 1 and 3). In this scenario, a model trained on data from panel 1 cannot be used on data from panel 2 or 3. A new model would have to be trained afresh for each panel, which would require large amounts of training data typically unavailable in routine diagnostics. Thus, any artificial intelligence (AI) model used for diagnostic prediction with FCS data must be robust and adapt to multiple FCS panels with fewer training data.

1.4 Transfer learning

In order to adapt and generalize existing models to multiple datasets and tasks, transfer learning is a sought-after method. Transfer learning (TL) is a technique to improve the performance of a new task by transferring knowledge from a related task that has already been learned. (Weiss et al., 2016) The new task (target task) to be learned usually has a smaller dataset than the base data with which the original task (base task) was learned (Figure 6).

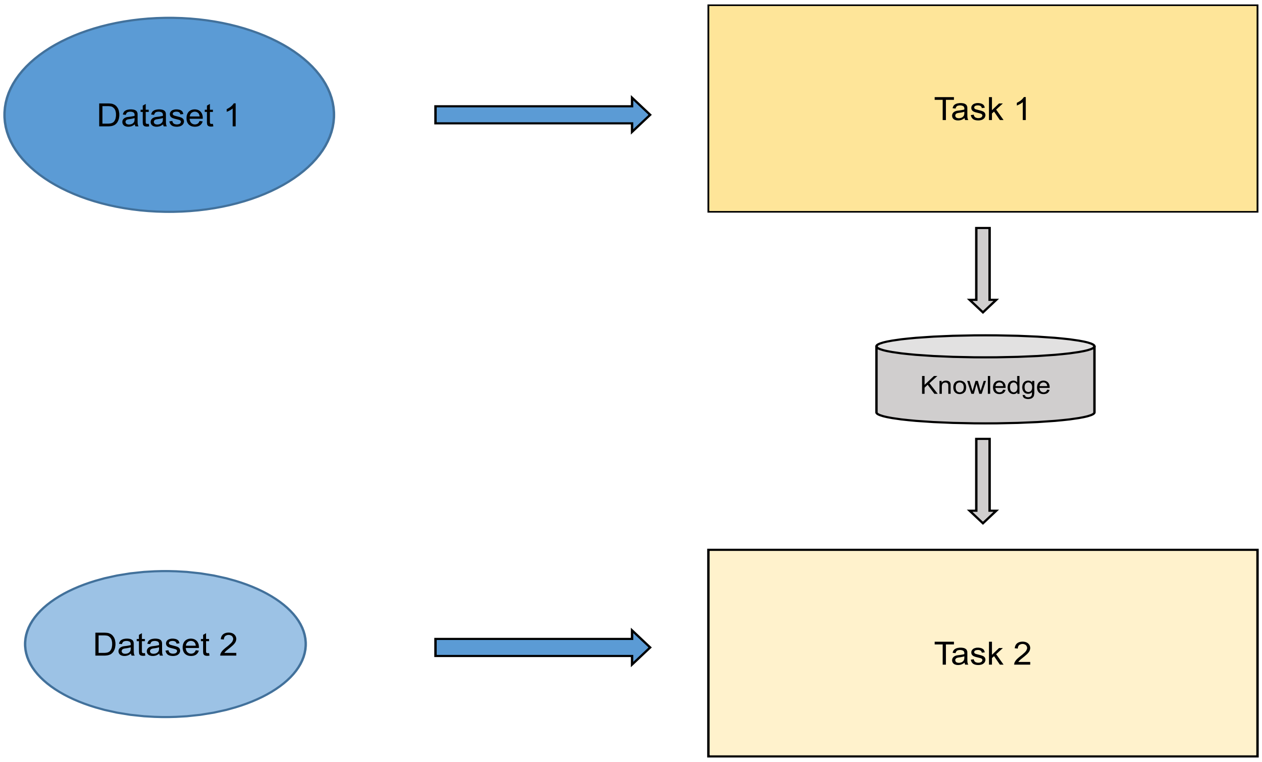


Figure 6: Transfer Learning. A schematic representation of the transfer learning process. Dataset 1 is the larger base data on which task 1 (base task) is learned. The target dataset, dataset 2, is usually much smaller than the base data and is used to learn a related task 2 (target task) by utilizing the knowledge from task 1.

The idea behind transfer learning is to pick up the training from where it was left off in the base model. This allows for a faster convergence for the loss function and higher performance with much less training data. The basic steps of transferring the already learned knowledge involve:

1. Developing a model for the chosen base task using a large dataset or selecting an available pre-trained model. The model must have learned the features sufficiently.

2. The base model can now be used as a starting point for the second task. Depending on the modeling technique, this may involve using all or parts of the base model.

3. Refine and tune the target model on the input data available for the target task.

The advantages of transfer learning are that the models achieve a higher start - initial performance, a higher slope - the rate of improvement, and a higher asymptote - converge faster to the optimal solution (Torrey and Shavlik, 2009). In cases where the problem to be solved does not have enough data to train a new model, transfer learning can enable models to be trained, which would otherwise not be possible. Furthermore, if the base model is trained on a large and general enough dataset, this model will effectively serve as a generic model that can be adapted to multiple related datasets and tasks.

1.5 Goals of the study

The main goal of this study is to create a fully automated AI model to classify FCS data directly into multiple diagnosis labels with expert-level accuracy. Next, we aim to extend and adapt the AI model to multiple datasets and FCS panels with fewer training data. Further, we make the AI's results interpretable and reliable by visualizing the AI's decision for a given prediction. Lastly, we also identify and flag samples that may need further manual analysis to achieve the correct diagnosis and provide valuable insights for such samples through our saliency analysis.

This study provides a “proof of concept” that shows it is possible to create an AI that can achieve expert-level accuracy in classifying B-cell neoplasms from FCS data without the need for manual gating or supervision. Furthermore, we create a pipeline that allows such models to be adapted to different datasets resulting from changes to the underlying FCS panels. By allowing models to adapt quickly to any changes, we make it possible for these models to move from a proof-of-concept stage to being implemented in routine diagnostics settings. We demonstrate that transfer learning makes it possible to train the newer models with far fewer training samples and achieve a higher learning rate and overall performance.

The subsequent chapters detail the methods used to train the models and our transfer learning process. The results are critically analyzed to show the crucial role of transfer learning and FCS data merging.

2. Material and Methods

The methods and results discussed in the subsequent sections are published in two papers listed below. This section describes the materials and methods in detail.

- Zhao, M., Mallesh, N., Höllein, A., Schabath, R., Haferlach, C., Haferlach, T., Elsner, F., Lüling, H., Krawitz, P., & Kern, W. (2020). Hematologist-Level Classification of Mature B-Cell Neoplasm Using Deep Learning on Multiparameter Flow Cytometry Data. Cytometry. Part A : the journal of the International Society for Analytical Cytology, 97(10), 1073-1080. https://doi.org/10.1002/cyto.a.24159.
- Mallesh, N., Zhao, M., Meintker, L., Höllein, A., Elsner, F., Lüling, H., Haferlach, T., Kern, W., Westermann, J., Brossart, P., Krause, S. W., & Krawitz, P. M. (2021). Knowledge transfer to enhance the performance of deep learning models for automated classification of B cell neoplasms. Patterns (New York, N.Y.), 2(10), 100351. https://doi.org/10.1016/j.patter.2021.100351.

2.1 Material

Five FCS datasets were acquired from four different laboratories and are described in detail below. Peripheral blood, bone marrow, or pleura samples were collected from patients with suspected leukemia in a routine diagnostic setting. All samples were prepared and stained according to the flow cytometry protocol of the respective laboratory and analyzed on different Navios cytometers (Beckman Coulter, Miami, Florida). All five datasets include the following eight subtypes of B-cell neoplasm: chronic lymphocytic leukemia (CLL), monoclonal B-cell lymphocytosis (MBL), marginal zone lymphoma (MZL), mantle cell lymphoma (MCL), prolymphocytic leukemia (PL), follicular lymphoma (FL), hairy cell leukemia (HCL), lymphoplasmacytic lymphoma (LPL), and healthy (normal) samples. The number of samples per class in each dataset is detailed in Table 1.

Table 1: Number of samples per class in each dataset

	MLL 9-color	MLL 5-color	Berlin	Bonn	Erlangen
CLL	4438	2277	420	96	72
MBL	1614	268	-	-	16
MCL	313	117	50	12	21
PL	588	200	-	-	-
LPL	726	130	3	6	9
MZL	1106	44	15	5	10
FL	246	142	49	20	10
HCL	225	290	54	13	2
normal	11366	5836	2182	404	107

Data distribution among the different classes for each dataset is shown here. The distribution reflects the presumable incidence for the listed subtypes. Only data samples with precise diagnosis, obtained with additional tests where necessary, were included. Further, only samples with the required panels (B1 and B2) are shown here for the Erlangen dataset. CLL and MBL are merged into a single class for classification.

2.1.1 MLL 9-color panel

FCS data was obtained from 20,622 routine diagnostic samples from patients with suspected B-cell neoplasm (B-NHL) that had been analyzed between January 01, 2016, and December 31, 2018, at Munich Leukemia Laboratory (MLL). For the assessment of B-NHL, a panel consisting of three 9-color combinations of monoclonal antibodies was used in all samples to analyze the surface expression of 21 antigens. The detailed antibody-color combination is reported in Table 2. In the following, we refer to this dataset as the MLL9F panel.

Table 2: Antibody-color combinations for MLL 9-color panel

	APCA750	KrOr	FITC	PE	ECD	PC5.5	PC7	APC	PacBlue
Tube 1	CD19	CD45	FMC7	CD10	IgM	CD79b	CD20	CD23	CD5
Tube 2	CD19	CD45	Kappa	Lambda	CD38	CD25	CD11c	CD103	CD22
Tube 3	CD19	CD45	CD8	CD4	CD3	-	-	CD56	HLA-DR

FCS panel used to acquire the MLL 9-color dataset. Colors are shown in the header row, and antibodies are shown for each tube. Further, forward scatter (FS) and side scatter (SS) were measured in all tubes.

2.1.2 MLL 5-color panel

A 5-color dataset consisting of 10,215 samples was acquired at MLL between January 1, 2011, and December 31, 2012. For the assessment of B-cell neoplasms, a panel consisting of seven 5-color combinations of monoclonal antibodies was used in all samples to analyze the surface expression of 20 antigens. A detailed antibody-color combination is given in Table 3. We refer to this dataset as the MLL5F panel.

Table 3: Antibody-color combinations for MLL 5-color panel

	FITC	PE	ECD	PC5.5	PC7
Tube 1	IgG1a	IgG1a	IgG1a	IgG1a	CD45
Tube 2	CD79b	CD5	CD19	CD20	CD45
Tube 3	FMC7	IgM	CD19	CD10	CD45
Tube 4	CD103	CD23	CD19	CD22	CD45
Tube 5	Kappa	Lambda	CD19	CD38	CD45
Tube 6	CD8	CD4	CD3	CD56	CD45
Tube 7	-	CD11c	CD19	CD25	CD45

FCS panel used to acquire the MLL 5-color dataset. Further, forward scatter (FS) and side scatter (SS) were measured in all tubes. Tube 1 is used for isotope control and is not considered for further processing.

2.1.3 Bonn panel

The third dataset was obtained from the University Hospital Bonn, consisting of 556 samples measured between January 1, 2018, and December 31, 2018. A panel composed of two 9-color combinations of monoclonal antibodies was used to analyze 16 antigens' surface expression for B-NHL assessment. Detailed FCS panel information is given in Table 4.

Table 4: Antibody-color combinations for Bonn panel

	FITC	PE	ECD	PC5.5	PC7	APC	AA700	AA750	PB
Tube 1	FMC7	CD23	CD19	CD11c	CD200	CD79b	CD5	CD43	CD20
Tube 2	Kappa	Lambda	CD19	CD10	CD22	CD103	CD25	CD38	CD20

Antibody-fluorochrome combinations used in the Bonn panel. Additionally, forward scatter (FS), and side scatter (SS) were measured in all tubes.

2.1.4 Berlin panel

For the fourth dataset, an 8-color panel consisting of 2,773 routine diagnostic samples from patients with suspected B cell neoplasms analyzed between January 1, 2016, and December 31, 2018, was obtained from the Berlin Hematology laboratory. The B-NHL assessment panel consisted of four 8-color combinations of monoclonal antibodies. Table 5 details the FCS panel used.

Table 5: Antibody-color combinations for Berlin panel

	FITC	PE	ECD	PC5.5	PC7	APC	PB	KrOr
Tube 1	IgG	IgG	IgG	IgG	IgG	IgG	IgG	CD45
Tube 2	Kappa	Lambda	CD19	CD5	CD38	CD10	CD20	CD45
Tube 3	FMC7	CD23	CD19	CD3	-	CD79	CD22	CD45
Tube 4	CD43	IgM	CD19	CD25	CD11C	CD103	CD5	CD45

Antibody-fluorochrome combinations used in the Berlin panel. Tube 1 is used as isotope control and is not considered in the workflow. Forward scatter (FS), and side scatter (SS) were measured in all tubes.

2.1.5 Erlangen panel

A fifth dataset was obtained from the University Hospital Erlangen. The dataset consisted of 1,626 routine diagnostic samples from patients with suspected B-NHL analyzed between January 1, 2014, and July 31, 2020. The assessment panel consisted of a screening panel (B1), with one ten-color combination of monoclonal antibodies used to analyze the surface expression of nine antigens. Next, a secondary panel (B2) was used to identify the B-NHL subtype where necessary. Finally, for the identification of HCL (hairy cell leukemia), a third panel (B3) was used. All three panels are described in detail in Table 6. We only consider the 247 samples with B1 and B2 panels for this study.

Table 6: Antibody-color combinations for Erlangen panel

	FITC	PE	ECD	PC5.5	PC7	APC	APC750	PB	KrOr
Tube B1	Kappa	Lambda	CD3	CD20	CD19	CD10	CD5	CD23	CD45
Tube B2	CD38	CD79b	-	CD11c	CD19	CD103	CD43	HLA-DR	CD45
Tube B3	Kappa	Lambda	CD3	CD11c	CD19	CD103	CD25	HLA-DR	CD45

Antibody-fluorochrome combinations used in the Erlangen panel. B1 is used as the screening panel, B2 for subtype identification, and B3 is the HCL panel.

All the markers and their functionality, as well as the fluorochromes used, are detailed in the supplementary information (Tables 12 and 13).

2.2 Methods

The study is divided into two phases. In the first phase, an automated AI model is trained on a single FCS dataset and is shown to achieve expert-level accuracy in classifying B-cell neoplasms (Zhao et al., 2020). The study's second phase extends the AI model to various FCS panels and datasets using transfer learning (Mallesh et al., 2021). This section describes the workflow, model architectures, and training procedure, along with all the analyses performed in both phases of the study.

2.2.1 System requirements

All analyses were performed with Python (Van Rossum and Drake, 2009) version 3.6 and Tensorflow (Abadi et al., 2016) version 1.12. All models were generated and trained using Tensorflow and Keras (Chollet et al., 2015) in the backend. An NVIDIA GPU is preferable for running all computations. We used a Tesla P40 GPU with 24 GB GDDR5X memory on an Ubuntu 16.04 Linux machine. In addition, at least 500 GB of HDD storage for the entire dataset is necessary. The computation time required for analysis depends on the size of the dataset. For our largest dataset, approximately 32 hours were required to train the model on the specified GPU.

2.2.2 Data

The five FCS datasets described above, along with ground truth diagnosis labels, were used to train the models and perform all analyses. All diagnoses were verified with additional tests from histology, cytomorphology, and *in situ* fluorescence hybridization. Furthermore, only cases obtained from peripheral blood or bone marrow aspirate with unambiguous labels were used to train the models. All FCS data were stored in the FCS 2.0 format (Dean et al., 1990), and the compensated FCS 2.0 data segment was used in the analysis.

Further quality control was performed by visually inspecting the channel plots to detect issues in machine calibration and compensation in the dataset. Next, the scaling used for each marker channel was checked to ensure all samples had identical parameters: PnG-scaling (linear gain with a factor) was used for SS and FS, and PnE-scaling (exponential scaling) for all color channels. No re-linearization or additional transformation was applied to the data. The channel intensities ranged between 0 and 1000, the max range defined by PnR. As the SOM and CNN weights are initialized with random values between 0 and 1, the channel intensity ranges were rescaled by a factor of 0.001 for an efficient start of the training process. However, this rescaling did not impact the data distribution and was only used to correct the intensity range to “0” and “1” for computational speed-up of the training process.

Additionally, the number of events acquired per tube for each dataset was examined and is described in Table 7.

Table 7: Event counts

	Tubes	Mean	SD	Max
MLL9F	Tube 1	48308.38	4788.90	50000
	Tube 2	46298.75	7544.50	50000
	Tube 3	48784.08	3936.46	50000
MLL5F	Tube 1	49942.99	4885.04	500000
	Tube 2	49993.30	4901.87	500000
	Tube 3	49958.13	4819.49	500000
	Tube 4	49978.72	4694.58	500000
	Tube 5	49903.96	4916.55	500000
	Tube 6	49942.94	4843.74	500000
	Tube 7	49969.46	5848.06	500000
Berlin	Tube 1	28995.20	4365.79	30000
	Tube 2	29261.80	3669.18	30000
	Tube 3	29320.63	3551.20	30000
	Tube 4	80920.36	25404.84	100000
Bonn	Tube 1	87525.58	24397.48	100000
	Tube 2	85555.61	27039.23	100000
Erlangen	Tube B1	86449.25	70766.70	250000
	Tube B2	81817.47	43784.52	125000
	Tube B3	82169.85	84764.58	226050

The mean number of events acquired per tube in each panel, the standard deviation amongst the samples, and the maximum events recorded per tube are reported here.

Base dataset:

The MLL9F panel was used as the base dataset to train a base model on which all initial evaluations were performed during the first phase of this study.

Target datasets:

The other four datasets: MLL5F, Bonn, Berlin, and Erlangen panels, were used later as target datasets to extend and adapt the base model to different FCS panels using transfer learning. These datasets were used to train the target models in the study's second phase.

2.2.3 Workflow

The general workflow is shown in Figure 7. The FCS data was converted into a low-resolution image using a self-organizing map (SOM). The generated SOM node weights, which are n-dimensional vectors of the original FCS data arranged on a two-dimensional grid, were used as input to the CNN that generates class predictions.

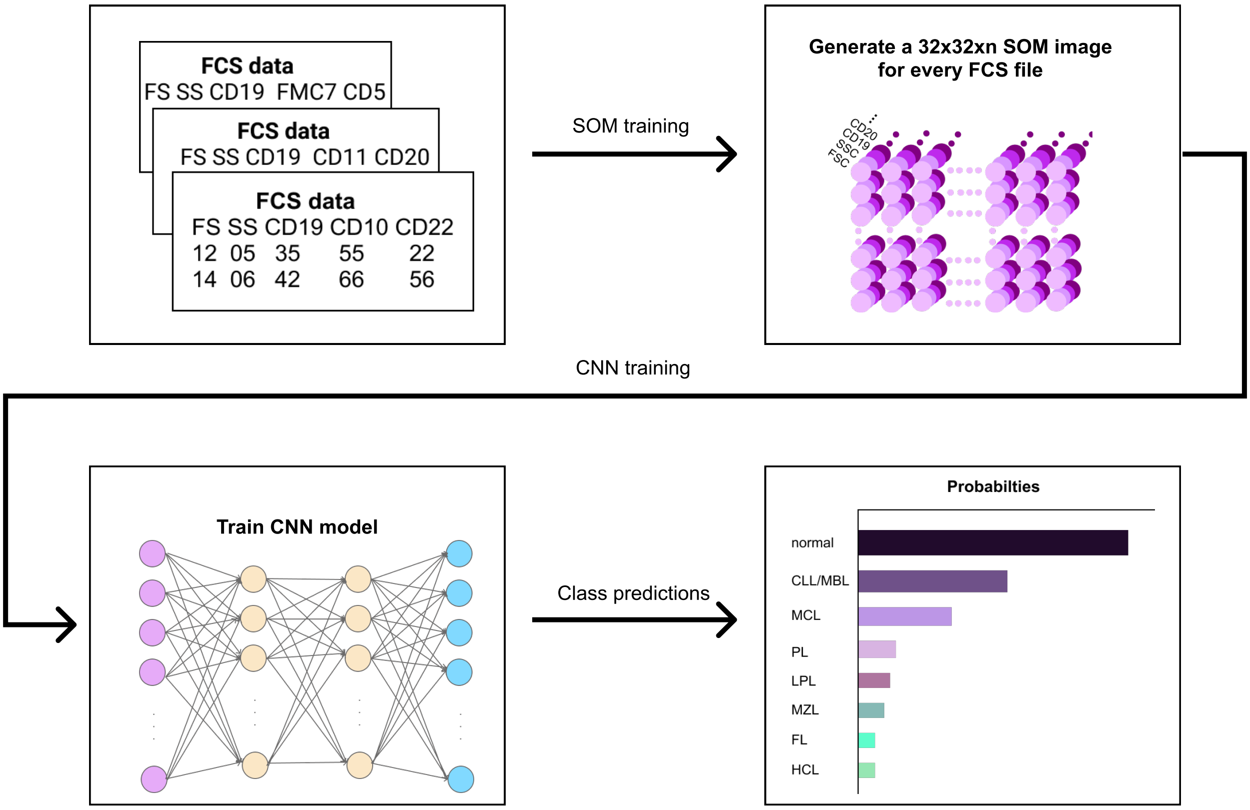


Figure 7: Workflow. A workflow diagram summarizing the steps for automated class prediction.

2.2.4 Phase 1 - AI model for classification of B-cell neoplasms

This section describes the base model generation and training. The trained model can classify the FCS data into eight classes: chronic lymphocytic leukemia and its predecessor monoclonal B-cell lymphocytosis (CLL/MBL), marginal zone lymphoma (MZL), mantle cell lymphoma (MCL), prolymphocytic leukemia (PL), follicular lymphoma (FL), hairy cell leukemia (HCL), lymphoplasmacytic lymphoma (LPL), and normal.

Self-organizing map (SOM)

A SOM is a network of interconnected nodes, ordered in a two-dimensional topology, which can be used for unsupervised clustering of high-dimensional data (Kohonen, 1990). SOMs were used as a method to reduce the dimensionality of the data while preserving its spatial structure. The SOM model was adapted from an implementation using Tensorflow for GPU-based training (Gorman, 2019). For each sample, data from individual tubes were independently transformed into separate SOMs. All FCS events were mapped onto a 32x32 grid of nodes. Each node in the SOM is associated with a “weight” vector representing the n-dimensional FCS data.

The mapping of events onto the SOM nodes was done in batches to increase the throughput of the training algorithm by leveraging efficient vectorizations (Fort et al., 2001). Euclidean norm was used to calculate the distance between input vectors and single SOM nodes. A radius parameter was used to set the width of the Euclidean neighborhood function for calculating weight updates. An initial assessment of training parameters showed that a higher learning radius correlated with lower topographic error (TE) and higher mean quantization error (MQE), which are favorable for good training. MQE is the average Euclidean distance of each input to their best matching node; it measures the quality of clusters. TE describes the proportion of input entries where the first and second best matching nodes are non-adjacent (Kiviluoto, 1996). Thus, a lower TE conserves the maps’ spatial relationship and neighborhood quality, while a higher MQE produces better clusters. A larger number of epochs only played a minor role in increasing MQE after a certain threshold. Based on these initial assessments, training parameters - learning radius, number of epochs - were set, and SOMs were generated for each tube in a given sample.

Furthermore, individual SOM transformation used pre-initialized weights from a reference SOM trained on a small subset of samples. The reference SOM was generated using a random sample from each diagnostic class in the training dataset, with at least 20% infiltration - the proportion of pathological events amongst the total events - reported in the manual analysis. The selected samples were then used collectively to train the reference SOM with random weight initialization for each tube.

The reference SOMs were trained using 10 epochs and an initial radius of 24, linearly decreasing to 2 at the end of training. A toroidal neighborhood function was used to avoid edge artifacts caused by a planar map (Mount and Weaver, 2011). Individual SOM for every sample was generated using the reference SOM node weights and four training epochs with a starting radius of 4, linearly decreasing to 1.

Additionally, the performance of various SOM grid sizes, such as 32x32 and 10x10, was compared. The 32x32 grid achieved a higher classification score (F1 score of 0.93 compared to 0.88) with only marginal performance penalties in the SOM training and thus was chosen as the base SOM size for further analysis. Larger SOM sizes, such as 48x48, were not attempted because of performance considerations.

The generated SOMs serve as low-resolution images of the FCS data. Figure 8 shows an example visualization of the generated SOM images. The three markers CD45/SS/CD19 shown here act as the three color channels of the image. The images here are limited to three colors for ease of visualization. The SOMs can be considered an image with “n” color channels corresponding to the number of markers measured in the FCS panel.

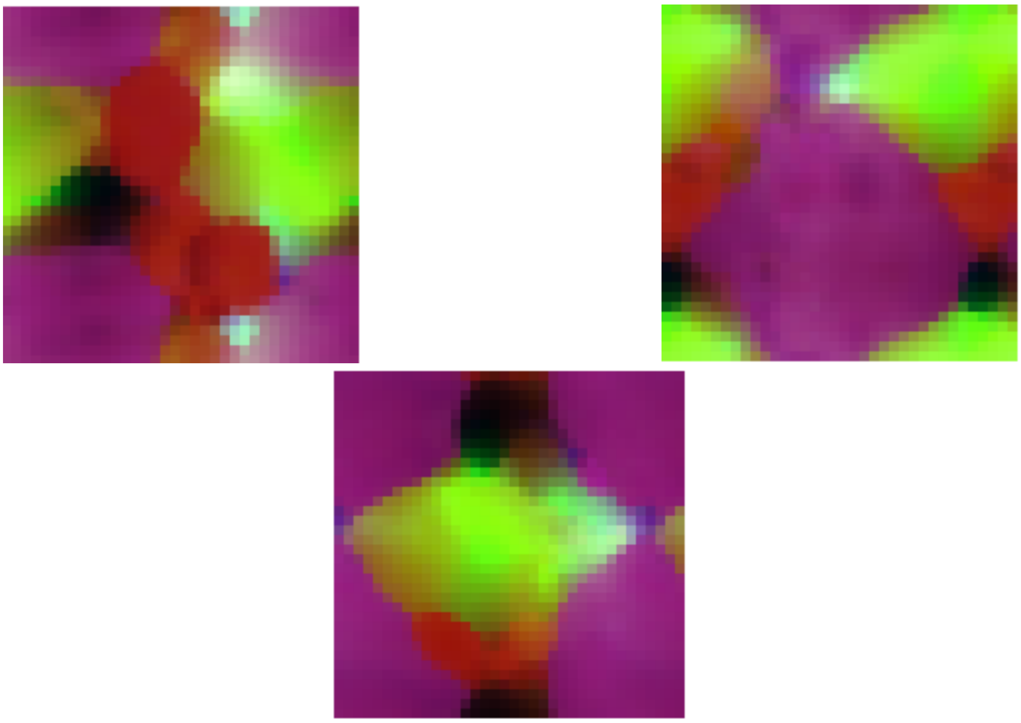


Figure 8: SOM images. Examples of SOM images with three markers/color channels - CD45 (red), SS (green), and CD19 (blue) are shown here. Nodes with cells that are CD45+ SS- CD19+ appear as magenta in these images.

As a QC step, the generated SOMs were compared to self-organizing maps generated by previously published algorithms such as flowSOM (Van Gassen et al., 2015). FlowSOM is an automated clustering algorithm that uses self-organizing maps and minimal spanning trees to cluster and visualize events. While the minimal tree clustering allows visualizing the clusters of different cell types, it does not yield an image-like representation required to train the CNN.

We compared both SOMs by using a random selection of 5 samples from each of CLL, MBL, MCL, PL, MZL, LPL, FL, HCL, and normal, a total of 45 samples. As described before, data were scaled by subtracting the mean and dividing by the standard deviation of the sample. A SOM was generated for each tube. Quantization error was calculated as the mean Euclidean distance of each input data point to its closest node on the mean. For both 10x10 and 32x32 maps, our implementation (flowCAT) achieved similar performance as the published flowSOM algorithm on the sample data as shown in Figure 9A. In order to further visualize the behavior of SOM nodes in approximating the input data, trained node weights were plotted as scatterplots and compared to the scatter plot of the original scaled FCS data. Both flowSOM and our SOM preserve the data structure and the existing spatial relationship to a similar extent. Figure 9B-D shows the scatterplots obtained for the markers CD19 and CD103. As seen in the plots, the SOM nodes have a similar spatial position and structure compared to the FCS data points.

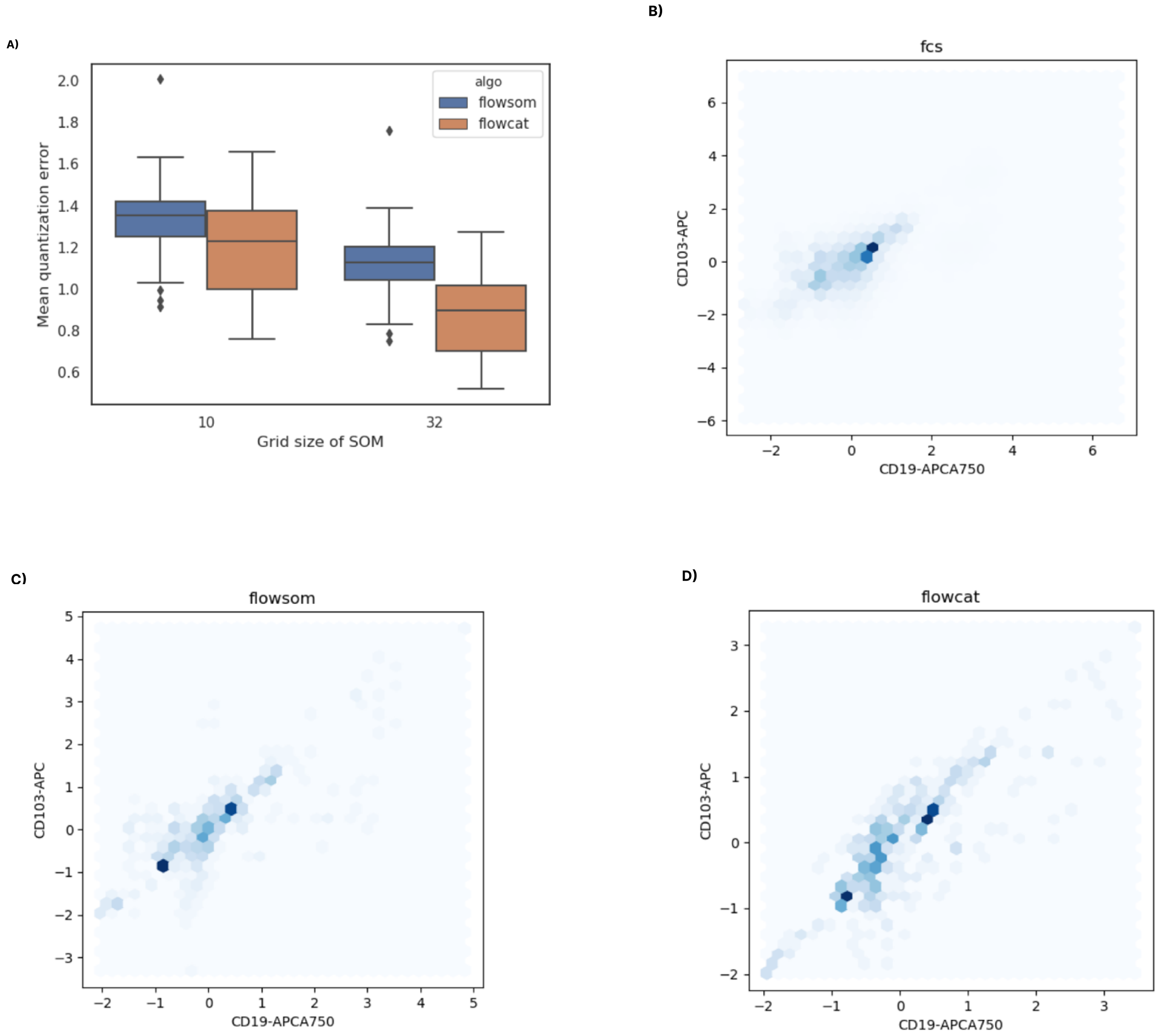


Figure 9: SOM validation. Comparison between quantization quality of the SOM generation in flowSOM and flowCAT. A) The mean quantization error over all samples and tubes for SOMs generated by flowSOM and the current implementation. B), C), and D) show the scatter plot comparison of the scaled data. B) shows the original scaled FCS data. In comparison, both flowSOM shown in C) and our SOM implementation (flowCAT) shown in D) perform a transformation that largely preserves the original distribution.

Convolutional neural network (CNN)

The CNN architecture is shown in Figure 10. The model generates predictions using SOM node weights for a number of classes. The weights are initially processed in three convolutional layers with decreasing filter sizes. A filter is a collection of kernels - the small pre-defined matrices that extract the features. The number and size of kernels for each convolutional layer were chosen after multiple rounds of hyperparameter tuning. A *hyperparameter* is a parameter that controls the learning process. Hyperparameter tuning involves choosing a set of optimal parameters for learning the given problem.

The CNN has three convolutional layers followed by a global max pooling layer that summarizes filters across the spatial dimension of the SOM map. A global max pool layer was used instead of multiple pooling layers to reduce the number of untrainable parameters and minimize computational overhead. The removal of intermediate pooling layers did not affect the model's performance. In order to merge information from multiple tubes, the convolutional layers and the final max pooling layer were replicated for each tube. The result from each max pooling layer was concatenated across all tubes and processed further in the two subsequent dense layers that combine information for class prediction. Thus, the dense layers in this architecture merge information from all provided tubes.

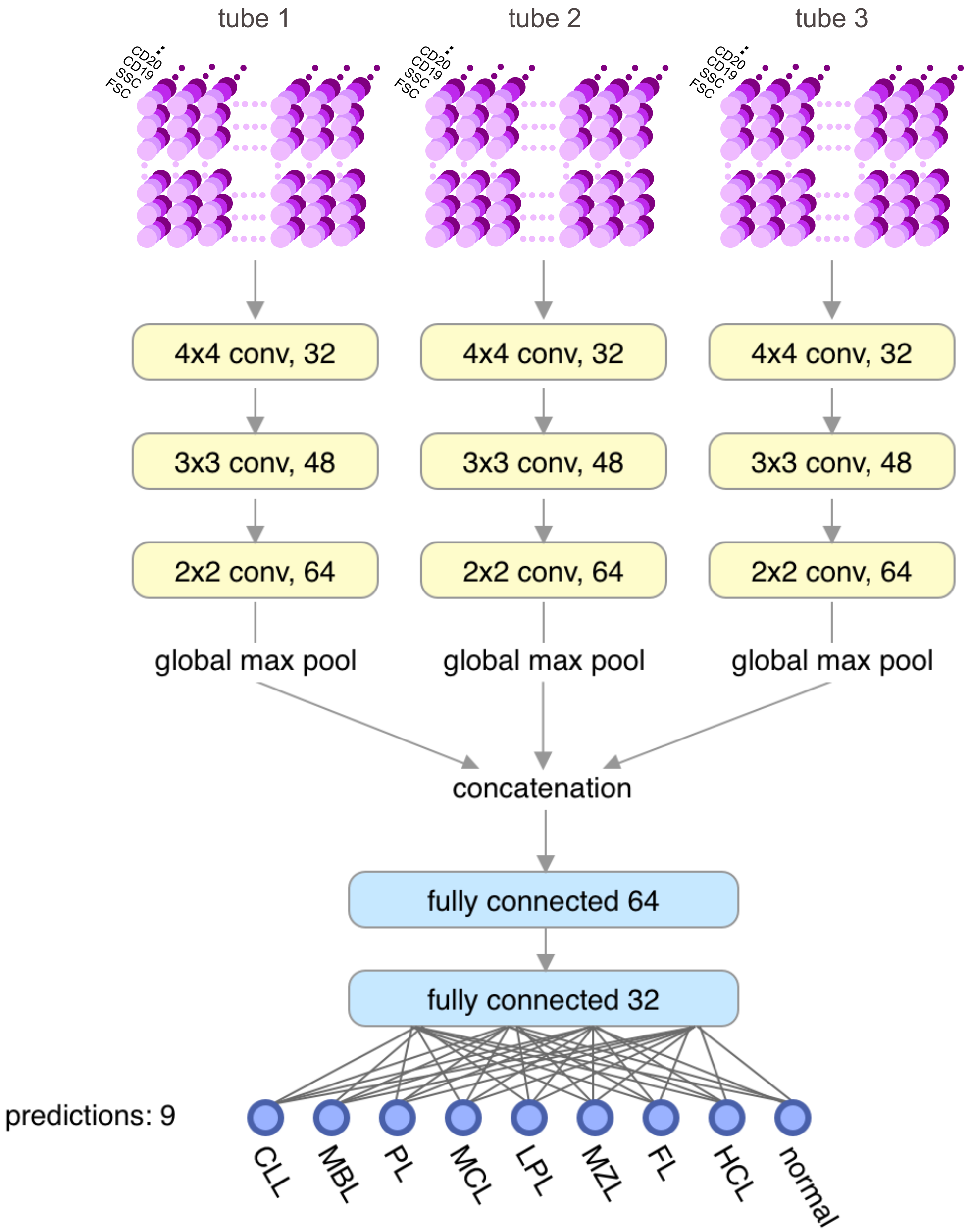


Figure 10: CNN architecture. First, the original 32x32 SOMs are toroidally wrapped by two pixels on each edge to produce a 36x36 input matrix, fed into convolutional layers with 32 4x4 filters. The input from each SOM is processed individually in a sequence of convolutional layers (conv), followed by a global max pooling and concatenation layer. This vector is further processed in two fully connected hidden layers, resulting in a softmax prediction layer.

Training

The base dataset (MLL9F) was split into training and hold-out test sets based on the sample acquisition timeline. Unlike randomly splitting data into training and test sets, preserving chronological order represents a more realistic situation in a diagnostic workflow. All samples in the MLL9F panel acquired before July 1, 2018, were used to train the model, while samples obtained after July 1, 2018, were set aside as a hold-out test set. The hold-out test set was used to assess the model performance. Further, classification accuracies were evaluated on a 10% validation split of the training set. They were used to optimize the network architecture and tune the hyperparameters.

The model was implemented using the Keras framework (Chollet et al., 2015). The model was trained for 15 epochs using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001. *Adam* is an optimization algorithm that is used instead of the classical stochastic gradient descent to update the network weights iteratively. Further, a global weight decay of 5e-6 was applied to all layers. Figure 11 below shows the training graph with the chosen parameters, the training and validation loss and accuracies converge well, and there is no over-fitting with the chosen parameters.

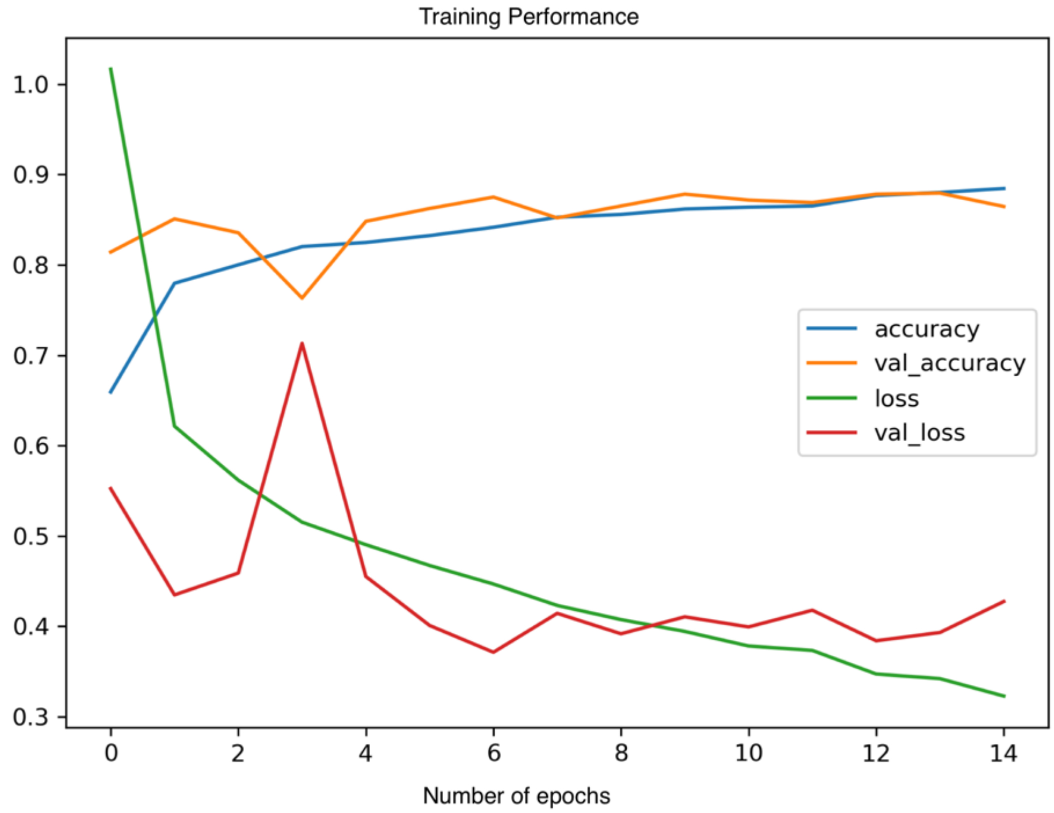


Figure 11: Model evaluation curves. The training and validation loss and accuracy over the number of epochs are monitored to find the optimal number of training epochs to avoid overfitting.

Performance metrics

Prediction performance was evaluated using F1 scores. The F1 score is the harmonic mean between recall and precision and places equal importance on both measures. We use the F1 score as a performance metric to reflect the real-world diagnostic scenario where precision and recall are equally important. Precision and recall per class were defined on the true label of each case. The overall average F1 score is calculated as the average of the per-class F1 scores given by the formula defined below.

$$avg\ f1 = \frac{1}{|C|}\sum_{c \in C} f_c, \text{ with } f_c = \frac{Precision_c.Recall_c}{Precision_c + Recall_c} \text{ where C is the set of all classes} \tag{2.1}$$

However, the average or macro F1 score does not account for class imbalance. Thus, we compute the overall weighted F1 score for the classifier. The weighted F1 score calculates the average, considering the proportion of samples for each class in the dataset. The weighted F1 score was calculated as the class-size-weighted average of the per-class F1 scores.

$$weighted\ f1 = \frac{1}{\sum_{c \in C} s_c}\sum_{c \in C} s_c f_c, \text{ with } s_c \text{ as the number of samples in class c} \tag{2.2}$$

We calculate the top 1 accuracy rate of the classifier for the eight classes: chronic lymphocytic leukemia and its predecessor monoclonal B-cell lymphocytosis (CLL/MBL), marginal zone lymphoma (MZL), mantle cell lymphoma (MCL), prolymphocytic leukemia (PL), follicular lymphoma (FL), hairy cell leukemia (HCL), and lymphoplasmacytic lymphoma (LPL) and healthy controls.

Although the model is trained on samples from the nine classes defined in the "Materials" section, we consider CLL and MBL as a single class to evaluate the model performance. MBL is a diagnostic finding that is regarded as a potential preneoplasia and precursor of CLL in most cases (Swerdlow et al., 2016). Both MBL and CLL, therefore, share a similar immunophenotype (CD5+/CD19+/CD20 low/CD23+/Ig low). Accordingly, we combine MBL and CLL into a single class for classification.

2.2.5 Phase 2 - Extending and adapting the AI model to multiple datasets and laboratories

As described in chapter 1, the FCS panel can change over time based on the cytometer used and the diagnostic workflow and goals. Thus, workflows to adapt existing AI models to multiple FCS panels are essential. This section describes our workflow to extend and adapt our AI described above. In order to handle the differences in the panels, we merge multiple tubes per sample into a single large FCS data matrix using the nearest neighbor (NN) approach. This method assumes that an event (cell) in one tube is identical to its NN in another tube in terms of the shared markers and can thus be used to impute missing marker values (Pedreira et al., 2008; Abdelaal et al., 2019). The expression vectors of all the NNs across tubes are merged, creating a single, high-dimension matrix of cellular expression across all tubes. NN merging has proven effective as part of classification pipelines (Van Dongen et al., 2012; Costa et al., 2010), while other merging methods are better suited for deep profiling (O'Neill et al., 2015). Deep profiling analyzes the relationship between phenotype and function of the various cell types and thus requires a more accurate measure of the cell properties. On the other hand, classification tasks aim to identify the patterns and different cell types present and thus only require a close estimate of all the cell properties. We use NN merge with TL to extend and adapt our AI model and achieve a higher learning rate with fewer training samples.

Modified Workflow

An overview of the TL process is shown in Figure 12. The workflow from the previous phase was adapted to facilitate transfer learning by adding an initial merge step. The SOM training was further adapted to account for marker discrepancies.

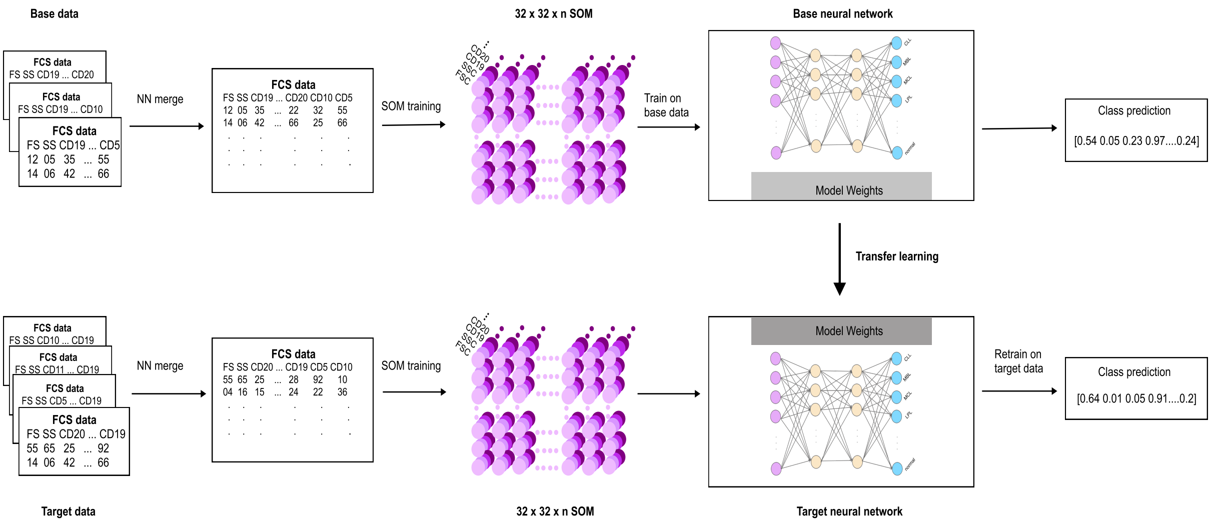


Figure 12: TL workflow. For each dataset, FCS files from different tubes for each sample are merged using the NN merge. Next, individual SOMs are generated for each of the merged FCS samples. The SOM nodes are arranged in a 32x32 grid where each node is associated with an n-dimensional weight vector, where n equals the number of channels in the original FCS events. The SOM node weights are then used as input to the CNN. The weights from the base model trained on the base dataset are transferred to each target network. The target networks are then retrained on the respective target dataset to generate class predictions.

Before knowledge transfer, we merged multiple aliquots (tubes) per sample into a single FCS data file using the NN merge. As described in the previous section, we processed individual tubes of each sample separately in our AI model, resulting in a CNN architecture that depends on the number of tubes per sample. Such a network's transferability between datasets with a different number of tubes per sample is very low - we can only transfer knowledge from the dense layers (see Figure 13). Merging multiple aliquots before the CNN training leads to an architecture independent of the number of tubes in the dataset and allows maximum transfer between the networks - weights from all layers can now be transferred.

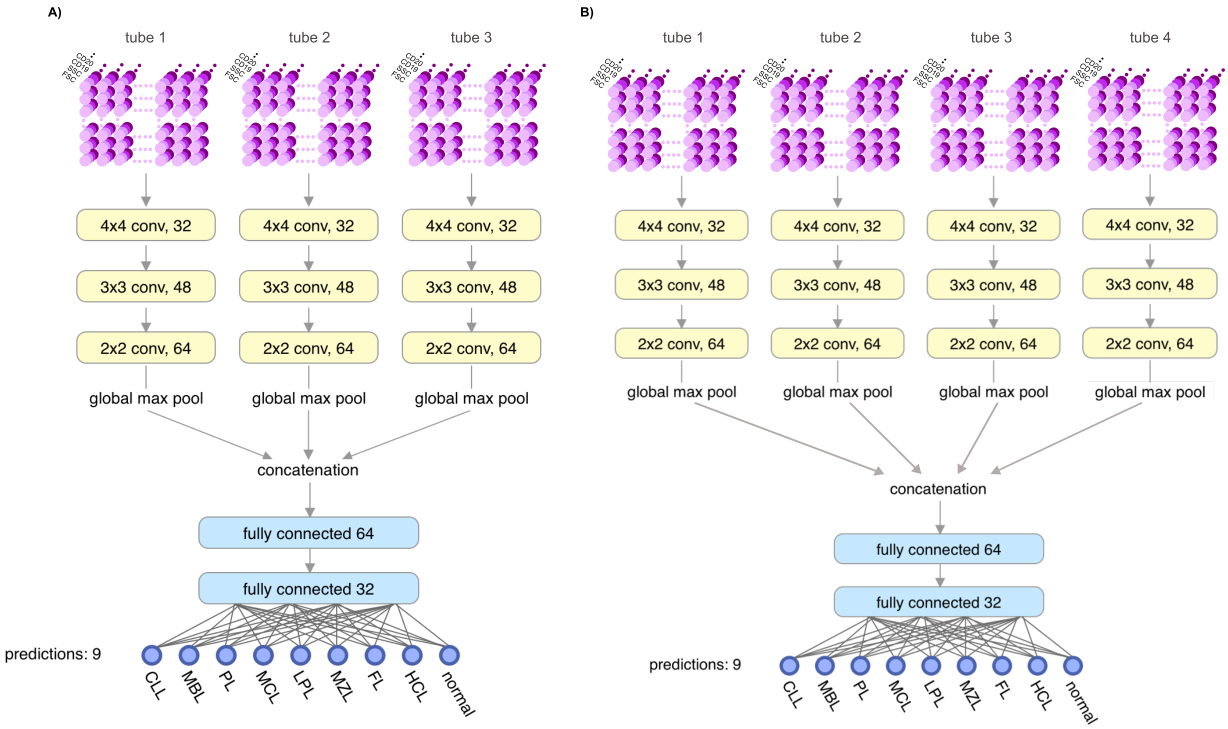


Figure 13: Need for merge. For each sample, the tubes are processed separately in the convolution layers. There are as many convolution stacks as the number of tubes. When the new dataset has a different number of tubes per sample, the convolution stacks change accordingly. Consider two datasets with the same CD markers measured over a different number of tubes: Fig A) shows the CNN for a dataset with three tubes per sample; we have three convolution stacks, one per tube. B) CNN for a dataset with four tubes is shown. Here, only the weights from dense layers can be transferred between networks. The convolution layers have different data dimensions corresponding to the number of markers per tube and are thus not transferable.

Next, a self-organizing map was generated as in the previous case. However, a single SOM was generated for each merged sample instead of one for each tube per sample. The SOMs were then used as input to the CNN that generates class predictions. While the base model was trained with random weight initialization, for each target dataset, target models were trained by initializing the weights with the final weights from the base model. The following sections describe each step of the modified workflow in detail.

Merge

As the first step of the modified workflow, FCS data from multiple tubes were merged. The merge process is depicted in Figure 14. The steps for matching events between different data files are as follows.

Step 1: determine the shared markers for the dataset (Table 8). The shared markers are used as the vector to calculate the distance between events in different data files.

Step 2: take tube *i* (start with the first tube; *i* = 1), and iterate over all the remaining tubes *j*.

Step 3: for each event in tube *i*, calculate the NN in tubes *j*.

Step 4: copy the tube-specific marker (non-shared marker) values from the computed NNs in tube *j* to the events in tube *i*.

Step 5: increment *i*, repeat the above steps. Events in each tube will now have imputed values for markers measured in a different tube.

Step 6: merge all the events across all tubes into a single large matrix.

The resulting data file obtained after merging the original data files and calculating each event's values is a file containing information about all parameters measured in all multicolor staining for each of the events recorded. For the MLL5F panel, we merged tubes 2, 3, 4, 5, and 7. Thus, each merged/calculated data file contained all 18 parameters measured for each of the 2.5 x 10^5 events analyzed per sample (5 aliquots/sample x 5 x 10^4 events/aliquot). The tubes merged for the different datasets, and the merge parameters are described in Table 8. We implemented the merge using scikit-learn API (Pedregosa et al., 2011).

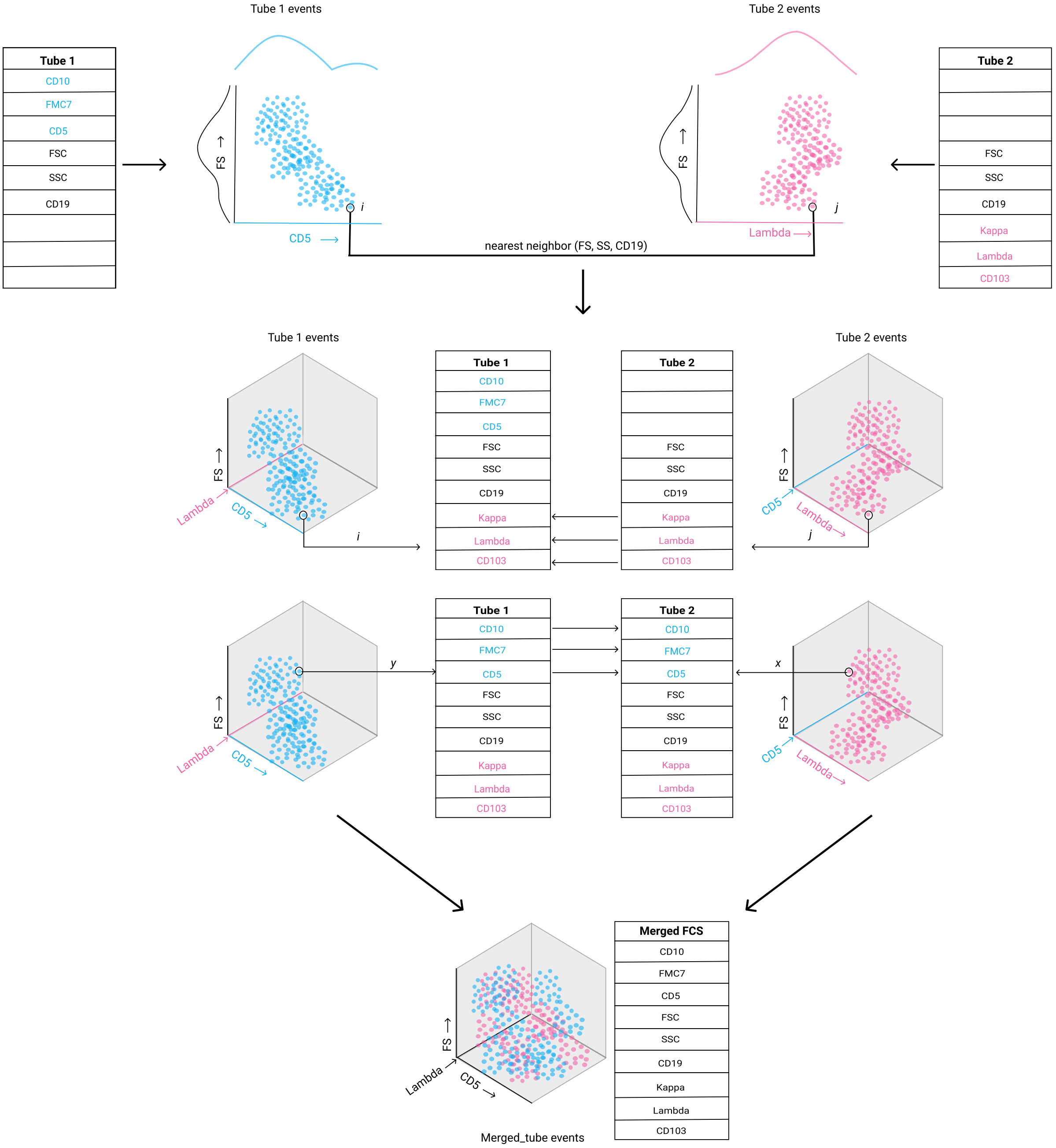


Figure 14: Overview of NN merge. NN merge is shown here for two tubes with three shared markers. Each tube has three tube-specific markers: CD10, FMC7, and CD5 are tube 1-specific markers, while tube 2-specific markers are Kappa, Lambda, and CD103. Events are shown in a two-dimensional space with one shared marker (FS) and one tube-specific marker (CD5 for tube 1 and Lambda for tube 2). For each event "i" in tube 1, the NN in tube 2, "j," is computed in terms of the shared markers (FS, SS, and CD19). Next, tube 2-specific markers from "j" are copied over to "i." The process is repeated for all events in tube 1 so that all the tube 1 events will have imputed values for tube 2-specific markers (Kappa, Lambda, and CD103). Next, for each event, "x," in tube 2, the nearest neighbors in tube 1, say "y," are computed, and tube-1 specific markers are copied over to tube 2. After this step, all tube 2 events will have imputed tube 1-specific markers (CD5, CD10, and FMC7). The events can now be analyzed for the imputed markers that were previously missing. Finally, the expression vectors of all events across tubes are merged, resulting in a combined FCS file consisting of events from both tubes with all the measured parameters.

Table 8: Merge parameters

Panel	Merged tubes	Shared markers
MLL 9F (base data)	1, 2	FS INT LIN, SS INT LIN, CD19, CD45
MLL 5F	2, 3, 4, 5, 7	FS INT LIN, SS INT LIN, CD19, CD45
Bonn	1, 2	FS INT LIN, SS INT LIN, CD19, CD20
Berlin	2, 3, 4	FS INT LIN, SS INT LIN, CD19, CD45
Erlangen	B1, B2	FS INT LIN, SS INT LIN, CD19, CD45

The tubes merged, and the shared markers for each dataset are reported here. The tubes to be merged for each dataset are chosen so that all five datasets have the same CD marker set. The merge resulted in a combined FCS file with 18 parameters for each event.

Merged datasets

Multiple tubes per sample were merged into a single FCS file for all the five datasets described in the “Material” section. The tubes merged were chosen such that the datasets had maximum overlap in terms of the number of markers. The merged MLL9F panel was used to train the new merged base model, while the merged target datasets were used to train the respective target models. The compensated data was used with no additional re-linearization or transformation. The channel intensities were rescaled to “0” and “1” for all the datasets.

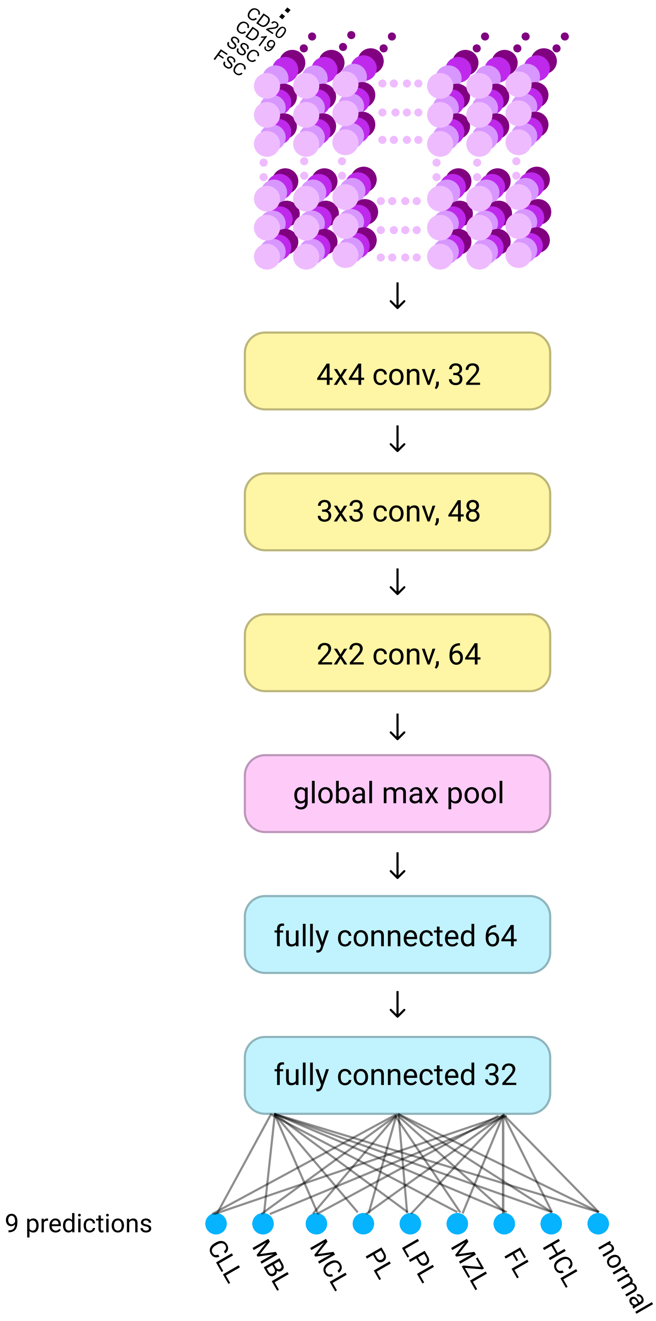
Extended SOM training

The SOM training was updated to account for marker variances between the datasets. As before, individual SOM transformation used pre-initialized node weights from a reference SOM. The reference SOM from the merged base dataset was used as the pre-initialized weights for the target datasets to ensure the same initial tree structure. By disregarding the associated fluorochromes, markers were aligned to the base dataset by matching FS, SS, and as many CD markers as possible. In case of missing markers in the target dataset, they were set to “n/a”; any new CD markers in the target set that were not found in the base dataset were ignored. The SOM implementation was adapted to account for missing data values by modifying the training process (Samad and Harp, 1992). We introduce a masking matrix with values 0 and 1 for each value in the original data: “1” indicates that the data value is valid, and “0” indicates that the data value is invalid, and hence the data point should be ignored for any calculations. The SOM training was then adjusted to use the mask values to ignore invalid data points for the best-matching unit calculation and weight updates.

All training parameters for the SOM generation were kept the same as described in the previous SOM section in phase 1.

Transfer learning

Figure 15 shows the modified CNN architecture for merged samples. Three convolution layers process the merged SOM with decreasing filter sizes. The convolution layers are followed by a global max pooling layer that summarizes filters across the SOM map's spatial dimension. Two fully connected dense layers then combine the information to generate class predictions. All the network parameters, such as the number of kernels, filter sizes, and the number of nodes in each layer, were kept the same as the unmerged model. The only change to the architecture is that there was no need to replicate the convolutional layers for each tube; thus, the concatenation layer was redundant and thus removed.



Modified CNN architecture

Figure 15: Modified CNN architecture. SOM generated for the merged FCS sample is processed by three convolution layers with varying filter sizes. A single global max pool layer is used after the convolution layers, followed by two fully connected layers that combine the features and generate class predictions. The CNN architecture remains the same for both standalone and transfer learning protocols. Each layer is initialized with weights from the base model for transfer learning. The two fully connected layers are frozen and not retrained to avoid overfitting.

The merged base dataset was then used to train a base model with the new CNN architecture for 20 epochs using the Adam optimizer with a learning rate of 0.001 and a global weight decay of 5e-6. The weights were initialized randomly for the base model. We refer to this model as MLL9F_base.

Two models were trained for each target dataset: a standalone model without knowledge transfer and a second model with knowledge from the base model (MLL9F_base). The weights for each layer in the target model with TL were initialized with trained weights from the base model's corresponding layer, while for the standalone models, these were randomly initialized. The standalone models’ hyperparameters were kept identical to the base model - 20 epochs, a learning rate of 0.001, and a global decay of 5e-6. For the second set of models with TL, we used the same learning rate and global decay while the number of epochs was reduced to 15.

Furthermore, the two dense layers were frozen by setting the “trainable” hyperparameter as false. When using TL, the norm is to freeze the convolution layers and retrain only the dense layers to avoid overfitting. However, in our case, the FCS panel composition differs from the base data. Therefore, to account for changes in the panel, we keep the convolution layers unfrozen and retrain them to learn the filters for the target FCS panel. Instead, we freeze the two dense layers that combine information for generating class prediction since the classes to be predicted are the same as in the base task.

The training curves show that the TL models converge with the standalone models with the chosen parameters (Figure 16). The TL models have a lower initial validation loss and reach the asymptote faster than the standalone models showing that the knowledge transfer from a pre-trained base model adds to the training of the CNN. While the TL loss for the Erlangen panel does not converge with the standalone model, the classification performance is still improved with TL. The lack of convergence in model loss could result from Erlangen’s different diagnostic setup, resulting in a small, highly imbalanced dataset that significantly diverges from the base data.

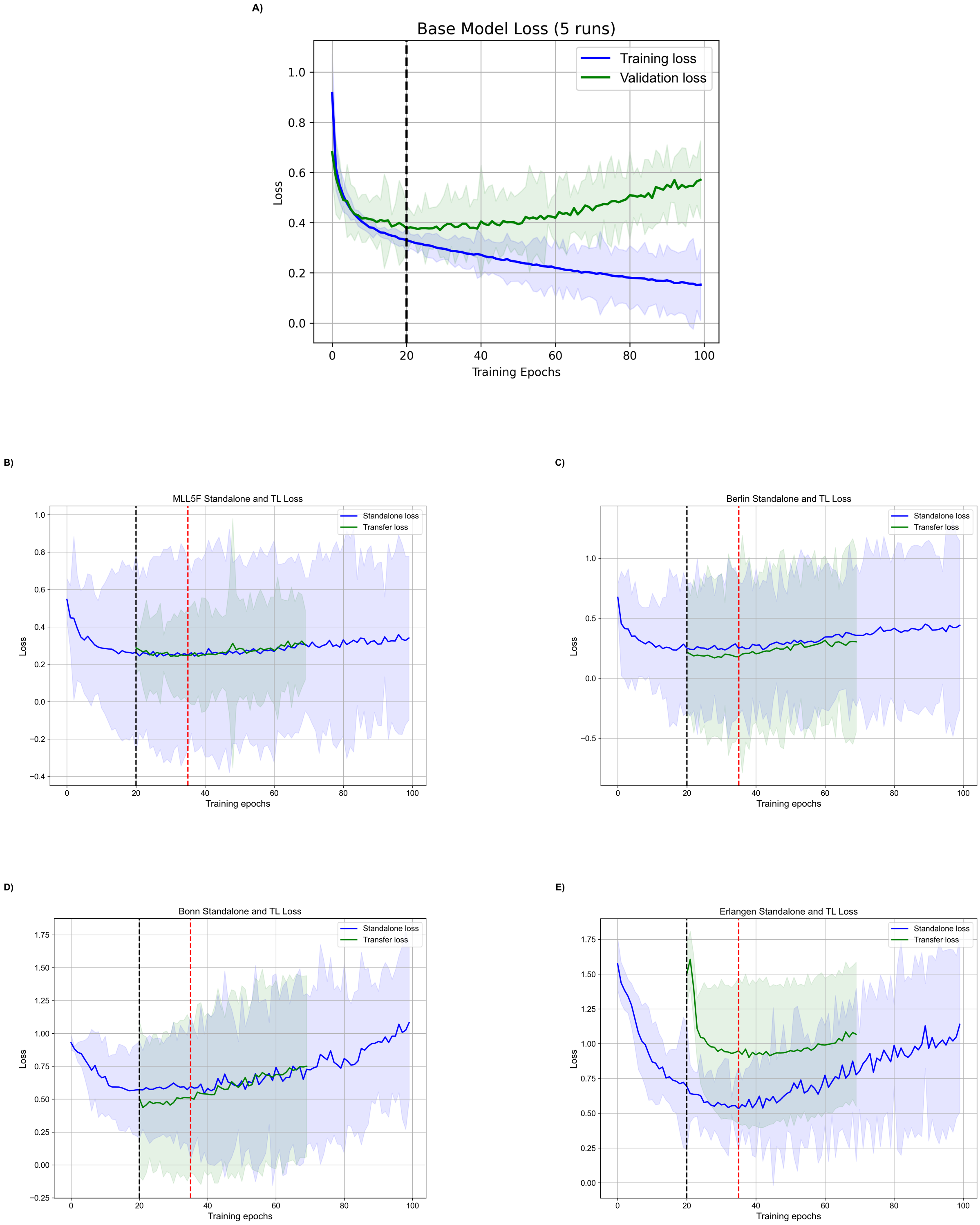


Figure 16: Model convergence plots. A) shows the training and validation loss for the base model. The dotted line represents the cutoff for the number of epochs, after which the model starts overfitting. B), C), D), and E) show the convergence of the transfer learning model to the standalone model for MLL5F, Berlin, Bonn, and Erlangen panels. The dotted lines represent the cutoff for the number of epochs for standalone (black) and transfer learning (red). For all target panels other than Erlangen, the TL model's validation loss starts lower than the standalone models' loss and converges to the standalone model's. While the TL loss for the Erlangen panel does not converge with the standalone, 15 epochs are still used as the cutoff, as the loss flattens out after this threshold, and the models overfit and do not benefit from additional training epochs.

We perform 10-fold validation for all the target datasets to avoid any bias resulting from a single random train-validation split, especially for the smaller datasets. Each target model was trained on the training split, and performance metrics were calculated for the validation split of the respective target dataset. The average scores across the 10-fold validation are reported as the final performance measure.

3. Results

Various performance metrics were analyzed for both unmerged and merged models. Transfer learning results were thoroughly evaluated and compared to standalone models to assess the effect of knowledge transfer. Further, the quality of the merged datasets was critically evaluated to ensure no unwarranted artifact was introduced by the data merging. For all the models generated, we make the decision processes of the AI available for better interpretation of the results through saliency analysis. Saliency analysis makes the AI explainable by tracing the model's decision for a given prediction. All results and the evaluation criteria used are detailed in this chapter.

3.1 Phase 1 - Model performance

The result of the classification process for a given sample is a score for every class learned by CNN. The subtype with the highest score is the likeliest diagnosis and is used for performance readout (top-1 accuracy). The performance metrics were computed on a hold-out test set of 2,378 samples, resulting in an average F1 score of 0.78 and a weighted F1 score of 0.94 for the eight-class classification (CLL/MBL, MCL, PL, LPL, MZL, FL, HCL and normal). When distinguishing only between B-cell neoplasms and healthy control, the average and weighted F1 score of the two classes of comparable size increases to 0.98. The confusion matrix for the CNN indicates that misclassifications are non-uniformly distributed (Figure 17A). Especially the subtypes PL/MCL and MZL/LPL are more likely to be mistaken, which is representative of their high flow cytometric profile similarity.

The percentage of lymphoma cells was estimated by human experts for each sample, with the lowest being 0.1% lymphoma cells. As MBL is defined by fewer than 5,000 cells with the typical CLL profile in flow cytometry, we list MBL as a separate class in the confusion matrix to allow for a more fine-grained analysis of classification sensitivity. While there are no false negatives for CLL, some cases labeled as MBL were misclassified. Most of the MBL cases misclassified as CLL had the number of lymphoma cells close to the distinguishing threshold of 5,000/50,000 = 10% in the MLL9F panel.

Furthermore, the high number of MBL cases classified as CLL and vice versa is a technical artifact that will only decrease the F1 scores of the nine-class problem but does not affect the F1 scores of classification problems where these subtypes are merged. In comparison, the error rates between the other B-cell neoplasm classes reflect a meaningful phenotypic similarity of these disorders. This similarity can be seen in the hierarchical clustering of the confusion matrix and the t-SNE visualization of SOM and intermediate model embeddings in Figure 17B. The t-SNE plot also shows cases with unseen diagnoses, such as multiple myeloma(MM), acute myeloid leukemia (AML), and hairy cell leukemia (HCLv), clustering with the normal class as expected.

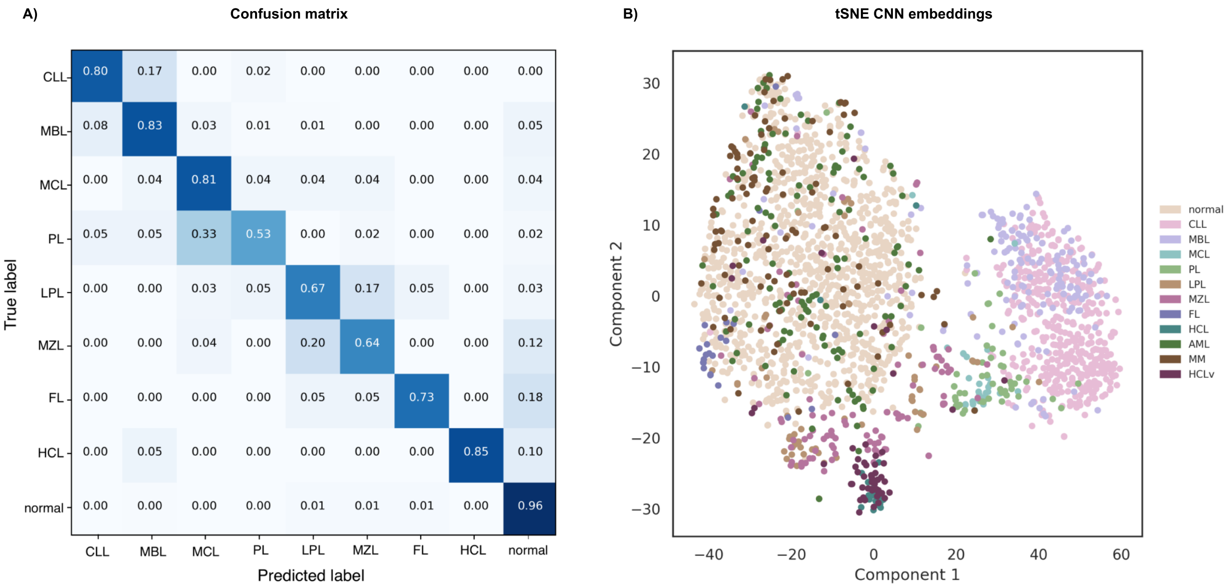


Figure 17: Base model performance. A) The confusion matrix shows the classification of each of the classes. CLL and MBL are shown as separate classes for fine-grained analysis. The confusion matrix also shows higher error rates between clinically similar subtypes. B) This similarity can also be observed for single cases in t-SNE embeddings of the intermediary output from the concatenation layer, showing clusters of MCL/PL and MZL/LPL.

Additionally, ROC curves were generated for the CNN model's prediction. ROC curves show the separability of the classes at all possible thresholds; that is, how well the classifier can separate the classes. The curves were generated for each class in a one-vs-rest approach (Figure 18). The ROC curves show high separability for each class compared to the others, indicating that the trained model is able to classify samples of each class with high confidence. The ROC curves and AUC scores provide an evaluation metric for the usability of the classifier. The usability metric can increase confidence in the model's prediction in a diagnostic setting.

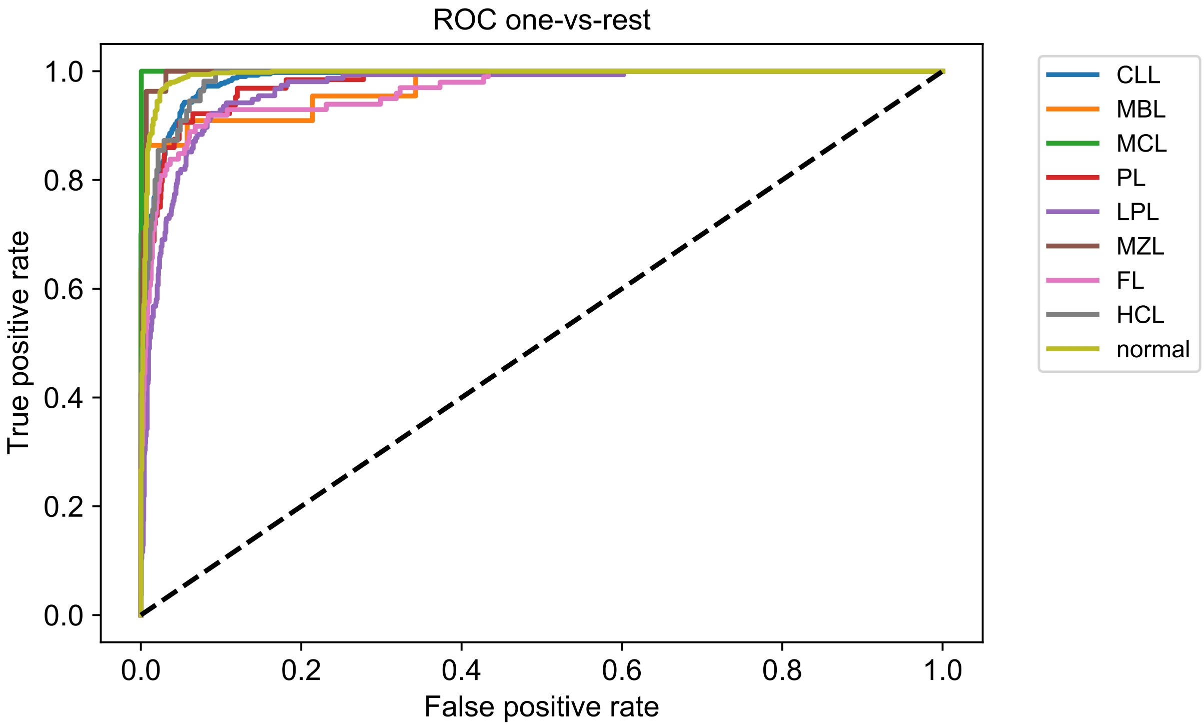


Figure 18: ROC curve. The receiver operator characteristic (ROC) curve is calculated for each class in a one-vs-rest fashion and shows a high response for all classes.

3.1.1 Comparison with ML models

The CNN model was compared to classical machine learning methods such as random forest, kNN, naive Bayes classifier, and a simpler dense neural network implementation with a similar number of trainable parameters compared to our CNN. Table 9 below reports the results on the 10% validation dataset for the CNN model (F1 0.70) in comparison to dense neural nets (F1 0.61), random forest (F1 0.45), and other alternative models. All models were evaluated on randomly upsampled data for each class with 6000 CLL/MBL (CM), 6000 normal, and 1000 each for LPL, HCL, MCL, MZL, PL, and FL. The upsampling process creates additional data samples by duplicating random samples that can be used for evaluation. Further, a grid search was used to find the number of neighbors and the number of estimators for kNN and random forest. The CNN (109,336 parameters) model was trained with the architecture shown in Figure 10, whereas the dense neural net was trained with two hidden layers on the concatenated SOM data for all three tubes for 15 epochs.

Table 9: Comparison of different classification models

Classification model	average F1	weighted F1
kNN	0.43	0.76
naive Bayes classifier	0.41	0.76
Random Forest	0.45	0.84
Dense Neural Net	0.61	0.84
CNN (10x10)	0.70	0.88
CNN (32x32)	0.76	0.93

Comparison of different classification models. All models were trained with a 10x10 SOM. Additionally, the CNN was evaluated for both 10x10 and 32x32 SOMs.The neural networks were trained for 15 epochs.

The comparison shows that CNN outperforms the other classical models. Furthermore, learning curves were analyzed for the top 3 models - CNN, dense neural net, and the random forest classifiers to evaluate the learning in these models. The CNN classifier again shows a superior overall accuracy and average F1 scores compared to alternative classifiers. Particularly noteworthy is the remarkable gain in performance for a growing number of training samples that has not reached a plateau (Figure 19), indicating that an even more extensive training set may further increase the performance of the CNN model.

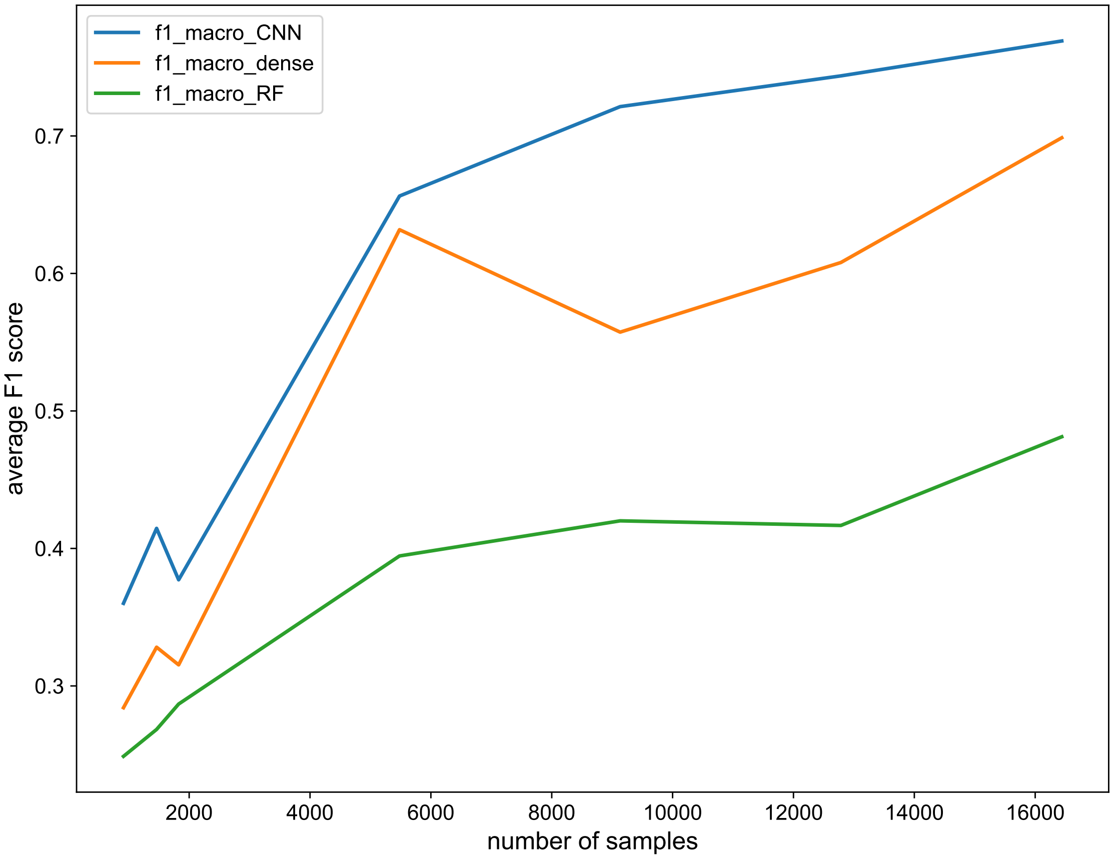


Figure 19: ML model comparison. Comparison of average F1 score against the number of training samples for different models. All three models were trained with 32x32 SOMs as inputs, and the overall average F1 score for the given 8-class classification problem is computed on a randomized 10% validation split of the training data. The models are trained with an increasing number of samples, starting with 913 samples and increasing every iteration to 16500 samples. The F1 score was computed for each training iteration to generate the learning curves.

3.1.2 Saliency analysis

A saliency map is a visualization technique that allows us to gain insights into the decision-making process of a neural network. These maps help understand what each layer of a CNN focuses on, thereby making it possible to explain and interpret the prediction of the network. There are several ways to generate saliency maps. We used the gradient-based approach introduced by Simonyan et al. (Simonyan et al., 2014). The gradient-based method involves computing the spatial support of a particular class in an input image using a single back-propagation pass through the CNN.

Given an image, a class “c,” and a trained CNN classifier with the class score function, the pixels in the image can be ranked based on their influence on the class score. The ranks can then be used to generate maps indicating the spatial region of the image that was the most influential for the given class prediction. We adapted the software implementation of the Keras visualization toolkit (Kotikalapudi and contributors, 2017) to compute saliency plots of a given FCS sample. All FCS events are assigned to the SOM nodes, which are used as the input to our CNN classifier. Therefore, we first compute the saliency of SOM nodes of a given SOM image by defining the saliency of each node as the maximum gradient over all input channels. SOM node saliency values were then mapped back to single FCS events by assigning the saliency value of the nearest SOM node to each event.

Selected scatter plots are shown for a representative CLL sample, which has correctly been classified (Figure 20). The results of the standard manual gating strategy are compared to the populations highlighted by saliency analysis. A likely pathological population has been identified by positivity for CD5, CD19, and CD20 after gating on lymphocytes using CD45 and SS. The corresponding saliency map shows a similar cell cluster that yields the strongest signal for the CLL subtype without prior gating. These maps allow the user to validate whether, for the predicted class, the AI is identifying the appropriate cell clusters.

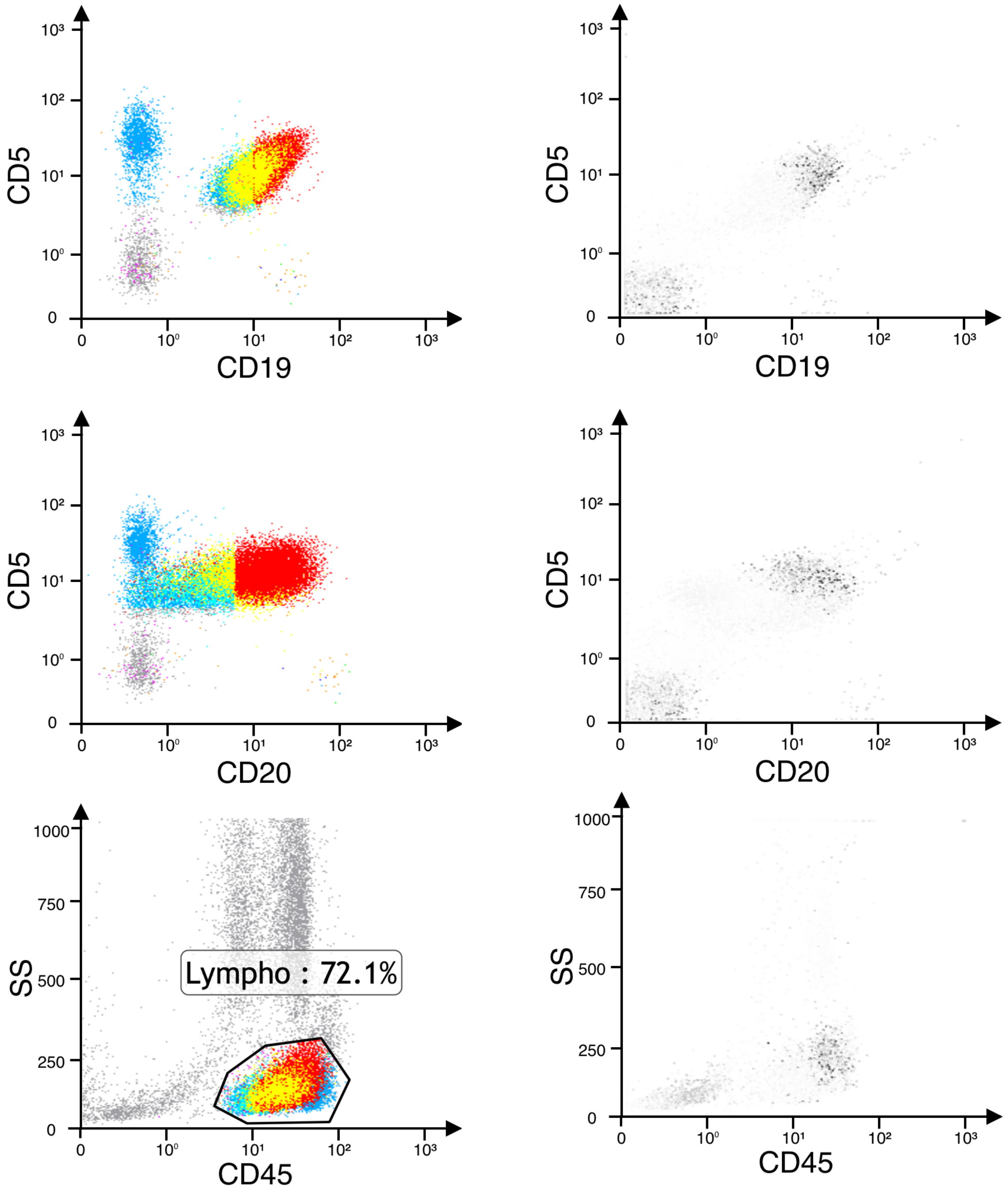


Figure 20: Saliency maps. Human experts identify pathogenic cell populations by a complex gating strategy over multiple 2D scatter plots. The result of the manual gating is shown for a CLL sample in the first column. Cells are colored according to their gate properties in different scatter plots. The saliency plots computed for the CNN in the second column show higher importance assigned to cells in the same region as the gated population. Darker colors represent higher gradients and, thus, higher importance in saliency analysis.

3.2 Phase 2 - Transfer learning results

Before evaluating transfer learning, data merging, a preprocessing step for transfer learning, is evaluated to ensure the NN merging process does not introduce significant artifacts that could impact the classification.

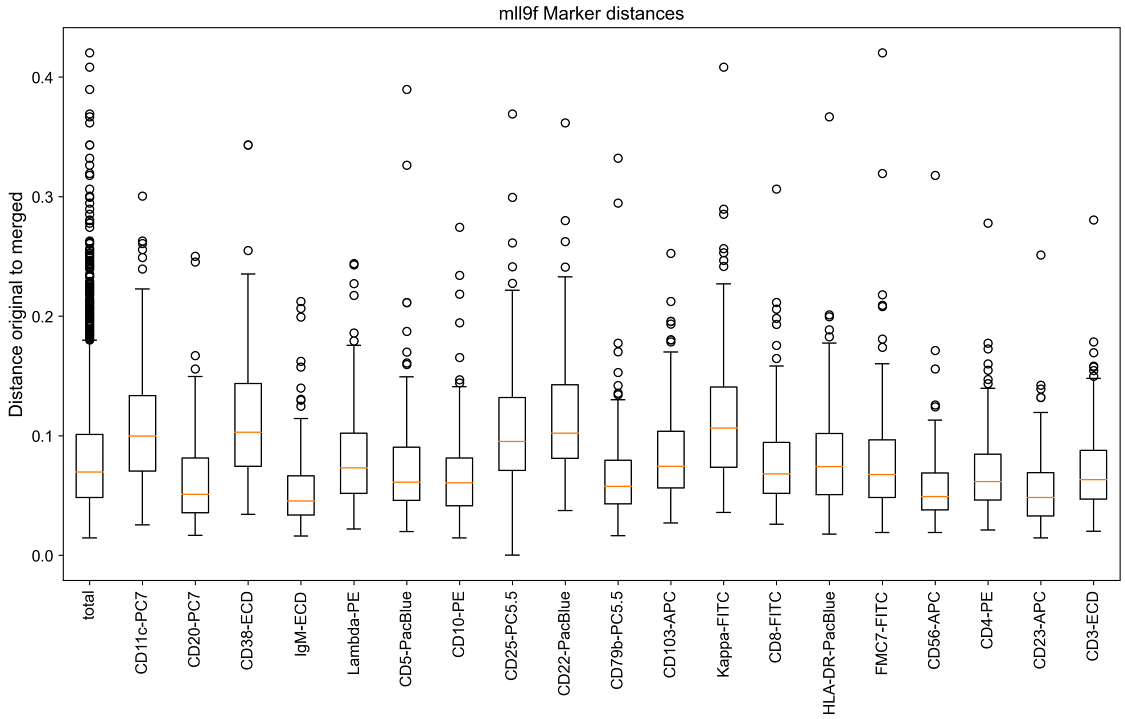
3.2.1 Merge Evaluation

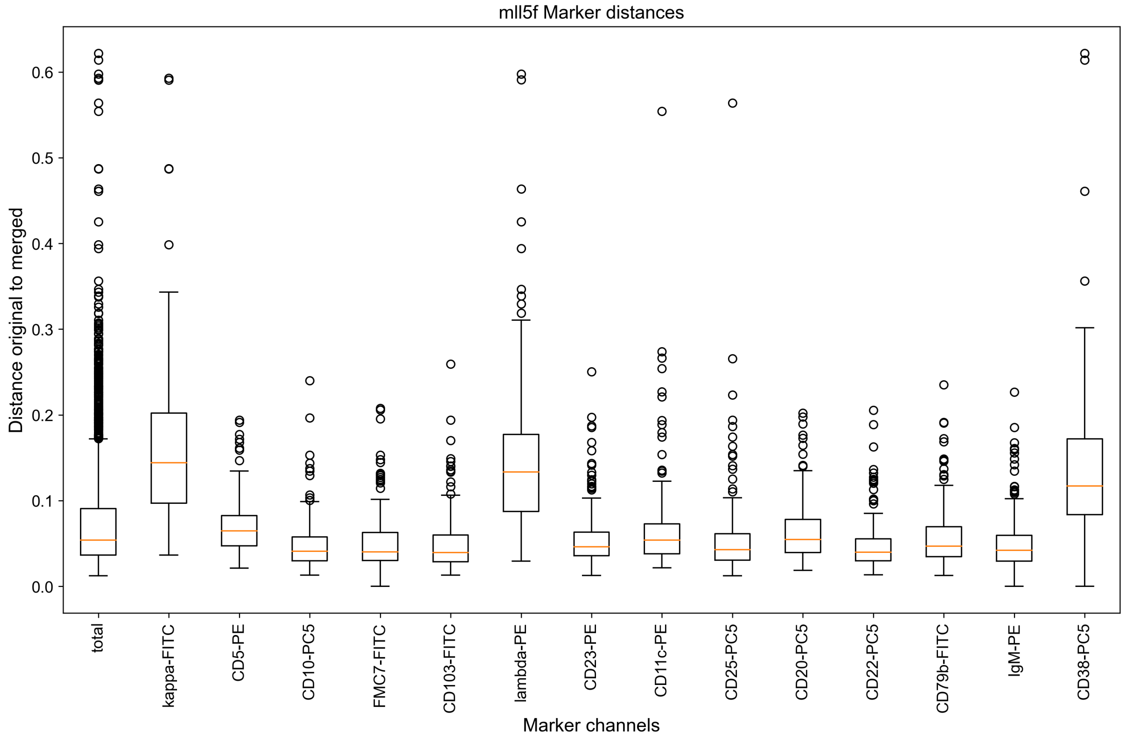
To evaluate the quality of the merged dataset, we use Jensen-Shannon distance (JSD) to quantify the similarity between the distributions of markers in the original and merged datasets, resulting in values between 0 (identical distributions) and 1 (totally disjoint distributions). If p and q are the probability distributions of a marker in the original and merged data, then the JSD is calculated as the square root of Jensen-Shannon divergence (Naghshvar et al., 2015):

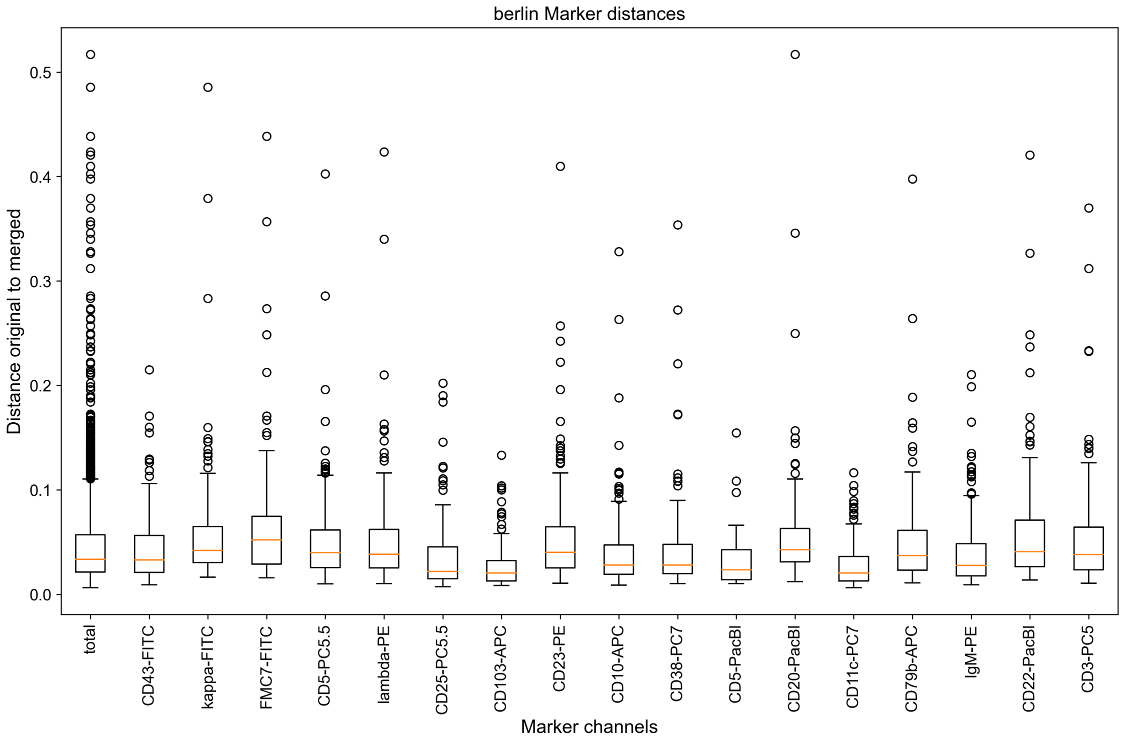
$$\sqrt{\frac{D(p||m) + D(q||m)}{2}}, \tag{3.1}$$

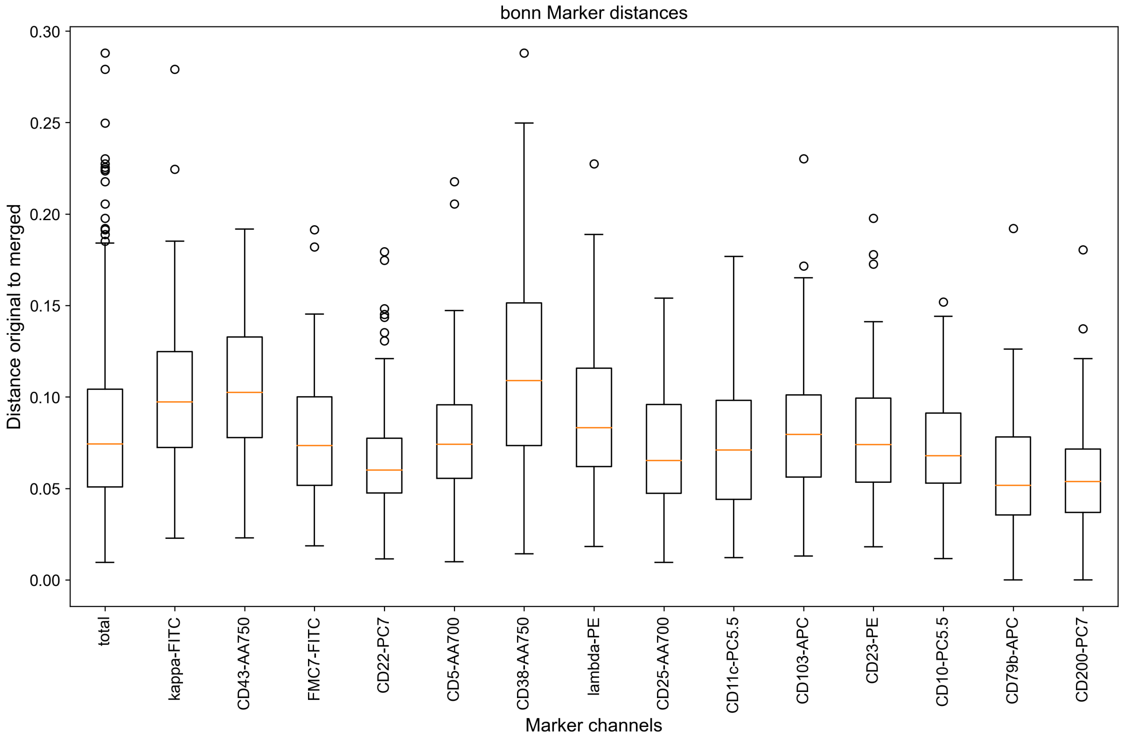
where m is the pointwise mean of p and q, and D is the Kullback-Leibler divergence (Kullback and Leibler, 1951).

We computed the JSD for each non-shared marker between the original and merged sample for all datasets. For each non-shared marker in the merged tubes, the JSD metric was computed using equation 3.1, defined above. We obtained a mean JSD score of less than 0.1 for all the datasets, indicating good agreement between the merged and original datasets in terms of marker distribution. The individual JSD score for each non-shared marker and the average JSD for each dataset are reported in Figures 21 and 22.


mll9f Marker distances
Distance original to merged
0.0
0.1
0.2
0.3
0.4
total
CD11c-PC7
CD20-PC7
CD38-ECD
IgM-ECD
Lambda-PE
CD5-PacBlue
CD10-PE
CD25-PC5.5
CD22-PacBlue
CD79b-PC5.5
CD103-APC
Kappa-FITC
CD8-FITC
HLA-DR-PacBlue
FMC7-FITC
CD56-APC
CD4-PE
CD23-APC
CD3-ECD
Marker channels


mll5f Marker distances
Distance original to merged
0.0
0.1
0.2
0.3
0.4
0.5
0.6
total
kappa-FITC
CD5-PE
CD10-PC5
FMC7-FITC
CD103-FITC
lambda-PE
CD23-PE
CD11c-PE
CD25-PC5
CD20-PC5
CD22-PC5
CD79b-FITC
IgM-PE
CD38-PC5
Marker channels


berlin Marker distances
Distance original to merged
0.0
0.1
0.2
0.3
0.4
0.5
total
CD43-FITC
kappa-FITC
FMC7-FITC
CD5-PC5.5
lambda-PE
CD25-PC5.5
CD103-APC
CD23-PE
CD10-APC
CD38-PC7
CD5-PacBl
CD20-PacBl
CD11c-PC7
CD79b-APC
IgM-PE
CD22-PacBl
CD3-PC5
Marker channels


bonn Marker distances
Distance original to merged
0.00
0.05
0.10
0.15
0.20
0.25
0.30
total
kappa-FITC
CD43-AA750
FMC7-FITC
CD22-PC7
CD5-AA700
CD38-AA750
lambda-PE
CD25-AA700
CD11c-PC5.5
CD103-APC
CD23-PE
CD10-PC5.5
CD79b-APC
CD200-PC7
Marker channels

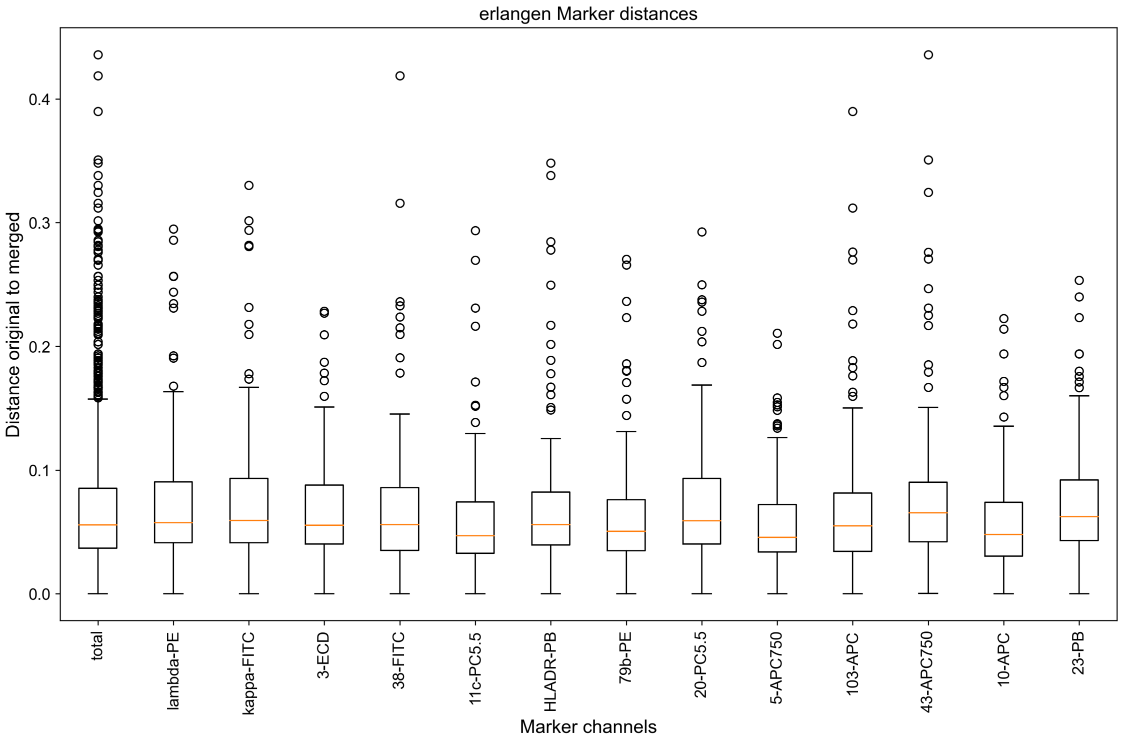


Figure 21: JSD scores. Jensen-Shannon divergence (JSD) scores for each of the imputed, non-shared markers in all five datasets are shown here.

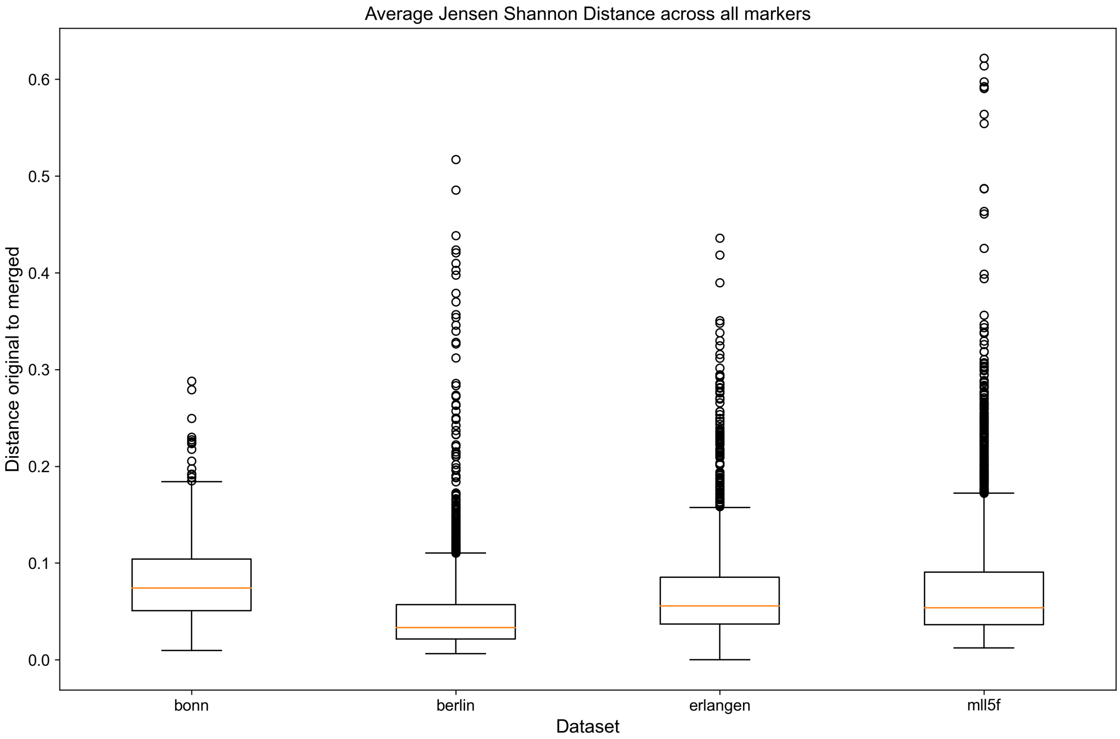


Figure 22: Average JSD scores. The average JSD score across all non-shared markers for each target dataset is computed. All datasets have a mean JSD score of less than 0.1, indicating good agreement between the merged and original data for all four datasets. The outliers were further examined by computing the correlation between JSD scores and classification accuracy using Pearson correlation (PC). PC coefficients were calculated for each dataset's true label score in a k-fold transfer learning experiment. PC for each dataset was found to be slightly negative, indicating that the lower the agreement between merged and original data (high JSD), the lower the classification score.

In manual FCS analysis, cell populations are defined based on the co-expression of markers measured in the same tube. During our merge process, all the non-shared markers between the different tubes are imputed using the nearest neighbor approach, making it possible to define cell populations that were not otherwise possible in the unmerged dataset.

For instance, in the merged MLL9F dataset, a CD5+/CD22+ cluster can be defined. During the NN merge, CD5 expression values are imputed for tube two events and CD22 values for tube one events. Due to the inherent characteristic of the NN computation, some CD22+ events in tube 2 can have neighbors, which are CD5+, and some CD5+ events in tube 1 can have neighbors, which are CD22+, making these events both CD22+ and CD5+. However, these events might not have been positive for both markers if the two markers were originally measured in a single tube. Thus a CD5+/CD22+ cluster in the unmerged dataset would have fewer cells and might not have been a significant cluster. The imputation, in this case, would result in a large “pseudo” CD5+/CD22+ cluster leading to a false pathology.

Although JSD scores do not directly evaluate the extent to which cell populations (based on the co-expression of markers) are preserved in the merged data, the scores provide a way of assessing how the imputation affected individual marker distributions. Given that the density distributions for the markers in both merged and unmerged datasets are very close, the relative number of positive cells for any given marker remains the same. Thus, a population defined by the positive co-expression of two markers is unlikely to have a significantly large pseudo-population in the merged data.

To this end, we manually evaluated the merge process to check for significant artifacts by creating an artificial panel from the unmerged MLL9F dataset. Random samples from tube 1 of the MLL9F panel were artificially split into two tubes (tube a: CD19, CD45, FMC7, CD10, IgM and tube b: CD19, CD45, CD20, CD23, CD5) and then merged using the NN merge algorithm. Cell populations such as CD19+/CD5+, CD19+/CD20+, and many more were manually defined and quantified before and after the merge. No large “pseudo” populations were found in the merged data, confirming the quality agreement shown in the JSD analysis. Moreover, the unmerged and merged models’ performance for the base data was compared to evaluate the effect of NN merge on the CNN classification. The merged base model achieved an overall weighted F1 score of 0.94 and an average F1 score of 0.74. In comparison, the model trained with the unmerged FCS data from tubes 1 and 2 achieved an overall weighted F1 score of 0.94 and an average F1 score of 0.75, indicating that the NN merge did not introduce significant artifacts that negatively impacted the CNN classification.

Further, we evaluated the effect of the number of shared markers on the quality of the merged dataset. The nearest neighbor is calculated based on the expression vector of the shared markers. If the number of shared markers between the tubes is sparse, the probability of introducing “pseudo” clusters are high, thus, making the imputation more error-prone.

To evaluate if there is a minimum number of shared markers necessary for imputing the values of non-shared markers using the nearest neighbor (NN) merge approach, we set up an analysis with various numbers of shared markers. We explored five possible cases for the number of shared markers used to compute the nearest neighbor:

Case 1: All shared markers - FS/SS/CD19/CD45

Case 2: leukocyte markers - FS/SS/CD45

Case 3: B-cells markers - FS/SS/CD19

Case 4: only the scatter measures - FS/SS

Case 5: only CD19

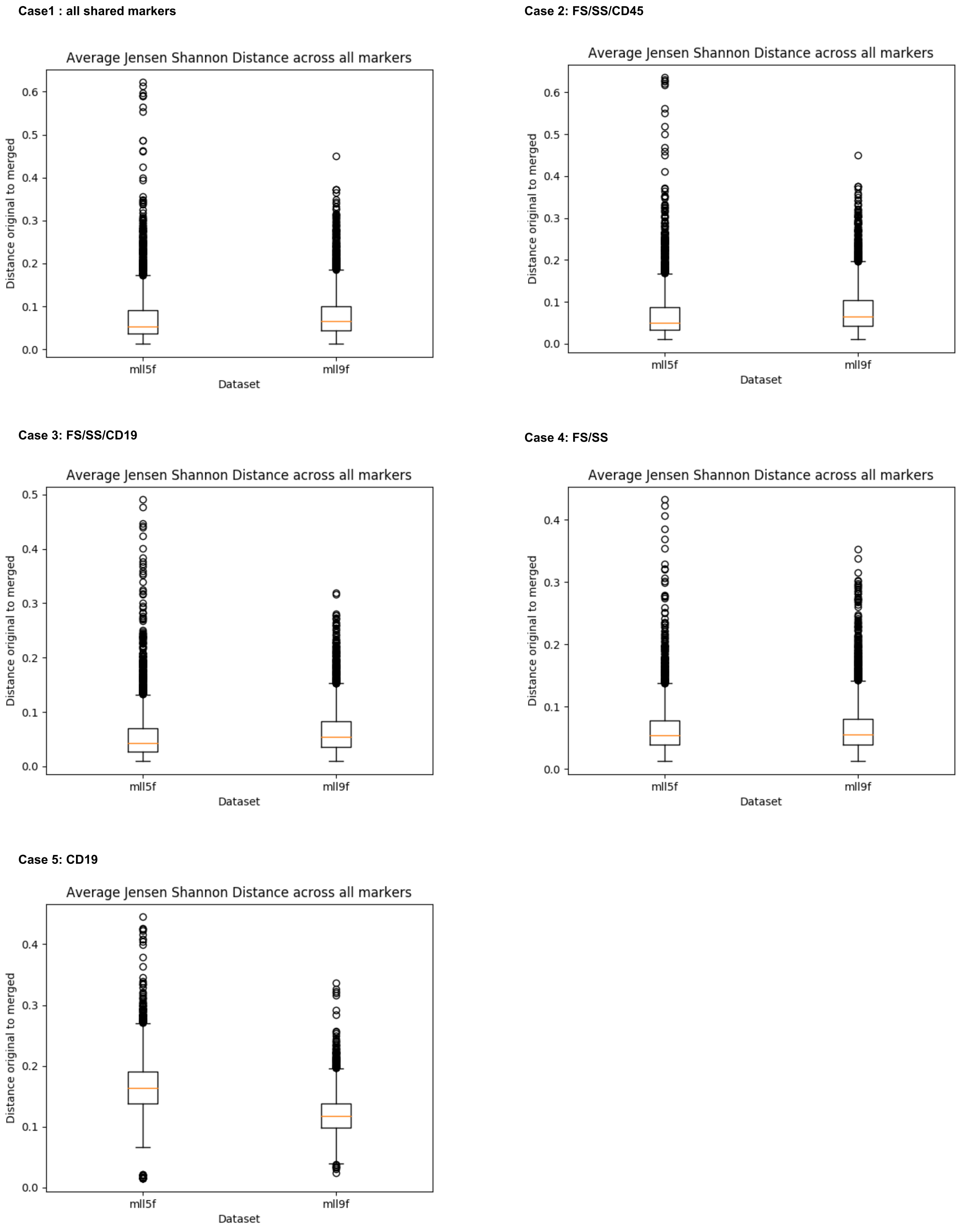
For each case, the merge algorithm described in the "Methods" section was used to merge multiple aliquots and thus impute values for each non-shared marker between these tubes. The imputation quality was again evaluated by computing Jensen-Shannon distance (JSD) scores for each non-shared marker. The distribution of the non-shared marker in the original, unmerged dataset is compared to the distribution after merging and imputation. If the imputations were bad, the two distributions would be distinct and result in a higher JSD score. We used a random subset of 30 samples per class to evaluate the cases for two of our datasets: MLL9F and MLL5F panels. JSD scores were calculated for all the non-shared markers for each case. The average score across all markers for each dataset was used to evaluate the effect of the number of shared markers on imputation quality.

The analysis indicates that the number of shared markers only marginally affects the nearest neighbor calculation in the context of flow cytometry events. Only when a single population marker was used (case 5) did we see a more noticeable reduction in the quality of imputed values. The JSD values for each case are summarized in Table 10 and visualized in the box plots below (Figure 23).

Table 10: JSD scores

	MLL5F	MLL9F
Case 1: all shared markers	0.075	0.079
Case 2: FS/SS/CD45	0.072	0.080
Case 3: FS/SS/CD19	0.079	0.075
Case 4: FS/SS	0.075	0.076
Case 5: CD19	0.168	0.119

Average JSD scores across all non-shared markers for MLL5F and MLL9F datasets are reported here.



Case1 : all shared markers

Case 2: FS/SS/CD45

Case 3: FS/SS/CD19

Case 4: FS/SS

Case 5: CD19

Figure 23: JSD scores comparison. Boxplots for each case show the average JSD score for MLL5F and MLL9F panels across all non-shared markers.

As seen in the plots, in the context of FCS events, standard markers such as FS/SS could be used as a minimal set of shared markers to impute non-shared markers between events in different tubes without significantly affecting the imputed data quality.

3.2.2 TL model performance

We compared the performance of the target models with and without TL. A 10-fold validation was performed on both the standalone and TL models for each target dataset. For each model, weighted and average F1 scores were calculated. The models with TL showed a significant improvement in F1 scores, especially the average F1 scores for all the datasets (Figure 24). Even for the Erlangen panel, where the TL model loss did not converge with the standalone loss (Figure 16E), we still see the added benefit of transfer learning for classification.

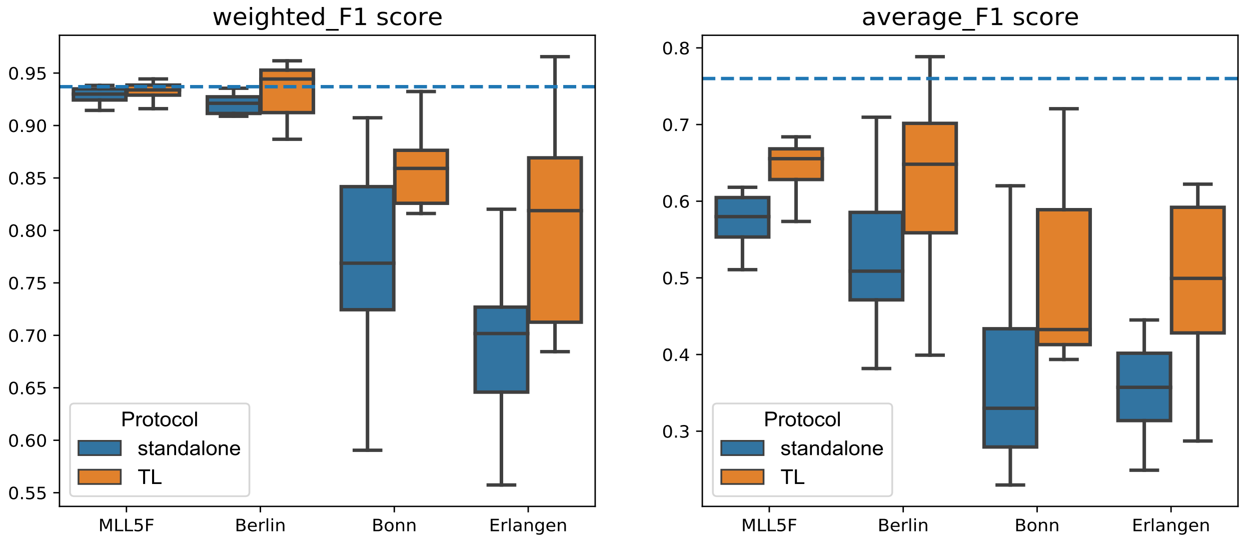


Figure 24: Performance for standalone versus transfer learning. The boxplots show F1 scores obtained: on the left, weighted F1 scores are plotted for each dataset, and on the right, the average F1 scores are shown.The blue dotted line across the plots represents the previously reported base model's performance, considered expert-level accuracy here. The transfer learning models perform better in all four datasets. These models achieve a higher F1 score, especially the average F1 score. A significant increase in average F1 score is seen for MLL5F (p = 1.805 x 10^{-3}) and Erlangen (p = 3.194 x 10^{-2}) panels. For Bonn and Berlin panels, we achieved a p-value of 6.838 x 10^{-1} and 1.659 x 10^{-1}, respectively. All p-values were computed using an independent t-test with Bonferroni correction.

Two of the four target models were able to reach expert-level accuracy with TL. The delta in the performance between the datasets may be attributed to the size of the dataset, the quality of the original data, and the quality of the merged data with imputed marker values. The overall scores obtained by averaging the F1 scores over the 10-fold validation and the 95% CI values for the four datasets are reported in Table 11.

Table 11: Performance metrics

Protocol	Scores	MLL 5F	Berlin	Bonn	Erlangen
With_ TL	f1_ weighted (95% CI)	0.93 (0.92, 0.93)	0.93 (0.91, 0.95)	0.85 (0.81, 0.88)	0.80 (0.73, 0.87)
	f1_ avg (95% CI)	0.64 (0.61, 0.66)	0.62 (0.54, 0.71)	0.50 (0.41, 0.59)	0.52 (0.40, 0.64)
	Precision	0.91	0.93	0.82	0.71
	Recall	0.92	0.93	0.83	0.76
Standalone	f1_ weighted (95% CI)	0.92 (0.91, 0.93)	0.92 (0.90, 0.93)	0.76 (0.69, 0.83)	0.69 (0.63, 0.74)
	f1_ avg (95% CI)	0.57 (0.54, 0.59)	0.52 (0.45, 0.59)	0.40 (0.26, 0.53)	0.35 (0.31, 0.40)
	Precision	0.90	0.92	0.75	0.58
	Recall	0.91	0.92	0.82	0.73

Weighted and Average F1 score along with 95% confidence interval (CI) values for the four target datasets for models with knowledge transfer and standalone models without transfer learning. The F-scores were calculated as an average of the 10-fold scores for each dataset. Precision and recall are calculated as the weighted average per class scores for each fold and then averaged over the 10-folds.

In addition, ROC curves were compared for both the standalone and TL models. The ROC curves are generated for the validation predictions from the CNN's softmax layer for each fold with a one-vs-rest approach and averaged to get the mean ROC curve. The comparison of these plots clearly shows that transfer learning not only benefits the larger classes but also increases the classification performance for many of the smaller classes (Figure 25).

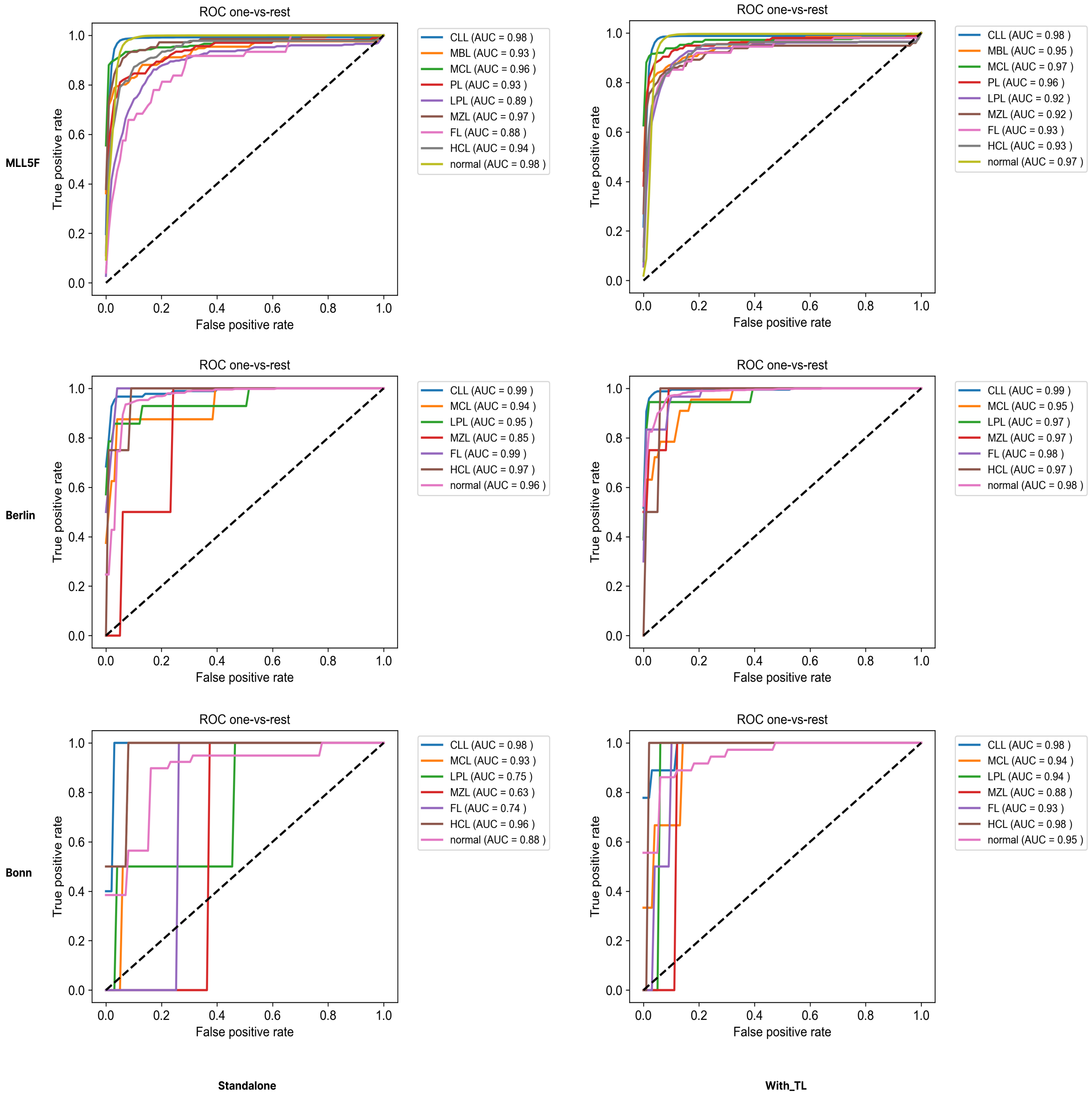


Figure 25: ROC curves for standalone versus transfer learning. The mean ROC curves and AUC for each class across the ten folds are generated in a one-vs-rest manner. Since the datasets are highly imbalanced, there were no validation samples for some of the classes in some folds for the Bonn and Berlin panels. Such folds were not used to compute the mean ROC. The first column shows the ROC curves for standalone models; ROC curves with transfer learning are shown in the second column. ROC curves for the MLL5F panel are in the first row, followed by the Berlin panel in the second row. The third row above is for the Bonn panel. As seen in all the rows, the models with transfer learning achieve a higher AUC.

3.2.3 Learning curve analysis

As described in the previous section, TL increases the overall performance of the models by allowing the already learned knowledge to be utilized. In this case, the target datasets were entirely available for training the new model. However, in an already established diagnostic workflow, any changes to the FCS panels happen in real-time. During this transition from the old to the new FCS panels, very little data from the new panels are available, and it may take a while for a laboratory to switch entirely to the new protocol.

Here, we describe two use cases that result in significant changes to the FCS diagnostic panel and require an AI model to be adapted quickly. We use our current workflow to adapt the base model for both cases and analyze the model's learning curves for each case. A learning curve shows the model's score for varying numbers of training samples and can be used to compare different settings or algorithms and determine the amount of data used for training (Meek et al., 2002). We demonstrate that TL with merge increases the models’ overall performance, and the models have a higher start on the learning curve for smaller sample sizes.

Case 1: Transition to a new cytometer within the same laboratory

In FCS diagnostics, switching to a device that supports more fluorochromes per measurement is a common transition in a diagnostic laboratory that optimizes its workflows by updating its equipment. Usually, this process involves a few weeks, during which samples are measured with both protocols, the old one validating the new one. However, this means that only a few samples from the new protocol are available to train a new classifier. Using knowledge transfer, we show that transition can be handled quickly by adapting an existing AI model.

We set up a transition scenario from a five-color cytometer to a nine-color one using our MLL5F and MLL9F merged datasets. We trained a model with the MLL5F panel and used this as the base network to train a new model for the MLL9F panel. We used an increasing number of samples in the training set for each iteration of the learning curve while the validation set for each iteration was kept the same. We started with five random samples per class and iteratively increased the number of training samples by five in each class until fifty random samples per class. F1 scores were recorded for each iteration. The learning curve (Figure 26A) with TL shows a higher start and asymptote for the target network; the confusion matrix obtained with five training samples per class (Figure 27A) shows a significant improvement in classification, especially for the smaller classes.

Case 2a: Model adaptability across laboratories

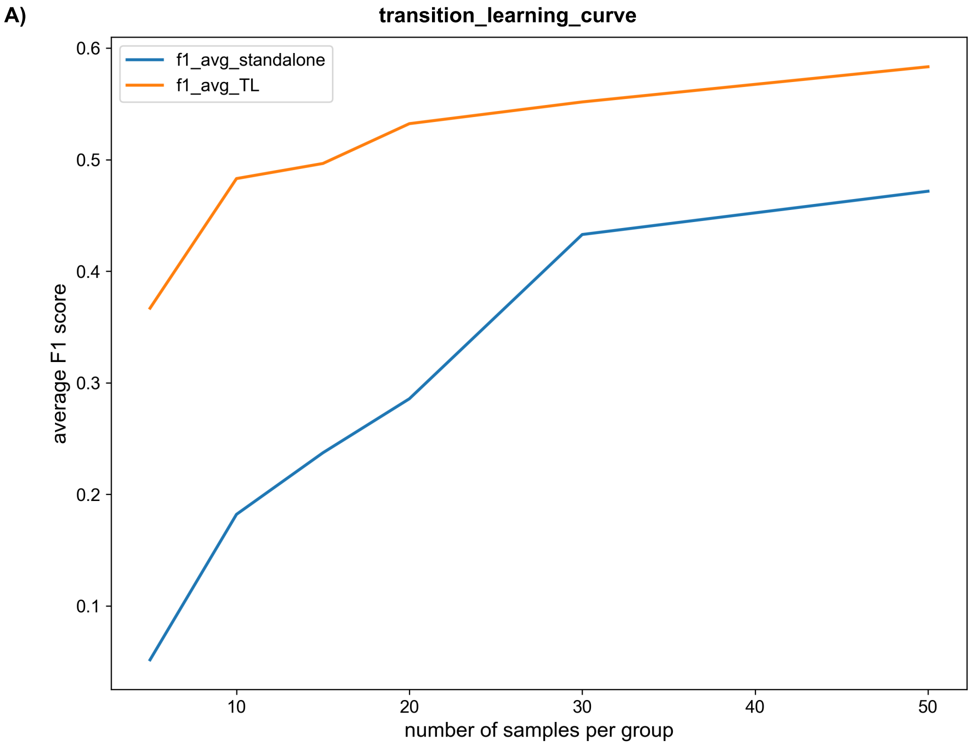
FCS diagnostic workflows are relatively similar across laboratories. However, the FCS panel used for diagnosis varies depending on the cytometer and antibodies measured. For an AI model, the reported performance is valid for the given FCS panel. When the model is used to interpret different FCS data, the performance drops significantly without changes to the underlying architecture and parameters. Training a new model requires a longer training time and large datasets.

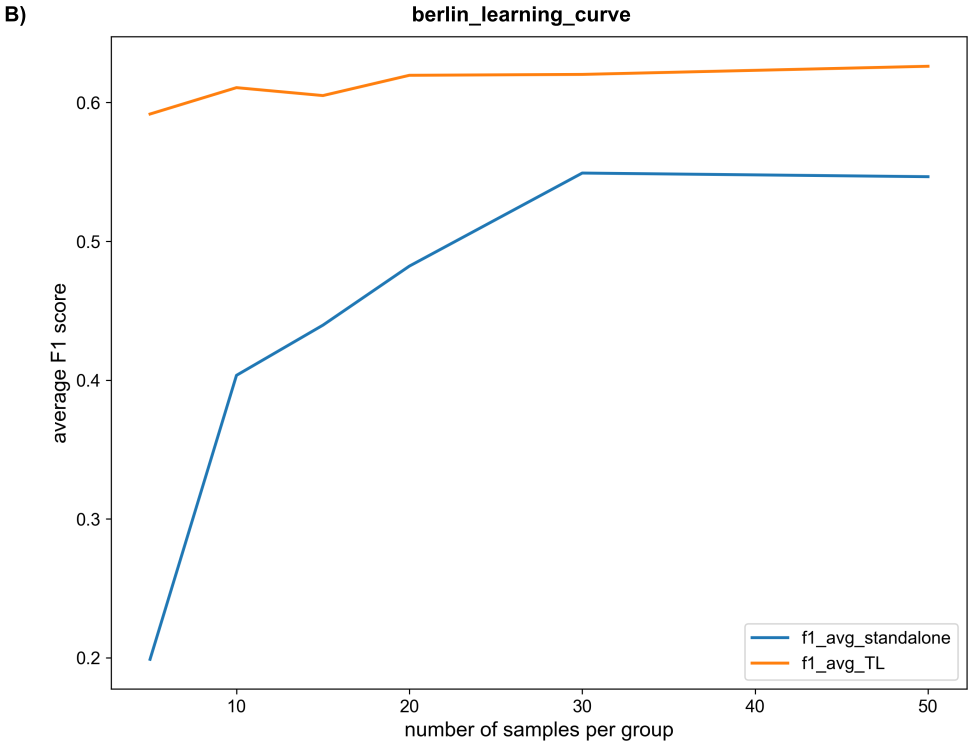
Here, we demonstrate that our workflow can extend a model trained on a specific FCS panel with an extensive training dataset to different FCS panels with lesser data (Table 1). We used our merged base model (MLL9F_base) to train new models for Bonn and Berlin panels. Both target models showed a significant increase in overall performance with TL. As with the previous experiment, the learning curves were obtained for an increasing number of training samples in each class. The target models were trained with five random samples per class, which were gradually increased to fifty samples per class. The F1 scores showed a significantly higher start and overall performance in inter-laboratory adaptation with our workflow (Figures 26B and 26C).

Case 2b: Cross-laboratory adaptation with different diagnostic setting

A screening panel was used for the Erlangen dataset to diagnose B-cell neoplasms with a separate classification panel for further subtype determination. In this scenario, most samples would only have a single screening panel which is different from the previous setting. In order to show that the existing model could still be beneficial, we trained a new model with the same architecture and parameters as our MLL9F_base model for the screening panel to obtain a “normal” versus “pathological” binary classification. The model was trained as a standalone model without transfer learning to verify the usability of the model for the primary classification of the screening panel. The resulting model could correctly classify 86% of pathological and 96% of normal samples with only a single screening panel. This primary model could be used to flag pathological samples that would require further examination.

Furthermore, we used our extended transfer learning with data merging workflow for the 247 samples (see Table 1) with both screening and the classification panel (B1 and B2). We show transfer learning benefits in this setting similar to the previous case, even though the second panel was measured much later, resulting in variations in the data acquired between the screening and classification panels because of sample freezing and preparation. We employed knowledge transfer as described and saw an overall gain in the average F1 score from 0.33 to 0.52. The learning curve (Figure 26D) showed a higher start and asymptote, similar to the other three datasets. This demonstrates that transfer learning can benefit very small datasets with more significant variations.


A)
transition_learning_curve
f1_avg_standalone
f1_avg_TL
average F1 score
0.6
0.5
0.4
0.3
0.2
0.1
10
20
30
40
50
number of samples per group


B)
berlin_learning_curve
average F1 score
0.6
0.5
0.4
0.3
0.2
f1_avg_standalone
f1_avg_TL
10
20
30
40
50
number of samples per group

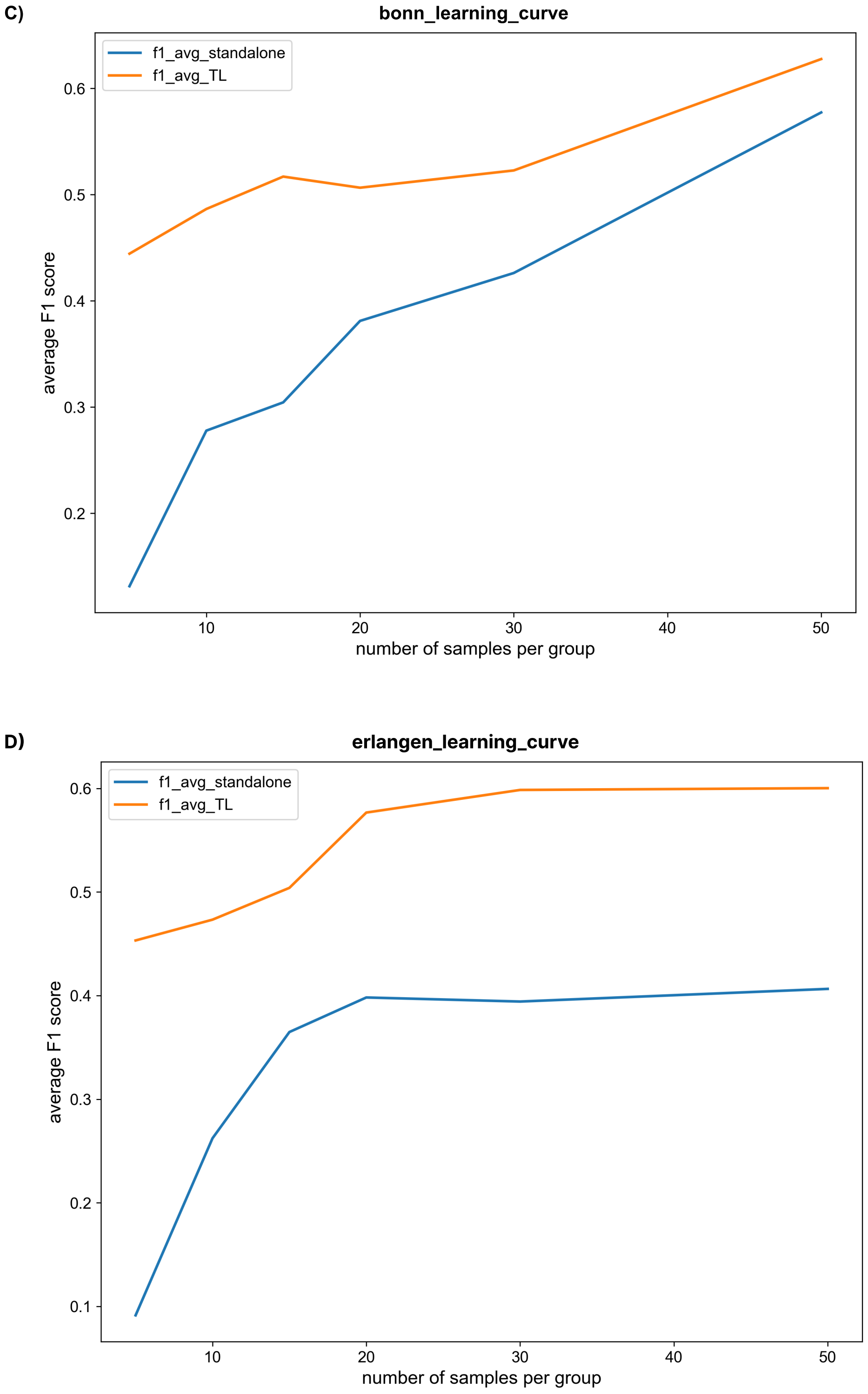
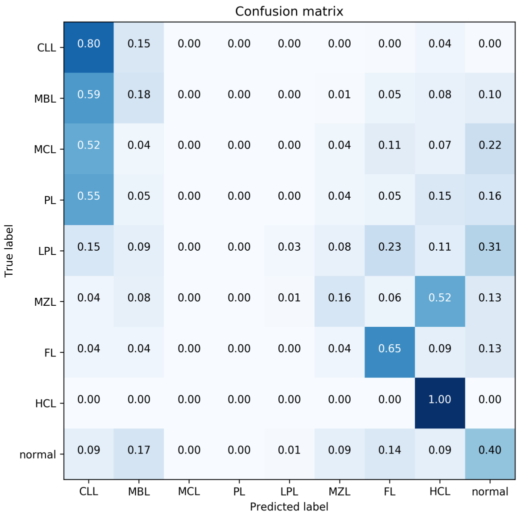
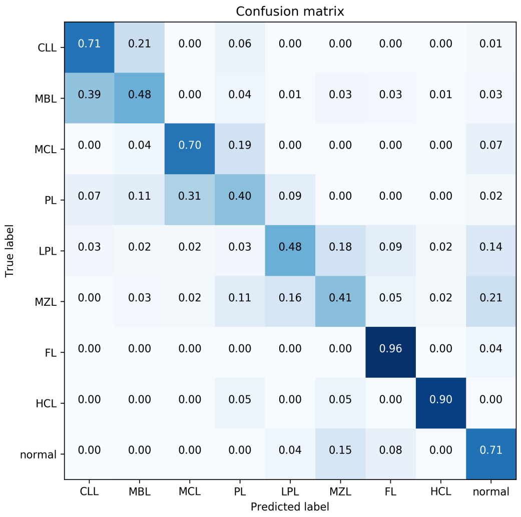


Figure 26: Learning Curves. The learning curve for average F1 scores with various training sizes for all four target datasets is shown here. The curves were obtained with randomly sampled training examples. We start with five training samples in each class and iteratively increase them to 50 samples per class. In cases where 50 samples are unavailable for a given class, existing samples are randomly resampled to create up to 50 samples for the learning curve analysis. The curve for the transition experiment is shown in A), while the curves for cross-laboratory experiments with Berlin, Bonn, and Erlangen panels are shown in B), C), and D), respectively. The learning curves for all panels show a higher start and asymptote with transfer learning and an overall performance enhancement.

A) transition_learning_curve analysis

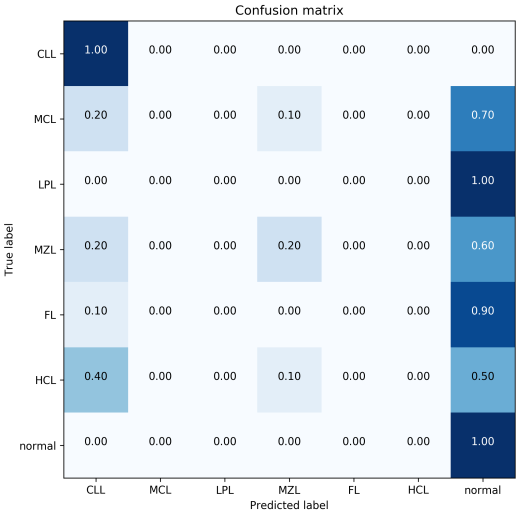


Standalone

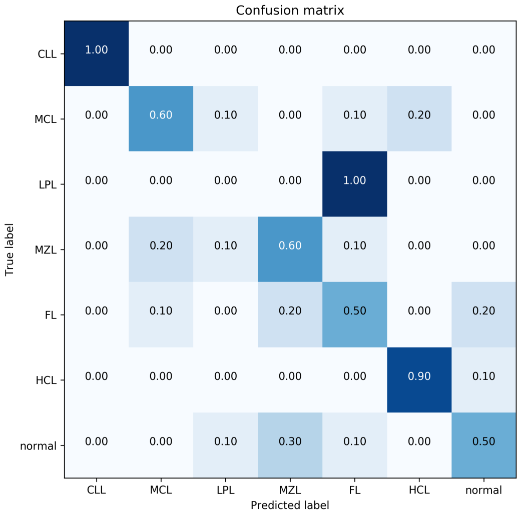


With_TL

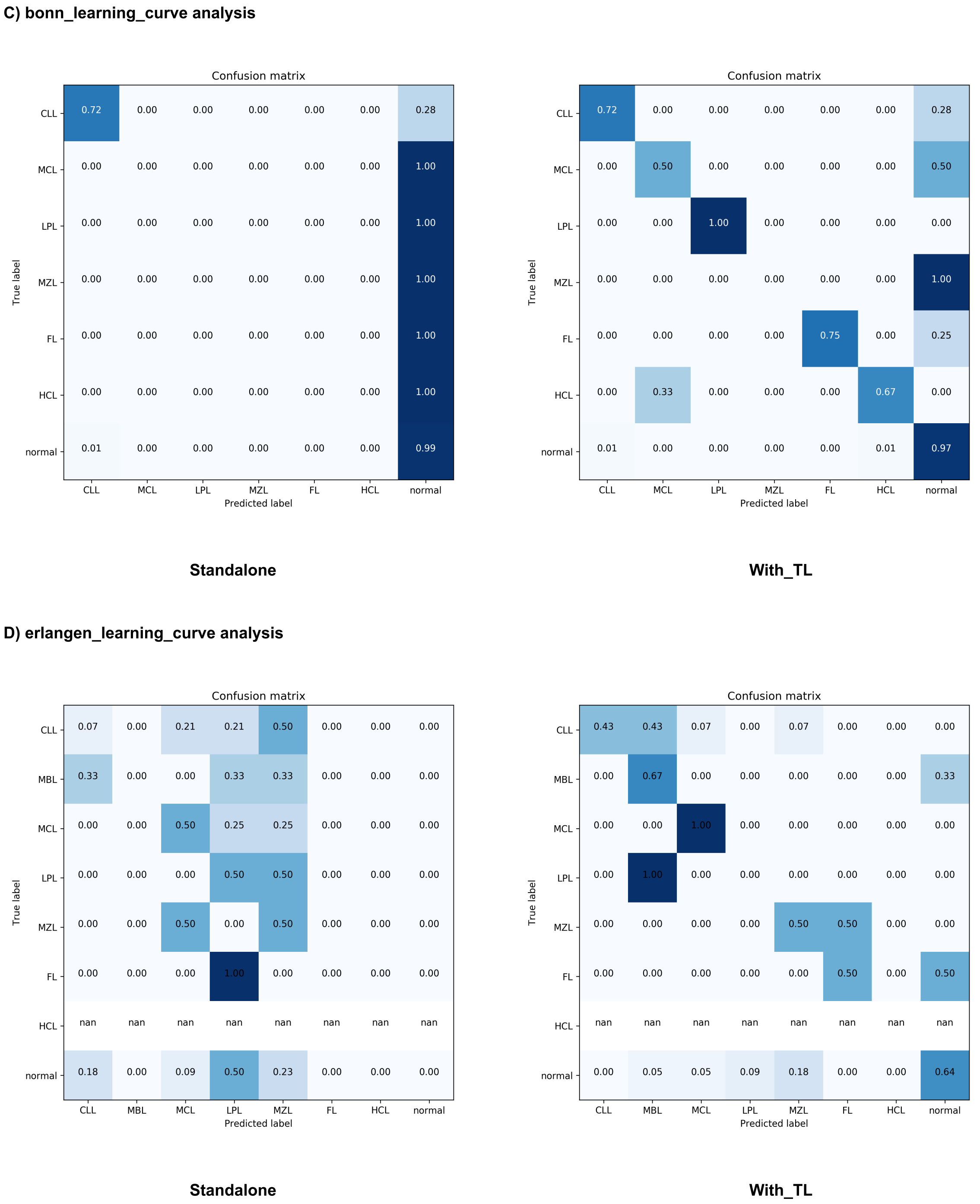
B) berlin_learning_curve analysis



Standalone



With_TL



C) bonn_learning_curve analysis

Standalone

With_TL

D) erlangen_learning_curve analysis

Standalone

With_TL

Figure 27: Confusion matrices for standalone versus transfer learning. The confusion matrices shown here are a snapshot of the classification performance with the least number of training samples on the learning curve. A) shows the difference between standalone models and models with transfer learning for the transition experiment with five training samples per class. The confusion matrices for the cross-laboratory adaption with Berlin, Bonn, and Erlangen panels are shown in B), C), and D), respectively.

3.3 Additional analysis

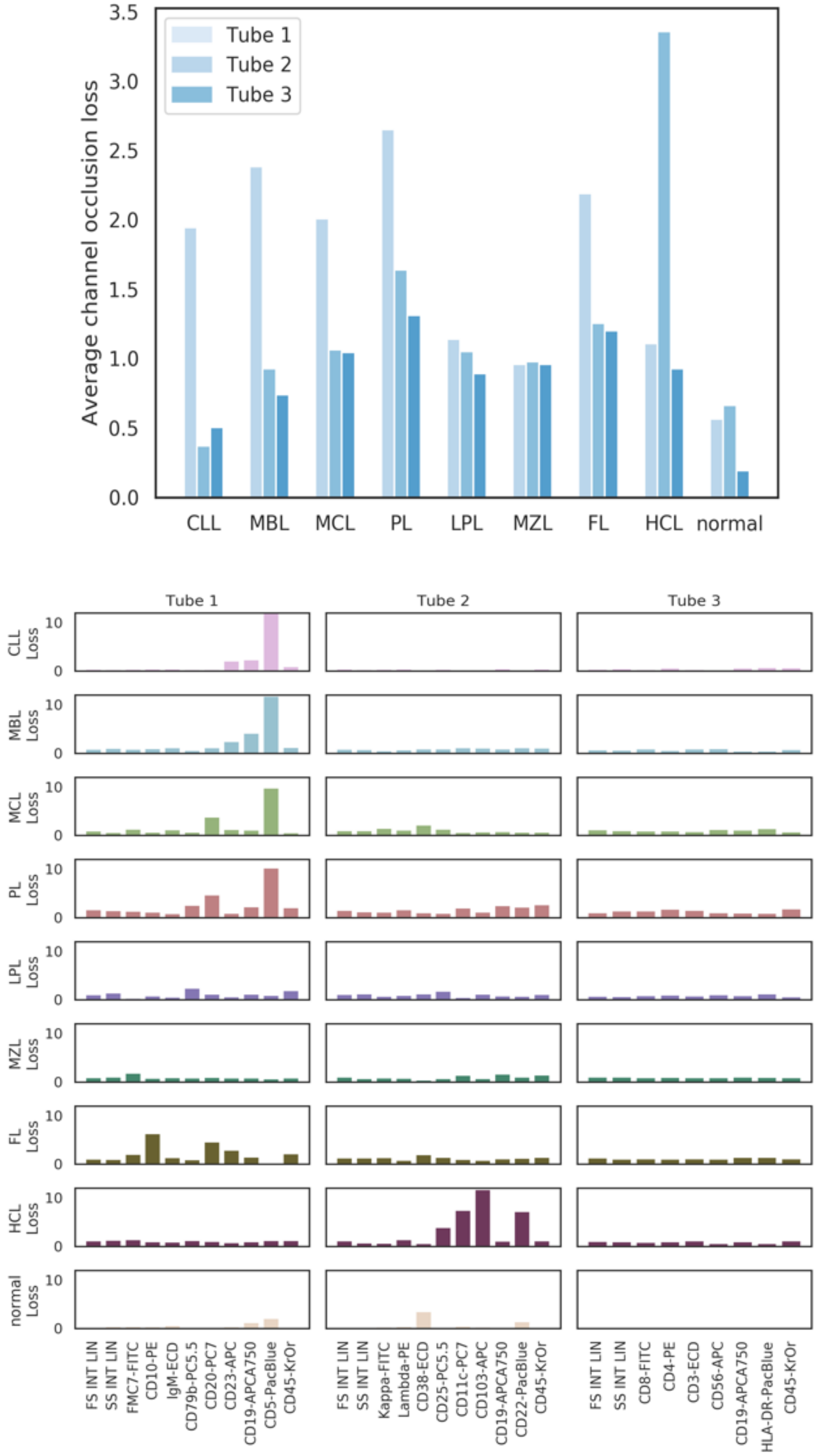
In addition to evaluating the model performance and class sensitivity, occlusion analysis and threshold estimation were performed to provide relevant insight that could help the diagnostic process. All additional analyses discussed here were done using the unmerged AI model described in phase 1 of the previous chapter. The unmerged MLL9F panel with the training and 10% validation split described in “Methods” was used. This section describes the additional analyses and the results.

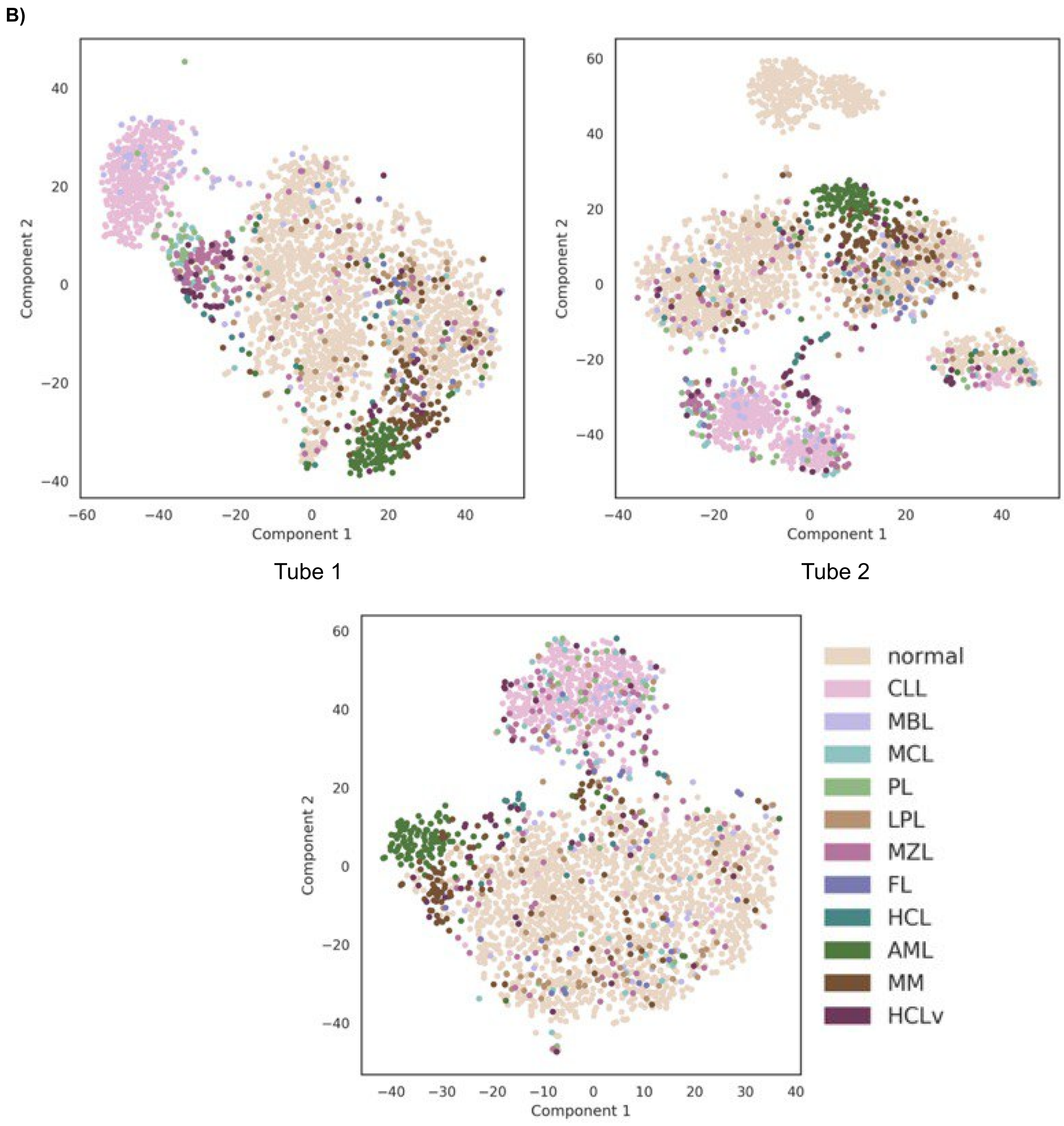
3.3.1 Marker importance

Occlusion analysis systematically determines the important features for classification by eliminating one feature at a time from the input. The approach has been previously described in the model analysis of image classifiers (Zeiler and Fergus, 2013). While occlusion analysis is another method to understand and visualize the classification process, it could also be used to gain vital insights relevant to the data domain. We implemented occlusion analysis for our unmerged AI model described in phase 1 of the previous chapter to determine the essential markers that can be used to inform panel design.

The importance of individual FCS markers for prediction accuracy in the trained model was measured by zeroing all values in the respective marker channel in the input SOM. More important information in the original input data will decrease prediction accuracy more strongly and thus increase the measured loss. Important markers for each class were calculated using average per class cross-entropies for all markers. Predictions were generated for samples in the validation set after setting all values for one marker channel or an entire tube to zero. For each occluded marker or tube, categorical cross entropy (loss) was calculated between the obtained prediction after occlusion and the ground truth. Losses were calculated for all samples in the validation split and grouped by diagnosis. Figure 28 shows the occlusion plots for the unmerged MLL9F dataset.

A)


Average channel occlusion loss
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Tube 1
Tube 2
Tube 3
CLL
MBL
MCL
PL
LPL
MZL
FL
HCL
normal
Tube 1
Tube 2
Tube 3
CLL Loss
MBL Loss
MCL Loss
PL Loss
LPL Loss
MZL Loss
FL Loss
HCL Loss
normal Loss
10
0
FS INT LIN
SS INT LIN
FMC7-FITC
CD10-PE
IgM-ECD
CD79b-PC5.5
CD20-PC7
CD23-APC
CD19-APCA750
CD5-PacBlue
CD45-KrOr
FS INT LIN
SS INT LIN
Kappa-FITC
Lambda-PE
CD38-ECD
CD25-PC5.5
CD11c-PC7
CD103-APC
CD19-APCA750
CD22-PacBlue
CD45-KrOr
FS INT LIN
SS INT LIN
CD8-FITC
CD4-PE
CD3-ECD
CD56-APC
CD19-APCA750
HLA-DR-PacBlue
CD45-KrOr



Tube 1

Tube 2

Tube 3

Figure 28: Occlusion analysis. A) shows averaged loss values over each tube for each diagnosis and the averaged losses for each marker channel. Occluding tube 1 shows a higher loss for CLL, MBL, MCL, PL, and FL, which can be explained by a high loss for CD5 in tube 1 for the CD5+ subtypes such as in CLL, MBL, MCL, and PL. FL shows higher loss values for CD10 and CD20 and no loss for CD5. HCL has high losses in CD103, CD11c and CD22 contained in tube 2. B) The t-SNE plots were generated for single tubes while occluding the other two. Here, the embeddings show a loss of distinguishable clusters for CD5+ and other populations in tubes 2 and 3.

The occlusion analysis shows the importance of tube 1 followed by tube 2 for the

MLL9F dataset. Further, critical markers such as CD5 and CD103 are highlighted in the analysis. Such an occlusion analysis from a large trained model can be used to determine the important markers for any given set of B-cell neoplasm subtypes. This information could play a vital role in choosing and designing the FCS panel at a new diagnostic center intending to use AI models for automated classification.

3.3.2 Misclassification analysis and threshold estimation

The misclassified samples - healthy samples wrongly classified as B-cell neoplasm (pathological) and vice versa - from the CNN's predictions were identified and further analyzed. The predicted labels were compared with the ground truth label for each sample in the validation set to identify the wrongly classified samples. Most misclassified samples were shown to have a lower probability score making it possible to eliminate these wrong predictions by using a threshold for the predicted probability score. However, a few samples were misclassified with a high probability score. These samples were further analyzed and found to have a very low infiltration rate - the percentage of pathological cells - of less than 1%. The low number of pathological cells in a given sample is insufficient for the CNN to learn the respective representation, thereby making it likely for the sample to be misclassified.

In order to flag samples likely to be misclassified, the trustworthiness of the classifier was estimated by using confidence threshold scores that were computed using the classification score. The confidence threshold scores are not just a cutoff defined on the predicted probabilities; Rather, the thresholds are computed using the predicted probability score to ensure a given accuracy for the model. Figure 29 shows the various threshold scores for a minimum accuracy of 85%. At each threshold, the ratio of cases with at least an 85% predicted probability score that will be included in the prediction result is shown. From the range of computed threshold scores, an appropriate score can be chosen to group samples with lower accuracy than the one indicated for the chosen threshold into an “uncertain” class.

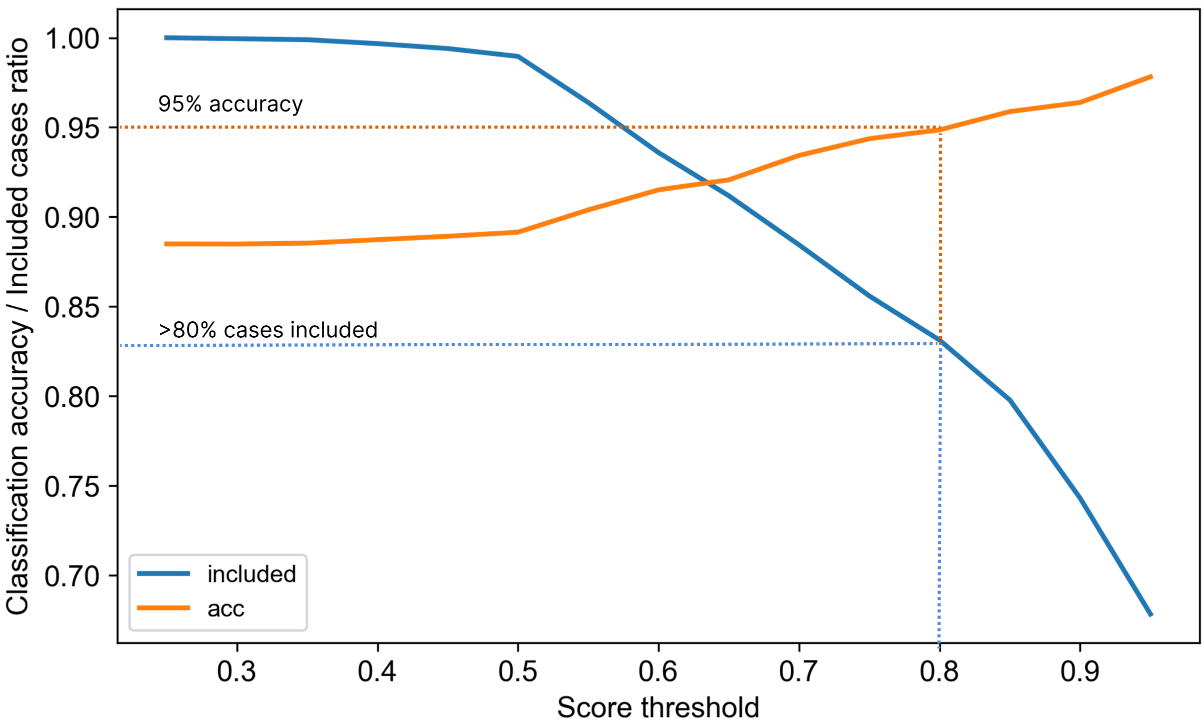


Figure 29: Confidence Threshold. The plot shows the various score thresholds used for inclusion against the given accuracy on a subset of predictions. The threshold scores were computed to obtain a minimum of 85% classification accuracy to separate high-certainty from low-certainty predictions. In addition, the plot also shows the percentage of cases that will be included for each threshold. For instance, a score threshold of 0.8 yields a 95% prediction accuracy for over 80% of cases.

Misclassification analysis and the confidence score threshold can be used in a routine diagnostic setting to avoid prediction errors and identify samples that may have untypical behavior. Any sample marked as “uncertain” can be flagged for manual analysis. Experts can further investigate such samples with additional tests and analysis as required to confirm the diagnosis. The predicted class and saliency maps of these “uncertain” samples may provide valuable insights that can guide the manual analysis.

4. Discussion

Hematological malignancies are increasingly being diagnosed using artificial intelligence models (Radakovich et al., 2020; Shouval et al., 2021). Several artificial intelligence models have been developed to diagnose various hematological diseases using high-throughput data such as FCS. Our SOM-CNN-based classification model can distinguish eight subtypes of B-cell neoplasms and normal controls with high precision. Our AI operates directly on compensated FCS data without the need for prior gating or manual data cleansing. The AI might therefore contribute to a speedier diagnosis process for certain samples that are thought to be easy to classify. Saliency maps may also provide a second opinion for complex cases. The AI may alert the clinician of lymphoma by identifying cell clusters and suggesting which type of lymphoma this pattern might fit. This might be an additional alert that could improve the sensitivity in detecting lymphoma before the clinician's final decision.

The hierarchical clustering of the confusion matrix and the t-SNE plots for the multiclass problem suggest that the AI learned a representation of the FCS data that reflects our knowledge about the subtypes' relatedness. We can think of the t-SNE embedding as a lower-dimensional cluster representation of the FCS data that preserves the properties we are interested in. As a result, clinically similar cases should be located close together. Indeed, samples with LPL/MZL and PL/MCL are part of the same cluster in the t-SNE plots, which concurs with the literature (Bassarova et al., 2015; Van der Velden et al., 2014).

Using the trained CNN embedding to represent subtypes works well if all the relevant markers from all the tubes are available. For instance, the trained embedding produces the best clustering of all the eight subtypes when all the relevant markers from all the tubes are used for training. If we limit the training and thus clustering to SOMs of single tubes in the unmerged data, the clear separation between CD5 positive and negative subtypes is lost, as can be seen in tSNE plots of tubes 2 and 3, where the CD5 marker is missing

(Figure 28B).

This brings us to the question of how machine learning can optimize marker selection in a diagnostic setting where the number of channels is limited. Our classifier's occlusion analysis (Figure 28) shows that marker CD5 plays a crucial role in identifying CLL, MBL, MCL, and PL. It is also possible to identify more subtype-specific markers, such as CD103 for HCL and CD10 and CD20 for FL (Figure 28A). Therefore, we may envision a set of hematologic ailments being screened with a broad panel of markers, then selecting the most appropriate markers for detecting it in a large-scale setting through occlusion analysis.

Whilst our initial AI achieves expert-level accuracy for classification, the model was trained on a single large dataset acquired with the same FCS protocol. The FCS protocol is not uniform between laboratories or the same laboratory over time, leading to changes in the data. Thus, a model trained on a specific FCS protocol cannot be applied to a dataset with a different protocol. In order to be successfully integrated into a routine diagnostic setting, the AI needs to adapt to multiple and smaller datasets. To this end, we developed a workflow with transfer learning to extend AI models trained on a specific FCS panel to multiple FCS panels and data sizes. Our workflow allows an existing model to adapt quickly to any changes in the data making it possible to be deployed in a routine diagnostic setting across different laboratories.

The knowledge from the base model trained on a single FCS panel is used to train target models for new FCS data. The extended workflow described in phase two applies TL to improve the performance and adaptability of AI to multiple datasets. Ideally, TL is applied in cases where the base and the target tasks are related yet different, whereas the datasets do not change in terms of composition. Our work shows that TL can be used successfully even when the base and target datasets change.

Our transfer learning workflow combines knowledge transfer with FCS data merging (Figure 14). Merging multiple aliquots is a known approach for increasing computational depth for deep phenotyping and FCS analysis (Robinson et al., 1991). In the context of a CNN, it increases the network's feature space by combining markers measured in different tubes. It also allows us to maximize our networks’ transferability, which is essential for a successful knowledge transfer.

The initial base AI model is extended to four additional datasets with a varying number of tubes per sample and markers with no changes to the model architecture and training parameters. Here, we show that knowledge transfer in conjunction with FCS data merging enhances the overall performance of target models by allowing already learned features from a large dataset to be transferred to smaller and different datasets.

With the TL workflow, the target models achieve an overall performance close to the previously reported expert-level accuracy of the base model. For the Berlin panel, the TL model achieved a median weighted F1 score of 0.94, the same as expert-level performance (Figure 24). This enhancement could only be achieved by combining FCS data merging with TL. While TL allows for features already learned to be transferred between models to enhance the overall performance of target models, merging multiple FCS tubes makes it possible to apply maximum TL between different FCS datasets.

Furthermore, the learning rate of target models with TL is much higher than the standalone models, as demonstrated by our learning curve analysis. The TL models achieve significantly higher performance for very small training sizes. In the context of a transition to a new cytometer, this would allow an already deployed AI model to be quickly adapted to the new protocol without having to wait for a considerable time for enough samples to become available for the new protocol.

Although the proposed workflow successfully allows the AI model to be adapted to different FCS data, it does not entirely address the inherent differences between various datasets. Each laboratory has a different diagnostic goal and expertise, leading to different panel designs and different data distributions among the classes for each dataset. The class imbalance within a given dataset can be accounted for in the CNN using appropriate class weights during training. However, these class weights are not transferrable; thus, the non-uniform imbalance between the various datasets cannot be addressed within the CNN. Advanced data augmentation strategies to artificially create more samples for the rare classes could allow for a uniform data distribution among the datasets. Future works should thus focus on various data augmentation strategies that can further improve classification performance by creating realistic samples for training.

The choice of marker combinations used for each FCS panel depends on the diagnostic workflow and preferences of the laboratories. While some markers are standard markers for B-cell neoplasm assessment, others are specific to certain subtypes, and different laboratories may use alternate markers for such cases. The differences in the marker combinations between the panels are addressed using NN merge and SOM training in our workflow. The overlapping CD markers between the base and target FCS panels are accounted for in the SOM calculation by reordering markers in the target panel to match the order in the base dataset. However, the missing markers in the target datasets are handled by setting these values to zero in SOM weight calculation. These markers may be necessary for specific subtype identification in base data and could impact the classification of these subtypes in the target models. For instance, IgM, a marker that Munich chose to improve LPL (lymphoplasmacytic lymphoma) classification in the MLL9F panel, is missing in the Bonn and Erlangen panels. We set the value of IgM to zero in Bonn and Erlangen panels, causing this information to be lost during the transfer. Although these panels use other known markers, such as CD38, for LPL identification, the information contained in IgM cannot be transferred easily to CD38. It can thus impact the classification performance for this class. This loss of information might also explain the decline in performance for LPL, which can be seen in confusion matrices for the Erlangen panel (Figure 27D). Despite these inherent biases that can confound the classification performance, we see an overall performance enhancement for all four target sets with the proposed workflow.

Even though TL helps adapt and improve model performance, the result must be carefully evaluated for each case. Especially, evaluation on small and highly imbalanced datasets often encountered in the routine laboratory setting can cause misleading results without a thorough assessment of different performance aspects.

In conclusion, our work is the first application of AI for the assessment of clinical flow cytometry data. However, it is just one additional piece in a long series of publications that showed how AI could increase sensitivity and specificity in health care (Topol, 2019). Further, we provide a workflow to extend deep learning models to multiple FCS panels and achieve high accuracy for multi-label classification across datasets. Here, we address some of the previous challenges for automated flow cytometry classification by allowing models to be trained with smaller training sizes and generalizing models to work with multiple FCS panels. Our transfer learning workflow is a step toward making deep learning models robust so that AI for diagnostic FCS can move from the “proof of concept” stage into routine diagnostics.

4.1 Limitations of the study

Herein, we address the limitation of this study in terms of known shortcomings of the merging approach, technical variance between datasets, and potential improvements. Although NN is a well-known method for data imputation, in the case of imputing markers for FCS events, NN merging is known to sometimes introduce a spurious combination of markers into the imputation results (Lee et al., 2011). However, this did not lead to a reduced performance of our classification model. Both merged and unmerged models produced nearly identical F1 scores for the base dataset. Furthermore, we also looked at the impact of the number of shared markers on the imputation quality and did not find any differences (Table 10). While TL accounts for some of the variability between the datasets, the technical variation arising from sample preparation and equipment calibration cannot be completely ruled out and could potentially affect the classification performance. A standardized normalization approach across datasets could improve the classification performance further. Although, this would add considerable computational overhead and may require a reference sample to be analyzed across various locations that can be used to remove all the technical variation. The other limitation of this study is that we align FCS channels between multiple datasets by matching CD markers while ignoring the fluorochromes for our knowledge transfer. While any missing markers are handled within the updated SOM training, the current workflow will ignore new markers. The information lost because of the marker alignment and ignoring new markers could impact the classification of specific subtypes and, thus, the overall performance. The performance may be improved further with partial knowledge transfer techniques, where features from existing channels are transferred while the model is trained to learn the new channels present in the new protocol (Hassan, 2019). Finally, all five datasets used in this study are from Navios cytometers. Although the workflow presented here is not limited to datasets acquired on a specific device, our models could have a potential vendor bias that should be considered when data are acquired on a device from a different vendor.

5. Abstract

B-cell neoplasms are the most prevalent type of non-Hodgkin lymphoma, including a diverse and heterogenous group of entities. Immunophenotyping with a high-throughput technology like flow cytometry is a standard diagnostic procedure in evaluating B-cell neoplasms. While multi-parameter flow cytometry (FCS) has become a cornerstone in clinical decision-making for leukemia and lymphoma, the data analysis requires manual gating of cell populations, which is time-consuming, subjective, and often limited to a two-dimensional space. In recent years, machine learning has become a popular approach for automating manual gating. Many automated gating algorithms require expert supervision and cannot classify the data into diagnosis labels. Furthermore, these algorithms still limit the analysis to a two-dimensional space, leading to the loss of information in the high-dimensional FCS data.

We hypothesize that the wealth of information captured in “n”-dimensional FCS data can be analyzed by current computer vision methods when represented as image data. We, therefore, transformed FCS raw data into a multicolor low-resolution image using self-organizing maps. These images are then analyzed and classified using a convolutional neural network. By this means, we built an artificial intelligence (AI) that not only can distinguish diseased from healthy samples but also differentiate seven subtypes of mature B-cell neoplasm. We trained our model with 18,274 cases, including chronic lymphocytic leukemia and its precursor monoclonal B-cell lymphocytosis, marginal zone lymphoma, mantle cell lymphoma, prolymphocytic leukemia, follicular lymphoma, hairy cell leukemia, lymphoplasmacytic lymphoma and achieved a weighted F1 score of 0.94 on a separate test set of 2,348 cases.

Next, we extend our AI model to multiple datasets and FCS panels using transfer learning in conjunction with FCS data merging. We demonstrate how transfer learning can be applied to boost the performance of models with much smaller datasets acquired with different FCS panels. We trained a new AI for four additional datasets by transferring

the features learned from our base model. Our workflow increased the model's overall performance and, more prominently, improved the learning rate for small training sizes.

6. Supplementary Information

Table 12: Markers and their function

CD marker	Function
CD3	A complex of subunits that mediates T-cell signal transduction
CD4	Initiates or augments the early phase of T-cell activation
CD5	Acts as a negative regulator of T-cell receptor signaling
CD8	May play an important role in T-cell mediated killing
CD10	Neutral endopeptidase that cleaves peptides and inactivates several peptide hormones
CD11c	Important for cell-cell interaction during inflammatory responses
CD19	Assembles with the antigen receptor of B lymphocytes to decrease the threshold for antigen receptor-dependent stimulation
CD20	Development and differentiation of B-cells into plasma cells
CD22	Mediates B-cell B-cell interactions. May be involved in the localization of B cells in lymphoid tissues. Modulates B-cell signaling
CD23	Key molecule for B-cell activation and growth. This receptor has essential roles in the regulation of IgE production and in the differentiation of B cells
CD25	Receptor for interleukin-2
CD38	Cell adhesion and signal transduction
CD43	Cell adhesion and T-cell activation
CD45	Leukocyte common antigen; Regulator of T- and B-cell antigen receptor signaling; regulator of cell growth and differentiation
CD56	Cell adhesion and neural plasticity
CD103	Promoting entry and retention of antigen specific CD8 effector molecules in epithelial compartments
CD200	Co-stimulates T-cell proliferation. May regulate myeloid cell activity
Kappa	Plays an important role in several immune responses
Lambda	Plays an important role in several immune responses
IgM	Primary immune response
FMC7	pan “B-cell” antigen; associated with late B cells that have features of activation

List of all the markers used in all five FCS panels and their functionality.

Table 13: List of fluorochromes

	Full name	Excitation wavelength (nm)	Emission wavelength (nm)
FITC	Fluorescein isothiocyanate	495	519
PE	R-phycoerythrin	565	578
ECD	R-phycoerythrin-Texas Red-X	565	613
PC5.5	Peridinin chlorophyll protein-Cyanine5.5	482	690
PC7	R Phycoerythrin Cyanin 7	565	770
APC	Allophycocyanin	650	660
APC750	Allophycocyanin - Alexa Fluor 750	749	775
PB	Pacific Blue	410	455
KrOr	Krome Orange	398	528
AA700	Alexa Fluor 700	702	723

List of fluorochromes and their excitation and emission peak wavelength.

7. List of figures

8. List of tables

9. References

Data file standard for flow cytometry. Cytometry 1990; 11(3): 323–332. doi:https://doi.org/10.1002/cyto.990110303

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray D G, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, and Zheng X. Tensorflow: A system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. USA: USENIX Association, OSDI'16. ISBN 9781931971331, 2016, 265–283

Abdelaal T, Höllt T, Unen V V, Lelieveldt B P, Koning F, Reinders M J, and Mahfouz A. CyTOFmerge: Integrating mass cytometry data across multiple panels. Bioinformatics 2019; 35. ISSN 14602059. doi:10.1093/bioinformatics/btz180

Abdelaziz Ismael S A, Mohammed A, and Hefny H. An enhanced deep learning approach for brain cancer MRI images classification using residual networks. Artificial Intelligence in Medicine 2020; 102: 101779. ISSN 0933-3657. doi:https://doi.org/10.1016/j.artmed.2019.101779

Aghaeepour N, Finak G, Consortium T, Consortium T, Hoos H, Mosmann T, Brinkman R, Gottardo R, and Scheuermann R. Critical assessment of automated flow cytometry data analysis techniques. Nat Methods 2013; 10: 228–238

Armitage J and Weisenburger D. New approach to classifying Non-Hodgkin's Lymphomas: clinical features of the major histologic subtypes. Non-Hodgkin's Lymphoma classification project. Journal of clinical oncology : official journal of the American Society of Clinical Oncology 1998; 16(8): 2780–2795. doi:10.1200/JCO.1998.16.8.2780

Bassarova A, Tøen G, Spetalen S, Micci F, Tierens A, and Delabie J. Lymphoplasmacytic lymphoma and marginal zone lymphoma in the bone marrow: Paratrabecular involvement as an important distinguishing feature. Am J Clin Pathol 2015; 143: 797–806

Belov L, Vega O, Remedios C, Mulligan S, and Christopherson R. Immunophenotyping of leukemias using a cluster of differentiation antibody microarray. Cancer research 2001; 61(11): 4483–4489

Bendall S and Nolan G. From single cells to deep phenotypes in cancer. Nat Biotechnol 2012; 30

Chollet F et al., 2015: Keras. https://keras.io

Costa E S, Pedreira C E, Barrena S, Lecrevisse Q, Flores J, Quijano S, Almeida J, García-MacIas M D C, Bottcher S, Dongen J J V, and Orfao A. Automated pattern-guided principal component analysis vs expert-based immunophenotypic classification of b-cell chronic lymphoproliferative disorders: A step forward in the standardization of clinical immunophenotyping. Leukemia 2010; 24. ISSN 14765551. doi:10.1038/leu.2010.160

Craig F and Foon K. Flow cytometric immunophenotyping for hematologic neoplasms. Blood 2008; 111

Dean P N, Bagwell C B, Lindmo T, Murphy R F, and Salzman G C. Introduction to flow cytometry data file standard. Cytometry 1990; 11(3): 321–322. doi:https://doi.org/10.1002/cyto.990110302

Fort J C, Cottrell M, and Letremy P. Stochastic on-line algorithm versus batch algorithm for quantization and self organizing maps. Neural Networks for Signal Processing - Proceedings of the IEEE Workshop 2001; doi:10.1109/nnsp.2001.943109

Gedye C, Hussain A, Paterson J, Smrke A, Saini H, and Sirskyj D. Cell surface profiling using high-throughput flow cytometry: A platform for biomarker discovery and analysis of cellular heterogeneity. PLoS ONE 2014; 9(8): 105602. doi:10.1371/journal.pone.0105602

Gorman C, 2019: tensorflow-som. https://github.com/cgorman/tensorflow-som.git

Greenspan H, Ginneken B, and Summers R. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. IEEE Trans Med Imaging 2016; 35: 1153–1159

Gurovich Y, Hanani Y, Bar O, Nadav G, Fleischer N, Gelbman D, Basel-Salmon L, Krawitz P, Kamphausen S, and Zenker M. Identifying facial phenotypes of genetic disorders using deep learning. Nat Med 2019; 25: 60–64

Hassan A, 2019: Transfer learning from RGB to multi-band imagery. https://www.azavea.com/blog/2019/08/30/transfer-learning-from-rgb-to-multi-band-imagery/.

Henel G and Schmitz J. Basic theory and clinical applications of flow cytometry. Lab Med 2007; 38

Kingma D P and Ba J. Adam: A method for stochastic optimization. In: Bengio Y and LeCun Y, eds., 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings. San Diego - CA, USA: arXiv, 2015

Kiviluoto K. Topology preservation in self-organizing maps. IEEE International Conference on Neural Networks - Conference Proceedings 1996; 1. doi:10.1016/b978-044450270-4/50022-x

Kohonen T. The self-organizing map. Proceedings of the IEEE 1990; 78(9): 1464–1480. doi:10.1109/5.58325

Kotikalapudi R and contributors: 2017, keras-vis. https://github.com/raghakot/keras-vis

Kullback S and Leibler R A. On information and sufficiency. The Annals of Mathematical Statistics 1951; 22(1): 79 – 86. doi:10.1214/aoms/1177729694

LeCun Y, Bengio Y, and Hinton G. Deep learning. Nature 2015; 521: 436–444. doi:10.1038/nature14539

Lee G, Finn W, and Scott C. Statistical file matching of flow cytometry data. J Biomed Inform 2011; 44: 663–676

Mallesh N, Zhao M, Meintker L, Höllein A, Elsner F, Lüling H, Haferlach T, Kern W, Westermann J, Brossart P, Krause S W, and Krawitz P M. Knowledge transfer to enhance the performance of deep learning models for automated classification of B cell neoplasms. Patterns (N Y) 2021; 2(10): 100351

Matek C, Schwarz S, Spiekermann K, and Marr C. Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks. Nat Mach Intell 2019; 1

Meek C, Thiesson B, Heckerman D, and Kaelbling P. The learning-curve sampling method applied to model-based clustering. Journal of Machine Learning Research 2002; 2: 397–418

Mount N J and Weaver D. Self-organizing maps and boundary effects: Quantifying the benefits of torus wrapping for mapping SOM trajectories. Pattern Analysis and Applications 2011; 14. ISSN 14337541. doi:10.1007/s10044-011-0210-5

Naghshvar M, Javidi T, and Wigger M. Extrinsic Jensen–Shannon Divergence: Applications to variable-length coding. IEEE Transactions on Information Theory 2015; 61(4): 2148–2164. doi:10.1109/tit.2015.2401004

O'Neill K, Aghaeepour N, Parker J, Hogge D, Karsan A, Dalal B, and Brinkman R R. Deep profiling of multitube flow cytometry data. Bioinformatics 2015; 31. ISSN 14602059. doi:10.1093/bioinformatics/btv008

O'Neill K, Aghaeepour N, Špidlen J, and Brinkman R. Flow cytometry bioinformatics. PLoS Comput Biol 2013; 9

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, and Duchesnay E. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 2011; 12: 2825–2830

Pedreira C E, Costa E S, Barrena S, Lecrevisse Q, Almeida J, Dongen J J V, and Orfao A. Generation of flow cytometry data files with a potentially infinite number of dimensions. Cytometry Part A 2008; 73. ISSN 15524922. doi:10.1002/cyto.a.20608

Perry A, Jacques D, Nathwani B, Maclennan K, Müller-Hermelink H, Boilesen E, Bast M, Armitage J, and Weisenburger D. Classification of Non-Hodgkin Lymphoma in seven geographic regions around the world: Review of 4539 cases from the international Non-Hodgkin Lymphoma classification project. Blood 2015; 126(23): 1484. doi: 10.1182/blood.V126.23.1484.1484

Radakovich N, Nagy M, and Nazha A. Artificial intelligence in hematology: current challenges and opportunities. Curr Hematol Malig Rep 2020; 15: 203–210

Robinson J, Durack G, and Kelley S. An innovation in flow cytometry data collection and analysis producing a correlated multiple sample analysis in a single file. Cytometry 1991; 12: 82–90

Samad T and Harp S A. Self-organization with partial data. Network: Computation in Neural Systems 1992; 3. ISSN 0954898X. doi:10.1088/0954-898X_3_2_008

Shapiro H M. Practical flow cytometry. John Wiley & Sons, 2005

Shen D, Wu G, and Suk H I. Deep learning in medical image analysis. Annu Rev Biomed Eng 2017; 19: 221–248

Shouval R, Fein J, Savani B, Mohty M, and Nagler A. Machine learning and artificial intelligence in haematology. Br J Haematol 2021; 192: 239–250

Simonyan K, Vedaldi A, and Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In: Bengio Y and LeCun Y, eds., 2nd International Conference on Learning Representations, ICLR 2014, Workshop Track Proceedings. Banff - AB, Canada: arXiv, 2014

Swerdlow S, Campo E, Harris N, Jaffe E, Pileri S, Stein H, Thiele J, Arber D, Hasserjian R, Le Beau M, Orazi A, and Siebert R. WHO classification of tumours of haematopoietic and lymphoid tissues. Lyon: International Agency for Research on Cancer, 2017

Swerdlow S H, Campo E, Pileri S A, Harris N L, Stein H, Siebert R, Advani R, Ghielmini M, Salles G A, Zelenetz A D, and Jaffe E S. The 2016 revision of the World Health

Organization classification of lymphoid neoplasms. Blood 2016; 127. ISSN 15280020. doi:10.1182/blood-2016-01-643569

Taheri Gorji H and Kaabouch N. A deep learning approach for diagnosis of mild cognitive impairment based on MRI images. Brain Sciences 2019; 9(9). ISSN 2076-3425. doi: 10.3390/brainsci9090217

Topol E. High-performance medicine: The convergence of human and artificial intelligence. Nat Med 2019; 25: 44–56

Torrey L and Shavlik J. Transfer learning. In: Soria E, Martin J, Magdalena R, Martinez M, and Serrano A, eds., Handbook of Research on Machine Learning Applications. IGI Global, 2009

Van der Velden V, Hoogeveen P, Ridder D, Schindler-van der Struijk M, Zelm M, M S, Karsch D, Beverlo H, Lam K, Orfao A, Lugtenburg P J, Böttcher S, Van Dongen J J, Langerak A, Kappers-Klunne M, and van Lom K. B-cell prolymphocytic leukemia: A specific subgroup of mantle cell lymphoma. Blood 2014; 124: 412–419

Van Dongen J, Lhermitte L, Böttcher S, Almeida J, Van Der Velden V H, Flores-Montero J, Rawstron A, Asnafi V, Lécrevisse Q, Lucio P, Mejstrikova E, Szczepaski T, Kalina T, De Tute R, Brüggemann M, Sedek L, Cullen M, Langerak A W, ca A M, MacIntyre E, Martin-Ayuso M, Hrusak O, Vidriales M B, and Orfao A. EuroFlow antibody panels for standardized n-dimensional flow cytometric immunophenotyping of normal, reactive and malignant leukocytes. Leukemia 2012; 26. ISSN 14765551. doi:10.1038/leu.2012.120

Van Gassen S, Callebaut B, Van Helden M, Lambrecht B, Demeester P, Dhaene T, and Saeys Y. FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry Part A : the journal of the International Society for Analytical Cytology 2015; 87(7): 636–645. doi:10.1002/cyto.a.22625

Van Rossum G and Drake F L. Python 3 Reference Manual. Scotts Valley - CA: CreateSpace, 2009. ISBN 1441412697

Weber L and Robinson M. Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. Cytometry Part A 2016; 89A:1084-1096

Weiss K, Khoshgoftaar T M, and Wang D D. A survey of transfer learning. Journal of Big Data 2016; 3. ISSN 21961115. doi:10.1186/s40537-016-0043-6

Zeiler M D and Fergus R. Visualizing and understanding convolutional networks. CoRR 2013; abs/1311.2901

Zhao M, Mallesh N, Höllein A, Schabath R, Haferlach C, Haferlach T, Elsner F, Lüling H, Krawitz P, and Kern W. Hematologist-level classification of mature B-Cell neoplasm using deep learning on multiparameter flow cytometry data. Cytometry Part A 2020; 97(10): 1073–1080. doi:https://doi.org/10.1002/cyto.a.24159

10. Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Dr. Peter Krawitz, for his invaluable supervision, support, and guidance during the course of my Ph.D. degree. His scientific inputs and suggestions were indispensable to the research and writing of this dissertation.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Peter Brossart, Prof. Matthias Schmid, and Prof. Wolfgang Kern, for their insightful comments and encouragement, as well as for prompting me to conduct a more extensive investigation.

My sincere thanks also go to Prof. Dr. Stefan Krause for providing valuable insights and feedback. His expertise and counsel were very helpful in completing the project.

I would also like to acknowledge all my colleagues at IGSB and my friends, who have always been eager to discuss new ideas and share feedback. Thank you for our exciting conversations and fun over the last four years.

Lastly, I would be remiss in not mentioning my family, especially my parents. Their belief in me has kept my spirits and motivation high during this process.