MSc DISSERTATION ASSESSMENT COVERSHEET

1ST (PROJECT SUPERVISOR) MARKER

Name of student:	Manikanta varma. Rudraraju
MSc Programme:	Bioinformatics (Msc)
Title of Dissertation:	Application of Supervised Machine Learning in Mass Cytometry
Year of Submission:	2022
Project Supervisor:	**William Alazawi & Hajar Saihi**
Declaration:	**'This research dissertation is submitted for the MSc in (insert title) at Queen Mary, University of London'**
Data sharing (delete appropriately)	I **do / ~~don't~~** allow consent for my project to be shared with future cohorts of students on MSc programmes in the School of Biological and Chemical Sciences and on institutional repositories and websites.

To be completed by Supervisor:

Second Marker Name	
Turnitin score if higher than 17%	

Section A. Continuous Assessment Mark (For supervisor only to assess the student's performance during the practical stages of the project. Note this scheme is inclusive of project type, for example, including field-based, laboratory, modelling and meta-analytic studies). Please highlight the words or phrases that are appropriate to justify the grade.

Mark (%)	Criteria
95 – 100%	Outstanding technical capacity, went beyond expectation in developing protocols, or analytical tools. The engagement was outstanding contributing to lab meetings, and the life of the hosting research group.
85 – 94 %	Exceptional performance showing outstanding technical ability, originality and initiative, high levels of commitment and application, ability to plan and organise the research programme and to contribute substantially to the development of the work.
70 – 84 %	Excellent performance, showing clear evidence of originality, initiative and ability to contribute to the development of the programme.
60 – 69 %	Good performance, notable for steady commitment, sound technical ability and some evidence of initiative and originality. Some contribution to the development of the project but mainly following the advisor's suggestions.
50 – 59 %	Performance generally satisfactory but with some deficiencies in technical ability and/or limited levels of commitment and application to the project. Little or no contribution to the development of the work and very little if any initiative and originality.
40 – 49 %	Performance somewhat weak, characterised by poor technical ability and low levels of commitment and application. Poor understanding of the project and effectively no contribution to the planning and organisation of the work.
20 – 39 %	Unsatisfactory performance. Very poor technical ability amounting to inability to perform routine tasks reliably. Very low levels of commitment and application including unacceptably low attendance.
0 – 19 %	Very poor performance! Careless and totally disorganised, and non-existent technical skills. Unacceptably low attendance.

Please comment in the box below on the student's coursework below, focusing on the student's performance productivity, attitude, commitment, timeliness, originality, organization, technical ability.

Section A. Supervisor's Mark for continuous assessment =_____%

Please complete the dissertation marking (to follow)

Section B (1st marker). Assessment of dissertation report

Please highlight the points that best describe the project

Mark (%)	Criteria
95 - 100%	Outstanding performance in producing a document which could be submitted as it for publication from a technical, analytical and editorial perspective.
85 – 94%	Exceptional project report showing very broad understanding of the project area and outstanding knowledge of the relevant literature. Exceptional presentation and analysis of results, logical organisation and ability to evaluate critically and discuss results with insight and originality.
70 – 84 %	An excellent project report showing evidence of wide reading, with clear presentation and thorough analysis of results and an ability to evaluate critically and discuss research findings. Clear indication of insight, understanding and originality. An extremely competent and well-presented report overall, excellent in most aspects.
60 – 69 %	A good project report which shows a clear understanding of the problem and sound knowledge of the relevant literature. Sound presentation and analysis but perhaps not exploiting the results to the full. Relevant interpretation and critical evaluation of results, though with some limitations regarding the scope. Good general standard of presentation and organisation.
50 – 59 %	A satisfactory project report which shows some understanding of the problem but limited knowledge and appreciation of the relevant literature. Presentation, analysis and interpretation of the results at a basic level and showing little originality or critical evaluation. Some weaknesses in the organisation and presentation of the report.
40 – 49 %	A weak project report showing only limited understanding of the problem and superficial knowledge of the relevant literature. Results presented in a somewhat confused or inappropriate manner and incomplete or erroneous analysis in places. Discussion and interpretation of results severely limited, including some basic misunderstandings, and with very little originality or critical evaluation. General standard of presentation weak.
20 – 39 %	An unsatisfactory project report containing substantial errors and omissions. Very limited understanding, or in some cases misunderstanding, of the problem and very restricted and superficial appreciation of the relevant literature. Very poor, confused and, in some cases, incomplete presentation of the results and limited analysis of the results including some serious errors. Severely limited discussion and interpretation of the results revealing little or no ability to relate experimental results to the existing literature. Very poor overall standard of presentation.

Mark (%)	Criteria
0 – 19 %	A very bad project report containing many errors and faults. Virtually no real understanding of the problem and of the literature pertaining to it. Haphazard presentation of results, and in some cases incompletely presented and virtually non-existent or inappropriate or plainly wrong analysis. Discussion and interpretation seriously confused or wholly erroneous revealing basic misconceptions.

Please comment on the student performance in the dissertation, with reference, as necessary, to each of the component parts, abstract, introduction, results, discussion, references (appropriateness and accuracy).

Section B. Supervisor's Mark (written report) = _____%

Section B. Second Marker (written report)* = _____%
***from 2nd Marker marksheet**

Agreed mark Section B (written report) = _____%

If the two marks for section B. (written report) differ by more than 10% a third marker will be needed to adjudicate and for a dissertation mark to be agreed between all three.

(if necessary third markers name: _____)

Overall mark:

= (0.75 x Agreed Mark Section B) + (0.25 x Supervisor's Mark Section A*)
***from page 3**

= _____

NB Masters students' overall dissertation marks relate to the University descriptive categories as follows:

Mark (%)	**Grade**
70.0% or above	Distinction
60.0 - 69.9%	Merit
50.0 - 59.9%	Pass
Below 49.9%	Fail

Please note the dissertations are second (blind) marked. Markers may mark the work in sequence, one after the other, or in parallel. If they differ in their mark for section B by more than 10%, then a third marker will be needed.

Application of Supervised Machine Learning in Mass Cytometry

Name: Manikanta varma. Rudraraju

Student No: 160259713

Supervised by: William Alazawi & Hajar Saihi

Abstract:

The Advent of single cell technologies has revolutionised the field of immunoinformatics. The recent developments in mass cytometry technologies now enable researchers to extensively phenotype immune cells by using mass isotopes. The increased number of dimensions necessitated the creation of unsupervised and supervised learning techniques for cell classification. However, current algorithms are only suitable for classification of a single experiment dataset with the need for prior knowledge in the immunological field. Such requirements deter novice researchers and the wider adoption of this technology in the clinical and research settings. Furthermore, researchers working on different species are often limited to manual gating due to lack of validation on such datasets. This work uses deep neural networks (DNN) to create a generalisable cross species model for classification of seven canonical cell types trained on multiple species and tissues. The model shows high classification accuracy with macro weighted f scores achieving 0.84 when tested on unseen dataset. With exception of B-Cells, this model also shows high f1, >0.75, scores across different classes, with all classes having high precision. This work also shows deep learning techniques perform slightly better than other supervised algorithms such as simple decision trees and support vector machines. The low entry barrier and simple steps needed for using the model can advantageous and an important step in automating the pipeline. Although this model can only classify canonical cell types, it can provide guidance for any future work for creating more complex models for rare cell classification.

Word Count: 9193 excluding Figures, Tables, Legends, References and Supplementary material

Table of Contents

Figures

Tables

Introduction

The human immune system comprises specialised cells and molecules that play a critical role in pathogenesis. Immune cells are generally categorised into two main groups based on their development and differentiation processes. Innate immune cells are derived from myeloid lineage and consist of Neutrophiles, Monocytes, and Dendritic Cells (DC) that use general conserved features of pathogens to provide a nonspecific phagocytic response. The lymphoid lineage cells consisting of T, B, and NK cells make up our adaptive immune response and provide cell-mediated and humoral immunity. The complex interactions between these systems protect us from foreign material (Chaplin DD, 2010). It is widely established that specific cells can dominate in certain conditions and significantly impact the disease prognosis. E.g., asthma is a chronic inflammatory disorder of the airways characterised by the dysregulation of helper T cells (Hye Young Kim et al, 2010). In Neutropenia, there is a significant reduction in neutrophil count due to various genetic and lifestyle factors, significantly increasing the risk of infection (Newburger PE & David DC, 2013). Therefore, it is imperative to understand the different types of immune cells and their roles in any novel treatments.

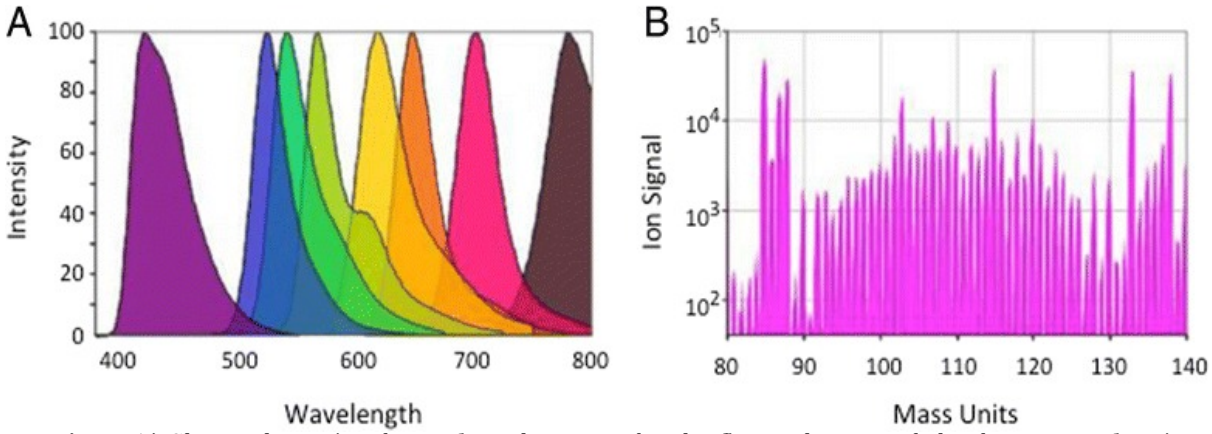


Fig.1: A) Shows the emitted wavelength spectra by the fluorophores and the degree overlapping regions causing spill over. **B)** Shows the spectra of the mass isotopes detected with greater resolution, allowing for the detection of more markers (from Maecker HT et al, 2015).

The advent of Single-cell technologies such as flow cytometry was at the forefront of immunophenotyping and aided researchers in understanding the immune system. Flow Cytometry uses fluorophores emitting different wavelengths of light. These are tagged to antibodies that bind to specific markers in a single cell suspension. Next, the suspension is introduced into a flow cytometer, where cells pass through multiple lasers, detecting scattered light and fluorescence. The fluorophores emit different light spectra in a specific wavelength corresponding to a specific marker, which can help researchers deduce the cell types and

quantities (McKinnon KM, 2018). However, the broad spectra of the detected wavelength have limited the number of fluorophores used in a single experiment as it can lead to spillover, a phenomenon where there is an overlap of spectra from two different fluorophores (see **Fig1A**). This reduces the number of parameters for downstream analysis. Although more advanced flow cytometry and computational techniques are being used to overcome such challenges, other cytometry techniques have been developed, which use heavy metal Isotopes known as mass cytometry (CyTOF) to improve spectral resolution. Instead of fluorochromes, single cell suspensions are stained by antibodies tagged with Lanthanide metal isotopes. They get introduced into CyTOF via nebuliser, causing the suspension to aerosolise. These droplets are vaporised and ionised using argon plasma, creating an ion cloud. The ion cloud, in theory, should consist of the ions from a single cell, stained antibodies, and their respective metal isotopes. Next, the cloud passes through a quadrupole with a magnetic field to remove low-mass ions such as nitrogen, oxygen, and carbon and retain heavy metal isotopes. Next, the cloud is introduced into a time-of-flight chamber where the ions are separated and detected based on their mass (**Fig2**) (Bendall SC et al, 2012). Since the spectra are resolved based on the isotope mass (Daltons), more parameters (30-50) can be detected without spillover issues (**Fig1B**). However, a significantly increased number of parameters creates new challenges for downstream analysis due to increased dimensions (Olsen LR et al,2019).

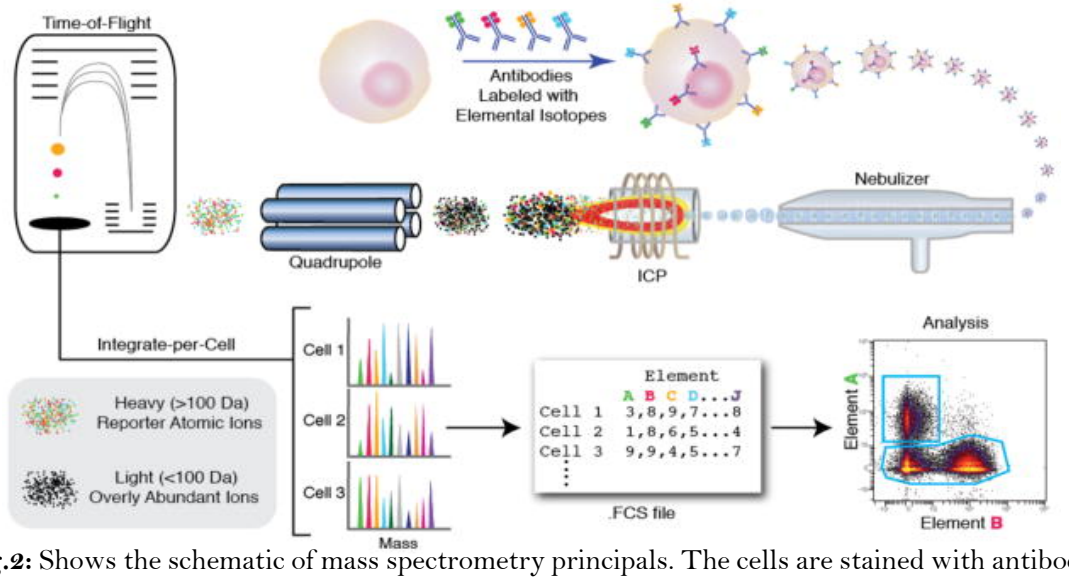


Fig.2: Shows the schematic of mass spectrometry principals. The cells are stained with antibodies conjugated to Lanthanide mass isotopes as they are not found in biological samples. The sample is introduced into ass cytometry where they get aerosolised, atomised and ionised. These ions are introduced into Time-of flight (TOF) chamber where a reflectron reverses the ion trajectory and resolving them based on mass to charge ratio. Detected ions induce an electrical pulse that is converted into ion count and recorded in the FCS file (from Bendall SC et al,2012).

CyTOF Pipeline

Data acquisition and File format:

The duration and intensity of the ion cloud detected above a certain threshold are retained and converted to raw ion counts by the instrument using the electrical impulse frequency and recorded in an FCS file, a standard file format developed by the International Society for Advancement of Cytometry (ISAC) (Olsen LR et al,2019). This file consists of metadata in the "HEADER segment" and contains information regarding the FCS format version and plain text to locate other sections. The expression set within the "DATA segment" consists of columns corresponding to mass isotopes and the duration of an event. These columns are also known as channels. The rows correspond to a single event containing their respective ion counts for each channel and the event duration. Finally, information about the experiment is contained within the "TEXT segment". It provides information about the isotope channels and their corresponding cell marker as a key-value pair, the instrument used, and the number of events. However, this is not an exhaustive list (Spidlen J et al, 2010). In addition, files obtained from the instrument contain noise and variation not representing the underlying biology that can impact downstream analysis. These problems are addressed using computational techniques to preprocess the data based on the sample preparation protocol.

Normalisation:

CyTOF data acquisition times can be lengthy due to their lower flow rate. This creates some challenges for ion detection due to the degradation of instrument sensitivity and mass calibration drift. In addition, a slower flow rate can cell introduction due to the blockage, further introducing noise and variation. As these changes do not reflect underlying biology, researchers must address these instrument variations (Rybakowska P et al,2021). Cleaning and recalibration are recommended for longer instrument runtime, and the use of a computational technique known as bead-based normalisation allows for intensity adjustment post-acquisition. To utilise bead-based normalisation, samples are mixed with polystyrene beads before introducing them to CyTOF. The mass ranges of the bead isotopes reflect the isotope masses used for antibody tagging. These are detected as bead events and differentiated from cell events. The instrument has an integrated algorithm, which uses a correction factor based on a "Global bead passport". However, it is not as robust at addressing variation in instruments with

a different overall signal intensity of the sample compared to the global bead passport, forcing the data to conform to the passport signal (Olsen LR et al, 2019).

Fink et al (2013) have published an open-source algorithm to overcome this problem. The algorithms first identify the bead events by plotting bi-axial plots using all bead isotopes against the DNA intercalator. Next, gates are applied to all plots that act as thresholds. An event within the thresholds of all bead channel plots is assigned as a "bead" event. These bead events are used to calculate a global mean for each bead isotope. To reduce the local variance, median intensities at a single time point are calculated by first summing the running median using a 500-event window. Finally, the median and global intensities are used to fit a regression line, which provides a correction factor for interpolating values for all mass isotopes within mass bead ranges and extrapolating for any mass isotopes falling outside the mass range of the beads.

Debarcoding:

It is well known that sample preparation protocol can significantly impact the detection of ions when running multiple samples. Slight variations caused by wet lab techniques such as pipetting, freeze-thaw cycles, numerous machines, and in-house antibody conjugation can introduce variation that does not represent any biological variation and can be misleading when analysed. This is known as the batch effect. To mitigate such issues, Zunder ER et al (2015) introduced a single-cell deconvolutional (SCD) algorithm that uses palladium-based barcoding to tag different samples and creates a single multiplexed sample. During sample preparation, the cells are first tagged with three unique palladium isotopes of different masses that cannot overlap with Lanthanide mass ranges. Since there are a total of six palladium isotopes used, a total of 20 combinations are created for barcoding using binomial coefficient $\binom{n}{k}$. After assigning a sample with a unique barcode, different samples are mixed together, followed by antibody staining resulting in reduced variation associated with protocol. Post-data acquisition, a single file will consist of cells from various sample types which need to be identified and separated. SCD algorithm first assigns preliminary barcodes by identifying three isotopes with the highest expression levels based on the user-provided barcoded channels list. It also calculates the distance between the third-highest and the fourth-highest expressed palladium isotopes. The algorithm then fits a linear and a three-parameter log-logistic function parameter that outputs cell yield based on the barcode separation distance. This yield curve can identify true barcode events (Crowell H et al,2022). It is important to note that more than three ions can

be used for barcoding, with no observed advantages. However, fewer combinations are available for multiplexing.

Data transformation and pre gating

Data transformation is a critical step in pre-visualisation due to the large dynamic range of detected Ions, making it impractical to visualise and distinguish positive/negative population distributions on a linear scale. With flow cytometry, it was a common practice to use log transformation to improve visualisation. Although it accurately represents distributions at higher transformed values, the values closer to 0 (accounting for a significant portion of the data) are confined into a compact bin creating a trimodal distribution instead of the bimodal that we expect (Dillon H, 2021). CyTOF data can also contain values of 0 due to lack of detection or negative values due to randomisation in the interval (x-1, x) to prevent picket fencing plots. This makes it impractical to apply log transformation, as $\log(0) = -\infty$. The current consensus is to use hyperbolic arcsine transformation, given by $\sinh^{-1}\left(\frac{x}{Cofac}\right)$ with a cofactor of 5, which was empirically derived from having an accurate representation on a biaxial plot for CyTOF data. This transformation allows the data to be displayed in a linear-like (affected by cofactor) scale at lower ion counts and log-like at higher counts, with a distribution similar to gaussian (Rybakowska P et al,2020).

Post transformation, the cells are preprocessed to remove noise, debris and doublets in a process known as pre-gating. This involves the usage of 4 parameters (1 bead isotope, DNA Intercalator-1, DNA Intercalator-2 and viability stain) to identify single viable cell events. It is important to note that single cell suspensions are prepared in the presence of DNA intercalators (Ir191, Ir193) that bind to DNA within the cells and viability stain such as Pt195, which can cross compromised cell membranes. Therefore, these isotopes can be used to identify any noise and debris as events without intercalator detections are unlikely to be a cell. In contrast, events with high viability stain are considered dead cells. Therefore, it is crucial to filter out debris, doublet and dead-cell events that make up a significant proportion of the dataset and can deviate from the single live cells affecting downstream analysis (Olsen LR et al, 2019).

Cell classification:

As the preprocessed data now consists of cell events, it is crucial to identify the different cell types before performing statistical tests accurately. Marker expression levels can provide such information for researchers to label cells accordingly; e.g., a T helper cell needs a high level of CD3 and CD4 antigens but low CD 8 levels (Maecker HT et al, 2012). Identified cell types can then be tested for changes in abundance or signalling markers associated with a phenotype. The gold standard procedure to classify cells is using "gating strategies", where a biaxial plot of a different combination of markers is used to create boundaries to label the cells that fall within the threshold (Rybakowska P et al,2020). Due to the limited number of parameters measured in flow cytometry, it was feasible to manually gate using GUI software such as FCS express or CytoExploreR (R package, Dillon H, 2021). However, as parameters can exceed 30 in CyTOF, manually gating different combinations can be laborious, especially when gating for multiple samples, leading to a phenomenon known as the "curse of dimensionality". Furthermore, selecting only a few markers for gating can lead to bias and omit potential discoveries. Manual gating also heavily depends on researchers' expertise and can further exacerbate this problem (Kimball AK et al, 2018).

The significant improvements in computational power and algorithms within machine learning and artificial intelligence helped researchers overcome these problems using unsupervised and supervised learning techniques. In unsupervised learning, the cells are clustered together based on their similarity within the feature space. The similarity is determined by a distance metric, with Euclidean being the most common. Although many packages exist that utilise unsupervised techniques for CyTOF, PhenoGraph and FlowSOM have been shown to outperform other tools in both accuracy and runtime, with PhenoGraph having the advantage of better identifying subclusters. PhenoGraph uses K-Nearest Neighbours algorithms to cluster cells (Levine JH et al,2015). FlowSOM uses self-organising maps that map high-dimensional data into a lower dimension of nodes representing similar cells (Van G.S et al,2015; Quintelier K et al, 2021; Amir A,2019). However, sample types can also have an impact, as shown by X-Shift having higher accuracy for mice data (Liu X et al, 2019).

On the other hand, supervised learning techniques have been shown to generally have a higher adjusted rand index (ARI), which shows the resemblance between two clusters, and F-measure, which shows the accuracy between model predictions and ground truth (Liu P et al, 2020). DeepCyTOF is an integrated deep learning model trained on manually gated reference samples

to classify the remaining samples. They incorporate autoencoders which encode the input data using an encoder and decode it into an output layer using a decoder. This is a denoising tool as the input values are reconstructed from code space. They also incorporate a residual net for batch correction by minimising the maximum mean discrepancy (MMD), which measures the distance between 2 distributions (Li H et al, 2017; Borgwardt KM et al, 2006). ACDC uses known prior biological knowledge to identify cells using sem-supervised learning. The algorithm first requires an input table of markers (columns) and cell types (rows) which are converted into landmark points that represent "fingerprints" for specific cell types. This can be achieved by assigning an integer value of -1,0, or 1 (never present, do not consider, or present) to each marker, creating a unique fingerprint for each specified cell type and using a score function that utilises two-mode Gaussian mixture model to create a posterior probability. Then random walks are used to evaluate the probability that an event belongs to a specific cell type (Lee HC et al, 2017; Liu P et al, 2020).

Current Challenges

To understand dynamic changes in cell population structure and their states under different conditions, e.g., disease vs healthy, researchers need to perform differential gene expression analysis. These techniques are well established and implemented in R packages such as **Limma**, **Deseq2**, and **EdgeR,** originating from the field of transcriptomics. These techniques have been adapted and streamlined into packages like **diffcyt** (R package), which test for cell differential abundance and differential states using empirical Bayes moderated tests. **Diffcyt** also incorporates CyTOF data preprocessing packages for a streamlined user experience (Weber LM et al, 2019). However, classifying cell types remains a hurdle for researchers in the field of immunoinformatics. Indeed, for unsupervised techniques, researchers must validate the clustering results by manually gating, as types of samples and species can influence the results. Furthermore, the stochastic nature built into these algorithms makes it hard for other researchers to replicate with the same dataset (Weber LM & Robinson MD, 2016). It might be necessary for the researchers to understand the algorithms to provide appropriate arguments to the algorithm, e.g., the number of neighbours to use for grouping, to avoid over/under classification, which many wet lab scientists are unaware of.

In comparison, supervised learning techniques require prior knowledge and expertise from the researcher for the models to train. Deep CyTOF needs a reference sample that needs to be

manually gated. Further complications arise for any meta-analysis studies, as at least 1 sample needs to be gated from each experiment, and the model needs to be trained separately for multiple experiments. In ACDC, researchers must also provide a cell type maker expression table. The algorithm also does not consider that marker expression can be a continuum instead of the "presence" or "absence" of a marker (Liu P et al, 2020). DGCyTOF, a Convolutional neural network (CNN), has also been utilised for sample classifications, but not at a single-cell level (Cheng L et al,2022). This is further exacerbated by the fact that different species have drastically different immune cell compositions and experimental approaches with no specific guidelines for a researcher to apply these tools(Mestas J & Hughes CC, 2004). Such limitations and the need for programming skills often deterred a greater adoption of these techniques.

Deep Neural Networks (DNN) For General Classification

Deep learning models have shown to generalise extremely well to classify a wide variety of data in many other fields, such as ImageNet (Alex K et al, 2017), given a large training dataset. In essence, DNNs are a collection of layers (known as hidden layers) consisting of densely interconnected nodes. These hidden layers take features from an input space and map features from an input layer to an output layer. The connections are associated with weights, with each node having its own bias. Since every node is connected to input space with multiple weights, the attributes are calculated and summed up to a single value. This value passes through an activation function to determine the extent of the signal propagation through the hidden layers transforming the values. The output layer connected to the hidden layers receives these values and returns an output value corresponding to the prediction of a response variable. The model calculates a loss function by comparing the predicted labels with the ground truth. The model aims to minimise this loss function by updating weight via backpropagation using the chain rule to calculate the derivative of the loss with respect to each weight and bias in the network, known as gradient descent. The network is considered "deep" when it contains more than one hidden layer (Jason B,2019; Mandy & Chris, 2017; Anne B, 2017).

Current Deep learning models for classifying cytometry data require prior knowledge and are built only to classify single experiment samples at any given time. However, these are not suitable for large-scale studies or for novice researchers. Since CyTOF instruments can record over 200,000 cell data points for a single sample, it allows for exploiting large datasets

deposited in public databases to create a generalisable model that can classify cells across multiple species. In addition, more complex architectures can be constructed for a robust model trained on multiple datasets. This allows the model to learn features across various experimental data and can be used for classifying unseen datasets.

The project aims to build a generalisable cross-species DNN model that can classify seven immune cell types and one unlabelled class across multiple species and samples. The model does not require prior knowledge and has a simple setup for use by non-computational researchers and can also be used for meta-analysis. As the dataset is a collection of independent cells with no topological information, we will use feed-forward DNN for feature mapping.

Materials and Methods

Data Collection:

In order to build a cross-species model, it was essential to collect a wide variety of datasets from multiple databases. The scope of the study was also minimised by focusing on non-stimulated healthy samples to avoid further variation, as the expression levels tend to change drastically based on experimental conditions and lead to a non-generalisable model. FlowRepository (Spidlen J et al,2012) is a free-to-use database using the International Society for Advancement of Cytometry (ISAC) guidelines. Even though it has an application programming interface (API) for quick and efficient file access, it requires an authorised account from the developers, and the database contains some datasets with missing metadata associated with files. To avoid omitting potential datasets, files were obtained via manual inspection of various repository IDs using the website (**https://flowrepository.org/**).

The database consists of FCS files from different cytometry technologies, including flow, mass and imaging cytometry. However, only mass cytometry FCS files were obtained by identifying tags associated with CyTOF within the metadata field, the methods section of the literature, or by hovering over the file with the "Cytometry" field filled as "DVSSCIENCES CYTOF", referring to the instrument. The datasets came from published literature, as the methods section was used to discern the usefulness of the data. Since the model will be trained on healthy samples, If the dataset is obtained from an experimental study (stimulated vs control) or case-

control studies (healthy vs tumour), only control samples and healthy samples were obtained, respectively. Any files that were pre-labelled and separated by the authors were excluded.

Database	Species	Tissue	N =75	Description
FlowRepository	Human	Peripheral Blood Mono Nuclear (PBMC)	9	ID: **FR-FCM-Z2L2** Healthy samples collected at different time points (Trussart M et al, 2020)
	Human	Whole Blood (WB)	12	ID: **FR-FCM-Z36F** Healthy samples from multiple sites. (Geanon D et al, 2020)
	Human	Bone Marrow (BM)	8	ID:**FR-FCM-ZYQB** Healthy volunteers for validating CyTOF concordance with flow cytometry (Oetjen KA et al,2018)
	Human	Lung	1	ID: **FR-FCM-ZY9W** Healthy sample acting as a control to Löfgren's Syndrome (Kaiser Y et al 2017)
	Human	Intestines	15	ID: **FR-FCM-ZYRM** Healthy samples acting as a control to inflammatory bowel disorder (van Unen V et al,2016)
	Mice	Lung	5	ID: **FR-FCM-ZYDW** Healthy samples compared to infected for testing various algorithms (Kimball AK et al,2018)
	Mice	Lacrimal Gland (LG)	1	ID: **FR-FCM-ZY4P** Healthy controls compared to tissue damage response at different time points (Hawley D et al, 2017)
	Macaques	Whole Blood	5	ID: **FR-FCM-ZZSR** Immunophenotyping of 5 healthy recess macaques (Elhmouzi-Younes J et al,2017)
	Macaques	Whole Blood	5	ID:**FR-FCM-ZYBG** Data from 5 healthy recess macaques pre vaccination (Palgen JL et al 2019)
HDCytoData	Human	Bone Marrow	2	Healthy control compared to acute myeloid lukemia and for validating PhenoGraph (Levine JH et al, 2015).
	Mice	Bone Marrow	10	Healthy population for validating X-Shift algorithm (Samusik N et al,2016)
CytoBank	Human	Bone Marrow	2	Healthy controls compared to different drug stimuli (Bendall SC et al,2011)

Table 1: Shows the source of each dataset, Species, and Tissue types. We obtained a total of N= 75 samples from 12 experiments. The description column provides any ID associated with datasets that can be used for repeating the procedure, and a brief description of samples chosen from the study.

Some datasets were deposited on Cytobank (Kotecha N, et al 2010), a cytometry data management and analysis web service. Although more datasets could be obtained with a premium account, data collection was limited to freely available files. These are saved as public experiments consisting of raw data, analysed data, and files split based on cell types. Only the raw CyTOF data files were used. Finally, data was also collected from HDCytoData (Weber LM et al 2019), an R dataset package maintained by **Bioconductors** experiment hub consisting of pre-cleaned data used for many benchmarks. Like FlowRepository, only mass cytometry data was retrieved from this source (See **Table1** for details on chosen datasets and their source database). The data was collected over a 40-day period from the end of April 2022 to mid-June, resulting in N= 75 from 12 studies.

Data Pre-processing

Normalisation:

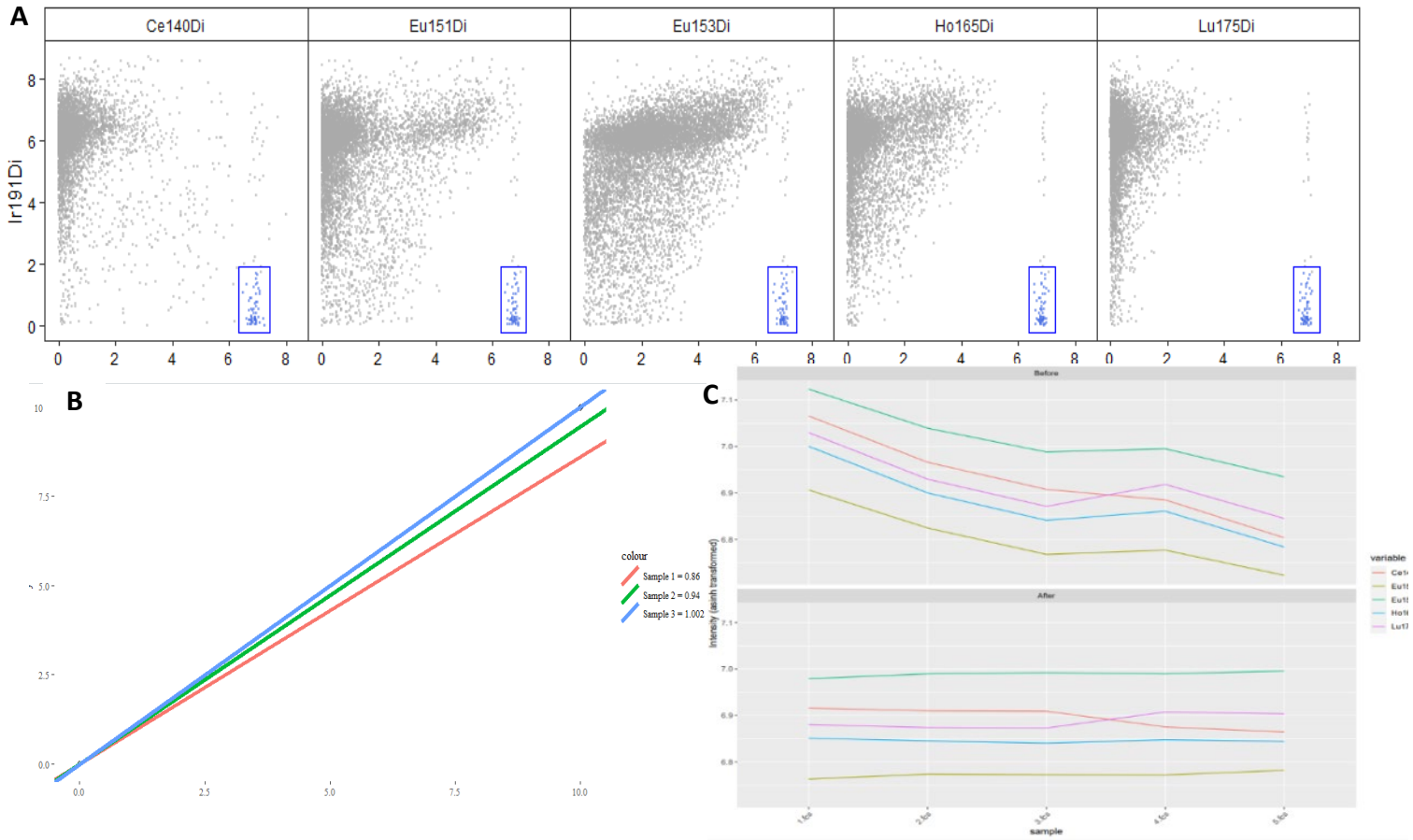


Fig.3: Shows diagnostic plots for the Mice Lung dataset normalisation (Kimball AK et al,2018). **A**) The algorithm automatically applies the gates across 5 bead channels to identify a single bead event. The events must be present within the gate across all 5 channels. **B**) Shows the fitted regression line coefficients that were extracted from for 3 out of 5 samples and manually plotted using **ggplot2**. These are used as correction factors. **C**) Shows the median intensity before and after normalisation, averaged across all 5 samples. **CATALYST** was used to obtain these plots

CATALYST (Crowell H et al,2022) is an **R package** that provides a pipeline for preprocessing steps for mass cytometry data. *norm_cytof* function used for normalisation takes a single cell experiment (SCE) object consisting of metadata, expression data and experiment information attributes. The function automatically uses default bead channels using the pre-configured EQ beads with mass isotopes Ce140, Eu151/153, Ho165, and Lu175 provided by Fluidigm. However, this can be changed by providing a list of channels when custom beads are used for calibration. These bead events were also filtered out post-normalisation, with files only consisting of cell events. Datasets that did not use EQ beads or were pre-normalised were excluded from this step. **Table2** shows the data that were normalised, pre-normalised, and non-

normalised due to calibration beads not being used. Diagnostic plots of an example dataset have been provided for illustration (**Fig3**).

Normalised	Pre-Normalised	Non-Normalised
Human Lung (Kaiser Y et al 2017)	Human Intestine (van Unen V et al,2016)	Human BM (Bendall SC et al,2011)
Human PBMC (Trussart M et al, 2020)	Human BM (Levine JH et al, 2015)	
Human WB (Geanon D et al, 2020)	Mice BM (Samusik N et al,2016)	
Mice Lung (Kimball AK et al,2018)	Mice LG (Hawley D et al, 2017)	
Macaques WB (Palgen JL et al 2019)	Macaques WB (Elhmouzi-Younes J et al,2017)	

Table.2: Normalisation performed using Fink's algorithm implemented in CATALYST. Pre-Normalised either using "Global bead passport" or Fink's algorithm implemented in MATLAB. No EQ beads used and cannot be normalised.

Debar coding:

A

	Mass ->	102	104	105	106	108	110
2	H1	1	1	1	0	0	0
3	H2	1	1	0	1	0	0
4	H3	1	1	0	0	1	0
5	H4	1	1	0	0	0	1
6	H5	1	0	1	1	0	0
7	H6	1	0	1	0	1	0
8	H7	1	0	1	0	0	1
9	H8	1	0	0	1	1	0
10	H9	1	0	0	1	0	1
11	CCL1	1	0	0	0	1	1
12	CCL2	0	1	1	1	0	0
13	CCL3	0	1	1	0	1	0

B

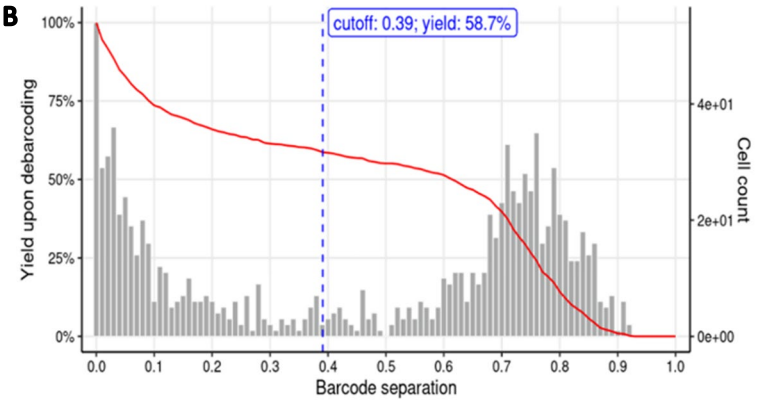


Fig.4: A) Shows the binary barcoded scheme for PBMC with 9 healthy and 3 CCL patients' scheme was used to assign preliminary barcodes to each event. **B**) Shows the linear and a three-parameter log-logistic plot that provides a Cell yield based on the barcode separation. The initial rapid drop in yield with small increases in the distance is due to the doublets or cell that have taken more than 3 barcoded isotope tags, causing the distance between the 3rd and 4th highest barcode to be extremely small. As the separation further increases the plateau phase shows no drastic changes in yield. Finally, the yield increases after a threshold distance represents the true single barcoded cells. The cut off can either be manually provided, or automatically calculated using the inflection point as shown in the plot. This diagnostic plot has been obtained using **CATALYST**.

SCD algorithm was also implemented in **CATALYST** for deconvoluting mixed sample type files into just one sample type file. This was achieved in three phases; first, by assigning preliminary barcodes using *assignPrelim* function using the binary barcoding scheme, second by calculating barcode separation cut-offs by *estCutoffs* function, and finally, by applying these cut-offs to deconvolute using *applyCutoffs* (Crowell H et al,2022) **(Fig4)**. The use of the SCD

algorithm depends on utilising cell multiplexing during sample preparation. Hence, only the Human PBMC dataset required debarcoding. Except for Human WB, Mice Lung, and Human BM (Bendall SC et al, 2011), which did not use multiplexing, the rest were pre-deconvoluted.

Transformation and Pre-gating:

All datasets have been transformed using the hyperbolic arcsine transformation using the **R**s built-in function with a cofactor of 5 prior to pre-gating and cell gating.

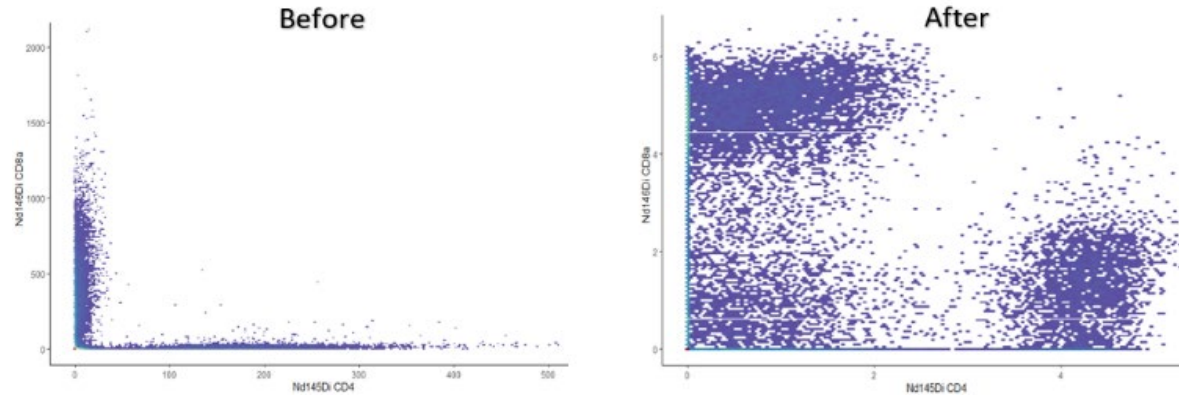


Fig.5: Shows biaxial plot of CD4 vs CD8 markers before and after transformation. The pre transformed plot shows large values dominating the scale, making it harder to distinguish different cell types. Whereas the post transformation plot show's clear separable populations. Data: Human BM (Oetjen KA et al,2018)

As previously described, pre-gating is a required step for filtering out any noise, debris and doublet events. This involved using a 4-step process. The first step involves using biaxial plots of the DNA intercalator and a single bead isotope, typically Ce140, to gate for cell events and remove any noise and potential bead events that were not identified during the normalisation step as shown in **Fig6A**. The second step involves using both DNA intercalators to gate for single cells within the dense region and filtering out any debris that falls below the lower bounds of the thresholds of the gate, and doublet events that exceed the upper bounds of the threshold, as shown in **Fig6B**. However, this does not eliminate the problem of doublet events. Therefore the third step involves specifically filtering out doublet events by using the event length and Intercalator, as longer event lengths are associated with doublets **(Fig6C)**. Finally, cells with excess levels of viability stain are filtered out as they are considered dead cells due to compromised membranes **(Fig6D)**. It is important to note that there are no guidelines for creating filtering thresholds. Gating strategies rely on researchers' expertise and the sample

dataset. For e.g., There seems to be a higher than usual uptake of viability stain in the Mice Lung Dataset (Kimball AK et al,2018) **(Fig6D)** compared to other datasets. The thresholds were created on dense regions based on the understanding that live cells are likely to have similar stain detection. All the final preprocessing steps were performed in **R** using **ggcyto** for visualisation and custom scripts for pre-gating.

Manual Gating and Batch Correction:

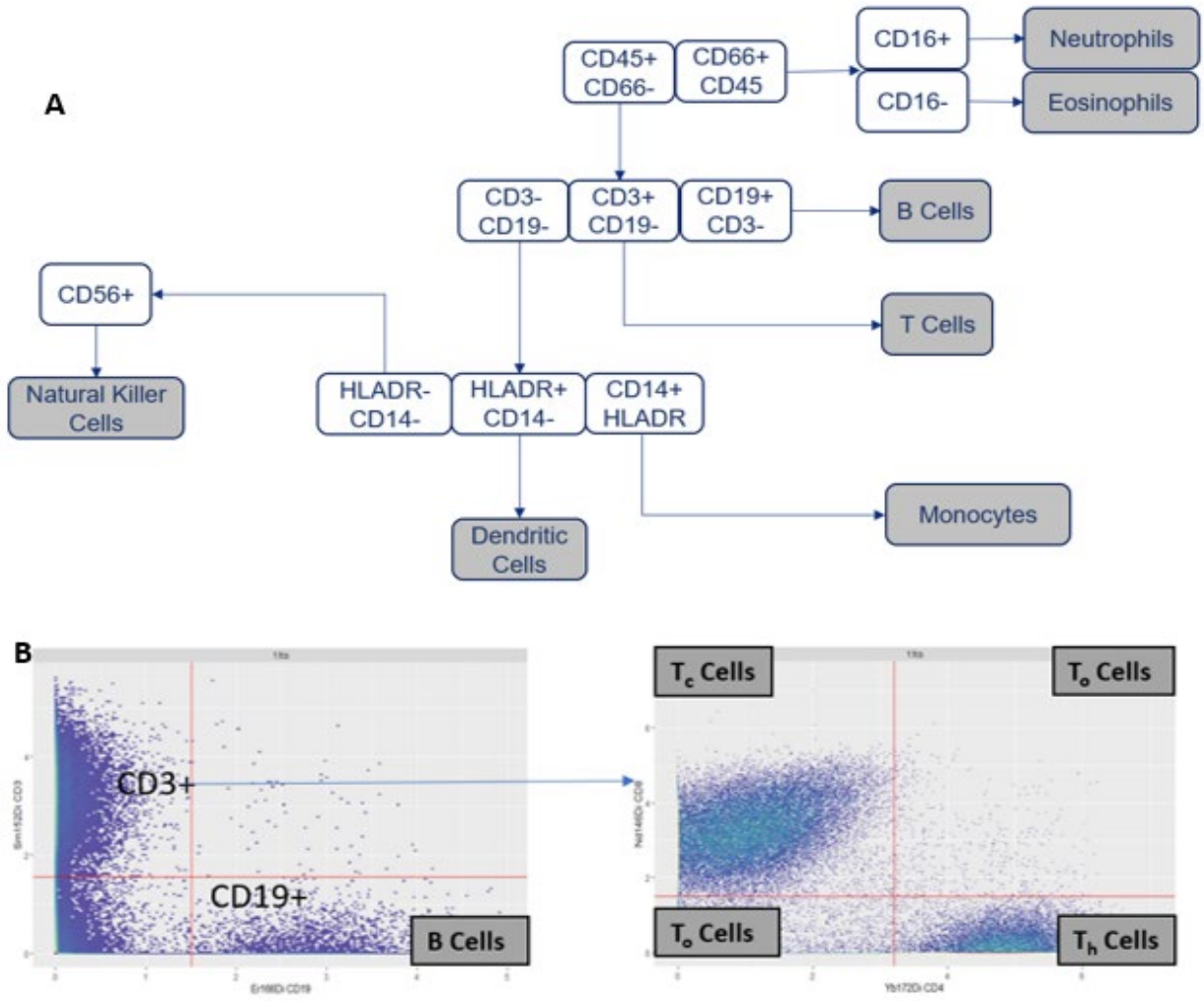


Fig.7: A) Shows the general strategy used for labelling classes (highlighted as grey) using 11 markers. If a cell does not fall within the strategy, they are classified as unlabelled. A box with 2 markers in it uses biaxial plot to gate for cell population. A positive marker requires high levels of the marker expression. A negative marker requires a low expression value, and a neutral marker does not have thresholds (can be high and low). A box with single marker uses a density plot showing a bimodal distribution, with higher value population gated accordingly. **B)** Shows an example of s mini gating strategy using mice lung dataset for illustration purposes. B cells require high CD19 but low CD 3 levels. In order to gate for cytotoxic T cells ($\mathbf{T}_c$), Helper T cells ($\mathbf{T}_h$), and other types of T cells ($\mathbf{T}_o$) first requires the gating of CD3+CD19- population.

Studies use distinct markers that are catered for their hypotheses. These marker panels are designed to allow researchers to identify the cell types in question. Consequently, studies often use custom gating strategies. These gating strategies are used to sequentially identify cellular populations using a different combination of markers using density and biaxial plots. E.g., To identify helper T Cells, cells are first gated to contain events with high CD3 expression levels (CD3+) followed by gating for high CD4 levels, thus making the cell CD3+CD4+. Whereas B cells are CD19+ CD3- **(Fig7B)**. Gating strategies used by the authors cannot identify the cell types required for building a cross-species model and also introduce batch effect. To maintain consistency, a standardised gating strategy proposed by Maecker HT et al (2012) for immunophenotyping was used to gate all the samples, with guidance from an expert. Although this was suggested for Human Immunology Project, this strategy has been extended to mice and macaques datasets. An overview of this strategy is shown in **Fig7A**. This strategy assumes 11 common markers to be shared across all the studies used for this project. However, this is not always true, so missing markers will be imputed as 0, and any associated gates for cell identification will be excluded, e.g., No B Cells have neem identified in Human BM (Oetjen KA et al,2018) dataset due to missing CD19. Other imputation methods such as mean or median, can lead to erroneous expression values. As cell markers expression depends on the cell type, e.g., a T cell with high CD3 levels cannot express high levels of CD19. Each sample from the datasets was gated separately using **R** *ggcyto* and custom scripts. The gating strategy for each dataset and the presence/absence of the panel markers are provided in the **supplementary material S1 and S2**.

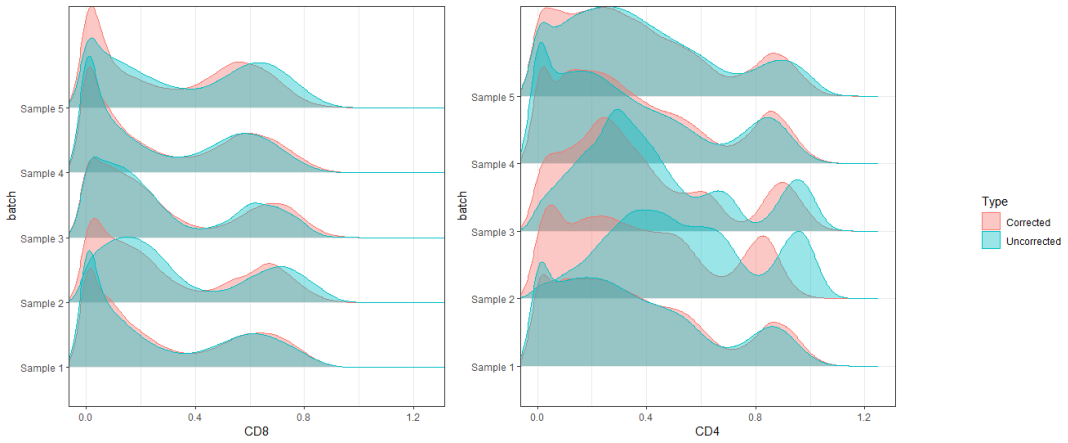


Fig.8: Shows an example of density plot of 2 markers (CD8 and CD4) out of 11 surface makers before and after batch correction using Mice Lung Dataset. Diagnostic plot obtain using cyCombine.

Prior to model training, a batch correction technique was applied using **Cycombine** (**R** package, Pedersen CB et al,2022), to Mice Lung and Human BM (Bendall SC et al, 2011) due to their sample preparation protocol, as these studies did not use multiplexing to address the variation associated with preparation techniques. Therefore, the samples were only batch-corrected within the studies but not across, i.e., Human BM sample 1 and sample 2 were batch corrected to each other and Mice Lung samples 1 to 5 were batch corrected to each other. Although Human WB samples were also prepared separately and across multiple sites. They used MaxPar Direct Immune Profiling Assay (MDIPA), a pre-configured dry tube containing a 30-marker broad immunophenotyping panel with optimal antibody concentrations. This allows samples to be directly added to the tube without needing antibody preparation and thus has a limited batch effect.

Feed Forward Neural Networks

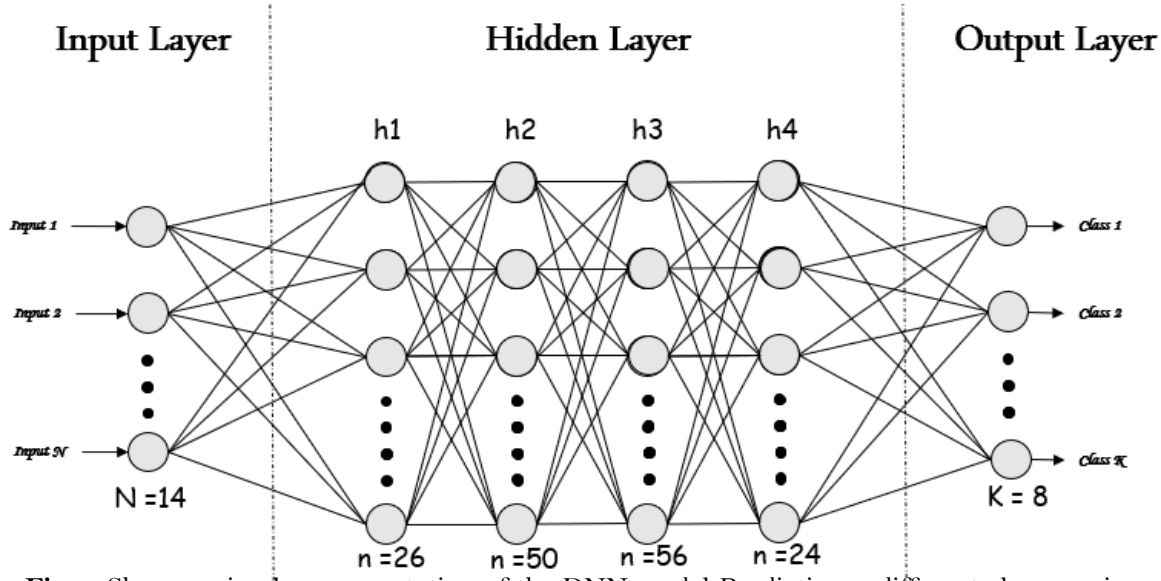


Fig.9: Shows a simple representation of the DNN model Predicting 8 different classes using 11 surface makers and one hot encoded species information.

The network consists of an input layer with 14 dimensions (11 markers and three species types that are one hot encoded). Four hidden layers, with 26 nodes in layer one, 50 nodes in layer two, 52 nodes in layer three and 24 nodes in layer four, with all hidden layers passing through a Rectified Linear Unit (ReLU) activation function, given by $\sigma(z) = \max(0, z)$ for a nonlinear transformation. Nodes are densely interconnected weights w_i and each node is associated with a bias b. The nodes receive input vector x such that the output of a node is given by $\sigma(\sum_{i=0}^{n} w_i x_i + b)$. The final output layer will consist of 8 nodes, each representing a different

class (Eosinophils, Neutrophiles, T-Cells, B-Cells, Monocyte, Dendritic cells, Natural Killer cells, and unassigned) and undergoes SoftMax activation, given by $S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$, transforming each output value x_i to a probability between 0 and 1, such that the sum of all output values across classes will equal 1. These probabilities are used to calculate sparse categorical cross entropy as the loss function given by $Loss = \sum_{i=1}^{n} y_i . log\hat{y}_i$, for weight updates via backpropagation to minimise the loss (Anne B, 2017; Jason B, 2020). Batch normalisation layers were also used to standardise the layer inputs of each mini-batch to prevent gradient decent oscillation caused by larger weight updates as a result of network learning from different scales. Regularisation techniques, dropout and L2 term were also utilised. The dropout layer randomly masks nodes based on user-provided proportion during batch training causing the model to generalise better. L2 regularisation, also known as ridge regression, adds the sum of the squared norms of the weight matrices given by $\left(\sum_{j=0}^{n} \|w_j\|^2\right)\lambda$, where w_j is the weight matrix that corresponds to each layer j in the network. This acts as a penalty term added to the loss term so that the loss function now becomes the sum of the original loss term and the L2 regularisation term. Hyperparameter λ, known as the regularisation rate, can be used to control the L2 term. Larger λ values cause the model to penalise complex models and incentivise the model weights to be closer to zero, preventing our model from overfitting the train data (Mandy & Chris, 2017)**.See Supplementary S8A for full architecture layout**. This model was implemented using **TensorFlow** and **Keras** API.

Usage	Species	Tissue	N	Datapoints	Study
Train	Human	Peripheral Blood Mono Nuclear	9	3208848	Trussart M et al, 2020
	Human	Whole Blood	12	3523364	Geanon D et al, 2020
	Human	Bone Marrow	8	3970683	Oetjen KA et al,2018
	Human	Lung	1	10405	Kaiser Y et al 2017
	Human	Intestines	15	444830	van Unen V et al,2016
	Mice	Lung	5	388515	Kimball AK et al,2018
	Mice	Lacrimal Gland	1	119649	Hawley D et al, 2017
	Human	Bone Marrow	2	244539	Bendall SC et al,2011
	Macaques	Whole Blood	5	619260	Palgen JL et al 2019
Test	Human	Bone Marrow	2	412762	Levine JH et al, 2015
	Mice	Bone Marrow	10	841644	Samusik N et al,2016
	Macaques	Whole Blood	5	531768	Elhmouzi-Younes J et al,2017

Table.3: Show the train and test split of the data and their respective studies. For training a total of **12530093** data points were available. The Training dataset was further split into 70% training and 30% validation using **sklearns** *train test split* function. The test set contain a total of **1776573** data points for model evaluation

The data will be split into train and test, as shown in Table 3. The test dataset was chosen to include one test dataset (Macaques) with similar sample distribution to one of the training datasets (Macaques), one test dataset (Human BM) that has different sample distribution to the training dataset but has been trained on the same species and sample type (Human BM), and one dataset with sample distribution that was not seen by the model (Mice BM). The test dataset will also be aggregated and tested as one large dataset. This is to improve the visualisation and interpretation of results without repetition.

The model described above was built using knowledge from the existing models, followed by hyperparameter tuning for an optimal architecture. **Hypopt** was utilised for grid search during hyper-parameter optimisation to test for different optimisers, learning rates, dropout rates, L2 penalty rates, batch sizes, and epochs. Although **sklearn** has well established grid-search function, it only allows for cross-validation (splitting data to proportions) of just the training dataset. In contrast, **hypot** will enable us to use the test dataset for validation. This allows us to optimise for generalisability instead of optimising for the training data. The TensorFlow model was wrapped in **scikeras** *KerasClassifier* class to maintain compatibility across the packages.

Domain Adversarial Neural Networks (DANN), a domain adaptation technique, were also tested for classification improvement. As the training set comes from a particular joint distribution represented by $trainset = \{(x^{train}, y^{train})\}_{i=1}^{n} \sim Q_{X,Y}$, where $Q_{X,Y}$ represents the training data distribution, although related, differs significantly from the test distribution, given by $testset = \{(x^{test}, y^{test})\}_{i=1}^{n} \sim Q_{X,Y}$. The model's ability to accurately classify the test data for its given distribution is much lower due to the model minimising the loss function only based on the training distribution. Computer vision neural networks successfully used domain adaptation to overcome this issue by using a discriminator that can identify the training data, referred to as source and test data, referred to as target, such that the feature extractor layer will be guided to learn features that cannot be used to discriminate the domain (Ganin Y et al, 2016; Harsh M, 2021). The work from Planche B & Andres ET (2019) was adapted and combined with the above model by adding a second network connected to the final feature extractor layer (hidden layer 4) using gradient reversal. As the secondary network minimises the domain classification loss, the feature extractor maximises it due to gradient reversal. This

causes the network to learn generalisable features across the domains. Note that this requires the training data, training labels and testing data, with training and testing data as source (coded as 1) and test as target (coded as 0). **See Supplementary S8B for full architecture layout**.

Finally, the DNN model will be compared with a simple random forest classifier model and support vector machines (SVM). Random forest is a machine learning algorithm made off multiple decision trees which aim to find a series of optimal split points of the input parameter until a class prediction. **Sklearn** implementation of this algorithm and its default setting (ensemble of 100 trees and Gini impurity for splitting criterion) was utilised for creating a decision tree classifier. SVMs algorithm finds an optimal hyperplane that separates class in N-dimensional space by maximising the margins. Similarly, **sklearn's** implementation and its default settings (radial bias kernel function) were used**.** For more information on these algorithms, refer to **sklearn's** documentation.

As the classes in the test dataset were imbalanced, model evaluation was done using the F1 score, which is a harmonic mean of precision calculated by, $precision = \frac{TP}{TP+FP}$ and recall, calculated by, $recall = \frac{TP}{TP+FP}$. Since this is a multiclassification model, we will use the macro average f measure given by $Macro\ F1 = \frac{1}{K}\sum_{i=0}^{K} f1_i$, where $f1_i$ is the f1 score for a class index and K the total number of classes, and the weighted f measure given by $Weighted\ F1 = \sum_{i=0}^{K} \frac{c_i}{N} f1_i$, where c_i is the number of cells for a given class, and $N$is the total number of cells (Harsh G, 2019). Area under the curve (AUC), receiver operating characteristic curve (ROC) was plotted, indicating the model's capability to distinguish between different classes. The ROC shows the performance of a classification model at different thresholds, and AUC allows us to compare different classes. These are plotted with the true positive rate (TPR) on the y-axis against the false positive rate (FPR) on the x-axis. Since these metrics can only be applied to binary classification, we will be comparing ground truth and predictions using 1 class vs all, i.e., if one of the eight classes is coded as 1(True positive), the rest will be coded as 0 (true negative) (Vinícius T, 2012).

Creating and testing machine learning models was done in python 3.9.11.

Computational resources: 8-core i7 CPU, 32GB RAM, RTX 2080 super. Cuda toolkit and CuDNN libraries were used for GPU computation.

Results and Discussion

The optimal DNN model described in the methods section was established after multiple iterations of training, validation, and testing. Since the training classes were imbalanced, (**Supplementary S3)**, the data was first under-sampled, so the predictions during the testing phase were not skewed or biased towards a specific cell type (Jason B, 2020) and significantly decreased the computation times for creating and testing multiple architectures. Next, under-sampling was performed using *RandomUndersampler* function from **Imblearn** to the minority class. Although it is well known that undersampling can lead to loss of information and impact the decision boundaries between classes (Tara B, 2019), the objective here is to establish a baseline model before fine-tuning using the complete dataset for better classification. The data was also not standardised due to lower performance (data not shown), likely due to arcsine transformation. One hot encoded species information was also initially omitted to test the model's ability to classify cells with fewer input features accurately. Since the weight and bias initialisation is also random, the training was repeated five times. This is to prevent a specific distribution of weight and bias values from performing exceptionally well on the limited test dataset but unable to generalise to other untested data in the future.

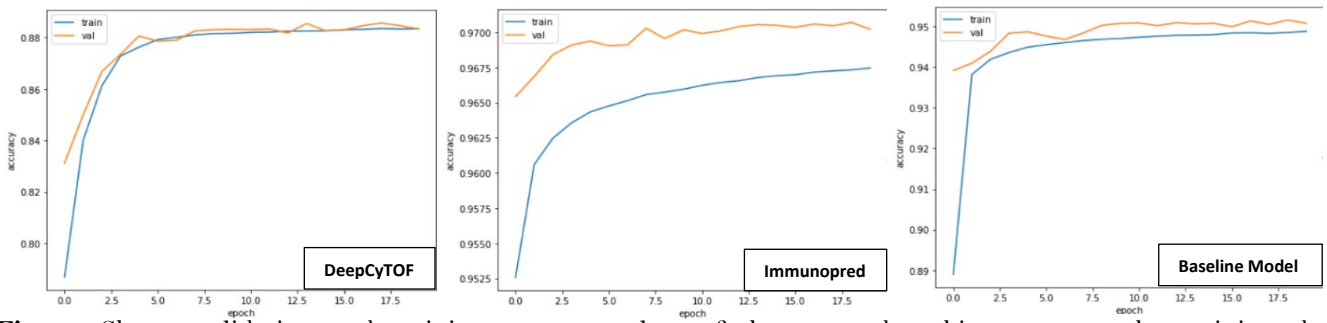


Fig.10: Shows validation and training accuracy plots of three tested architectures on the training data. DeepCyTOF had an average accuracy of 86.9% ± SD 3.38%. ImmunoPred had an average accuracy of 95.7 ± SD of 0.62%. Baseline model had an average accuracy of 92.5 ± SD of 0.98%. The plots above are taken from the best run out of five repeats. Plotted using TensorFlow, Python

The architecture from DeepCyTOF, the only published cell classification model, was used to fit the data. Note that only the cell classification architecture was used. The model consists of three layers. 12 nodes in layer one, 6 nodes in layer 2, and 3 nodes in layer three, with no batch normalisation or dropout layers. This model was built to classify four classes from a single experiment, which was modified to classify eight cell types. Training the model with our dataset with 20 epochs and a batch size of 64 showed the model to underfit the data, as shown in the training and validation accuracy plots, with an average accuracy of 86.9% being the

worst performer (**Fig 10**). The model was unable to learn the training dataset due to the fewer learnable parameters available for mapping features to the output node resulting in poor performance metrics for training and validation. Evaluating the model using the test data also shows the worst performance compared to other models with a macro f score of 0.56 and a weighted f score of 0.63, which vastly differs from the validation accuracy. The discrepancy between the macro and weighted f score suggests that the model was better at classifying only certain types of classes. Upon further inspection, the "unassigned" class, which comprised 41% of the test data, had the highest f1 score of 0.84 on average (**See Supplementary material S4**). This could be due to unassigned classes not having any features or patterns defining them and the simplistic nature of the model being unable to adequately fit our training data to separate the classes appropriately based on input features. However, T-Cells also had high f1 score of 0.81, which features associated with it.

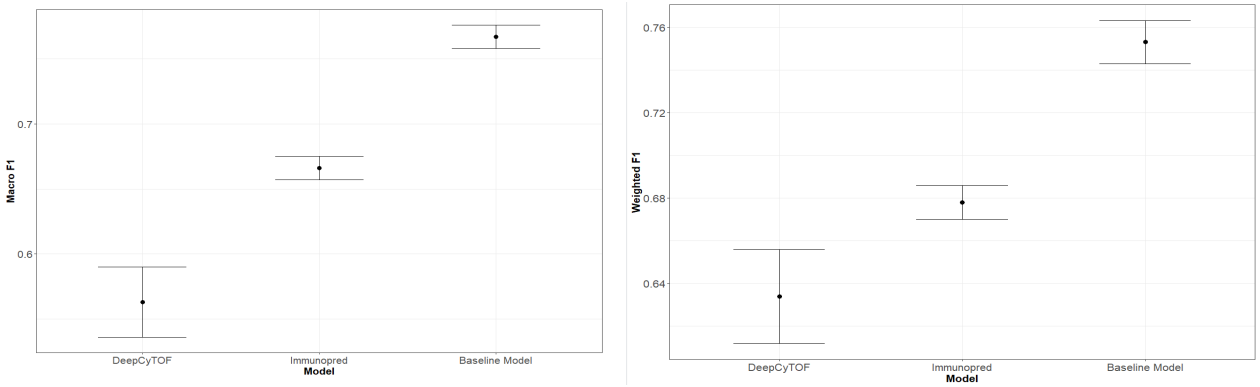


Fig.11: Shows the Macro f score and Weighted f score for each architecture using the test dataset, repeated five times. DeepCyTOF had an average Macro f score of 0.563 ± 0.027 and Weighted F score of 0.634 ± 0.022. ImmunoPred had an average Macro F score of 0.666 ± 0.009 and Weighted F score of 0.678 ± 0.008. Our baseline model had an average Macro F score of 0.767 ± 0.009 and Weighted F score of 0.753 ± 0.01. Plotted using **ggplot2, R**.

ImmunoPred is another model that was tested for its suitability to our dataset. This model was developed by the Alazawi group (William Alazawi, QMUL) but was not published. Therefore, the model's architecture details will not be described to protect intellectual property. The performance metrics of this model show substantial overfitting to our training data due to its high complexity (a large number of parameters). As shown in the accuracy curve (**Fig.10**), this model performed exceptionally well on the validation dataset compared to the other models. However, the macro and weighted f scores on the test dataset were relatively low,0.67 and 0.68 respectively, compared to the baseline model suggesting the model overfitting the training set and poor generalisability (**Fig 11**).

Optimiser	Learning rate	Dropout	L2	Epochs	Batch-size
Adam	0.01	**0.3**	0.1	20	1024
SGD	**0.001**	0.5	**0.001**	**30**	2046
RMSprop	0.0001			40	**4096**

Table.4: Show the hyper parameters tested for an optimal model during Grid search. All the combinations were iteratively searched using **hypot**. The green highlighted combination had the highest score of 0.78.

The Baseline model was built using the knowledge from the two previous architectures. The DeepCyTOF model was used as a starting point and increased its complexity to include more nodes and layers to account for the wide variety of input data, leading to an optimal architecture as described in the methods. Also, batch normalisation and dropout layers from the Immunopred model were adapted to improve generalisability. Post training, validation metrics show a high average accuracy of 92.5% on average, although 2% less than the Immunopred model (**Fig.10**). However, testing the model shows a considerable improvement in the macro and weighted f score when compared to other models (**Fig.11**). It is important to highlight that this architecture was the final iteration after testing multiple architectures with slight variation. The model was further optimised by tuning the hyperparameters using grid search from **hypopt,** which test for all combination of parameters as shown in **table 4**. Due to memory constraints and time limitations, the number of search parameters was limited to the methods shown in **table4**. The function outputs two variables. The first variable consists of a dictionary with the key-value pair containing information about the specific combination of hyperparameters, e.g. {learning rate: lr, epochs : ep1…} where lr1,ep1 represent the first value of the learning rate list and epoch list provided by the user. The second variable outputs the accuracy as a score for each model iteration. The highest score of 0.78125 was obtained for a model with hypermeters tuned to Adam optimiser with a rate of 0.001, dropout at 0.3, l2 regularisation rate of 0.01, 30 epochs, and a batch size of 4096. Since weight and bias initialisation can vary each iteration, **TensorFlows** global seed was set for reproducibility, so only our hyperparameters are tested**. See supplementary S5 material for examples.**

The large batch size was notable, as most models often recommend using a smaller batch size (32 to 512) for better generalisation. The exact reason behind performance degradation due to large batch size is still not well understood, but it is thought that large batch sizes are attracted

to sharp minima (Franck D 2017). A reason the hyper-tuned model was able to better classify with large size is likely due to different dataset sizes for each species, as human had significantly more datapoints than macaques. Therefore, with smaller batch size, the model was unable to use data from all species in some of the iterations during training of an epoch. However, this needs to be investigated further. Note that batch sizes of 128, 512, and 1028 were also tested separately but still showed lower performance (Data not shown). Finally, one hot encoded species information was introduced to the model, which showed the largest

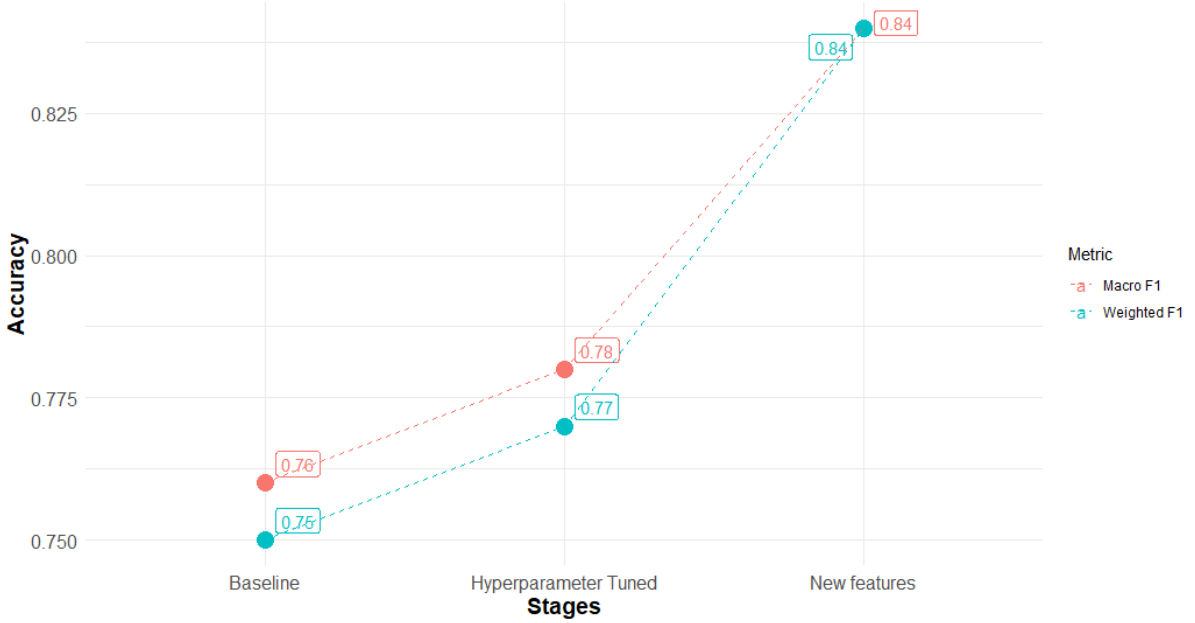


Fig.12: Shows the Macro F score and Weighted F score during different stages of training. There is a modest increase of 0.02 for Macro and Weighted F scores when hyper tuned. The model saw the largest increase when species information was added with Macro f1 increasing by 0.06 and weighted increasing by 0.07 from the hyper parameter tunes. Plotted using **ggplot2, R**.

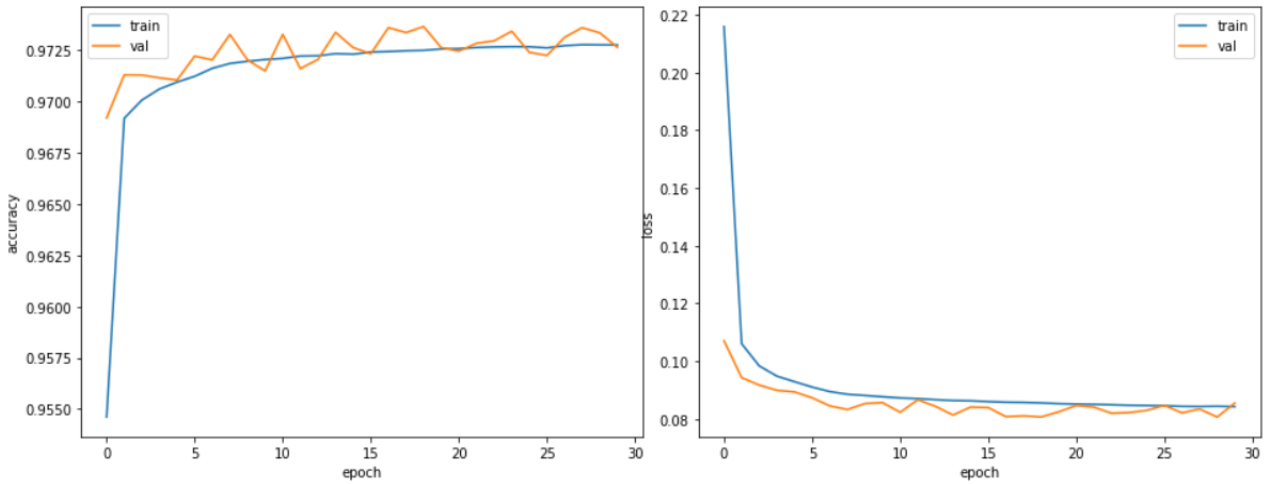
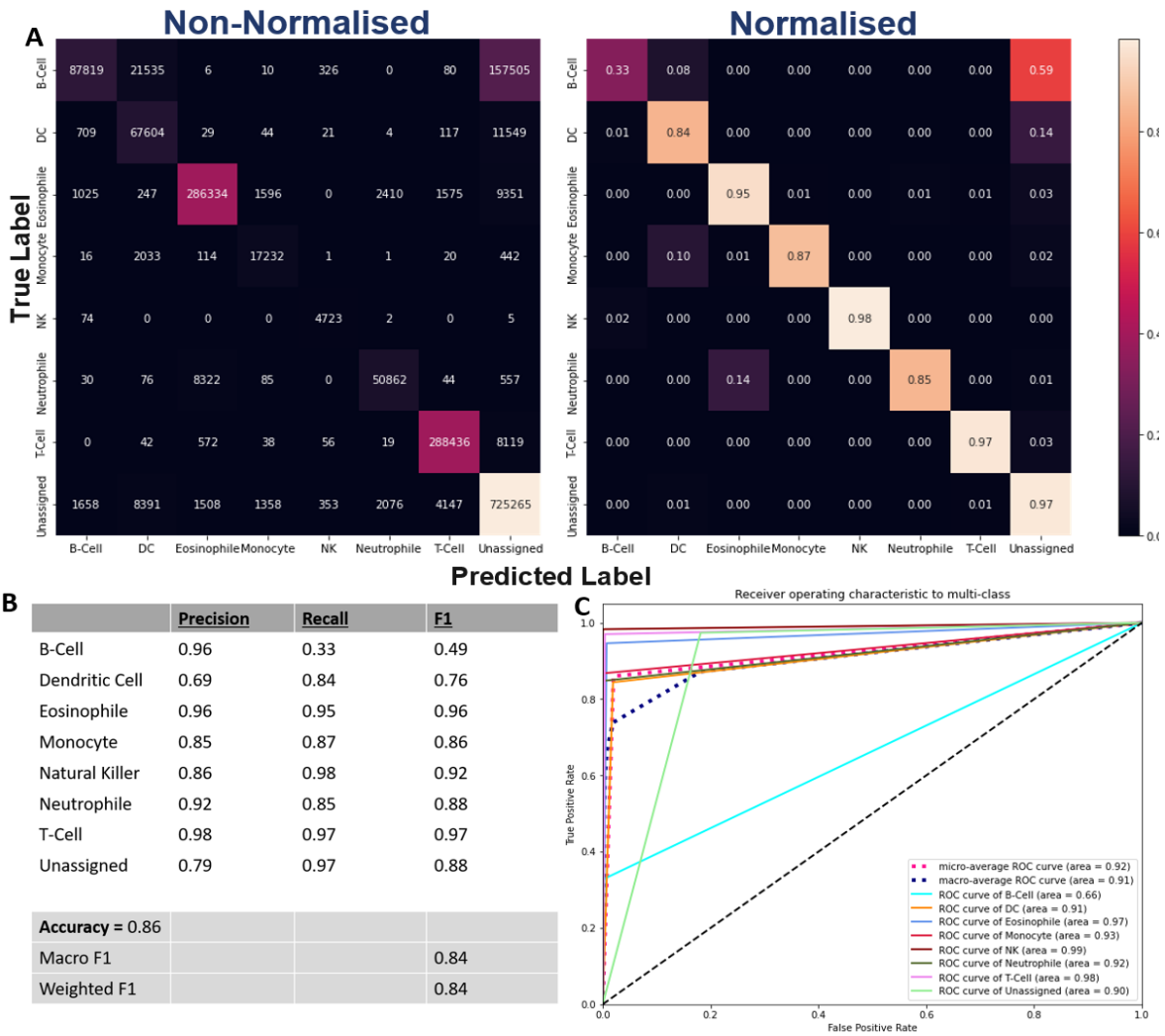


Fig.13: Show the accuracy and loss of the final model during training and validation. Best plot out of five repeats were taken with an average accuracy of 97% and a log loss of 0.06



	Precision	Recall	F1
B-Cell	0.96	0.33	0.49
Dendritic Cell	0.69	0.84	0.76
Eosinophile	0.96	0.95	0.96
Monocyte	0.85	0.87	0.86
Natural Killer	0.86	0.98	0.92
Neutrophile	0.92	0.85	0.88
T-Cell	0.98	0.97	0.97
Unassigned	0.79	0.97	0.88
Accuracy = 0.86			
Macro F1			0.84
Weighted F1			0.84

Fig.14: A) Shows the confusion matrix for each class predictions. Due to large imbalances in the test dataset, the values have been normalised for a more accurate map. **B)** Shows the class f1 score breakdown, with their precision and recall. The model generally has a high precision (>0.5), i.e., of all prediction for each cell class, most of them matched the true label. Low recall for B cell shows the misclassification, i.e., of all the true B cells, only some were predicted as a B cell. **C)** Shows the ROC-AUC curve, the diagonal line is a no-discrimination line, representing the model's inability to distinguish a specific class from the rest. For most thresholds the AUC is at 90% or above with exception of B cell. As the models had poor performance of B cells, it has the lowest AUC of 0.66. Plotted using Matplotlib & Seaborn, python.

The final model shows good performance metrics for classifying canonical cell types across humans, mice and macaques using the entire dataset. The model showed a slight improvement in validation metrics from the baseline model with an accuracy of 97% (4% increase) and a log loss of 0.06 (**Fig.13**), matching the ImmunoPred model training. However, the testing metrics showed a considerable improvement in the model's ability to classify, as shown in the **Fig13**

and **Fig14**, with both macro-F and weighted F scores achieving 0.84. Although significant improvements were made, the model could still not generalise fully. This is shown in the classification of B cells labelled as unassigned, despite the B cell marker (CD19) being present in 11 of 12 datasets. Upon further inspection, the model misclassified more cells from the Mice BM dataset than others. This needs further investigation to conclude any biological relevance or methodological errors associated with the model. This is also shown in the ROC plots with the exception of B cell curve, most curves being close to the top left corner and high area under the curve (90% or above). The B- Cell is much closer to the no-discrimination line (**Fig14C).**

Addressing imbalanced training dataset

It is understood that significant class imbalance can lead to a biased model towards the larger class. This is due to larger classes having more influence on weight updates during training (Jason B, 2020). This is clearly shown in the credit card dataset example (Jason B, 2020). To address this issue, we recruited three strategies to balance the dataset. Under-sampling using *RandomUnderSampler* shows lower macro and weighted f1 scores **(Fig 15B)**, which was expected due to loss of information as the total training data was down to 2167456 from 12530093 (Jason B, 2020). However, there is slight improvement in the B-cell f1 score compared to the unmodified dataset at the expense of other classes. Oversampling was omitted due to overfitting issues, as many repeats are needed for the minority class to obtain similar distribution to the majority class. Instead, Synthetic Minority Oversampling Technique (SMOTE), a data augmentation method, was used to generate synthetic data. This algorithm works by using the k nearest neighbours' algorithm and generating a new datapoint between the chosen datapoint and a neighbour. The the algorithm performs best when combined with under-sampling, outperforming pure under-sampling techniques (Chawla NV at al, 2002). Therefore, new data is only generated for classes with less than 10% of the data points from the majority class. Any classes that exceeded the 10% threshold were down-sampled only to contain 10% of the data points from the majority class**, see supplementary S6 for more details**. Surprisingly, this strategy performed worse than under-sampling, potentially due to large synthetic generation causing an overfit of our model **(Fig 15B)**. The synthetic generation threshold was also tested at 5% of the majority class with no improvement (Data not shown). Finally, cost-sensitive training was used by calculating the inverse proportion of class frequencies and providing these values as a dictionary to class weights argument when fitting the model. The model uses this to treat multiple instances of a larger class as equivalent to a single instance of a smaller class by weighting the loss function. However, the performance

metrics although better than other strategies, the performance was still lower than providing the model with the full dataset. This was unexpected, given the large difference between minority and majority class distributions. The f1 scores of each class also show that B cells were primarily misclassified despite not being the minority class. Increased misclassifications in other classes were observed when trying to correct for the imbalance **(Fig 15A)**

	Under sampled	SMOTE	Cost Sensitive
B-Cell	0.56	0.42	0.49
Dendritic Cell	0.62	0.56	0.71
Eosinophile	0.74	0.74	0.91
Monocyte	0.85	0.82	0.89
Natural Killer	0.80	0.77	0.85
Neutrophile	0.82	0.82	0.77
T-Cell	0.95	0.99	0.94
Unassigned	0.84	0.78	0.86

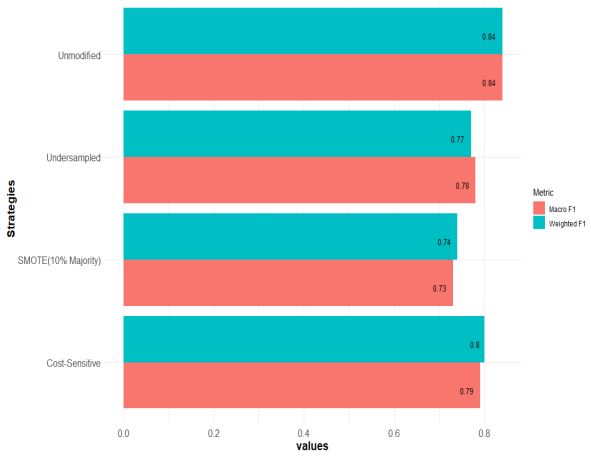


Fig.15: A) Show the f1 scores for each class for each strategy. When compared to unmodified (Fig 14B) we see slight improvements in B cell f1 scores but drastic decrease in other classes. Best plot out of five repeats were taken with an average accuracy of 97% and a log loss of 0.06. **B)** Shows the Macro and weighted f1 scores. The model had the best performance when the full input data without modification was used during training with macro and weighted f scores of 0.84. Plotted using **ggplot2**, R.

Domain Adversarial Neural Networks

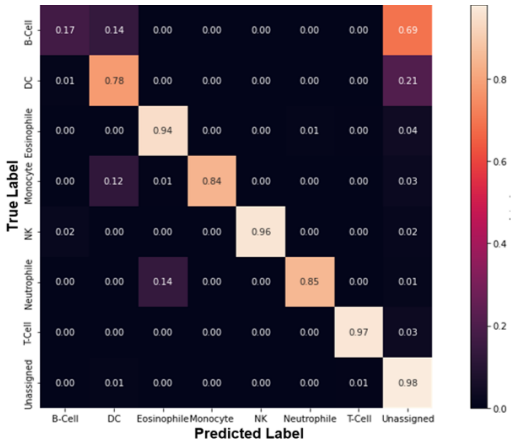
Testing DANN on our dataset showed no improvement in prediction accuracy of the test dataset, with macro and weighted f scores remaining unchanged **(S7)**. However, the secondary network's low complexity was initially considered an issue. Increasing the number of layers and nodes resulted in drastic performance loss (data not shown). The little to no performance gain can also be attributed to the violation of the assumption of only two distinct distributions for training and testing (Ganin Y et al, 2016) since our dataset is made up of multiple experiments, with each having a unique distribution. Combining the dataset could cause the model to not adequately discriminate the source and target domains due to multiple distributions. We observe little to no variation in the f scores of each cell **(S7)**. Although this technique showed drastic improvements in class prediction in the computer vision field, this was unsuccessful in predicting the test data.

Comparison with other supervised learning techniques.

Simple SVM and Decision trees also show excellent performance metrics achieving a macro f score of 0.82 and a weighted f score of 0.81, with SVM performing slightly better than the decision tree with a macro f score of 0.80 and 0.81. However, the DNN outperforms both algorithms by at least 0.02 points in both macro and weighted f scores. The DNN model generally has much higher f1 scores for cell classification. We see similar misclassification of B cells with SVM and decision trees, having the lowest f1 score and the most prominent difference from the DNN model. Important to note that SVM and Decision trees were built using default settings. Therefore, slight modifications in the parameter provided to the algorithm can yield similar results.

A

	Precision	Recall	F1
B-Cell	0.96	0.17	0.29
Dendritic Cell	0.63	0.81	0.71
Eosinophile	0.96	0.94	0.95
Monocyte	0.87	0.84	0.85
Natural Killer	0.86	0.96	0.91
Neutrophile	0.93	0.85	0.89
T-Cell	0.98	0.97	0.97
Unassigned	0.76	0.98	0.86
Accuracy = 0.83			
Macro F1			0.80
Weighted F1			0.81



B

	Precision	Recall	F1
B-Cell	0.96	0.23	0.37
Dendritic Cell	0.68	0.82	0.74
Eosinophile	0.97	0.92	0.94
Monocyte	0.84	0.84	0.87
Natural Killer	0.85	0.89	0.92
Neutrophile	0.96	0.99	0.90
T-Cell	0.98	0.98	0.98
Unassigned	0.76	0.98	0.86
Accuracy = 0.84			
Macro F1			0.82
Weighted F1			0.81

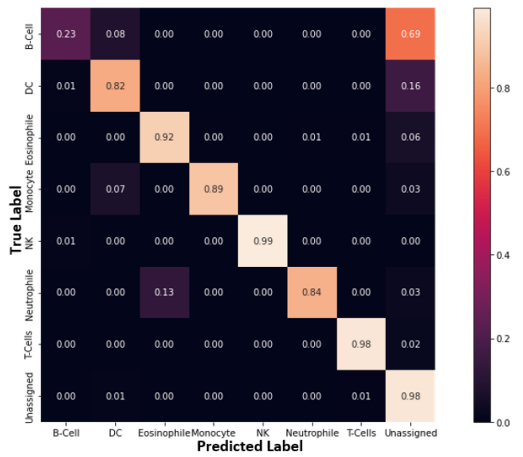


Fig.15: A) On the left shows the Precession, Recall, class f1, macro f1 and weighted f1 scores for the Decision tree. On the right shows the normalised confusion matrix. Compared to the DNN model, larges score difference was observed in B -cell and noticeable differences in Dendritic cell. **B)** On left is the Precession, Recall, class f1, macro f1 and weighted f1 scores for the SVM. The right shows the normalised confusion matrix. Largest difference was observed in B cell compared to the DNN model. Plotted using Matplotlib & Seaborn, python.

Conclusions:

Mass cytometry experiments can provide multivariate data with up to thirty parameters. Automated solutions for analysing such datasets are essential for CyTOF technology to become more widely adopted in research and clinical settings. However, current supervised and unsupervised learning for CyTOF data have shown limited use due to their high entry barrier, limiting them to manual gating for cell labelling. Researchers must also understand and use different algorithms based on their hypotheses and species in question. This work demonstrate the use of DNN, a powerful supervised machine learning technique that can be utilised to classify single-cell mass cytometry data. DNNs' ability to generalise and predict with high accuracy using large training datasets can be utilised for canonical cell classification in immunophenotyping with a limited number of markers. We address current challenges by creating a model that requires a simple setup, a CSV file with 11 specific markers with species information. Unlike other techniques, no prior field knowledge is needed from the researcher. With the use of standardised gating strategies, the model could predict 7 different cell types from unseen data with high accuracy despite missing markers in both training and testing data. Although it could not fully generalise, with many misclassifications observed specifically with B cells, the model's low barrier requirements for its utility can be highly advantageous for novice researchers with no computational experience. This is also the first cross-species model to demonstrate cell classification across three different species and different tissue with high accuracy despite the variation in immune cell composition and experimental procedure. This can help large-scale studies that use many datasets to identify these cell types without requiring manual gating and emphasising more on analysis. Although its current testing and validation are limited to humans, mice and macaques, further testing is required to check for its suitability from a different unknown species. Interestingly we see no advantages in using domain adaptation techniques using combined training and testing datasets from different studies. We also show that the DNN model had better performance metrics than simple SVMs and Decision trees.

Limitations and Future Work

Although the database consisted of large repositories of cytometry experiments, the metadata fields are often omitted when uploading. Therefore, manually examining a large number of repository IDs can be time consuming and error prone, potentially missing valuable datasets.

To overcome these issues, strict guidelines must be followed when uploading the files. Recently, ImmProt has implemented a database for sharing CyTOF data. Although it only consists of 16 datasets, a researcher can access all the granular information about the datasets with advanced search and visualisation filtering similar to the Genomics Data Commons data portal.

Although spillover was initially thought to have limited mass cytometry instruments, a recent study (Chevrier S et al, 2018) has shown to have a more profound effect on the CyTOF dataset. Spillover associated with CyTOF results from some mass ions being detected in the +/- 1 mass channels or the oxidation of an ion leading to the ions being detected in the channel that is 16 Daltons heavier. This can largely be negated with the appropriate concentration of the tagged antibodies, but the problem can persist, especially for less optimised novel protocols. A spillover matrix can correct such issues but has seen limited use within our datasets.

The pre-gating step is largely manual and depends on the researcher's expertise. There are no specific guidelines to implement thresholds during this step, which can impact the number of cells available for analysis. This is extremely important when the study involves dealing with rare cells. However, the recent updates in the instrument software now allow the researcher to record 4 new gaussian parameters derived from the event length. These parameters can be used to automatically pre-gate cells using probabilistic methods (Bagwell CB et al, 2019). Wider adaption of this process is needed for reproducible data.

Although the strategy recommended by Maecker HT et al (2012) was adapted to all our datasets, this gating strategy using specific markers was only validated against human WB and PBMC samples. The exact implications of using this on different tissue types and species are still unknown, as the immune composition can differ significantly. Using such a strategy also requires specific markers to be shared across all experimental studies, which makes it expensive. Furthermore, the lack of these specific markers could significantly impact the classification models and downstream analysis. Therefore, it is recommended that the manufacturer establish an assay similar to MAXPAR for study adaptability. This can also help independent researchers to use public data more widely since they contain the required markers, as many datasets were excluded during this project due to missing significant number of markers. Having not standardised assay also limits supervised learning techniques for classifying canonical cells, which are often not in the studies iinterest, as most novel researcher experiment on rare cells.

Due to limited time and experience in deep learning techniques, the model optimisation could be significantly improved, especially for B cell classification. Furthermore, domain adaptation techniques can be improved by finding a better architecture or different techniques, e.g. using discriminative adversarial neural networks.

Acknowledgements:

Special thanks to Professor William Alazawi for providing dissertation and presentation feedback. Hajari Saihi for supervising and guiding with her expertise in immunoinformatics and machine learning and providing crucial feedback for the dissertation. Dr WenHao Li for providing feedback on the presentation. Professor Conrad Bessant and the Bioinformatics (MSc) lecturers for organising and teaching adaptable skills.

Code: An example of trained model can be found at https://github.com/Mani-varma1/DNN_CyTOF/blob/main/Final_Model/Training_Model.ipynb. For using the model use "Using_Model.ipynb" with data stored in "data" folder. Request training and test dataset by emailing: m.v.rudraraju@se21.qmul.ac.uk

Refrences:

1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90. https://doi.org/10.1145/3065386
2. Amir A. Medium, Self Organizing Map (SOM) with Practical Implementation. May 2019. [online] Available at: https://medium.com/machine-learning-researcher/self-organizing-map-som-c296561e2117
3. Anne B. [online] Towards data science. The Complete Beginners Guide to Deep Learning . 2017 Jan. Available at: https://towardsdatascience.com/intro-to-deep-learning-c025efd92535
4. Bagwell CB, Inokuma M, Hunsberger B, Herbert D, Bray C, Hill B, Stelzer G, Li S, Kollipara A, Ornatsky O, Baranov V. Automated Data Cleanup for Mass Cytometry. Cytometry A. 2020 Feb;97(2):184-198. doi: 10.1002/cyto.a.23926. Epub 2019 Nov 18. PMID: 31737997.
5. Bendall SC, Nolan GP, Roederer M, Chattopadhyay PK. A deep profiler's guide to cytometry. Trends Immunol. 2012 Jul;33(7):323-32. doi: 10.1016/j.it.2012.02.010. Epub 2012 Apr 2. PMID: 22476049; PMCID: PMC3383392.
6. Bendall SC, Simonds EF, Qiu P, Amir el-AD, Krutzik PO, Finck R, Bruggner RV, Melamed R, Trejo A, Ornatsky OI, Balderas RS, Plevritis SK, Sachs K, Pe'er D, Tanner SD, Nolan GP. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. Science. 2011 May 6;332(6030):687-96. doi: 10.1126/science.1198704. PMID: 21551058; PMCID: PMC3273988.

7. Borgwardt KM, Gretton A, Rasch MJ, Kriegel HP, Schölkopf B, Smola AJ. Integrating structured biological data by Kernel Maximum Mean Discrepancy. Bioinformatics. 2006 Jul 15;22(14):e49-57. doi: 10.1093/bioinformatics/btl242. PMID: 16873512.
8. Chaplin DD. Overview of the immune response. J Allergy Clin Immunol. 2010 Feb;125(2 Suppl 2):S3-23. doi: 10.1016/j.jaci.2009.12.980. PMID: 20176265; PMCID: PMC2923430.
9. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002 Jun 1;16:321-57.
10. Cheng L, Karkhanis P, Gokbag B, Liu Y, Li L. DGCyTOF: Deep learning with graphic cluster visualization to predict cell types of single cell mass cytometry data. PLoS Comput Biol. 2022 Apr 11;18(4):e1008885. doi: 10.1371/journal.pcbi.1008885. PMID: 35404970; PMCID: PMC9060369.
11. Chevrier S, Crowell HL, Zanotelli VRT, Engler S, Robinson MD, Bodenmiller B. Compensation of Signal Spillover in Suspension and Imaging Mass Cytometry. Cell Syst. 2018 May 23;6(5):612-620.e5. doi: 10.1016/j.cels.2018.02.010. Epub 2018 Mar 28. PMID: 29605184; PMCID: PMC5981006.
12. Crowell H, Zanotelli V, Chevrier S, Robinson M (2022). CATALYST: Cytometry dATa anALYSis Tools. R package version 1.20.1, https://github.com/HelenaLC/CATALYST.
13. Dillon Hammill. [Online] CytoExploreR: Interactive Analysis of Cytometry Data. R package version 1.1.0. 2021. Available at: https://github.com/DillonHammill/CytoExploreR
14. Elhmouzi-Younes J, Palgen JL, Tchitchek N, Delandre S, Namet I, Bodinham CL, Pizzoferro K, Lewis DJM, Le Grand R, Cosma A, Beignon AS. In depth comparative phenotyping of blood innate myeloid leukocytes from healthy humans and macaques using mass cytometry. Cytometry A. 2017 Oct;91(10):969-982. doi: 10.1002/cyto.a.23107. Epub 2017 Apr 26. PMID: 28444973.
15. Finck R, Simonds EF, Jager A, Krishnaswamy S, Sachs K, Fantl W, Pe'er D, Nolan GP, Bendall SC. Normalization of mass cytometry data with bead standards. Cytometry A. 2013 May;83(5):483-94. doi: 10.1002/cyto.a.22271. Epub 2013 Mar 19. PMID: 23512433; PMCID: PMC3688049.
16. Franck D. [online] Stack Exchange. What is the trade-off between batch size and number of iterations to train a neural network?. 2017 April. Available at : https://stats.stackexchange.com/q/236393.
17. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. The journal of machine learning research. 2016 Jan 1;17(1):2096-30.
18. Geanon D, Lee B, Gonzalez-Kozlova E, Kelly G, Handler D, Upadhyaya B, Leech J, De Real RM, Herbinet M, Magen A, Del Valle D, Charney A, Kim-Schulze S, Gnjatic S, Merad M, Rahman AH. A streamlined whole blood CyTOF workflow defines a circulating immune cell signature of COVID-19. Cytometry A. 2021 May;99(5):446-461. doi: 10.1002/cyto.a.24317. Epub 2021 Feb 16. PMID: 33496367; PMCID: PMC8013522.
19. Hawley D, Ding J, Thotakura S, Haskett S, Aluri H, Kublin C, Michel A, Clapisson L, Mingueneau M, Zoukhri D. RNA-Seq and CyTOF immuno-profiling of regenerating lacrimal glands identifies a novel subset of cells expressing muscle-related proteins. PLoS One. 2017 Jun 29;12(6):e0179385. doi: 10.1371/journal.pone.0179385. PMID: 28662063; PMCID: PMC5491009.
20. Harsh G. [online] Medium. Evaluating Multi-Class Classifiers. 2029 Jan. Available at: https://medium.com/apprentice-journal/evaluating-multi-class-classifiers-12b2946e755b
21. Harsh M. [online] Towards data science. What is Domain Adaptation . 2021 Jan. Available at: https://towardsdatascience.com/understanding-domain-adaptation-5baa723ac71f
22. Jason B. [online] Machine learning mastery. Machine Learning Project in Python Step-By-Step. 2019 Feb. Available at: https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

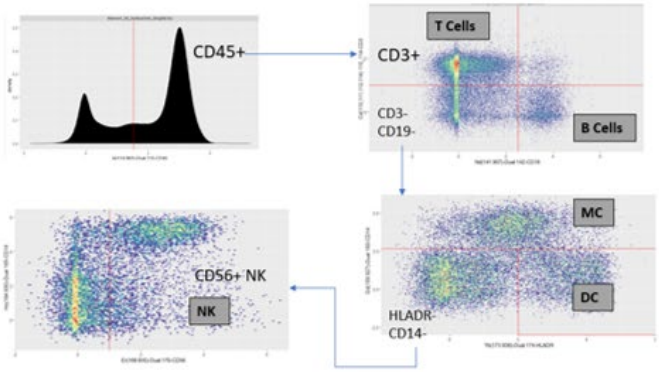
23. Jason B. [online] Machine learning mastery. What is machine learning. 2020 Feb. Available at: https://machinelearningmastery.com/what-is-deep-learning/
24. Jason B. [online] Machine learning mastery. Imbalanced Classification with the Fraudulent Credit Card Transactions Dataset. 2020 march. Available at: https://machinelearningmastery.com/what-is-deep-learning/
25. Jason B. [online] Machine learning mastery. Multi-Class Imbalanced Classification. 2020 Aug. Available at: https://machinelearningmastery.com/multi-class-imbalanced-classification/
26. Kaiser Y, Lakshmikanth T, Chen Y, Mikes J, Eklund A, Brodin P, Achour A, Grunewald J. Mass Cytometry Identifies Distinct Lung CD4+ T Cell Patterns in Löfgren's Syndrome and Non-Löfgren's Syndrome Sarcoidosis. Front Immunol. 2017 Sep 12;8:1130. doi: 10.3389/fimmu.2017.01130. PMID: 28955342; PMCID: PMC5601005.
27. Kimball AK, Oko LM, Bullock BL, Nemenoff RA, van Dyk LF, Clambey ET. A Beginner's Guide to Analyzing and Visualizing Mass Cytometry Data. J Immunol. 2018 Jan 1;200(1):3-22. doi: 10.4049/jimmunol.1701494. PMID: 29255085; PMCID: PMC5765874.
28. Kim HY, DeKruyff RH, Umetsu DT. The many paths to asthma: phenotype shaped by innate and adaptive immunity. Nat Immunol. 2010 Jul;11(7):577-84. doi: 10.1038/ni.1892. Epub 2010 Jun 18. PMID: 20562844; PMCID: PMC3114595.
29. Kotecha N, Krutzik PO, Irish JM. Web-based analysis and publication of flow cytometry experiments. Curr Protoc Cytom. 2010 Jul;Chapter 10:Unit10.17. doi: 10.1002/0471142956.cy1017s53. PMID: 20578106; PMCID: PMC4208272.
30. Lee HC, Kosoy R, Becker CE, Dudley JT, Kidd BA. Automated cell type discovery and classification through knowledge transfer. Bioinformatics. 2017 Jun 1;33(11):1689-1695. doi: 10.1093/bioinformatics/btx054. PMID: 28158442; PMCID: PMC5447237.
31. Levine JH, Simonds EF, Bendall SC, Davis KL, Amir el-AD, Tadmor MD, Litvin O, Fienberg HG, Jager A, Zunder ER, Finck R, Gedman AL, Radtke I, Downing JR, Pe'er D, Nolan GP. Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. Cell. 2015 Jul 2;162(1):184-97. doi: 10.1016/j.cell.2015.05.047. Epub 2015 Jun 18. PMID: 26095251; PMCID: PMC4508757.
32. Li H, Shaham U, Stanton KP, Yao Y, Montgomery RR, Kluger Y. Gating mass cytometry data by deep learning. Bioinformatics. 2017 Nov 1;33(21):3423-3430. doi: 10.1093/bioinformatics/btx448. PMID: 29036374; PMCID: PMC5860171.
33. Liu X, Song W, Wong B.Y, et al. A comparison framework and guideline of clustering methods for mass cytometry data. Genome Biol 20, 297 (2019). https://doi.org/10.1186/s13059-019-1917-7.
34. Liu P, Liu S, Fang Y, Xue X, Zou J, Tseng G, Konnikova L. Recent Advances in Computer-Assisted Algorithms for Cell Subtype Identification of Cytometry Data. Front Cell Dev Biol. 2020 Apr 28;8:234. doi: 10.3389/fcell.2020.00234. PMID: 32411698; PMCID: PMC7198724.
35. Maecker HT, Harari A. Immune monitoring technology primer: flow and mass cytometry. J Immunother Cancer. 2015 Sep 15;3:44. doi: 10.1186/s40425-015-0085-x. PMID: 26380089; PMCID: PMC4570613.
36. Maecker HT, McCoy JP, Nussenblatt R. Standardizing immunophenotyping for the Human Immunology Project. Nat Rev Immunol. 2012 Feb 17;12(3):191-200. doi: 10.1038/nri3158. Erratum in: Nat Rev Immunol. 2012 Jun;12(6):471. PMID: 22343568; PMCID: PMC3409649.
37. Mandy, Chris. [online] Deeplizard. Deep Learning Fundamentals. 2017. Available at: https://deeplizard.com/learn/playlist/PLZbbT5o_s2xq7LwI2y8_QtvuXZedL6tQU
38. Mestas J, Hughes CC. Of mice and not men: differences between mouse and human immunology. J Immunol. 2004 Mar 1;172(5):2731-8. doi: 10.4049/jimmunol.172.5.2731. PMID: 14978070.
39. McKinnon KM. Flow Cytometry: An Overview. Curr Protoc Immunol. 2018 Feb 21;120:5.1.1-5.1.11. doi: 10.1002/cpim.40. PMID: 29512141; PMCID: PMC5939936.

40. Newburger PE, Dale DC. Evaluation and management of patients with isolated neutropenia. Semin Hematol. 2013 Jul;50(3):198-206. doi: 10.1053/j.seminhematol.2013.06.010. PMID: 23953336; PMCID: PMC3748385.
41. Oetjen KA, Lindblad KE, Goswami M, Gui G, Dagur PK, Lai C, Dillon LW, McCoy JP, Hourigan CS. Human bone marrow assessment by single-cell RNA sequencing, mass cytometry, and flow cytometry. JCI Insight. 2018 Dec 6;3(23):e124928. doi: 10.1172/jci.insight.124928. PMID: 30518681; PMCID: PMC6328018.
42. Olsen LR, Leipold MD, Pedersen CB, Maecker HT. The anatomy of single cell mass cytometry data. Cytometry A. 2019 Feb;95(2):156-172. doi: 10.1002/cyto.a.23621. Epub 2018 Oct 2. PMID: 30277658.
43. Palgen JL, Tchitchek N, Elhmouzi-Younes J, Delandre S, Namet I, Rosenbaum P, Dereuddre-Bosquet N, Martinon F, Cosma A, Lévy Y, Le Grand R, Beignon AS. Prime and Boost Vaccination Elicit a Distinct Innate Myeloid Cell Immune Response. Sci Rep. 2018 Feb 15;8(1):3087. doi: 10.1038/s41598-018-21222-2. PMID: 29449630; PMCID: PMC5814452.
44. Pedersen CB, Dam SH, Barnkob MB, Leipold MD, Purroy N, Rassenti LZ, Kipps TJ, Nguyen J, Lederer JA, Gohil SH, Wu CJ, Olsen LR. cyCombine allows for robust integration of single-cell cytometry datasets within and across technologies. Nat Commun. 2022 Mar 31;13(1):1698. doi: 10.1038/s41467-022-29383-5. PMID: 35361793; PMCID: PMC8971492.
45. Quintelier K, Couckuyt A, Emmaneel A et al. Analyzing high-dimensional cytometry data using FlowSOM. Nat Protoc 16, 3775–3801 (2021). https://doi.org/10.1038/s41596-021-00550-0
46. Rybakowska P, Alarcón-Riquelme ME, Marañón C. Key steps and methods in the experimental design and data analysis of highly multi-parametric flow and mass cytometry. Comput Struct Biotechnol J. 2020 Mar 31;18:874-886. doi: 10.1016/j.csbj.2020.03.024. PMID: 32322369; PMCID: PMC7163213.
47. Rybakowska P, Van Gassen S, Quintelier K, Saeys Y, Alarcón-Riquelme ME, Marañón C. Data processing workflow for large-scale immune monitoring studies by mass cytometry. Comput Struct Biotechnol J. 2021 May 21;19:3160-3175. doi: 10.1016/j.csbj.2021.05.032. PMID: 34141137; PMCID: PMC8188119.
48. Samusik N, Good Z, Spitzer M, et al. Automated mapping of phenotype space with single-cell data. Nat Methods 13, 493–496 (2016). https://doi.org/10.1038/nmeth.3863
49. Spidlen J, Breuer K, Rosenberg C, Kotecha N and Brinkman RR. (2012), FlowRepository: A resource of annotated flow cytometry datasets associated with peer-reviewed publications. Cytometry, 81A: 727-731. https://doi.org/10.1002/cyto.a.22106.
50. Spidlen J, Moore W, Parks D, Goldberg M, Bray C, Bierre P, Gorombey P, Hyun B, Hubbard M, Lange S, Lefebvre R, Leif R, Novo D, Ostruszka L, Treister A, Wood J, Murphy RF, Roederer M, Sudar D, Zigon R, Brinkman RR. Data File Standard for Flow Cytometry, version FCS 3.1. Cytometry A. 2010 Jan;77(1):97-100. doi: 10.1002/cyto.a.20825. PMID: 19937951; PMCID: PMC2892967.
51. Tara B. [online] Towards data science. Dealing with Imbalanced Data. 2019 Feb. Available at: https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18
52. Trussart M, Teh CE, Tan T, Leong L, Gray DH, Speed TP. Removing unwanted variation with CytofRUV to integrate multiple CyTOF datasets. Elife. 2020 Sep 7;9:e59630. doi: 10.7554/eLife.59630. PMID: 32894218; PMCID: PMC7500954.
53. Van G.S, Callebaut B, Van H.M.J, Lambrecht B.N, Demeester P, Dhaene T, and Saeys Y. (2015), FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry, 87: 636-645. https://doi.org/10.1002/cyto.a.22625.
54. Van Unen V, Li N, Molendijk I, Temurhan M, Höllt T, van der Meulen-de Jong AE, Verspaget HW, Mearin ML, Mulder CJ, van Bergen J, Lelieveldt BP, Koning F. Mass Cytometry of the Human Mucosal Immune System Identifies Tissue- and Disease-Associated Immune Subsets.

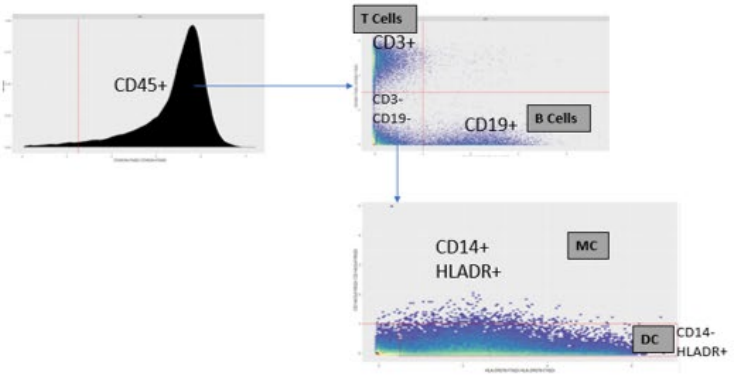
Immunity. 2016 May 17;44(5):1227-39. doi: 10.1016/j.immuni.2016.04.014. Epub 2016 May 10. PMID: 27178470.
55. Vinícius T. [online] Towards data science. Multiclass classification evaluation with ROC Curves and ROC AUC . 2012 Feb. Available at: https://towardsdatascience.com/multiclass-classification-evaluation-with-roc-curves-and-roc-auc-294fd4617e3a
56. Weber LM, Nowicka M, Soneson C, & Robinson MD. diffcyt: Differential discovery in high-dimensional cytometry via high-resolution clustering. Commun Biol 2, 183 (2019). https://doi.org/10.1038/s42003-019-0415-5.
57. Weber LM, Robinson MD. Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. Cytometry A. 2016 Dec;89(12):1084-1096. doi: 10.1002/cyto.a.23030. Epub 2016 Dec 19. PMID: 27992111.
58. Weber LM, Soneson, Charlotte. "HDCytoData: Collection of high-dimensional cytometry benchmark datasets in Bioconductor object formats." 2019. F1000Research, 8(v2), 1459.
59. Zunder ER, Finck R, Behbehani GK, Amir el-AD, Krishnaswamy S, Gonzalez VD, Lorang CG, Bjornson Z, Spitzer MH, Bodenmiller B, Fantl WJ, Pe'er D, Nolan GP. Palladium-based mass tag cell barcoding with a doublet-filtering scheme and single-cell deconvolution algorithm. Nat Protoc. 2015 Feb;10(2):316-33. doi: 10.1038/nprot.2015.020. Epub 2015 Jan 22. PMID: 25612231; PMCID: PMC4347881.

Supplimentary Material

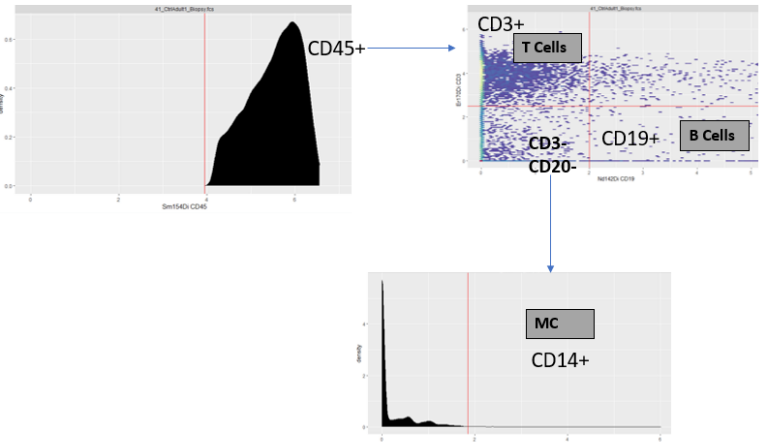
Human BM 11 (Levine JH et al, 2015)



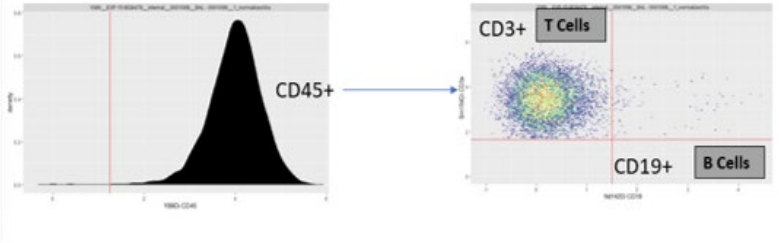
Human BM(Bendall SC et al,2011)



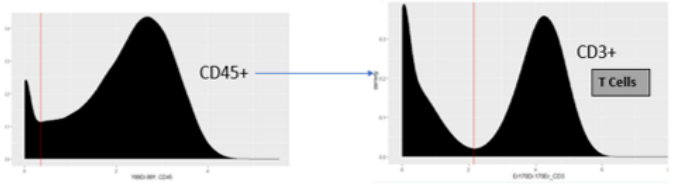
Human Intestine (van Unen V et al,2016)



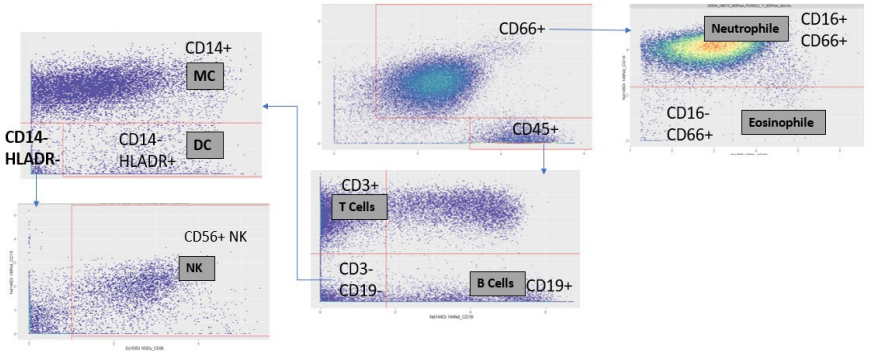
Human Lung (Kaiser Y et al 2017)



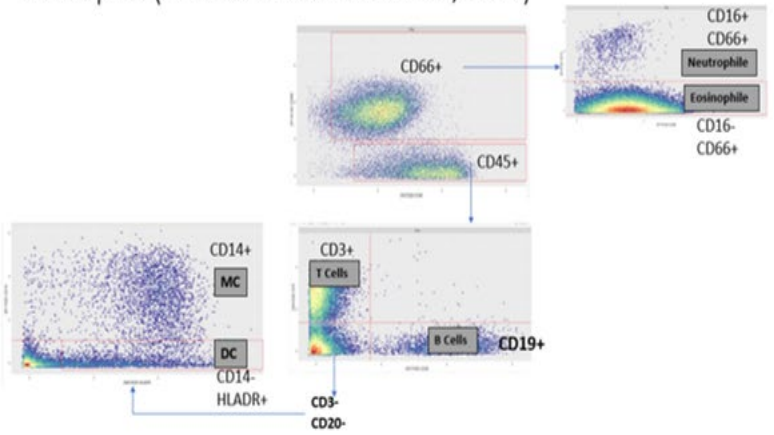
Human BM (Oetjen KA et al,2018)



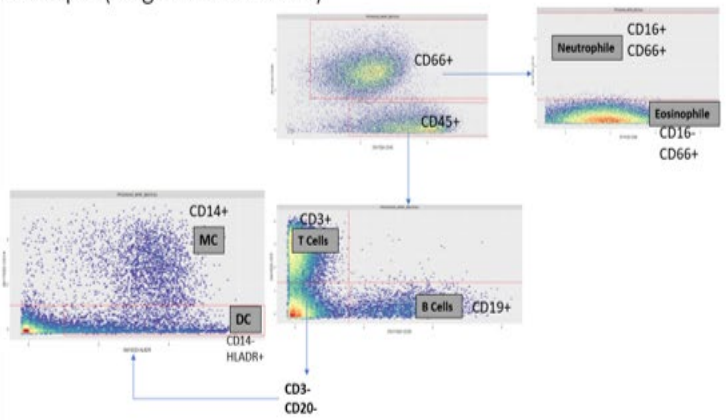
Human WB (Geanon D et al, 2020)



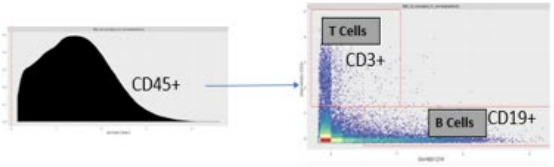
Macaques (Elhmouzi-Younes J et al,2017)



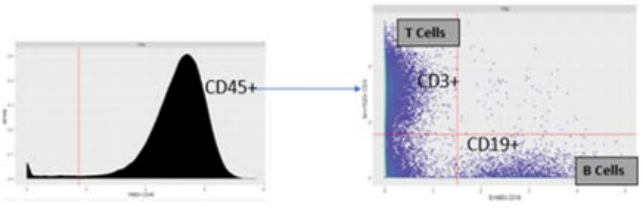
Macaque (Palgen JL et al 2019)



Mice BM (Samusik N et al,2016)



Mice Lung (Kimball AK et al,2018)

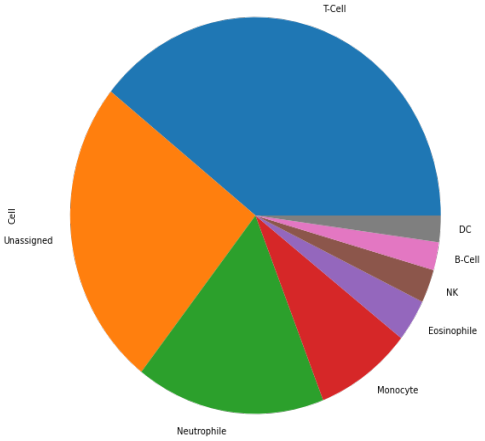


Mice Lacrimal Gland (Hawley D et al, 2017)

Supplementary S1: Outlines the gating strategy used for all studies. Plots are shown for a single example. The gating was done separately for all 75 samples. Any missing markers and their associated cells were not gated for.

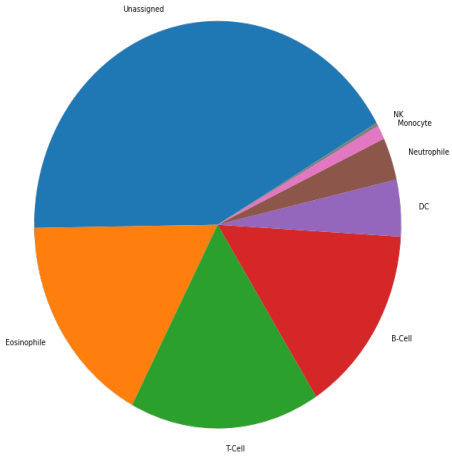
Marker	HUM WB	HUM PBMC	HUM BM 11	HUM BM 15	HUM BM 18	HUM LUNG	HUM INTESTINE	MICE LUNG	MICE LG	MICE BM	MAC 2018	MAC 2021
CD66												
HLADR												
CD3												
CD8												
CD4												
CD16												
CD14												
CD45												
CD19												
CD27												
CD56												

Supplementary S2: Shows the markers the required markers that were present in the study. Boxes with green highlights means the markers were present, and red means not present. HUM : Human, and MAC : Macaques. The year is provided to differentiate any studies coming from the same species and sample.



	Train
B-Cell	285374
Dendritic Cell	270932
Eosinophile	430788
Monocyte	1058238
Natural Killer	341458
Neutrophile	2068198
T-Cell	4918627
Unassigned	3156079

	Test
B-Cell	267281
Dendritic Cell	80077
Eosinophile	302538
Monocyte	19859
Natural Killer	4804
Neutrophile	59976
T-Cell	297282
Unassigned	744756



Supplementary S3: Shows the distribution of full training dataset on a pie chart on the left with the values in the Train table. On the right shows the distribution of the test dataset and its associated values.

	DeepCyTOF	ImmunoPred	Baseline-Model
B-Cell	0.12	0.56	0.43
Dendritic Cell	0.41	0.41	0.72
Eosinophile	0.71	0.69	0.86
Monocyte	0.47	0.65	0.78
Natural Killer	0.61	0.70	0.85
Neutrophile	0.53	0.77	0.72
T-Cell	0.81	0.97	0.94
Unassigned	0.84	0.64	0.83

Supplementary S4: Shows the f1 scores of each class during initial model exploration.

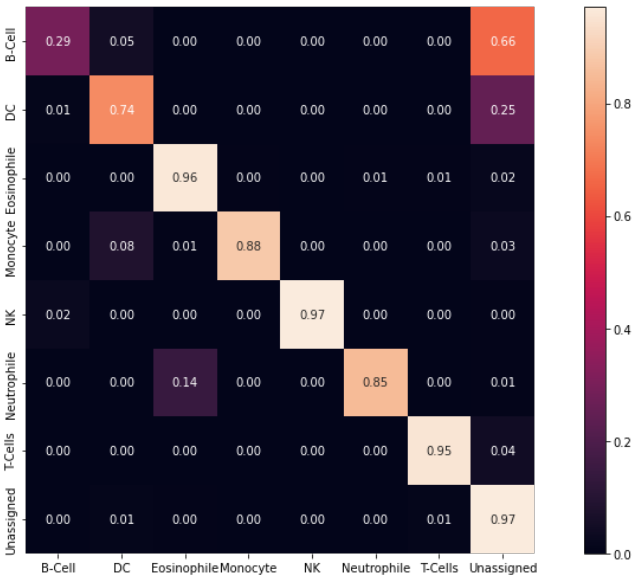
	Parameters	Score
2	Optimiser : Adam, Learning_rate :0.001,Dropout: 0.3, L2: 0.01 epochs : 30, BatchSize : 4096	0.78125
3	Optimiser : Adam, Learning_rate :0.001,Dropout: 0.3, L2: 0.01 epochs : 30, BatchSize : 2046	0.78034
4	Optimiser : Adam, Learning_rate :0.001,Dropout: 0.5, L2: 0.01 epochs : 30, BatchSize : 4096	0.77937
5	Optimiser : RMSprop, Learning_rate :0.001,Dropout: 0.3, L2: 0.01 epochs : 30, BatchSize : 4096	0.77125
6	Optimiser : RMSprop, Learning_rate :0.001,Dropout: 0.3, L2: 0.01 epochs : 30, BatchSize : 2096	0.77005

Supplementary S5: Shows the top 5 examples of scores. Values are printed as a standard output in Jupyternotebook, which were saved to csv file for visualisation.

	Unmodified	Under Sampling (Minority = DC)	SMOTE (10% of Majority = TC)
B-Cell	285374	270932	SMOTE = 491,862
Dendritic Cell	270932	270932	SMOTE = 491,862
Eosinophile	430788	270932	SMOTE = 491,862
Monocyte	1058238	270932	UnderSampled = 491,862
Natural Killer	341458	270932	SMOTE = 491,862
Neutrophile	2068198	270932	UnderSampled = 491,862
T-Cell	4918627	270932	UnderSampled = 491,862
Unassigned	3156079	270932	UnderSampled = 491,862

Supplementary S6: Shows the strategies when correcting for data imbalance. The under-sampling method used the minority class Dendritic cell to down sample the remaining class. SMOTE used a mix of synthetic generation and underdamping. The majority class was T cell with 10% of datapoint (491,862) used as threshold. If a class is below the threshold, new synthetic data was generated marked by having "SMOTE" in the table value. When the threshold was exceeded, the class was under sampled, marked by the "UnderSsampled"

A

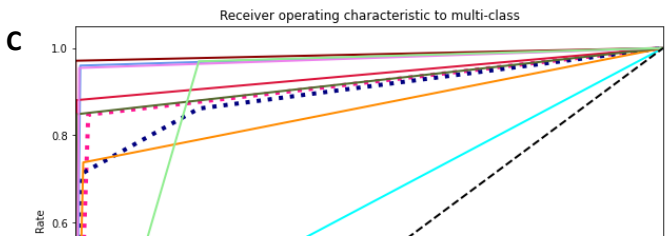


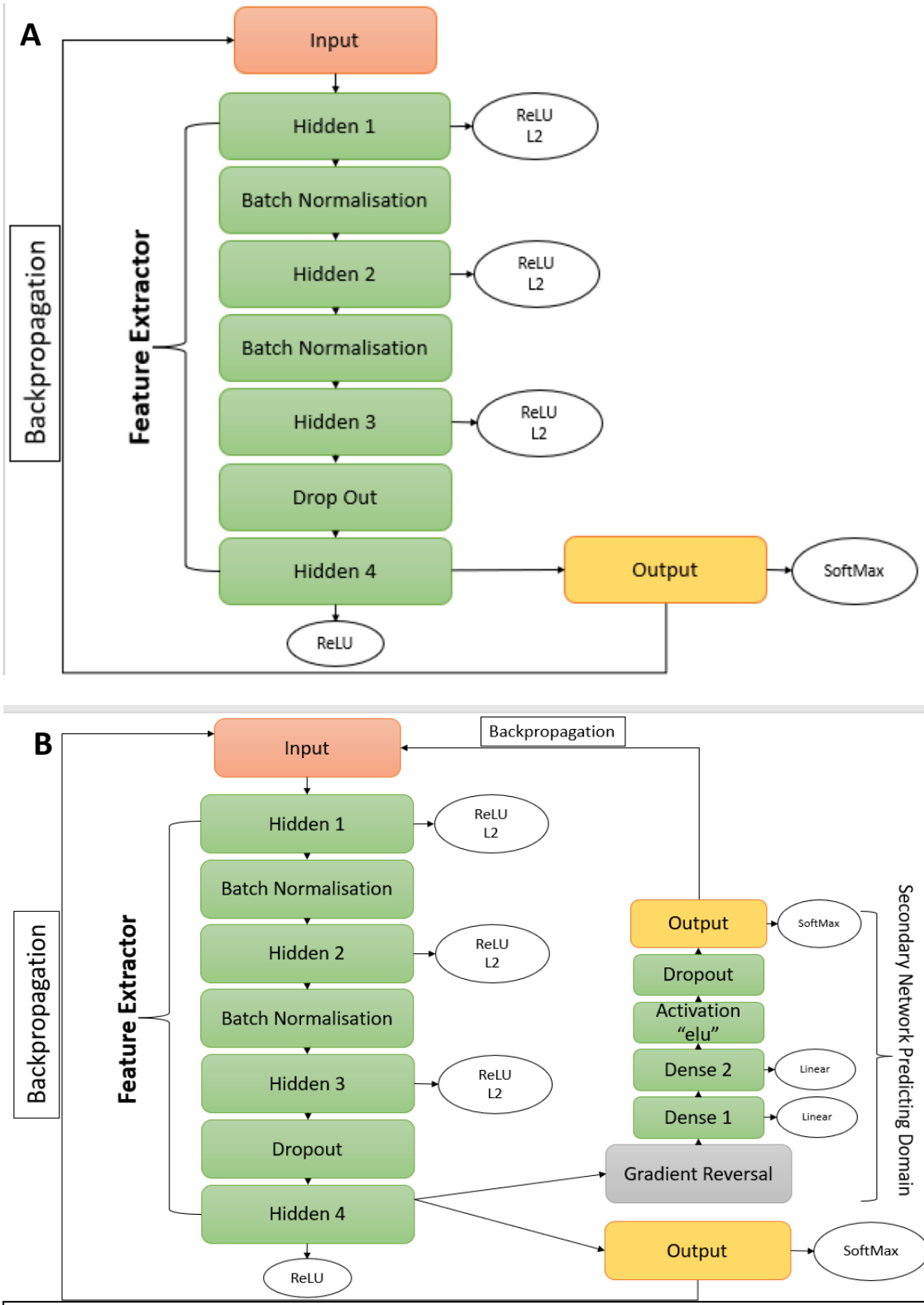
B

	Precision	Recall	F1
B-Cell	0.96	0.29	0.44
Dendritic Cell	0.72	0.74	0.73
Eosinophile	0.96	0.96	0.96
Monocyte	0.85	0.88	0.87
Natural Killer	0.89	0.97	0.93
Neutrophile	0.94	0.85	0.89
T-Cell	0.97	0.95	0.96
Unassigned	0.77	0.97	0.86

Accuracy = 0.85			
Macro F1			0.83
Weighted F1			0.83

C





Supplementary S8: A) Shows the architecture layout of the final model, with ovals indicating the activation and regularisation each layer undergoes. **B)** Shows the architecture of the Domain adversarial neural networks with a secondary network attached to the final feature extractor layer. This network was adopted from Planche B & Andres ET (2019)

Software/ Package	Version	Link
R	4.2	https://www.r-project.org/
Python	3.9.11	https://www.python.org/
Bioconductor	3.15	https://bioconductor.org/
HDCytoDATA	1.16.0	https://bioconductor.org/packages/release/data/experiment/html/HDCytoData.html
CATALYST	1.20.1	https://bioconductor.org/packages/release/bioc/html/CATALYST.html
Ggplot2	3.3.5	https://ggplot2.tidyverse.org/
ggcyto	1.24.1	https://bioconductor.org/packages/release/bioc/html/ggcyto.html
cyCombine	0.2.5	https://github.com/biosurf/cyCombine
Anaconda	2.1.0	https://www.anaconda.com/
JupyterNotebook	6.4.0	Provided by Anaconda
TensorFlow	2.9.1	https://www.tensorflow.org/
Pandas	1.4.3	https://pandas.pydata.org/
NumPy	1.23.0	https://numpy.org/
Scikit-learn	1.1.1	https://scikit-learn.org/stable/
SciKeras	0.9.0	https://www.adriangb.com/scikeras/stable/
hypopt	1.0.9	https://github.com/cgnorthcutt/hypopt
Matplotlib	3.5	https://matplotlib.org/
Seaborn	0.11.2	https://seaborn.pydata.org/