Machine learning and statistical analysis for diagnosis of hematological diseases

Author: Pol Benítez Colominas.
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.[*]

Advisor: Aurora Hernández-Machado

Abstract: Blood is a biological fluid composed mainly of water, red blood cells and other components and it is a non-Newtonian fluid. Red blood cells play an important role in the rheological properties of the blood and are the main responsible for the shear thinning behaviour of blood. Some hematological diseases can change the geometrical shape of red blood cells and thus their viscosity. In this work we have computed the viscosity of different blood samples that were obtained with a microfluidic device and normalized the viscosities for hematocrit using statistical analysis tools. We have also used different machine learning methods as Logistic Regressions or Artificial Neural Networks (ANN) to predict if a sample of blood corresponds to healthy blood or to a blood with an hematological disease. We have obtained different performance for the different methods, some of them with very good results and an accuracy of 94% of correct prediction has been achieved with an ANN model.

I. INTRODUCTION

We want to apply machine learning methods to a set of measures of different blood samples. Different methods will be used in order to test their performance to our data [1,2].

A microfluidic device has been used to generate the data that we need from different blood donors. This device consists in a microfluidic channel with a rectangular section connected to a pump that send some fluid to a given pressure. The fluid travels along the channel and a set of electrodes (distributed in four groups of six electrodes each group) can detect the advancement of the fluid front [3]. In figure 1 we can see the device.

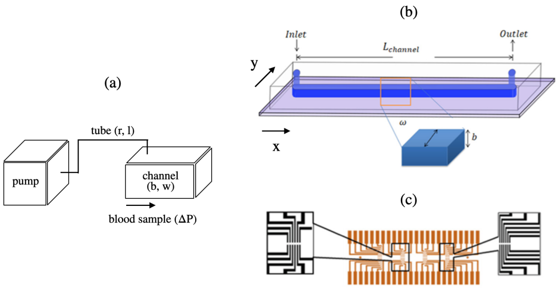

FIG. 1: Schematic representation of the principal components of the microfluidic device. (a) Pump connected to the channel by a micro tube with length l_t and radius r. (b) Microfluidic channel, with width w and height b, through which blood advances with ΔP [3]. (c) Pattern of electrodes printed on the glass substrate that work as switches and detect fluid front advancement [3].

We can record the time that fluid front passes for an electrode and thus to know the dynamics of the fluid inside the channel. If we solve the Navier-Stokes equation for the geometry of the microfluidic channel we can know the viscosity for a Newtoninan fluid, and if we use more than one pressure to pump the sample we can get a curve of viscosity for shear rate for a non-Newtonian fluid. Thus we can use the microdevice as a rheometer.

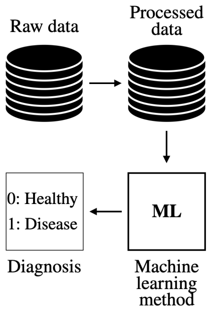

FIG. 2: Simplified diagram of the process where the principal steps are shown. We start with raw data generated by the microfluidic device and finish with a diagnosis.

The data set obtained with the microfluidic device and from anonymous donors has measures from 274 donors (some of them healthy and the rest have an hematological disease), and for every donor, we have between 5 and 10 data rows corresponding to different pressures, in order to get a viscosity curve. For every measure we have 24 times (related to the electrode time detection), a pressure, the hematocrit concentration, and other values containing information about the acquisition process. So in total our initial data set has 2018 rows and 37 columns.

Our objective (we can see a diagram of the process in figure 2) is to process this raw data set in order to get n and m coefficients that gives us a viscosity curve and then to normalize this coefficient for hematocrit to get n_{htc} and m_{htc}. With this processed data we want to use a machine learning model to automatizes the diagnosis of hematological disease.

[*] Electronic address: pbenitco7@alumnes.ub.edu/polbeco@gmail.com

II. THEORETICAL MODEL AND VISCOSITY COMPUTATION

A. Newtonian and non-Newtonian fluid behaviour in the microfluidic device

We can use the Navier-Stokes equation and an expression that relates viscosity, η, with shear rate, $\dot{\gamma} \equiv \frac{\partial v_x}{\partial z}$, to obtain the viscosity coefficients n and m [4-6].

For the relation $\eta\left(\dot{\gamma}\right)$ we can use the power-law model:

$$\eta\left(\dot{\gamma}\right) = m\dot{\gamma}^{n-1}, \tag{1}$$

that is suitable for both cases, Newtonian fluids, $n = 1$, and non-Newtonian fluids, $n \neq 1$. For non-Newtonian fluids we can distinguish two behaviours depending on the n value, for $n < 1$ we have shear-thinning behaviour, and for $n > 1$ we have shear-thickening behaviour. Blood presents a shear-thinning behaviour [6].

If we use both expressions (1) and Navier-Stokes equation for the boundary conditions of our microdevice we will get [5,6]:

$$\Delta P = A(n, m)\dot{h}^n, \tag{2}$$

where ΔP contains all the pressures involved in the process, $\dot{h}$ is the mean velocity of the fluid front on the channel, and $A(n, m)$ is a parameter that depends on the geometry of the system and the power-law model parameters n and m. For our system $A(n, m)$ can be expressed as [6]:

$$A(n, m) = m\frac{2l_t\left(\frac{1}{n}+3\right)^n}{r^{n+1}}\left(\frac{bw}{\pi r^2}\right)^n, \tag{3}$$

where l_t is the length of the tube that connects the pump with the channel, r is the radius of this tube, w is the width of the channel and b is the height of the channel.

B. Blood and plasma viscosity determination

Using expressions (2) and (3) we can get n and m if we know the other variables (are provided by the microdevice).

For Newtonian fluids we have $n = 1$ and with one measure we can get the viscosity of the fluid:

$$\eta = m = \frac{\Delta P \pi r^4}{8\dot{h}l_t wb}. \tag{4}$$

We can use this expression to compute the viscosity of the plasma (blood without red cells) since it behaves as Newtonian fluid.

For non-Newtonian fluids we need more than one measure in order to determine n and m. If we have more than one point $\left(\Delta P, \dot{h}\right)$, and using (2) and (3), we can do a linear regression $\ln\left(\Delta P\right) = \alpha \ln \dot{h} + \beta$, and with α and β obtained we can get the power-law model parameters:

$$\begin{cases} n = \alpha \\ m = e^{\beta}\frac{r^{n+1}}{2l_t\left(\frac{1}{n}+3\right)^n}\left(\frac{\pi r^2}{bw}\right)^n \end{cases}. \tag{5}$$

With these we can determine the viscosity curve for non-Newtonian fluid.

C. Blood viscosity normalization

In the last subsection we have studied how to get viscosity curves for our blood samples, but in order to compare between different viscosity curves it would be interesting to normalize by hematocrite (percentage of red cells) since we have samples of blood with different concentration of red cells.

We propose the following model [7]:

$$\eta_{htc} = 1 + \left(\frac{\eta}{\eta_p} - 1\right)\frac{\phi_{control}}{\phi}, \tag{6}$$

where η_{htc} is the normalized viscosity, η is the viscosity we want to normalize, η_p is the viscosity of the plasma, $\phi \in [0, 1]$ is the concentration of hematocrite, and $\phi_{control} = \max\left(\phi_i\right)$, i.e. the maximum concentration of red cells from a sample present in our data.

We are interested in obtaining the parameters of power-law model for the normalized situation, thus if we have $\eta_{htc} = m_{htc}\dot{\gamma}^{n_{htc}-1}$ and we have points $\left(\eta_{htc}, \dot{\gamma}\right)$ we can do a linear regression $\ln\left(\eta_{htc}\right) = \alpha \ln \dot{\gamma} + \beta$, so that:

$$\begin{cases} n_{htc} = \alpha + 1 \\ m_{htc} = e^{\beta} \end{cases}. \tag{7}$$

III. MACHINE LEARNING METHODS

In this section we describe briefly the different machine learning methods that we will use to automatize the diagnosis of hematological diseases.

We suppose that exists a function $f(X)$ that given the rheological data of our blood samples, X, is capable to classify as healthy blood, 0, or hematological disease, 1. Thus:

$$f(X) \mapsto \{0, 1\}. \tag{8}$$

We can use machine learning methods [1,2] to estimate this function and get $\hat{f}(X) \equiv \hat{Y}$, so that $Y = \hat{Y} + \varepsilon$, where we want ε to be as small as possible in order to get a model that classifies samples with the minimum number of errors. In our case ε is related with classifying incorrectly the samples in 0 or 1.

To do that we can define [1] an error function $E\left(Y - \hat{Y}\right)$ that gives us bigger values when $\hat{Y}$ is less similar to Y. The purpose of the machine learning method is to minimize this function and thus to obtain a good $\hat{Y}$.

If we have a data set with (Y, X), i.e. we know the inpunt and output data, we can use supervised methods [1,2], where we split the data set into train set that is used to train the model and to get $\hat{f}(X)$, and test set that is used to verify if $\hat{f}(X)$ is a good prediction of $f(X)$ with data never seen in the training in order to avoid overfitting.

Now we can describe in a few words, some classification models that can help to solve our problem.

A. Logistic Regression

We expect $\hat{Y}$ can only take two values 0 or 1, so we can try to model a sigmoid function as it saturates in 0 and 1. So we can use [1]:

$$\hat{Y} = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}, \tag{9}$$

and find β_0 and β_1 that minimize the error function, E.

B. K-Nearest Neighbors

The idea of this method [1] is to classify an unknown X the same class that belongs to their k nearest neighbors. So if we want to classify X_x, $Y\left(X_x\right)$ will be the Y more frequent in $Y\left(X_1\right)$, $Y\left(X_2\right)$,..., $Y\left(X_k\right)$, where $Y\left(X_i\right)$ for $i = 1, ..., k$ are known.

C. Classification Tree and Random Forest

In classification tree model [1] the method splits the data to predict asking diatomic questions about X (for example, if X is bigger than 1), until it is capable to assign the correct class to the maximum amount of splits.

Random forest model [1] generates a big number of classification trees simultaneously and assigns the class most predicted by the trees.

D. Deep Learning. Artificial Neural Networks

The minimum element of a neural network [1,2] is a perceptron or neuron, that receives n inputs, x_i, and add them with weights, w_i, and a bias, b, so $\sum_{i=1}^{n} w_i x_i + b$. To allow non-linear behaviour an activation function, σ, is used, and there are some functions specially interesting for this [2] like sigmoid function or rectified linear unit function. The output from one perceptron will be:

$$\sigma\left(\Sigma\right) = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right). \tag{10}$$

To create an artificial neural network (ANN) [2] we only have to create layers of neurons (the neurons in the same layer are not connected between them) that receive inputs and every one of them generates and output. The first layers of neurons receive the inputs from X and the following layers receive the outputs generated by the previous layer of neurons. The last layer is the output layer that gives us $\hat{Y}(X)$. If we have more than one layer of neurons, we talk about deep learning [1,2].

The ANN has to adjust w_i and b_i in order to minimize the error function E. Due the big size of the ANNs, backpropagation algorithm [2] is used adjust the weights and the biases.

IV. RESULTS

A. Viscosity curves

We have samples of blood from 274 donors (every donor with more than one measure for different pressures) with eight different conditions: healthy blood, iron-deficiency anemia, beta thalassemia, hereditary spherocytosis, sickle cell, vitamin B12 deficiency, mechanical hemolysis and hepatic cirrhosis.

117 samples correspond to blood with some concentration of red cells and the rest correspond to plasma for the different conditions.

TABLE I: Mean viscosity value of plasma for healthy blood and for each disease. Plasma is a Newtonian fluid so its viscosity is constant for shear rate.

Disease	$\bar{\eta}\ [mPa \cdot s]$
Healthy blood	2.2 ± 0.5
Iron-deficiency anemia	2.3 ± 1.0
Beta thalassemia	2.1 ± 0.3
Hereditary spherocytosis	1.79 ± 0.08
Sickle cell	2.0 ± 0.7
Vitamin B12 deficiency	2.1 ± 0.4
Mechanical hemolysis	1.82 ± 0.15
Hepatic cirrhosis	2.4 ± 0.6

Using equation (4) we can obtain the viscosity for plasma, and we can observe the results in table I. We can see that all the values are similar and around 2 $mPa \cdot s$, as we expect because we suppose that rheological differences between diseases are consequence of red cells.

Now we can use equation (5) to obtain n and m. With equation (7), the viscosities of plasma showed in table I, and the n and m coefficients that we have just computed we can obtain n_{htc} and m_{htc}.

We can observe the results in table II. We have plotted the viscosity curves for each disease in the same scale to compare. In figure 3 we have viscosity curves for n and m, and in figure 4 we have normalized viscosity curves obtained with n_{htc} and m_{htc}.

In this work we are interested in be able to distinguish between healthy blood and non-healthy blood, so in ta-

TABLE II: Mean values of n, m, n_{htc} and m_{htc} for the different diseases with their respective errors.

Disease	$\bar{n}$	$10^3 \cdot \bar{m}$	$\bar{n}_{htc}$	$\bar{m}_{htc}$
Healthy blood	0.82 ± 0.11	19 ± 15	0.81 ± 0.11	12 ± 9
Iron-deficiency	0.88 ± 0.12	10 ± 20	0.87 ± 0.12	7 ± 9
Beta thalassemia	0.88 ± 0.06	10 ± 5	0.88 ± 0.06	6 ± 3
H. spherocytosis	0.91 ± 0.04	11 ± 7	0.90 ± 0.03	7 ± 2
Sickle cell	0.98 ± 0.02	4.7 ± 0.3	0.976 ± 0.19	2.5 ± 0.3
B12 deficiency	0.88 ± 0.6	11 ± 4	0.87 ± 0.07	6 ± 2
M. hemolysis	0.92 ± 0.03	6.2 ± 1.3	0.92 ± 0.03	3.7 ± 0.9
Hepatic cirrhosis	0.92 ± 0.04	8.8 ± 1.3	0.92 ± 0.04	4.4 ± 0.7

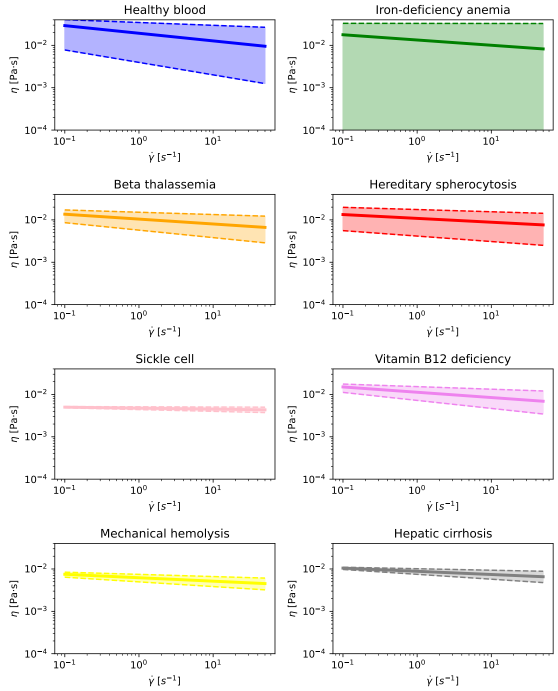

FIG. 3: Viscosity curves for healthy blood and different diseases with a logarithmic scale. The thick straight line is the viscosity curve for n and m mean value, and the striped straight lines are viscosity curves for n and m plus one standard deviation (top) and n and m minus one standard deviation (down).

ble 3 and figure 5 we have n, m, n_{htc}, m_{htc} and normalized viscosity curves but now for healthy blood and for non-healthy blood that merge the values for the different diseases.

TABLE III: Mean values of n, m, n_{htc} and m_{htc} for healthy blood and for non-healthy blood with their respective errors.

Disease	$\bar{n}$	$10^3 \cdot \bar{m}$	$\bar{n}_{htc}$	$\bar{m}_{htc}$
Healthy blood	0.82 ± 0.11	19 ± 15	0.81 ± 0.12	12 ± 9
Non-healthy blood	0.87 ± 0.12	13 ± 19	0.89 ± 0.08	6 ± 5

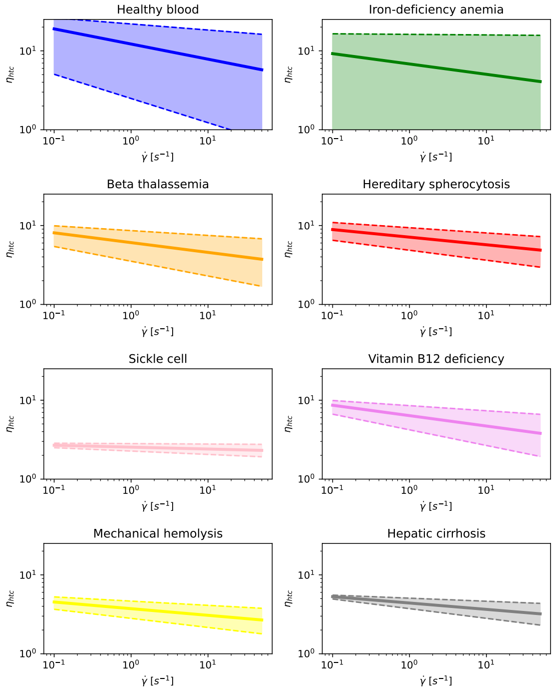

FIG. 4: Viscosity curves normalized by hematocrite for healthy blood and different diseases with a logarithmic scale. The thick straight line is the viscosity curve for n_{htc} and m_{htc} mean value, and the striped straight lines are viscosity curves for n_{htc} and m_{htc} plus one standard deviation (top) and n_{htc} and m_{htc} minus one standard deviation (down).

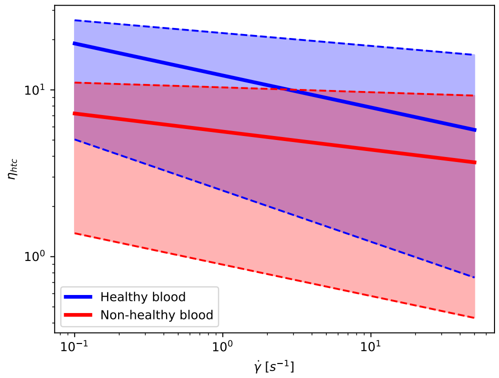

FIG. 5: Comparison between viscosity mean curve and one standard deviation normalized by hematocrit for healthy blood and for non-healthy blood in logarithmic scale.

The most important viscosity parameter is n because it gives us information about how fast or slow the shear-thinning behaviour happens for shear rate, m is a prefactor that move $\eta(\dot{\gamma} = 1)$ along the vertical axis. We can see in table III that mean n value is different for healthy or non-healthy blood and it is more clear for n_{htc}, although if we take into account the error values for healthy and non-healthy both values are compatible. Despite this, in figure 5 we can see that for small shear rate values the viscosity curves are different.

We expect that with the values of n, m, n_{htc} and m_{htc}, a machine learning method has enough information to classify a donor as healthy or non-healthy.

B. Machine learning methods performance

Finally we have applied machine learning methods to our raw and processed data. To evaluate our models we count the number of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) applying the model to a test set.

We can start applying methods to raw data. Results are showed in table IV.

TABLE IV: Machine learning performance for raw data. The raw data set has 2018 samples, train set has 1412 samples and test set has 606 samples.

ML method	TP	FP	FN	TN	Accuracy
Logistic Regression	63	182	38	323	0.64
Artificial Neural Network	121	124	94	267	0.64

Finally, in table V we can see the results for data with n, m, n_{htc} and m_{htc} using five different ML methods. The best results are obtained with an ANN that has 4 layers of 16, 8, 4 and 2 neurons, respectively, and an output layer of 1 neuron.

TABLE V: Machine learning performance for processed data, using only n, m, n_{htc} and m_{htc} coefficients for the four groups of electrodes. The processed data set has 117 samples, train set has 81 samples and test set has 36 samples.

ML method	TP	FP	FN	TN	Accuracy
Logistic Regression	14	3	2	17	0.86
K-Nearest Neighbors	13	4	10	9	0.61
Classification Tree	13	4	7	12	0.69
Random Forest	13	4	2	17	0.83
Artificial Neural Network	16	1	1	18	0.94

V. CONCLUSIONS

We have obtained the best performance for data with n, m, n_{htc} and m_{htc} and it is something that we could expect because machine learning methods have more information to learn, and because with n_{htc} and m_{htc} we can train for samples with any hematocrit concentration without taking into account the value of this concentration. We can see that not all methods have the same performance and that is something positive because means that the model is not overfitting.

With raw data we get a good accuracies (more than 50%), but we have to note that a lot of samples are classified incorrectly as FP.

So the best performance is for the normalized data using an ANN, despite logistic regression and random forest also perform with an accuracy above 80%.

Thus, it seems that is a good idea to use machine learning techniques to solve classification problems in medical sciences doing a preprocessing to our data using physics knowledge.

We can conclude that we can use the ANN model, the preprocessing necessary to get normalized viscosity coefficients and a not very big number of donors, to classify the data generated by the microdevice as healthy or non-healthy.

Acknowledgments

I would like to thank Dra. Aurora Hernández-Machado and Josep Ferré-Torres for giving me their time and the opportunity to contribute to their project. I would also like to thank my friends, my classmates and my family, especially my parents who always have trusted me and gave me all the possible support.

[1] G. James, D. Witten, et al. *An Introduction to Statistical Learning*, 2nd Ed. Springer Texts in Statistics, 2021.

[2] F. Chollet. *Deep Learning with Python*, 1st Ed. Manning Publications Co., 2018.

[3] L. Méndez-Mora, M. Cabello-Fusarés, et al. Microrheometer for Biofluidic Analysis: Electronic Detection of the Fluid-Front Advancement. *Micromachines*, 2021, 12, 726.

[4] H. Bruus. *Theoretical Microfluidics*, 1st Ed. Oxford University Press, 2007.

[5] C. Trejo-Soto, E. Costa-Miracle, et al. Capillary Filling at the Microscale: Control of Fluid Front Using Geometry. *PLOS ONE*, 2016, 11, e0153559.

[6] C. Trejo-Soto, E. Costa-Miracle, et al. Front microrheology of the non-Newtonian behaviour of blood: scaling theory of erythrocyte aggregation by aging. *Soft Matter*, 2017, 13, 3042-3047.

[7] C. Trejo-Soto and A. Hernández-Machado. Normalization of Blood Viscosity According to the Hematocrit and the Shear Rate. *Micromachines*, 2022, 13, 357.