Programming and Algorithms (M1 CMB)
Platelets subpopulation clustering

Polina Soloveva
January 2024

Table of Contents

Introduction **2**
1 Data Preprocessing **3**
1.1 Source Files 3
1.2 Data Scaling 3
1.3 Isolation of Platelet Population 3
1.3.1 Statistical Methods 3
1.3.2 Isolation Forest 4
1.3.3 DBSCAN 6
2 Clustering of Platelet Subpopulations **7**
2.1 Overview of Clustering Methods 7
2.1.1 K-means 7
2.1.2 Expectation–Maximization (EM) Algorithm 7
2.1.3 Agglomerative Clustering 8
2.1.4 DBSCAN 8
2.2 Clustering Quality Metrics 8
2.2.1 Davies-Bouldin Index (DBI) 8
2.2.2 Calinski-Harabasz (CH) Index 8
2.2.3 Silhouette Coefficient 8
2.2.4 Time 9
2.3 Clustering Quality Assessment 9
2.4 Subpopulation Means Fluorescence 11
3 Platelets Clustering Cytometry Data Script **12**
Conclusion **15**
References **15**

Introduction

To study the membrane reactions of blood coagulation on the platelet membrane, we used the method of flow fluorescence cytometry, which allows us to measure the degree of binding of fluorescently labeled coagulation factors to the cell membrane. Even though the cytometry method is familiar, using it to study platelets is fraught with some difficulties. So platelets are the smallest blood cells (which requires special cytometer settings), and when activated they are divided into two subpopulations of cells.

To study each platelet subpopulation, a protocol was developed that allows these subpopulations to be identified by fluorescence through various channels. However, due to the novelty of this protocol, no commonly known software can automate the gating of platelet subpopulations on cytometric data. Since clusters can change their location depending on the physiological characteristics of the blood donor and the number of associated proteins (examples in Fig.1). Currently, gating is carried out manually, which significantly slows down the processing and introduces processing errors. The average parameters of subpopulations depend significantly on the choice of the gate regions.

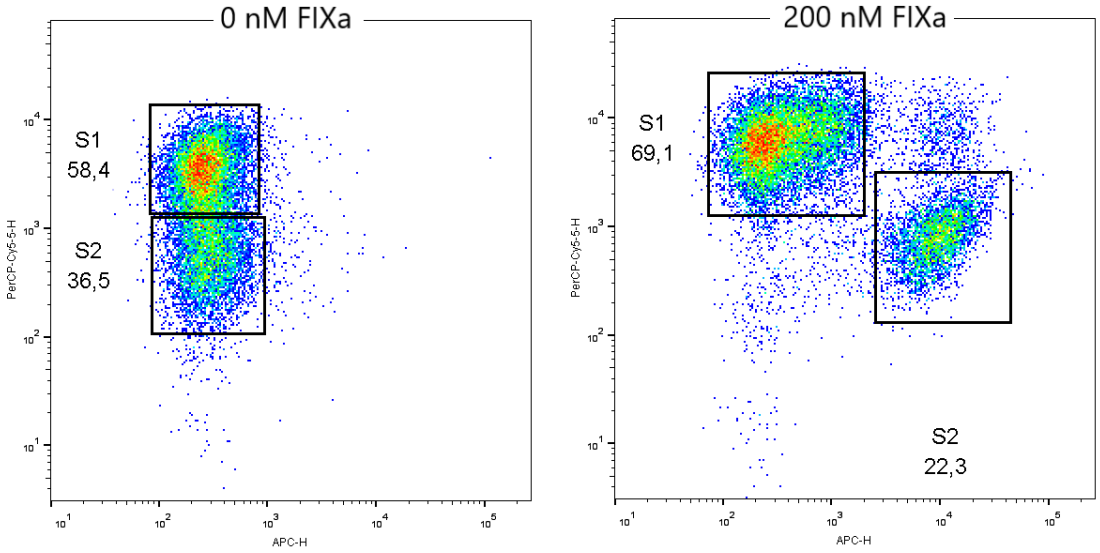

Fig.1 Examples of manual gating of platelet subpopulations under different experimental conditions.

This project aims to select the most plausible and fast clustering algorithm that would automate the problem of separating platelet subpopulations.
For this purpose the following tasks are set:

1. Develop a procedure for preprocessing cytometric data (separation of platelets)
2. Compare the performance quality of different machine learning algorithms that allow clustering platelet subpopulations using special metrics
3. Prepare a script to process data using the most suitable algorithm(s)

1 Data Preprocessing

1.1 Source Files

Initial data were obtained on a cytometer BD FACSCantoII. Data present the results of experiments on the equilibrium binding of coagulation factor IXa at various concentrations (from 0 to 200 nM) to activated platelets. Isolation of subpopulations is carried out through two fluorescence channels: PerCp-Cy5-5, corresponding to the level of calcium in the population (FuraRed dye), and APC, corresponding to the amount of bound coagulation factor (AlexaFluor647 dye). Direct (FSC) and side (SSC) light scattering channels can be used to separate platelets from debris. The data are presented without compensation since the selected dyes do not have a significant absorption/emission spectra intersection.

Cytometric files are saved in the FCS format. To read them, the open library ‘readfcs’ was used, presenting the data in AnnData format. For the convenience of further work, data of the measured cell parameters were placed into a Pandas Datarame and visualized using Seaborn and Matplotlib.

1.2 Data Scaling

The parameters of the studied cells vary from 0 to 10^6 standard units, therefore, it was decided to use a logarithmic scale. Before this, all parameter values less than or equal to zero were discarded as negative values sometimes occur due to internal software processing of the cytometer.

1.3 Isolation of Platelet Population (Anomalies Detection)

During the analysis process, excess particles may enter the detector, and various cell aggregates may arise, which can affect further clustering of subpopulations. The task of separating an entire population of platelets can be reduced to the task of getting rid of noise. Platelet separation occurs according to size and density according to parameters FSC, SSC. Three different approaches were employed to identify anomalies: 1) a percentile-based statistical method, 2) an isolation forest algorithm, and 3) density-based spatial clustering of applications with noise (DBSCAN).

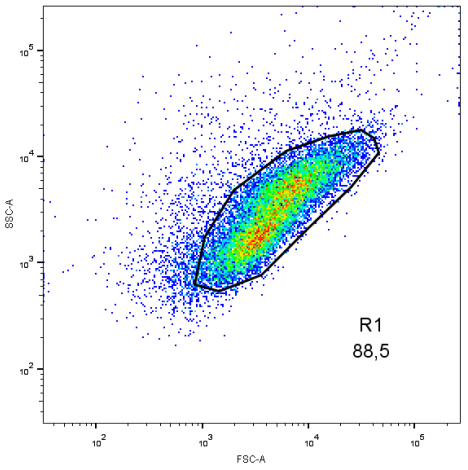

Fig.2 Sample of manual platelet population gating.

1.3.1 Statistical Methods

To select appropriate statistical methods, it is necessary first to determine whether the distribution of each parameter is normal. This can be done using a graphical method, such

as constructing a Q-Q plot (quantile-quantile plot), and also by conducting a formal statistical test like the Kolmogorov–Smirnov test to assess the normality of the distribution.

A Q–Q plot is a probability plot for comparing two probability distributions by plotting their quantiles against each other. If the two distributions being compared are similar, the points in the Q–Q plot will approximately lie on the identity line y = x. The distributions of cell fluorescence parameters were compared with normal ones using the statsmodels library (Fig.3)

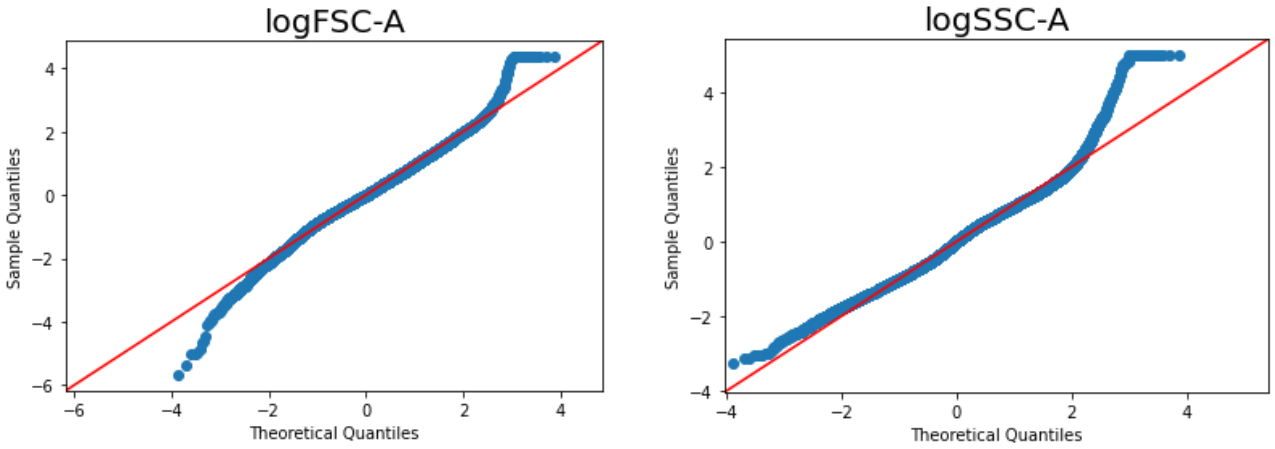

Fig.3. Q–Q plots

The Kolmogorov–Smirnov test is a hypothesis test in which the null hypothesis asserts that the sample originates from a specific distribution (here, a normal distribution). A high p-value suggests that the data set conforms to a normal distribution, while a low p-value indicates non-normal distribution. The Scipy library was utilized for statistical calculations. In both instances, the p-value was zero, falling below the significance threshold of 0.05.

For logFSC-A: KstestResult(statistic=0.9870229453310306, pvalue=0.0)
For logSSC-A: KstestResult(statistic=0.9919391491541164, pvalue=0.0)

Based on the results of both methods, we can conclude that the data is not normally distributed. Both parameters deviate from normal at the edges of the distribution (small and large values), which can significantly affect the average value. Therefore, statistical methods for normally distributed values cannot be used to get rid of noise in this case.

1.3.2 Isolation Forest

Isolation forest is an anomaly detection algorithm that identifies outliers in a dataset. The algorithm constructs random decision trees and measures the number of splits required to isolate an instance. Anomalies, being less frequent, are typically isolated more quickly than normal instances. The isolation path length serves as a measure of anomaly, with shorter paths indicating potential outliers. By aggregating these measures across multiple trees, the

algorithm identifies instances with shorter average path lengths as anomalies in the dataset. However, this algorithm does not take into account the density of data distribution.

The main parameter of the algorithm that affects the number of anomalies found is contamination. In the case of cytometric data, this parameter can be selected based on empirical data. Thus, it is known that on average 80% to 90% of events are platelets, respectively, the degree of contamination can vary from 10% to 20%. Several parameter values were considered (0.05, 0.1, 0.13, 0.15, 0.18, 0.2) on one of the datasets (Fig.4). For each value, comparisons with a manual gating sample (Fig.2) were made.

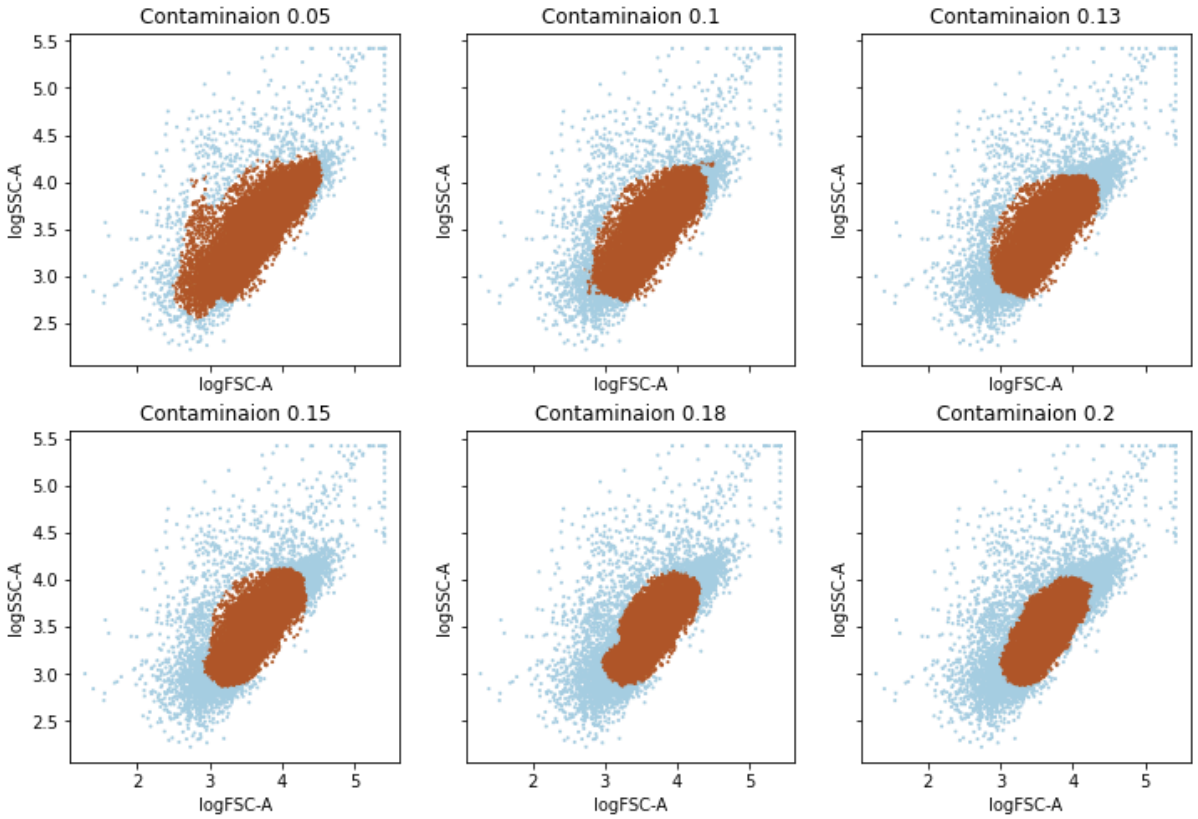

Fig.4. Isolation of platelet clusters using the Isolation Forest method with different values of the contamination parameter.

Thus, it can be seen that the platelet cluster narrows more noticeably at the ends than at the sides as the contamination parameter increases. This removes valid events at the poles (at a high contamination value) and leaves noisy data on the sides of the cluster (at a low contamination value). This method could be used to separate noise, but it requires manual selection of the contamination parameter using an empirical method, which varies randomly depending on the experiment. Therefore, this method loses the advantages of machine methods, since it does not allow for complete automation of the process.

1.3.3 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identifies noise in a dataset by distinguishing points that are not part of any dense cluster. Specifically, DBSCAN classifies points as core points, border points, or noise points. Core points are dense enough and form the core of a cluster, while noise points fail to meet the density requirements for cluster membership.

The algorithm has two important parameters. Epsilon specifies the radius of the neighborhood for each point. Within this radius, neighboring points are searched. If there are more of these points than the specified minimum number, the point is considered part of the cluster, otherwise, it is noise. In this case, all points in the epsilon neighborhood of this point are added to the cluster.

The optimal epsilon value for noise reduction: The method for determining the optimal epsilon value for DBSCAN for noise reduction involved a two-step process. The first step was calculating the distance between each point and its nearest neighbor. This was achieved by fitting a NearestNeighbors model from the sklearn.neighbors module to the dataset, and then using the kneighbors method to find the distances and indices of the nearest neighbors. After computing these distances, they were sorted in ascending order (Fig.5)

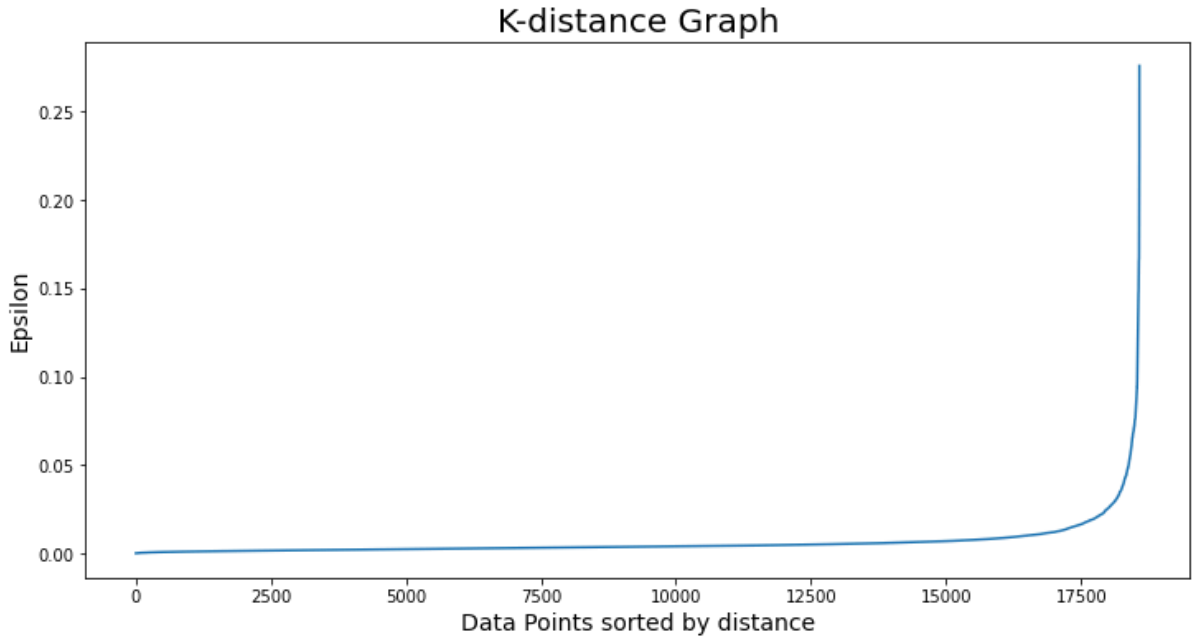

Fig.5. The distances of the nearest neighbors

The second step involved finding the point of maximum curvature on the K-distance plot, which indicates the optimal epsilon value. This point represents a balance between including core points (those within dense regions) and excluding noise (points in sparse regions). In simpler terms, beyond this point of maximum curvature, adding more points to the neighborhood (increasing epsilon) would start to include points from less dense regions, diluting the core clusters. This is accomplished by first normalizing the sorted distances and then calculating their gradient. The point of maximum curvature was identified as the point

where the absolute difference between the normalized gradient and a specified tangent value is minimized. This point corresponds to the optimal epsilon value for the DBSCAN algorithm on a particular dataset.

Having determined the optimal epsilon, the clustering method was applied (Fig.6). Each point received a label, with the desired cluster corresponding to the platelet population being the most frequent. This method provides the most optimal cutting of noise data, due to the complex location of the desired cluster. It is this that will be used for further processing.

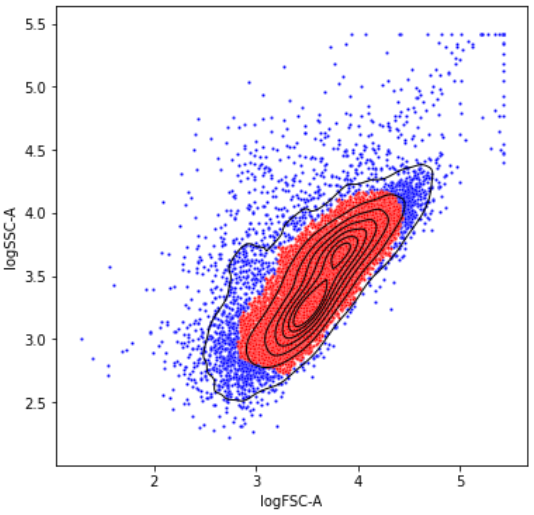

Fig.6. Clustering of platelet subpopulations using the DBSCAN method at optimal epsilon.

2 Clustering of Platelet Subpopulations

2.1 Overview of Clustering Methods

2.1.1 K-means

K-means is a clustering algorithm that partitions a dataset into K clusters, where each data point belongs to the cluster with the nearest mean. It's important to note that K-means assumes that clusters are spherical and equally sized. Thus, this algorithm works as hard clustering. The main configurable parameters of the algorithm are the number of expected clusters (in our case, always 2) and the method of initializing the starting point (selected k-mean++).

2.1.2 Expectation–Maximization (EM) Algorithm

The EM algorithm is used for clustering and is particularly associated with Gaussian Mixture Models. EM is a soft clustering algorithm, meaning that each data point has a probability distribution over all clusters rather than being assigned to a single cluster. The key idea behind EM is to iteratively refine the parameter estimates by alternately estimating the values of the latent variables (E-step) and updating the model parameters based on these estimates (M-step). The main configurable parameters of the algorithm are the number of expected clusters (in our case, always 2), the method of initializing the starting point (selected k-mean++), covariance of the difference classes estimated (selected spherical).

2.1.3 Agglomerative Clustering

Agglomerative Clustering is a hierarchical clustering algorithm that starts with individual data points as separate clusters and merges them progressively until a single cluster encompasses all the data. This is a "bottom-up" approach. Agglomerative clustering is commonly employed as a hard clustering method. The main configurable parameters of the algorithm are the number of expected clusters (in our case, always 2), the metric for calculating the connection (selected euclidean), and the linkage criterion (selected wand).

2.1.4 DBSCAN

In addition to being used for noise removal, this algorithm is also used to identify clusters of complex shapes. It also belongs to the group of soft clustering algorithms. Unlike being used to cut noise, epsilon was chosen empirically to not only cut noise but also to separate clusters (higher epsilon value).

2.2 Clustering Quality Metrics

To assess the clustering quality in this task, it is possible to use only metrics that do not require labeled data. Since the clustering problem in this case is not clear-cut, it does not have a uniquely correct answer that can be relied upon in data labeling. When assessing clustering, we will rely on the visual similarity of the expected clusters, as well as internal quality assessment metrics.

2.2.1 Davies-Bouldin Index (DBI)

The Davies-Bouldin index is an evaluation metric for clustering algorithms, measuring the average similarity between clusters, where the similarity is a measure that compares the distance between clusters with the size of the clusters themselves. A lower Davies-Bouldin index indicates better partitioning, meaning the clusters are more compact and better separated.

2.2.2 Calinski-Harabasz (CH) Index

The Calinski-Harabasz (CH) index assesses clustering quality by the ratio of the sum of between-clusters variance to the sum of within-cluster variance, with higher values indicating better-defined clusters. This index favors clusters that are dense and well-separated.

2.2.3 Silhouette Coefficient

The Silhouette Coefficient is a measure of cluster cohesion and separation; it quantifies how similar an object is to its cluster compared to other clusters. A higher Silhouette score suggests a better-defined cluster structure, with values ranging from -1 for incorrect clustering to +1 for highly dense clustering.

2.2.4 Time

Since the calculations are performed on a local personal computer and not on a server, the execution time of the algorithm can be an important parameter limiting the application of the proposed method.

2.3 Clustering Quality Assessment

Four different datasets were used to test the performance of all algorithms. The first of them shows a low concentration of the protein being bound, so the second subpopulation (to which the protein binds more actively) is slightly shifted along the x-axis. Therefore, the two subpopulations are close to each other. The last dataset is a case of high concentration of the protein of interest, so the subpopulations are separated in space, and the second subpopulation is strongly shifted to the right along the x-axis. These two examples represent two extremes of the possible relative arrangement of clusters. The remaining two datasets display intermediate states.

To work, we provide the path to the folder with dataset files in fsc format, load them, logarithm the necessary parameters and cut off the platelet population using the DBSCAN method, as was shown in the previous chapter. Then we apply each algorithm, visualize the resulting clusters (Fig.6), and calculate the metrics. (Table 1)

To determine the possible scatter in the quality of clustering in all algorithms, clustering was repeated 10 times on each dataset, and the mean and standard deviation were calculated. (Table 2) As expected, the Agglomerative and DBSCAN methods have stable metrics without variability, since these methods are deterministic for given parameters. The k-means and EM methods are regulated by an internal random-state parameter, which randomizes the setting of the initial value, so they are considered stochastic methods.

According to all the obtained metrics, the k-mean and EM algorithms are in the lead, however, EM shows more variability. Agglomerative Clustering shows similar values of quality metrics but is significantly inferior in execution time (a difference of 3 orders of magnitude). DBSCAN is inferior in all metrics except time. Presumably, this is due to the isolation of a separate noise cluster. However, DBSCAN is better than other algorithms in identifying dense clusters based on the visual principle. To test the hypothesis about the influence of the noise cluster on the metrics, all metrics for DBSCAN were also calculated for the two main large clusters, which are the required subpopulations. (Table 1) After calculating metrics for DBSCAN after noise removal, its quality turns out to be the best among all algorithms.

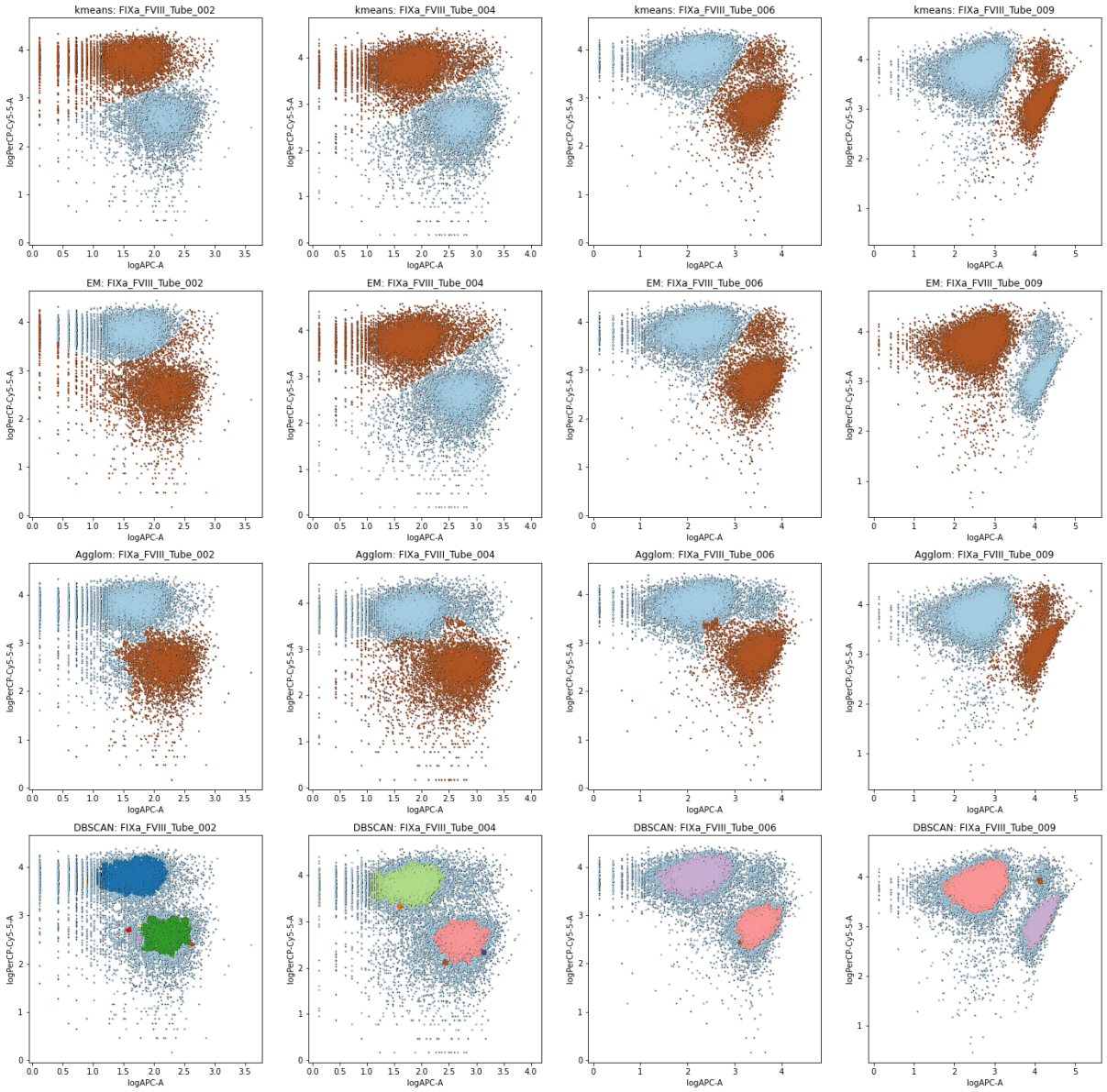

Fig.6. The result of clustering on four different datasets (columns are datasets, rows are algorithms)

Test\FIXa_FVIII_Tube_002.fcs

	Algorithms	DBI_score	CHI_score	Silh_Coef	Time
0	kmeans	0.691380	17268.574844	0.541367	0.462716
1	EM	0.688505	17198.846773	0.540690	0.020893
2	Agglom	0.718851	14381.922245	0.510721	7.579934
3	DBSCAN	2.388955	976.038538	-0.098519	0.220512
4	DBSCAN-2max-clusters	0.419609	28401.416607	0.706718	0.220512

Test\FIXa_FVIII_Tube_004.fcs

	Algorithms	DBI_score	CHI_score	Silh_Coef	Time
0	kmeans	0.683176	21857.130010	0.546117	0.035044
1	EM	0.696839	21195.929080	0.540016	0.017280
2	Agglom	0.717041	19944.562247	0.526007	12.725560
3	DBSCAN	1.572579	1782.822915	-0.118476	0.308793
4	DBSCAN-2max-clusters	0.399237	43061.186776	0.721351	0.308793

Test\FIXa_FVIII_Tube_006.fcs

	Algorithms	DBI_score	CHI_score	Silh_Coef	Time
0	kmeans	0.591921	27231.220389	0.591216	0.057606
1	EM	0.587311	27150.837326	0.592111	0.024429
2	Agglom	0.612514	21838.996774	0.557298	12.134375
3	DBSCAN	1.751783	4059.432333	0.204678	0.267903
4	DBSCAN-2max-clusters	0.373629	48482.152605	0.732619	0.267903

Test\FIXa_FVIII_Tube_009.fcs

	Algorithms	DBI_score	CHI_score	Silh_Coef	Time
0	kmeans	0.535661	31755.052924	0.619342	0.041153
1	EM	0.523453	31126.882635	0.618260	0.016194
2	Agglom	0.535567	31163.709314	0.616359	16.298126
3	DBSCAN	2.304744	5837.711612	0.357475	0.469183
4	DBSCAN-2max-clusters	0.359837	55656.381768	0.731057	0.469183

Table 1. Metrics for assessing the quality of algorithm performance on four different datasets (based on the results of one run)

Test\FIXa_FVIII_Tube_002.fcs

Algorithms	DBI_score_mean	DBI_score_std	CHI_score_mean	CHI_score_std	Silh_Coef_mean	Silh_Coef_std	Time_mean	Time_std
Agglom	0.718851	0.000000	14381.922245	0.000000	0.510721	0.000000	12.877501	0.335599
DBSCAN	2.388955	0.000000	976.038538	0.000000	-0.098519	0.000000	0.296445	0.022081
EM	0.713458	0.079604	16263.101525	2859.149939	0.529052	0.035584	0.079412	0.038759
kmeans	0.691397	0.000021	17268.573625	0.002371	0.541362	0.000005	0.092288	0.022657

Test\FIXa_FVIII_Tube_009.fcs

Algorithms	DBI_score_mean	DBI_score_std	CHI_score_mean	CHI_score_std	Silh_Coef_mean	Silh_Coef_std	Time_mean	Time_std
Agglom	0.535567	0.000000	31163.709314	0.000000	0.616359	0.000000	27.656802	1.334345
DBSCAN	2.304744	0.000000	5837.711612	0.000000	0.357475	0.000000	0.572454	0.044260
EM	0.523267	0.000226	31089.024767	48.148199	0.618128	0.000165	0.109922	0.055927
kmeans	0.535661	0.000000	31755.052924	0.000000	0.619342	0.000000	0.101206	0.003661

Table 2. Average metrics with standard deviation for assessing the quality of algorithm performance on two different datasets (based on the results of ten runs)

2.4 Subpopulation Means Fluorescence

The scientific problem for which clustering is performed is to determine the average fluorescence of each subpopulation. To understand how the automatically calculated parameters compare with those obtained manually, let us plot the dependence of the average fluorescence of subpopulations on the concentration of the added protein (the so-called equilibrium binding curve). Two sets of data, differing in the characteristics of the experimental setup, were tested using the DBSCAN and k-means algorithms.

The first data show the binding of coagulation factor IXa to platelets in the presence of cofactor VIIIa. Under such conditions, a weaker activator is used, which affects the proportion of separation into subpopulations; a higher affinity of the protein for the membrane of procoagulant platelets is observed, which causes a more rapid increase in average fluorescence. The second data shows the equilibrium binding of only factor IXa to platelets after the more potent activation. These data are characterized by a higher density of the procoagulant platelet subpopulation.

On the first data, both algorithms (DBSCAN, k-mean) showed similarity with the values obtained manually. (Fig.7) The same thing is observed in the second data (Fig.8.); however, if we look at the cluster graphs separately, this similarity in this case is simply a coincidence. Due to the different densities of subpopulations in the second dataset, both DBSCAN and k-means separate clusters with errors. (Fig.9.) The coincidence of the averages is possible because, due to the large number of points, the cluster centers are very stable. However, for scientific accuracy, clustering was also carried out using the Agglomerative Clustering algorithm, which showed the coincidence of both averages and cluster shapes. (Fig.9)

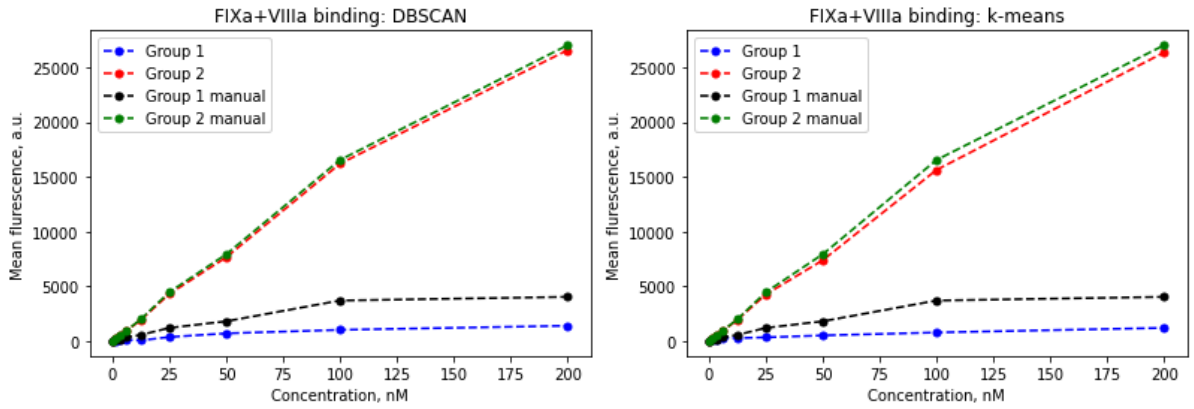

Fig.7. Equilibrium binding of factor IXa in the presence of cofactor VIIIa to activated by TRAP-6 platelets. Average fluorescence values for subpopulations calculated by manual gating or automatic clustering method (DBSCAN or k-means)

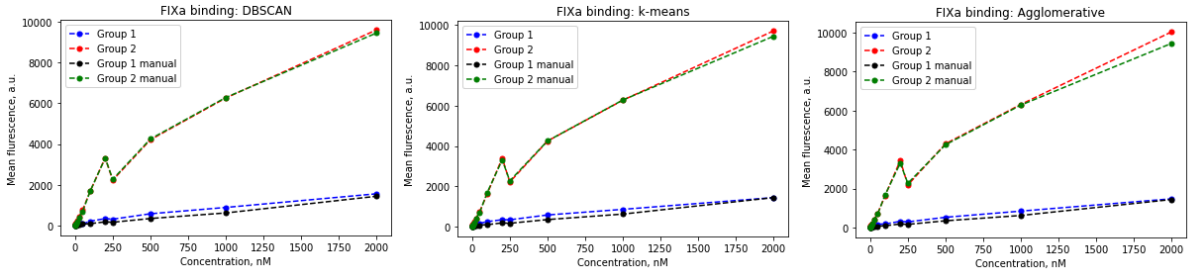

Fig.8. Equilibrium binding of factor IXa to activated by thrombin platelets. Average fluorescence values for subpopulations calculated by manual gating or automatic clustering method (DBSCAN or k-means or Agglomerative Clustering)

3 Platelets Clustering Cytometry Data Script

The final script for the automated processing of cytometric data obtained on the platelet population consists of the following main parts.

1. Uploading files from the folder in the FCS format. Preprocessing (separation of the platelet population from noise) using the parameters of forward and side light scattering using the DBSKAN method.
2. Clustering of platelets into two subpopulations using two fluorescence channels (PerCP-Cy5-5 channel corresponds to FuraRed, APC channel corresponds to AlexaFluor647) by the selected algorithm (DBSCAN, k-means or Agglomerative Clustering).
3. Calculation of average fluorescence values and standard deviation for each population, each dataset. Saving a pivot table in CSV format. (Table 3)
4. Visualization of clustering quality for each dataset. Saving a summary figure as PNG.
5. Visualization of the dependence of the average fluorescence of subpopulations on the concentration of the added factor (equilibrium curve). (Fig.10.)

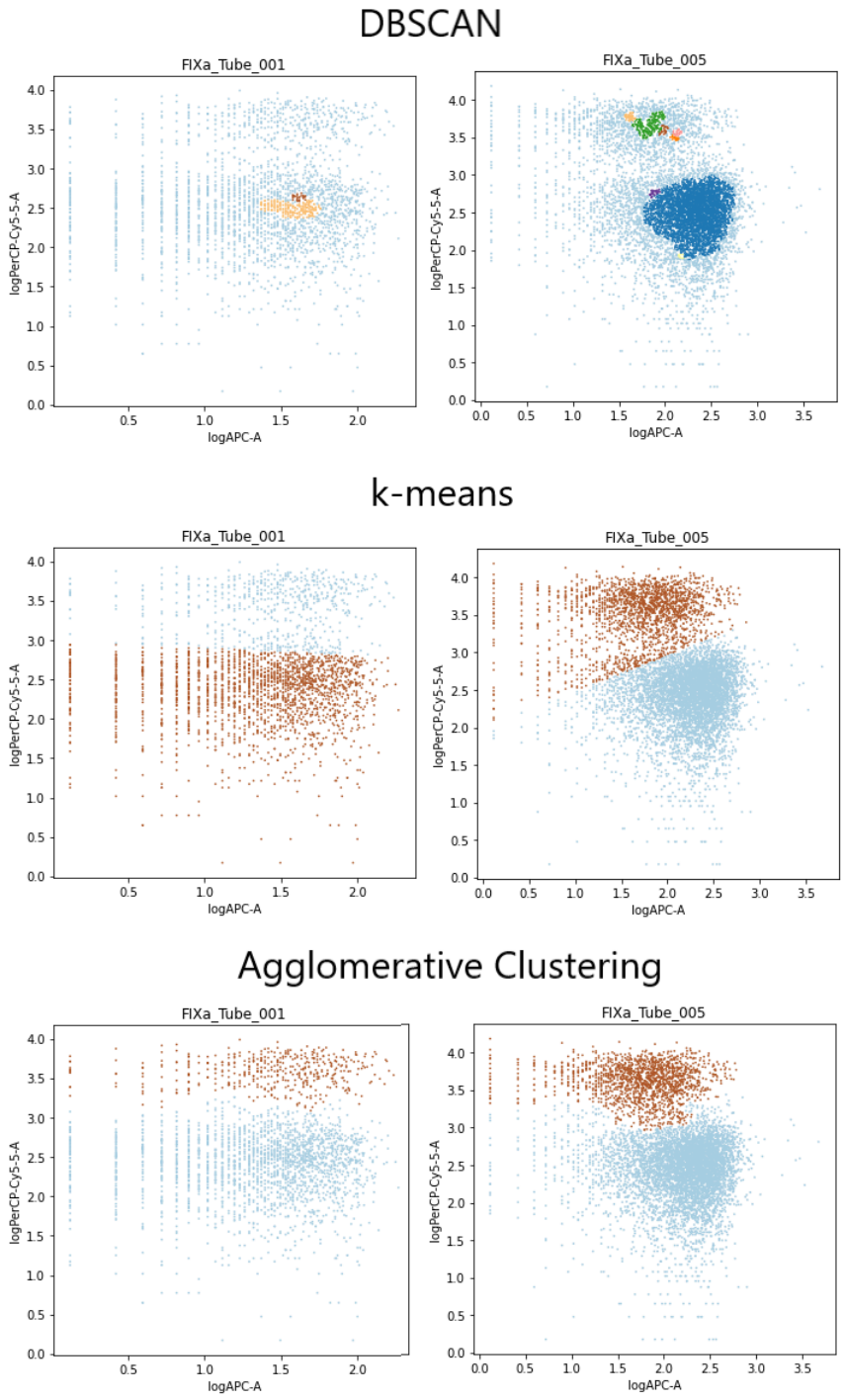

Fig.9. Equilibrium binding of factor IXa to activated by thrombin platelets. Examples of datasets clustering by different algorithms (DBSCAN or k-means or Agglomerative Clustering).

The script takes two main parameters: the path to the data folder, and the path to the output folder. The script also takes four optional parameters: algorithm type (by default 'DBSCAN'), whether to save an image of clusters ('yes' by default), and a list of concentrations for

constructing an equilibrium curve ('None' by default). To obtain complete information, the information '- -help' or '-h' command can be used.

	Subpopulation	Mean	std	sem
FIXa_FVIII_Tube_002	Group 1	55.824	32.931	0.484
FIXa_FVIII_Tube_002	Group 2	174.672	80.817	1.648
FIXa_FVIII_Tube_003	Group 1	60.557	36.117	0.498
FIXa_FVIII_Tube_003	Group 2	338.618	161.432	2.778
FIXa_FVIII_Tube_004	Group 1	74.465	47.82	0.635
FIXa_FVIII_Tube_004	Group 2	620.612	317.06	5.363
FIXa_FVIII_Tube_005	Group 1	108.577	76.97	1.0
FIXa_FVIII_Tube_005	Group 2	1536.053	743.623	13.141
FIXa_FVIII_Tube_006	Group 1	210.624	167.039	2.08
FIXa_FVIII_Tube_006	Group 2	3511.511	1798.906	29.94
FIXa_FVIII_Tube_007	Group 1	332.334	261.136	2.965
FIXa_FVIII_Tube_007	Group 2	6361.903	3061.845	50.425
FIXa_FVIII_Tube_008	Group 1	471.034	350.021	3.846
FIXa_FVIII_Tube_008	Group 2	10718.463	4604.413	76.954
FIXa_FVIII_Tube_009	Group 1	571.406	408.646	4.179
FIXa_FVIII_Tube_009	Group 2	14829.698	6462.531	104.425

Table 3. Pivot table in .csv format, saving by the script.

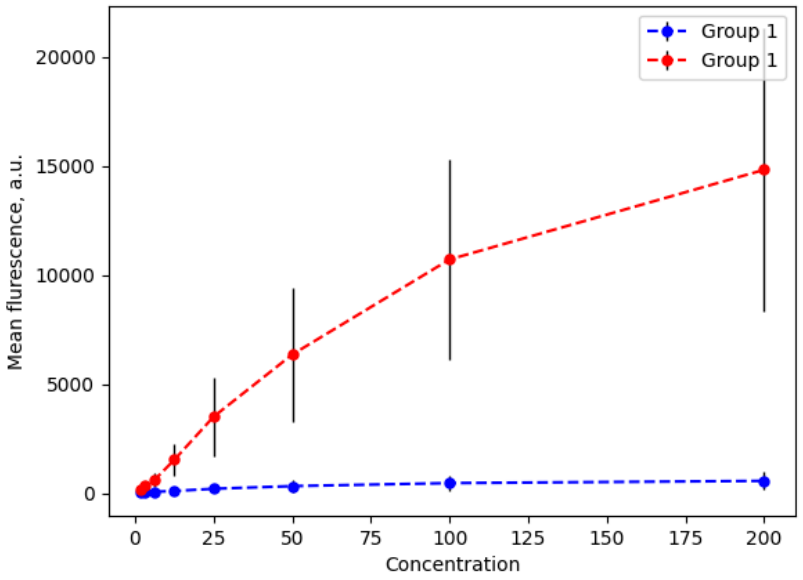

Fig.10. Equilibrium binding of coagulation factor to two subpopulations of platelets, obtained by data processing using the clustering method and implemented in a script.

Conclusion

Clustering biological data, which has a large amount of noise and significant variability in the shape and size of clusters, is a challenging task. This work describes an attempt to develop a general approach to automate the separation of platelet subpopulations in coagulation protein binding experiments. Unfortunately, data variability makes it difficult to choose a single method that will work perfectly for all data with the same specified parameters. However, this work offers several basic approaches for the user to choose from. With a large number of points in the studied dataset, as well as with comparable cluster sizes, it is acceptable to use the DBSCAN or k-means methods. With a reduced number of points in the dataset, as well as with a significant difference in cluster densities, it is proposed to use a longer time, but in this case more accurate, method of Agglomerative Clustering.

Using the proposed script, the user can, in one step, obtain clustering of all data, calculated average values, and errors, and a preliminary graph of equilibrium binding (which can be replaced by any other parameter specified by a discrete list).

As an extension of this work, it is possible to use a neural network that determines the type of data and selects the most suitable machine learning algorithm for clustering.

References

U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 12, pp. 1650-1654, Dec. 2002, doi: 10.1109/TPAMI.2002.1114856.

Hassani, M., Seidl, T. Using internal evaluation measures to validate the quality of diverse stream clustering algorithms. *Vietnam J Comput Sci* 4, 171–183 (2017). https://doi.org/10.1007/s40595-016-0086-9

Baumgaertner, P., Sankar, M., et al. (2021). Unsupervised Analysis of Flow Cytometry Data in a Clinical Setting Captures Cell Diversity and Allows Population Discovery. *Frontiers in Immunology*, *12*, 633910. https://doi.org/10.3389/fimmu.2021.633910

Aghaeepour N, Nikolic R, Hoos HH, Brinkman RR. Rapid cell population identification in flow cytometry data. Cytometry A. 2011 Jan;79(1):6-13.
doi: 10.1002/cyto.a.21007. PMID: 21182178; PMCID: PMC3137288.

Abe, K., Minoura, K., Maeda, Y. *et al.* Model-based clustering for flow and mass cytometry data with clinical information. *BMC Bioinformatics* 21 (Suppl 13), 393 (2020). https://doi.org/10.1186/s12859-020-03671-7

Pedersen CB, Olsen LR. Algorithmic Clustering Of Single-Cell Cytometry Data-How Unsupervised Are These Analyses Really? Cytometry A. 2020 Mar;97(3):219-221. doi: 10.1002/cyto.a.23917. Epub 2019 Nov 5. PMID: 31688998.

Cheung M, Campbell JJ, Whitby L, Thomas RJ, Braybrook J, Petzing J. Current trends in flow cytometry automated data analysis software. Cytometry A. 2021 Oct;99(10):1007-1021. doi: 10.1002/cyto.a.24320. Epub 2021 Feb 19. PMID: 33606354.

Ronan T, Qi Z, Naegle KM. Avoiding common pitfalls when clustering biological data. Sci Signal. 2016 Jun 14;9(432):re6. doi: 10.1126/scisignal.aad1932. PMID: 27303057.