Comparative analysis of experimental data clustering in MATLAB and Python environment

Gennady Seroklinov[1,*], and *Andrew* Goonko[2]

[1]Siberian Federal Scientific Centre of AgroBioTechnologies of RAS, 630501 Krasnoobsk, Russia
[2]Novosibirsk State Technical University, 630073 Novosibirsk, Russia

Annotation. The paper compares normalization and clustering methods for processing experimental data in the MATLAB package and a Python program using the scikit-learn library. Recommendations on further application of Python programs and selection of normalization and clustering methods are provided. The choice of clustering methods for further research is discussed.

1 Introduction

During the vegetation process, plants are exposed to various environmental influences. The changes that occur under the action of these stressors are accompanied by a change in the ionic conductivity of the plant cells, and as a result, a change in its biopotentials, which are involved in its vital activity. It has been proven that the level of plant resistance to different stressors is genetically determined and can be altered during ontogenesis. Therefore, by analyzing the changes in plant biopotentials , it is possible to assess the resistance of plants to stressors (in particular, high and low temperatures),develop quantitative criteria for intervarietal differences, and divide them into groups (clusters) with different or similar resistance characteristics. This especially applies to cereal crops when evaluating different varieties at the stage of their creation.

2 Problem definition

In this work the task is to identify plants with the same level of resistance to high or low temperature effects (that is, to divide them into two groups by varieties) by means of cluster analysis based on experimental data of biopotentials of two wheat varieties, as it was described in [1]. Next step is to compare the results of clustering of one set of experimental data by algorithmically identical (if possible) methods in MATLAB and Python environments. Data preprocessing (normalization) is carried out using the available tools in the selected environment.

The research was conducted using the experimental setup (the structural diagram of which is shown in Figure 1) on 7-14 daily seedlings of wheat varieties "Novosibirskaya 18" and "Omskaya 18" under the influence of high (up to 40^0 C) and low (down to 8^0 C)

[*] Corresponding author: seroklinov@mail.ru

temperatures. In the course of experimental studies, each sample was subjected to temperature effects twice, while changes in biopotential signals were recorded at three points of the seedling using a three-channel multimeter IPL 113 connected to a PC, and temperature change signals were formed using the automated system "AutoExpI" [2].

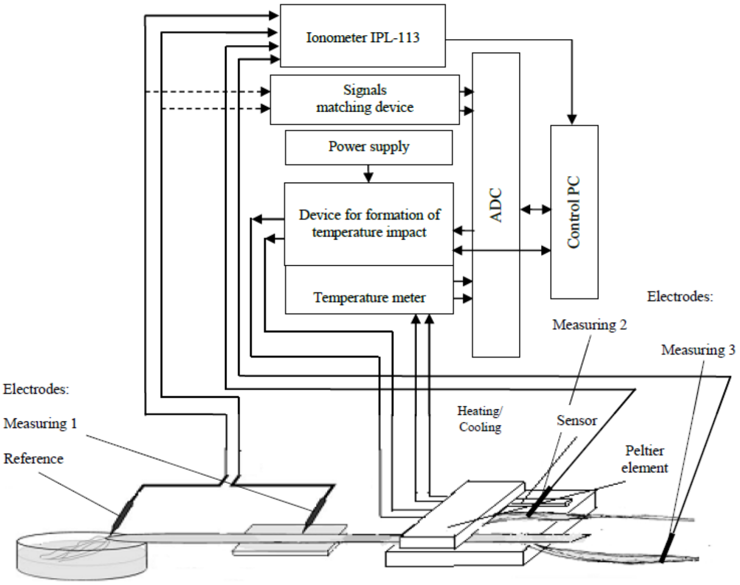

Fig. 1. Structural diagram of the experimental setup.

The biopotential realizations obtained for the "Omskaya 18" wheat seedlings under the influence of high temperature have a form shown in Figure 2. From the entire array of collected data, those that allow to identify the distinctive features of the plants were selected. These data of different wheat varieties were combined in pairs for comparison and shown in Tables 1, 2. Based on a wheat variety with a known level of resistance from varietal tests, it is necessary to distinguish plants of the same variety with the same reaction to temperature changes from the total population of the studied plants.

3 Theory

Clustering of samples in MATLAB environment was performed using such clustering methods as clusterdata (hierarchical classification method) [3], kmeans (k-means method) [4], and fcm (fuzzy c-means, C-means fuzzy clustering method) [5]. Prior to applying these methods, previously calculated parameters of biopotential signals were normalized using the mapminmax function (scales data to the range [-1, + 1]) [6], formerly known as premnmx, and the mapstd function (converts data to zero mean and unit standard deviation) [7] (formerly known as prestd). Both normalized and non-normalized parameters were clustered using these methods.

When writing a data clustering program in Python, the KMeans and MeanShift methods were selected (to replace the absent analogue of clusterdata) from the sklearn.cluster [8], which contains up to one and a half dozen methods, as well as the fuzzy c-means method available for installation [9].

For data normalisation the scikit-learn library provides a sklearn.preprocessing [10] package, which containes 5 methods: StandardScaler (scaling data to zero mean and unit variance), MinMaxScaler (scaling data to fit between a given minimum and maximum value, often between zero and one), MaxAbsScaler (scaling the maximum absolute value of each data point scaled to unit size), RobustScaler (recommended for data containing many outliers), Normalizer (scaling individual samples to unit norm).

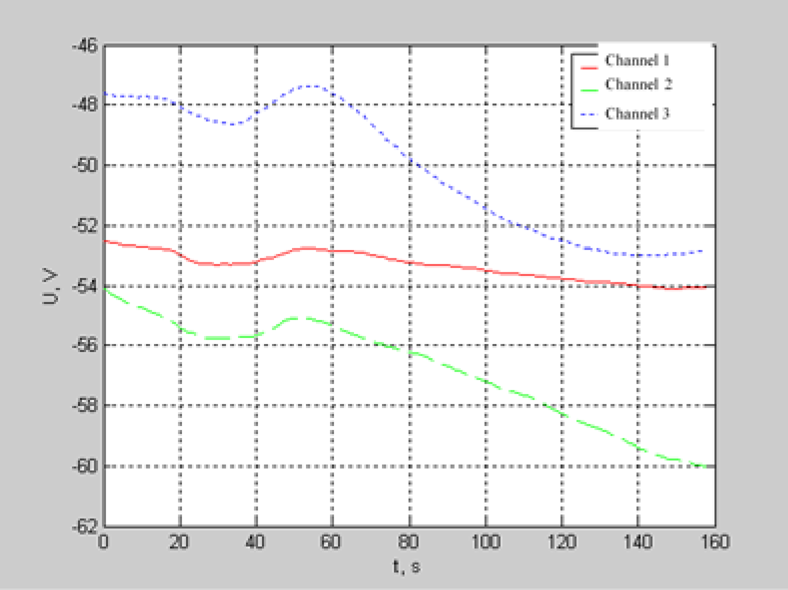

Fig. 2. Realization of “Omskaya 18” wheat seedling biopotentials when exposed to high temperatures.

Table 1. Parameters of biopotential signals when exposed to high temperature.

ID	Sample	Exposition	Channel	dfmin	Dfmax	dcfmin	dcfmax
Variety	Novosibirskaya 18						
5418111	1	1	1	-0.114819	0.001697	-0.070216	0.046300
			2	-0.267722	0.183197	-0.229290	0.221629
			3	-0.108534	0.094186	-0.065029	0.137691
5418211	2	1	1	-0.023800	0.001140	-0.013673	0.011268
			2	-0.498144	0.308028	-0.368780	0.437392
			3	-0.213870	0.187296	-0.211658	0.189508
5418311	3	1	1	-0.028549	0.003468	-0.017297	0.014719
			2	-0.317385	0.092181	-0.285775	0.123791
			3	-0.022745	0.119235	-0.039719	0.102261
Variety	Omskaya 18						
5518111	1	1	1	-0.065298	0.008046	-0.027183	0.046160
			2	-0.229450	0.050306	-0.152782	0.126974
			3	-0.271843	0.238216	-0.224021	0.286038
5518211	2	1	1	-0.069627	0.019570	-0.051980	0.037218
			2	-0.116183	0.208763	-0.088121	0.236825
			3	-0.181325	0.361543	-0.189974	0.352894
5518311	3	1	1	-0.061594	0.038802	-0.045618	0.054778
			2	-0.147744	0.070986	-0.120999	0.097731
			3	-0.129863	0.240503	-0.133663	0.236703

4 Analysis of results

The results of applying the above-described clustering methods in MATLAB environment are shown in Table 3, and the results obtained in Python are shown in Table 4. These tables indicate how many times out of a total of 540 experiments with samples of 2 varieties it was possible to correctly (in bold) or partially correctly (when samples of different varieties fell into one cluster, but the cluster sizes were determined correctly, indicated in parentheses) to divide the samples by species (varieties). Voltage signals were subjected to low-frequency filtering, centering (elimination of the constant component), differentiation, after which the minimum, average, maximum values of the initial, centered, filtered, differentiated signals were calculated for each of the three measurement channels, along which clustering was carried out.

Table 2. Parameters of biopotential signals when exposed to low temperature.

ID	Sample	Exposition	Channel	dfmin	dfmax	dcfmin	dcfmax
Variety	Novosibirskaya 18						
5418110	1	1	1	-0.410626	0.062935	-0.364945	0.108616
			2	-0.195732	0.356326	-0.190102	0.361955
			3	-0.131672	0.434186	-0.154692	0.411167
5418210	2	1	1	-0.009296	0.006444	-0.008468	0.007273
			2	-1.139404	2.344527	-1.182653	2.301277
			3	-0.672142	0.646041	-0.691294	0.626890
5418310	3	1	1	-0.011752	0.012585	-0.009519	0.014818
			2	-0.258846	0.168389	-0.201975	0.225260
			3	-0.047830	0.203645	-0.070253	0.181222
Variety	Omskaya 18						
5518110	1	1	1	-0.036320	0.016556	-0.037217	0.015660
			2	-0.094567	0.197817	-0.097955	0.194429
			3	-0.136728	0.198679	-0.135262	0.200145
5518210	2	1	1	-0.234799	0.051583	-0.212337	0.074045
			2	-0.156897	0.329994	-0.148876	0.338015
			3	-0.156145	0.228404	-0.143218	0.241331
5518310	3	1	1	-0.080877	0.014258	-0.057371	0.037765
			2	-0.207150	0.267091	-0.176486	0.297754
			3	-0.065337	0.116126	-0.058625	0.122837

Table 3. A summary table of sample clustering for all types of data normalization and clustering methods in the Matlab environment.

Methods Normalization	clasterdata	Kmeans	Fcm	By type:
No	**0**(12)	**0**(14)	0(14)	**0**(40)
Premnmx	**1**(2)	**2**(8)	**2**(10)	**5**(20)
Prestd	**1**(2)	**2**(8)	**2**(8)	**5**(18)
By method:	**2**(17)	**4**(30)	**4**(32)	**10**(78)

Table 4. Summary table of sample clustering for all types of data normalization and clustering methods in Python environment.

Methods Normalization	KMeans	Fcm	MeanShift	By type:
No	**0** *0* (12)	**0** *0* (9)	**0** *0* (12)	**0** *0* (33)
StandardScaler	**0** *0* (8)	**0** *0* (8)	**0** *1* (4)	**0** *1* (20)
MinMaxScaler	**1** *0* (7)	**1** *0* (8)	**0** *2* (4)	**2** *2* (19)
MaxAbsScaler	**3** *0* (6)	**3** *0* (9)	**0** *3* (4)	**6** *3* (19)
RobustScaler	**0** *0* (4)	**0** *0* (5)	**0** *1* (4)	**0** *1* (13)
Normalizer	**0** *0* (10)	**0** *0* (10)	**0** *3* (8)	**0** *3* (28)
By method:	**4** *0* (47)	**4** *0* (49)	**0** *10* (36)	**8** *10* (132)

Table 4 additionally indicates in italics the number of times that all samples of the same variety fell into one cluster, and the number of clusters turned out to be more than 2, since the MeanShift method does not specify the number of clusters. Typically, an extra third cluster included a sample whose signal parameters had significant outliers.

As an illustration of the clustering results, Figure 3 shows the separation of samples into 2 clusters using fcm method with data normalization by the premnmx function from the MATLAB package.

Figure 4 shows the results of Fcm clustering of data normalized by MaxAbsScaler in a Python program.

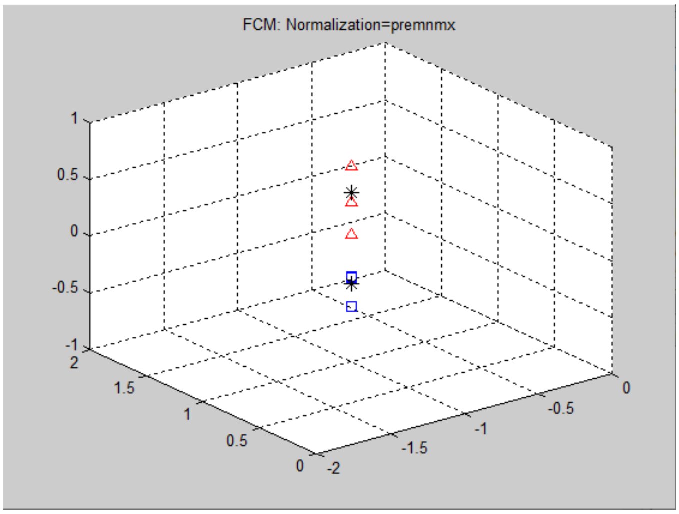

Fig. 3. Splitting samples into 2 clusters by fcm normalized by the premnmx function from the MATLAB package.

5 Discussion of results

In the MATLAB environment, the best clustering results are obtained using the kmeans and fcm methods, provided the data are normalized by any of the functions used. Clustering of non-normalized data, as well as the application of the clusterdata method, seems impractical.

In the Python environment, the MeanShift method allows samples to be partially correctly divided into clusters. However, since it does not scale by the number of clusters, its use is advisable only for processing data that do not have significant outliers. The clustering methods implemented by the Kmeans and Fcm functions allow us to explicitly divide the samples into clusters and give almost the identical results.

The StandardScaler normalization method (as well as normalization absence) give practically no results. The RobustScaler and Normalizer methods only partially separate samples into clusters and only when using the MeanShift method. The best clustering results by all three methods are achieved by the applying of MaxAbsScaler normalization. MinMaxScaler normalization gives the second best result.

6 Conclusion

The presented results indicate that the clustering methods implemented in Python language libraries provide generally no worse results than the methods proposed in the MATLAB package. At the same time, using of Python programs does not require the installation of proprietary (commercial) and resource-intensive software, and the sklearn.cluster clustering methods (available in the package, but not discussed here) allow us to hope for better results.

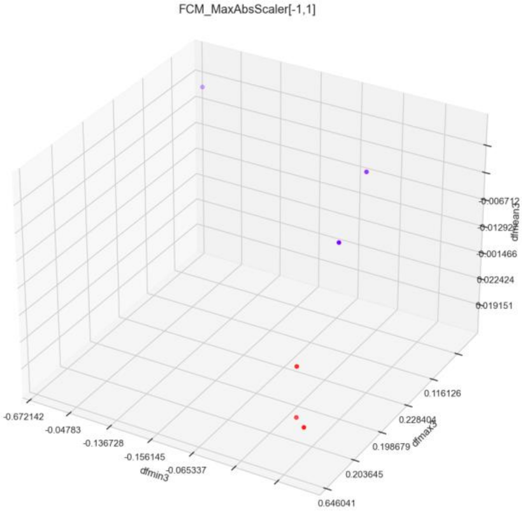

Fig. 4. Splitting samples into 2 clusters by FCM of data normalized by MaxAbsScaler in Python.

In particular, in future studies we plan to apply spectral and agglomerative clustering methods, clustering based on affine propagation density-based spatial clustering applications with noise (DBSCAN), and clustering based on the Gaussian mixture model.

References

1. G. V. Seroklinov, A. V. Gunko, IOP Conference Series: Earth and Environmental Science **848** (2021)
2. G. V. Seroklinov, A. V. Gunko, N. A. Dobrovolsky, *Methods and technical means of research of physical processes in agriculture* (SibPTI, Novosibirsk, 2011)
3. Construct agglomerative clusters from data – MATLAB clusterdata, https://www.mathworks.com/help/stats/clusterdata.html
4. k-means clustering – MATLAB kmeans, https://www.mathworks.com/help/stats/kmeans.html
5. Fuzzy c-means clustering – MATLAB fcm, https://www.mathworks.com/help/fuzzy/fcm.html

6. Process matrices by mapping row minimum and maximum values to [-1 1] – MATLAB mapminmax, https://www.mathworks.com/help/deeplearning/ref/mapminmax.html
7. Process matrices by mapping each row's means to 0 and deviations to 1 – MATLAB mapstd, https://www.mathworks.com/help/deeplearning/ref/mapstd.html
8. Clustering - scikit-learn 1.2.1 documentation, https://scikit-learn.org/stable/modules/clustering.htm
9. Fuzzy-c-means PyPi, https://pypi.org/project/fuzzy-c-means/
10. Preprocessing data - scikit-learn 1.2.1 documentation, https://scikit-learn.org/stable/modules/preprocessing.html