A comparative study of machine learning and deep learning algorithms to classify cancer types based on microarray gene expression data

Reinel Tabares-Soto [Corresp., 1] , **Simon Orozco-Arias** [Corresp., 2, 3] , **Victor Romero Cano** [4] , **Vanesa Segovia Bucheli** [5] , **Jose Luis Rodriguez-Sotelo** [1] , **Cristian Felipe Jiménez Varón** [6]

[1] Department of Electronics and Automation, Universidad Autónoma de Manizales, Manizales, Caldas, Colombia

[2] Department of Computer Science, Universidad Autónoma de Manizales, Manizales, Caldas, Colombia

[3] Department of Systems and informatics, Universidad de Caldas, Manizales, Caldas, Colombia

[4] Department of Automatics and Electronics, Universidad Autónoma de Occidente, Cali, Valle del Cauca, Colombia

[5] İzmir International Biomedicine and Genome Institute, Izmir, Turkey

[6] Department of Physics and Mathematics, Universidad Autónoma de Manizales, Manizales, Caldas, Colombia

Corresponding Authors: Reinel Tabares-Soto, Simon Orozco-Arias
Email address: rtabares@autonoma.edu.co, simon.orozco.arias@gmail.com

Cancer classification is a topic of major interest in medicine since it allows accurate and efficient diagnosis and facilitates a successful outcome in medical treatments. Previous studies have classified human tumors using a large-scale RNA profiling and supervised Machine Learning (ML) algorithms to construct a molecular-based classification of carcinoma cells from breast, bladder, adenocarcinoma, colorectal, gastro esophagus, kidney, liver, lung, ovarian, pancreas, and prostate tumors. These datasets are collectively known as the 11_tumor database, although this database has been used in several works in the ML field, no comparative studies of different algorithms can be found in the literature. On the other hand, advances in both hardware and software technologies have fostered considerable improvements in the precision of solutions that use ML, such as Deep Learning (DL). In this study, we compare the most widely used algorithms in classical ML and DL to classify the tumors described in the 11_tumor database. We obtained tumor identification accuracies between 90.6% (Logistic Regression) and 94.43% (Convolutional Neural Networks) using k-fold cross-validation. Also, we show how tuning process may or may not significantly improve algorithms' accuracies. Our results demonstrate an efficient and accurate classification method based on gene expression (microarray data) and ML/DL algorithms, which facilitates the prediction of the type of tumor from a multi-cancer-type scenario.

1 **A comparative study of machine learning and deep**
2 **learning algorithms to classify cancer types based on**
3 **microarray gene expression data**

5 Reinel Tabares-Soto[1], Simon Orozco-Arias[2,3] , Victor Romero Cano[4] , Vanesa Segovia Bucheli[5],
6 Jose Luis Rodriguez-Sotelo[1], Cristian Jiménez-Varón[6]

8 [1] Department of Electronics and Automation, Universidad Autónoma de Manizales, Manizales,
9 Colombia.
10 [2] Department of Computer Science, Universidad Autónoma de Manizales, Manizales, Colombia.
11 [3] Department of Systems and Informatics, Universidad de Caldas, Manizales, Colombia
12 [4] Department of Automatics and Electronics, Universidad Autónoma de Occidente, Cali,
13 Colombia.
14 [5] İzmir International Biomedicine and Genome Institute, Izmir, Turkey
15 [6] Department of Mathematics and Physics, Universidad Autónoma de Manizales, Manizales,
16 Colombia.

18 Corresponding Authors:
19 Reinel Tabares-Soto[1]
20 Antigua estación del ferrocarril, Manizales, Caldas, 170002, Colombia
21 Email address: rtabares@autonoma.edu.co

23 Simon Orozco-Arias[2,3]
24 Antigua estación del ferrocarril, Manizales, Caldas, 170002, Colombia
25 Email address: simon.orozco.arias@gmail.com
26

28 **Abstract**

29 Cancer classification is a topic of major interest in medicine since it allows accurate and efficient
30 diagnosis and facilitates a successful outcome in medical treatments. Previous studies have
31 classified human tumors using a large-scale RNA profiling and supervised Machine Learning
32 (ML) algorithms to construct a molecular-based classification of carcinoma cells from breast,
33 bladder, adenocarcinoma, colorectal, gastro esophagus, kidney, liver, lung, ovarian, pancreas,
34 and prostate tumors. These datasets are collectively known as the 11_tumor database, although
35 this database has been used in several works in the ML field, no comparative studies of different
36 algorithms can be found in the literature. On the other hand, advances in both hardware and
37 software technologies have fostered considerable improvements in the precision of solutions that
38 use ML, such as Deep Learning (DL). In this study, we compare the most widely used
39 algorithms in classical ML and DL to classify the tumors described in the 11_tumor database.

40   We obtained tumor identification accuracies between 90.6% (Logistic Regression) and 94.43%
41   (Convolutional Neural Networks) using k-fold cross-validation. Also, we show how tuning
42   process may or may not significantly improve algorithms' accuracies. Our results demonstrate
43   an efficient and accurate classification method based on gene expression (microarray data) and
44   ML/DL algorithms, which facilitates the prediction of the type of tumor from a multi-cancer-type
45   scenario.
46
Introduction
48   Cancer is one of the most deadly diseases in human health caused by the abnormal proliferation
49   of cells, leading to malignant malformations or tumors with different pathology characteristics
50   (Varadhachary, 2007). Cancer-type classification is critical to increasing patient survival rates.
51   Molecular genetic analyses have discovered genetic alterations, or signatures, with different
52   biological characteristics that allow discerning the responses to several treatments (Greller &
53   Tobin, 1999). This enables early diagnosis and an accurate treatment; therefore, ensuring the
54   efficacy and reduction of side effects (toxicity) of the treatment (Wang et al., 2005).
55
56   Impaired gene expression is a characteristic of carcinogenic cells (Su et al., 2001). Accordingly,
57   microarray gene expression data from tumor cells provide an important source of information to
58   improve cancer in a cost-efficient manner, allowing the use of this strategy in developing
59   countries. Although microarray datasets contain thousands of different genes to be analyzed, an
60   accurate and efficient way of analyzing this amount of data is by Machine Learning (ML) and
61   Deep Learning (DL) algorithms (Motieghader et al., 2017). In particular, these algorithms have
62   been applied in other biological areas, including rules of association (Orozco-Arias et al., 2019).
63   Previous studies demonstrate the use of ML and DL in microarray gene expression to infer the
64   expression of target genes based on landmark gene expression (Chen et al., 2016), in feature
65   selection aimed at finding an informative subset of gene expression (Sharma, Imoto & Miyano,
66   2012), and in the diagnosis and classification of cancer types (Fakoor et al., 2013).
67
68   A well-known database of gene microarrays related to cancer is the 11_Tumors database (Su et
69   al., 2001), which is available at
70   https://github.com/simonorozcoarias/ML_DL_microArrays/blob/master/data11tumors2.csv. This
71   dataset is a good example of the curse of dimensionality due to the high number of
72   characteristics and few registers of this database. Therefore, most studies use it to test specific
73   data science techniques, such as feature selection methods (Bolón-Canedo et al., 2014; Wang &
74   Wei, 2017; Han & Kim, 2018; Perera, Chan & Karunasekera, 2018), dimension reduction
75   (Araújo et al., 2011), clustering methods (Sardana & Agrawal, 2012; Sirinukunwattana et al.,
76   2013; Li et al., 2017), preprocessing techniques (Liu et al., 2019), among others. The 11_Tumors
77   database has also been used in gene selection for cancer classification (Moosa et al., 2016;
78   Alanni et al., 2019). Although the authors achieved high accuracy in these publications, they
79   only used some ML algorithms, one preprocessing strategy, and one learning technique

80   (supervised or unsupervised), which could add bias to their methodology. Additionally, to date,
81   no comparative study on the application of machine learning in microarray datasets is found in
82   the literature.
83
84   In several machine learning studies, DL has proven to be a robust technique for analyzing large-
85   scale data sets (Bengio, Courville & Vincent, 2013). With these advances, DL has achieved
86   cutting-edge performance in a wide range of applications, including bioinformatics and genomics
87   (Min, Lee & Yoon, 2016; Yue & Wang, 2018), analysis of metagenomics samples (Ceballos et
88   al., 2019), identification of somatic transposable elements in ovarian cancer (Tang et al., 2017),
89   identification and classification of retrotransposons in plants (Orozco-Arias, Isaza & Guyot,
90   2019) and cancer classification using Principal Component Analysis (PCA) (Liu, Cai & Shao,
91   2011). Recent work by Guillen and Ebalunode demonstrated promising results for the
92   application of DL in microarray gene expression (Guillen & Ebalunode, 2016).
93
94   In general, there are two different tasks that ML algorithms can tackle: supervised and
95   unsupervised learning. In supervised learning, the goal is to predict the label (classification) or
96   response (regression) of each data point by using a provided set of labeled training examples. In
97   unsupervised learning, such as clustering and principal component analysis, the goal is to learn
98   inherent patterns within the data (Zou et al., 2018).
99
100   The main goal of any ML task is to optimize model performance not only on the training data but
101   also on additional datasets. When a learned model displays this behavior, it is considered to
102   generalize well. With this aim, the data in a given database are randomly split into at least two
103   subsets: training and validation sets (Zou et al., 2018). Then, a model as complex as possible is
104   learned (training set), tuned (validation set), and tested for generalization performance on the
105   validation set. This process is crucial for avoiding overfitting or underfitting. Therefore, a sound
106   learning algorithm must reach an appropriate balance between model flexibility and the amount
107   of training data. An overly simple model will underfit and make inadequate predictions, whereas
108   an overly flexible model will overfit to spurious patterns in the training data and not generalize
109   (Zou et al., 2018).
110
111   In this study, we compare the performance of the most commonly used ML and DL algorithms
112   in bioinformatics (Orozco-arias et al., 2019) in the task of classifying by supervised and
113   unsupervised techniques. We used the 11_Tumor database and applied different preprocessing
114   strategies. Our detailed evaluation and comparison illustrate the high accuracy of these
115   algorithms for tumor identification in a multiple-cancer-type scenario and the influence of
116   preprocessing strategies and tuning processes on these accuracies.
117

Materials & Methods

118

119   ML and DL techniques can learn the characteristics of a given problem from a certain amount of
120   data. These data are usually randomly subdivided into two groups: training and validation. A
121   training data set is used to calibrate the parameters of the model, and a validation data set is
122   utilized for evaluating model performance (Eraslan et al., 2019).
123
124   In this paper, we compared results obtained from classifying 11 different tumor classes through
125   different approaches of ML and DL. We began by evaluating two unsupervised methods; the
126   first method is the popular K-means algorithm, in which a given number of prototype samples,
127   also known as cluster centers, are estimated by iteratively assigning data points to prototype
128   samples and updating them as the mean of the assigned samples. The second clustering method
129   tested is hierarchical clustering, which is better suited for irregular shapes than K-means. After,
130   we tested eight different classification algorithms. The most popular one, and the standard
131   baseline in classification problems, is K-Nearest Neighbors (KNN), where classification
132   decisions are done through a voting mechanism and model training stores the dataset in a way
133   that queries can be done efficiently. Another family of classification methods comprises the so-
134   called linear models, for which a learning algorithm estimates as many weights as features from
135   the training data so classification prediction is done as a function of the dot product between the
136   weights and a test sample. Linear models are fast to train, fast to predict, and also scale well to
137   datasets in which the number of features is large compared to the number of samples. The linear
138   methods we tested are Linear Support Vector Classifier (SVC), Logistic Regression (LR), Linear
139   Discriminant Analysis (LDA), Naive Bayesian Classifier (NB), and Multi-Layer Perceptron
140   (MLP).
141
142   We also included Decision Tree-methods (DT) such as Random Forests (RF). Unlike linear
143   models, DTs and RFs are invariant to data scaling and work well with features on different
144   scales. Finally, we applied Deep Neuronal Networks (DNN), such as fully connected neural
145   networks, also known as Multi-Layer Perceptron (MLP) and Convolutional Neural Networks
146   (CNNs). MLPs are well-suited for non-linear data, whereas CNNs automatize the costly task of
147   engineering features; an unavoidable task in classical machine learning approaches. The above
148   algorithms are extensively explained in (Michie, Spiegelhalter & Taylor, 1994; Chollet, 2007).
149

150   **Datasets**
151
152   The datasets used represent measurements of gene expression using cancer microarrays and
153   normal biopsies (Statnikov et al., 2005; Bolón-Canedo et al., 2014), and are consolidated in the
154   "11 Tumors database", which is freely available online at
155   (https://github.com/simonorozcoarias/ML_DL_microArrays/blob/master/data11tumors2.csv).
156   This database consists of 174 samples with 12,533 gene expression microarrays for 11 different
157   types of cancer. The 12,533 microarrays of genetic expression are integers with positive and
158   negative values; these values represent the characteristics that allow the ML and DL algorithms

159   to learn how to classify by cancer type. The types of cancer and the number of patients for each
160   type are shown in table 1. The classes of each cancer type are unbalanced and remained so in the
161   experimentation.
162

163   **Preparing the data**
164   For the experiments, we divided the information into two groups; the first group corresponds to
165   the features (X) and the second group to the classes (Y). The features compose a matrix of size m
166   x n and the classes are a vector of size n x 1, where m is the number of samples and n is the
167   number of genes for each class (12,533). The dataset, containing 174 samples, is randomly
168   subdivided into two sub-sets (80% training and 20% validation), including 139 samples for
169   training and 35 samples for validation. Initial calibration of ML and DL algorithms (training)
170   was done using the training set; then, hyperparameter tuning was performed with the validation
171   set and measured the accuracy of the algorithms. We calculated the accuracy of each algorithm
172   using tuned hyperparameters with k-fold cross-validation and *k=10* to avoid overfitting.
173

174   The dataset used in this paper has the curse of dimensionality since the number of characteristics
175   (12,533) is higher than the number of samples (174) (Powell, 2007). Therefore, the data are
176   dispersed and the results are not statistically stable or reliable, directly affecting the accuracy
177   achieved by ML and DL algorithms. Two preprocessing techniques were used to solve this
178   problem: scaling (Géron, 2017) and principal component analysis (PCA) (Wold, Esbensen &
179   Geladi, 1987). The first technique guarantees that the data are in a range of suitable values to
180   calibrate the model. With the second technique, the statistical significance is improved and the
181   noise introduced by irrelevant characteristics during model training decreases. In this paper, we
182   worked with several combinations of the preprocessing techniques mentioned above to find the
183   best performance.
184

185   Four different datasets were created for the training and validation of each ML or DL algorithm.
186   For the first dataset, we did not apply any preprocessing operations; for the second, we
187   performed a scaling process; for the third, we applied PCA with a retained variance of 96% to
188   reduce data dimensionality, obtaining a dimensional reduction from 12,533 to 83 features.
189   Finally, for the last dataset, we applied both scaling and PCA, obtaining a dimensional reduction
190   from 12,533 to 113 features (principal components).
191

192   **Unsupervised learning experiments**
193

194   Classification performance is highly correlated with the degree of separability of a dataset;
195   therefore, we analyzed performance using clustering techniques. Based on data labels, we can
196   gain a priori insight into the algorithm that works best on the distribution of the gene expression
197   microarray dataset.
198

199   Before applying the classification algorithms (supervised learning), we performed a hierarchical
200   analysis to better understand the dataset. This hierarchical clustering used different distance
201   metrics, such as Ward, average, single, complete, weighted, centroid, and median. Further, as
202   input, we used a dataset with no preprocessing. These distance metrics serve to capture the
203   differences between the data samples and vary in their capacity to deal with large outliers (i.e.
204   between weighted, centroid, and median metrics) or if they allow choosing the number of
205   clusters to consider (e.g. Ward) (Foss, Markatou & Ray, 2019). After this clustering, we tested
206   all of the datasets created in the previous step to determine the best preprocessing methodology.
207   Finally, a dendrogram and a heat map were used to illustrate the separability attribute of our
208   dataset. Additionally, we performed a clustering analysis using the K-means algorithm with k
209   values of one to eleven clusters using all datasets. We plotted the behavior in terms of accuracy
210   and as a confusion matrix.
211
212   **Supervised learning**
213
214   We evaluated the performance of well-known ML classification algorithms, including KNN,
215   SVC, LR, LDA, NB, MLP, RF, and DT. Subsequently, we evaluated DL architectures, such as
216   fully connected neural networks (FNNs) and convolutional neural networks (CNNs).
217
218   **Neural network architecture**
219
220   Two types of networks were used for deep learning; the first is a fully connected neural network
221   and the second is a convolutional neural network. The FNN consists of three fully connected
222   layers of 100 neurons each and the Softsign activation function; then, a final layer of 11 neurons
223   is generated with the sigmoid activation function to generate the probability of the type of
224   cancer. The CNN consists of three convolutional layers with 128 filters each, with a kernel size
225   of 3 and a linear activation function; followed by a layer of 100 fully connected neurons with the
226   Softsign activation function and, finally, a layer of 11 neurons with the Softmax activation
227   function to generate the probability of the type of cancer. Figure 1 shows the architectures used
228   for the experiment, in which the top scheme is a FNN and the bottom scheme is a CNN.
229
230   **Tuning the algorithms**
231
232   Several algorithms were tested by varying or tuning parameter values to find the best
233   performance (Table 2). With these results, we plotted the accuracy values using all datasets
234   created in the training and validation processes and also created confusion matrices. Finally, we
235   did a cross-validation of each algorithm to find the accuracy that was less affected by bias.
236   Additionally, in FNNs and CNNs, we performed a hyperparameter search with a grid search
237   method (GridSearchCV) from the sklearn module, considering the variables shown in Table 3.
238   Due to the high number of parameters, the process of tuning FNNs and CNNs involved choosing

239 the parameter values that achieved the best accuracy and, then, using these values to find others.
240 The process of finding the best parameter values is presented as follows: 1) batch size and
241 epochs 2) training optimization algorithm 3) learning rate and momentum 4) network weight
242 initialization 5) neuron activation function 6) dropout regularization and 7) number of neurons in
243 the hidden layers.
244
245 **Significance Tests**
246
247 We performed a test for difference in proportions to determine whether the difference between
248 accuracies of the algorithms is significant. We calculated the differences between the observed
249 and expected accuracies under the assumption of a normal distribution. Given the number of
250 correct test predictions x and the number of test instances N, accuracy is defined as follows:

251 $$Acc_i = \frac{x}{N}$$

252 $$H_0 : Acc_i - Acc_j = 0$$
253 $$H_1 : Acc_i - Acc_j \neq 0$$

254
255 This test allowed determining if the accuracies of the algorithm change significantly after the
256 tuning process and also if there are significant differences between the two algorithms with the
257 highest average accuracies. Based on this, we evaluated whether the parameter tuning of the
258 algorithms was necessary or if the ML algorithm used was more relevant.
259
260 **Tools**
261
262 The algorithms were executed using Python programming language and sklearn libraries
263 (Pedregosa et al., 2011), which are explained in (Komer, Bergstra & Eliasmith, 2014) for ML
264 algorithms. PCA and scaling were executed with decomposition and preprocessing modules from
265 sklearn. Also, DNNs were implemented using Keras (Chollet & others, 2015). All images were
266 created with matplotlib (Hunter, 2007). The significance tests were performed using R software
267 (Supplementary material 1). The algorithms used here are available at
268 https://github.com/simonorozcoarias/ML_DL_microArrays.
269

270 # Results
271 **Hierarchical analysis**
272
273 Before evaluating the classification algorithms, we visualized the intrinsic groupings in the data
274 and determined how these groups are influenced by the different preprocessing methodologies
275 applied to our data (Figure 2). Using the downloaded raw data, we created a hierarchical graph
276 (unsupervised learning) using different methodologies (Fig. S1) and concluded that Ward's
277 method produced the most balanced clusters (Figure 3). Then, using only Ward's method, we

278 performed additional analyses using different datasets, including raw data, scaled data, data
279 transformed by PCA, and data scaled and transformed by PCA. Finally, we created a
280 dendrogram and a heat map to find whether data can be clustered into groups without any given
281 class with the best results. Figure 4 shows four well-separated groups, but the heat map
282 demonstrated other well-conserved groups, which may indicate that the four main clusters could
283 be divided into subgroups.
284
285 Ward's method created four groups, while the other methods clustered the individuals into fewer
286 groups and, in most cases, these groups are largely unbalanced. On the other hand, the raw data
287 and data transformed by PCA performed better in the hierarchical clustering analysis. Employing
288 these datasets, we were able to obtain four and five clusters, respectively. Finally, the heatmaps
289 plotted in Figure 4 showed one group greatly distant from the others (green in Figure 4A and
290 light blue in Figure 4B). On the other hand, the other clusters showed low intra-cluster distances,
291 which is an ideal feature in classification problems (clear blue in Figure 4A and green in Figure
292 4B).
293
294 Based on a priori knowledge that the number of cancer types is eleven (11), we were interested
295 in determining how the hierarchical clustering algorithm created the cluster assignments.
296 Therefore, we applied the best parameters found previously (clustering method: ward, and input:
297 raw data and data reduced by PCA). The results shown in Figure 5 and Tables 4 and 5
298 demonstrate that, although the hierarchical clustering algorithm displays good performance, it
299 does not group the data into the correct number of groups.
300
301 Another unsupervised learning assessment involved the implementation of the K-means
302 algorithm. We used all datasets and changed the number of clusters iteratively from one to
303 eleven, increasing by one cluster at a time. Then, we calculated the accuracy in each iteration and
304 a confusion matrix was plotted with the best results (Figure 6). Additionally, we calculated other
305 metrics, such as precision, recall, and f1-score for each class. Overall, the best results were
306 obtained by K-means using 11 clusters with input data processed by PCA, achieving an accuracy
307 of 68.34% (validation set, using the hold-out splitting method). Also, classes 6, 7 and 9 showed
308 precisions of 100% and class 5 of 91% (Table 5).
309
310 **Algorithm tuning**
311
312 The algorithms were tuned by setting several parameters between a given value range (Table 2)
313 to find the best behavior using all datasets. Through this, we aimed to calculate the best
314 hyperparameters for each algorithm and determine which dataset could be the most appropriate.
315 The results of the highest validation accuracies are shown in Table 6. To evaluate overfitting or
316 underfitting, we plotted the accuracy values of the training and validation processes on all
317 datasets described above (Figure 7). RF and DT were not plotted since more than one

318 hyperparameter were tuned. The best results were obtained using LG and raw data. We also
319 calculated a confusion matrix for these results, finding very good classification rates (Figure 8).
320
321 **Cross-validation**
322
323 KNN, SVC, LG, MLP, K-MEANS, LDA, NB, RF, and DT were trained and validated with the
324 same fraction of data and each experiment was repeated 10 times to obtain the standard
325 deviations using sklearn's cross-validation function with k=10 (Komer, Bergstra & Eliasmith,
326 2014). We used the entire dataset (174) for this procedure. The accuracy and standard deviation
327 results are shown in Table 7.
328
329 **Deep neural networks**
330
331   The grid-search method showed the hyperparameter values that provided the best accuracy in
332 FNN and CNN architectures (Table 8). Figures 9 and 10 show the training results of both
333 architectures, demonstrating how the loss function decreases when most epochs are used until a
334 specific number of epochs is reached (80 for FNN and 8 for CNN). Similarly, the accuracy
335 increases in both the training and validation data until reaching the same number of epochs
336 mentioned for the loss function. After this number of epochs, no significant changes were
337 observed for the loss and accuracy values. Using these parameters and cross-validation with
338 k=10, FNN and CNN achieved accuracies of 91.43% and 94.43%, respectively.
339
340 **Significance tests**
341
342 We performed a test of significant differences, with a 95% confidence level, between the two
343 best-performing ML algorithms (LG and CNN). Accordingly, we found no significant
344 differences between the accuracies of these two algorithms (p-value=0.447).
345

346 # Discussion
347 In this work, we show the application of unsupervised and supervised learning approaches of ML
348 and DL for the classification of 11 cancer types based on a microarray dataset. We observed that
349 the best average results using the training and validation data are obtained using the raw dataset
350 and the Logistic Regression (LR) algorithm, yielding an accuracy value of 100% (validation set,
351 using the hold-out splitting method). One could assume there is overfitting since the confusion
352 matrix showed an extremely good behavior; however, the comparison of the training and
353 validation accuracies between parameters using the entire dataset may indicate perfect accuracy
354 in both training and validation datasets. Additional tests with independent data should be done to
355 discard potential overfitting.
356

357 On the other hand, Multi-Layer Perceptron (MLP) and Linear Discriminant Analysis (LDA)
358 showed a high accuracy value of 97.14% in the validation dataset. This improvement in accuracy
359 was obtained by optimizing several parameters (number of neurons in MLP) and preprocessing
360 the data set with PCA.
361
362 After tuning four parameters, Random forest (RF) obtained high results, with a maximum
363 accuracy of 85.71%. In contrast, Decision trees (DT) obtained 51.14% accuracy, demonstrating
364 that DT does not work properly for the datasets used in this study, despite tuning several
365 parameters (in our case, three).
366
367 Our findings demonstrate that the various algorithms work better by preprocessing the datasets
368 differently. Our results show that MLP, DT, and LDA improved in performance if PCA was
369 applied in advance. However, LG, KNN, NB, RF, and K-means worked better using no
370 preprocessing. Only SVC improved when using scaling and, interestingly, none of the other
371 algorithms showed better results using scaling and PCA on the datasets.
372
373 Parameter tuning can improve the accuracy of the algorithm used (Table 7). For instance, SVC
374 obtained a low accuracy of 10.82% before preprocessing but increased to 81.98% after tuning.
375 Although most of the algorithms improved their accuracies after the tuning process, only two of
376 them (SVC and K-means) showed significant changes. We conclude that Logistic Regression
377 (LG) is the best ML algorithm for the test dataset in this study, providing an accuracy of 90.6%
378 with a standard variation of 5.94 from the cross-validation analysis based on ten times.
379 Nevertheless, we recommended using it with moderation. On the other hand, for DL
380 architectures, CNN obtained the best accuracy with 94.43% (Figure 10). The grid search
381 technique enabled parameter tuning and improved the results, allowing us to propose new DNN
382 architectures (i.e. the architectures showed in Figure 1). Finally, we found no significant
383 difference between the accuracies obtained by LG and CNN.
384
385

386 ## Conclusions
387 Cancer is predicted to become the most deadly disease for humans in the future (Dagenais et al.,
388 2019); therefore, early diagnosis, identification, and treatment are needed to control the disease.
389 ML and DL techniques are promising tools for the classification of cancer types using complex
390 datasets, such as microarrays. In this study, we obtained predictions with as high as 93.52 and
391 94.46% accuracies, which will allow patients with these types of pathologies to receive an early
392 and precise detection of their disease, and will also contribute to the discovery of new selective
393 drugs for the treatment of these types of tumors.
394
395 ## References
396

Alanni R, Hou J, Azzawi H, Xiang Y. 2019. A novel gene selection algorithm for cancer classification using microarray datasets. *BMC medical genomics* 12:10.

Araújo D, Neto AD, Martins A, Melo J. 2011. Comparative study on dimension reduction techniques for cluster analysis of microarray data. In: *The 2011 International Joint Conference on Neural Networks*. 1835–1842.

Bengio Y, Courville A, Vincent P. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35:1798–1828. DOI: 10.1109/TPAMI.2013.50.

Bolón-Canedo V, Sánchez-Marono N, Alonso-Betanzos A, Ben'\itez JM, Herrera F. 2014. A review of microarray datasets and applied feature selection methods. *Information Sciences* 282:111–135.

Ceballos D, López-Álvarez D, Isaza G, Tabares-Soto R, Orozco-Arias S, Ferrin C. 2019. A Machine Learning-based Pipeline for the Classification of CTX-M in Metagenomics Samples. *Processes* 7:235. DOI: 10.3390/pr7040235.

Chen Y, Li Y, Narayan R, Subramanian A, Xie X. 2016. Gene expression inference with deep learning. *Bioinformatics* 32:1832–1839.

Chollet F. 2007. *Deep Learning with Python*. DOI: citeulike-article-id:10054678.

Chollet F, others. 2015. Keras.

Dagenais GR, Leong DP, Rangarajan S, Lanas F, Lopez-Jaramillo P, Gupta R, Diaz R, Avezum A, Oliveira GBF, Wielgosz A, others. 2019. Variations in common diseases, hospital admissions, and deaths in middle-aged adults in 21 countries from five continents (PURE): a prospective cohort study. *The Lancet*.

Eraslan G, Avsec Ž, Gagneur J, Theis FJ. 2019. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*. DOI: 10.1038/s41576-019-0122-6.

Fakoor R, Ladhak F, Nazi A, Huber M. 2013. Using deep learning to enhance cancer diagnosis and classification. In: *Proceedings of the international conference on machine learning*.

Foss AH, Markatou M, Ray B. 2019. Distance Metrics and Clustering Methods for Mixed-type Data. *International Statistical Review* 87:80–109.

Géron A. 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc."

Greller LD, Tobin FL. 1999. Detecting selective expression of genes and proteins. *Genome Res* 9:282–296.

Guillen P, Ebalunode J. 2016. Cancer Classification Based on Microarray Gene Expression Data Using Deep Learning. In: *2016 International Conference on Computational Science and Computational Intelligence Cancer*. 208–216. DOI: 10.1109/CSCI.2016.269.

Han D, Kim J. 2018. Unified Simultaneous Clustering and Feature Selection for Unlabeled and Labeled Data. *IEEE transactions on neural networks and learning systems* 29:6083–6098.

Hunter JD. 2007. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* 9:90–95. DOI: 10.1109/MCSE.2007.55.

Komer B, Bergstra J, Eliasmith C. 2014. Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn. *Scipy 2014*:33–39.

Li J, Liu R, Zhang M, Li Y. 2017. Ensemble-based multi-objective clustering algorithms for gene expression data sets. In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. 333–340.

Liu J, Cai W, Shao X. 2011. Cancer classification based on microarray gene expression data using a principal component accumulation method. *Science China Chemistry* 54:802–811.

443   Liu S, Zhang J, Xiang Y, Zhou W, Xiang D. 2019. A Study of Data Pre-processing Techniques
444   for Imbalanced Biomedical Data Classification. *arXiv preprint arXiv:1911.00996*.

445   Michie ED, Spiegelhalter DJ, Taylor CC. 1994. Machine Learning , Neural and Statistical
446   Classification. *Technometrics* 37:459. DOI: 10.2307/1269742.

447   Min S, Lee B, Yoon S. 2016. Deep learning in bioinformatics. *Briefings in
448   bioinformatics*:bbw068.

449   Moosa JM, Shakur R, Kaykobad M, Rahman MS. 2016. Gene selection for cancer classification
450   with the help of bees. *BMC medical genomics* 9:47.

451   Motieghader H, Najafi A, Sadeghi B, Masoudi-Nejad A. 2017. A hybrid gene selection
452   algorithm for microarray cancer classification using genetic algorithm and learning
453   automata. *Informatics in Medicine Unlocked* 9:246–254. DOI: 10.1016/j.imu.2017.10.004.

454   Orozco-Arias S, Isaza G, Guyot R. 2019. Retrotransposons in Plant Genomes: Structure,
455   Identification, and Classification through Bioinformatics and Machine Learning.
456   *International Journal of Molecular Sciences* 20:3837. DOI: 10.3390/ijms20153837.

457   Orozco-arias S, Isaza G, Guyot R, Tabares-soto R. 2019. A systematic review of the application
458   of machine learning in the detection and classi fi cation of transposable elements. *Peerj*:1–
459   29. DOI: 10.7717/peerj.8311.

460   Orozco-Arias S, Núñez-Rincón AM, Tabares-Soto R, López-Álvarez D. 2019. Worldwide co-
461   occurrence analysis of 17 species of the genus *Brachypodium* using data mining. *PeerJ*
462   6:e6193. DOI: 10.7717/peerj.6193.

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

467   Perera K, Chan J, Karunasekera S. 2018. Feature Selection for Multiclass Binary Data. In:
468   *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 52–63.

469   Powell WB. 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality*.
470   John Wiley & Sons.

471   Sardana M, Agrawal RK. 2012. A comparative study of clustering methods for relevant gene
472   selection in microarray data. In: *Advances in Computer Science, Engineering &
473   Applications*. Springer, 789–797.

474   Sharma A, Imoto S, Miyano S. 2012. A top-r feature selection algorithm for microarray gene
475   expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics
476   (TCBB)* 9:754–764.

477   Sirinukunwattana K, Savage RS, Bari MF, Snead DRJ, Rajpoot NM. 2013. Bayesian hierarchical
478   clustering for studying cancer gene expression data with unknown statistics. *PloS one*
479   8:e75748.

480   Statnikov A, Tsamardinos I, Dosbayev Y, Aliferis CF. 2005. GEMS: a system for automated
481   cancer diagnosis and biomarker discovery from microarray gene expression data.
482   *International journal of medical informatics* 74:491–503.

483   Su AI, Welsh JB, Sapinoso LM, Kern SG, Dimitrov P, Lapp H, Schultz PG, Powell SM,
484   Moskaluk CA, Frierson HF, Hampton GM. 2001. Molecular classification of human
485   carcinomas by use of gene expression signatures. *Cancer Research* 61:7388–7393.

486   Tang Z, Steranka JP, Ma S, Grivainis M, Rodic N, Huang CRL, Shih I-M, Wang T-L, Boeke JD,
487   Fenyo D, Burns KH, Rodić N, Huang CRL, Shih I-M, Wang T-L, Boeke JD, Fenyö D,
488   Burns KH. 2017. Human transposon insertion profiling: Analysis, visualization and

489   identification of somatic LINE-1 insertions in ovarian cancer. *PROCEEDINGS OF THE*
490   *NATIONAL ACADEMY OF SCIENCES OF THE UNITED STATES OF AMERICA*
491   114:E733–E740. DOI: 10.1073/pnas.1619797114.
492 Varadhachary GR. 2007. Carcinoma of unknown primary origin. *Gastrointestinal cancer*
493   *research : GCR* 1:229–35.
494 Wang Y, Makedon FS, Ford JC, Pearlman J. 2005. HykGene: A hybrid approach for selecting
495   marker genes for phenotype classification using microarray gene expression data.
496   *Bioinformatics* 21:1530–1537. DOI: 10.1093/bioinformatics/bti192.
497 Wang S, Wei J. 2017. Feature selection based on measurement of ability to classify subproblems.
498   *Neurocomputing* 224:155–165.
499 Wold S, Esbensen K, Geladi P. 1987. Principal component analysis. *Chemometrics and*
500   *intelligent laboratory systems* 2:37–52.
501 Yue T, Wang H. 2018. Deep Learning for Genomics: A Concise Overview. :1–40.
502 Zou J, Huss M, Abid A, Mohammadi P, Torkamani A, Telenti A. 2018. A primer on deep
503   learning in genomics. *Nature Genetics*. DOI: 10.1038/s41588-018-0295-5.
504

Figure 1

Artificial neural network architectures used for cancer classification.

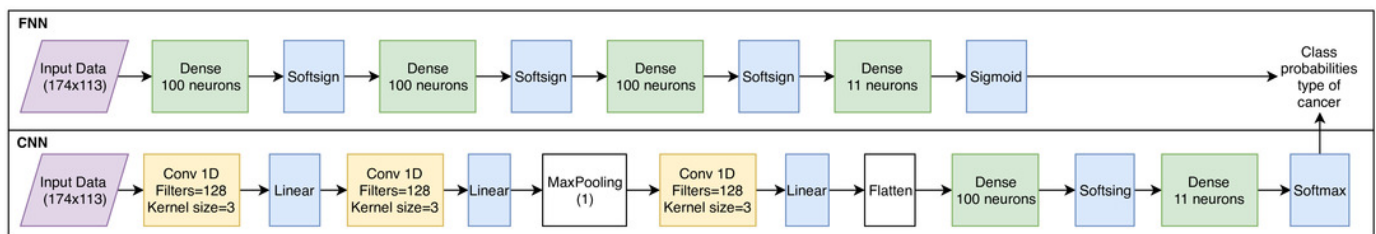



Figure 2

Hierarchical maps using Ward as clustering method and A) raw data B) scaled data, C) data reduced by PCA and D) data scaled and reduced by PCA.

Due to the large number of characteristics of the data set, it is recommended that you transform the data set to use only the most relevant and informative variables, which is called the preprocessing step.

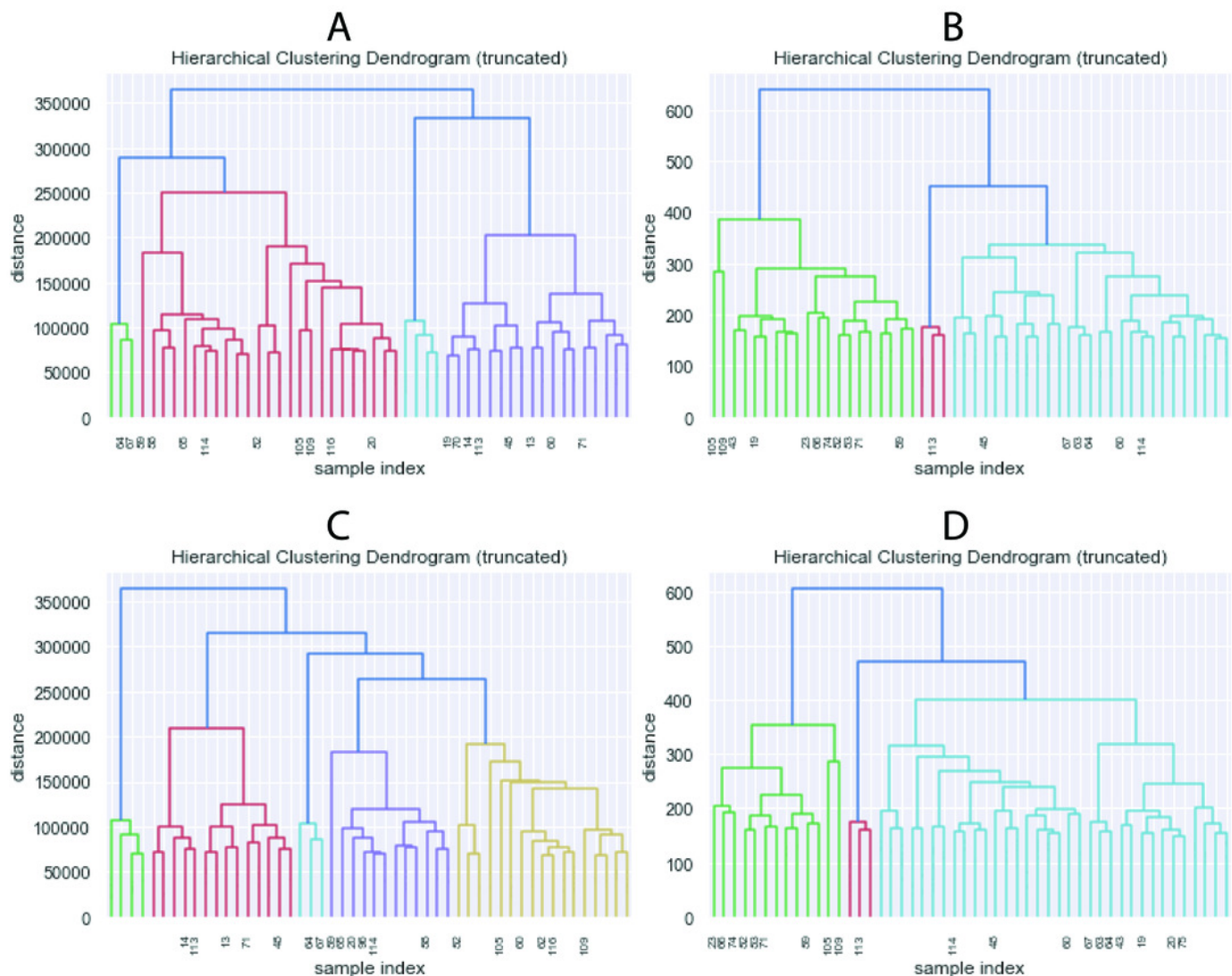



Figure 3

Hierarchical maps using Ward method as the criterion for choosing the pair of clusters to merge at each step.

This hierarchical map was generated by data without transformation and deleting their labels. Clustering approaches demonstrate whether the data contain relevant patterns for grouping.

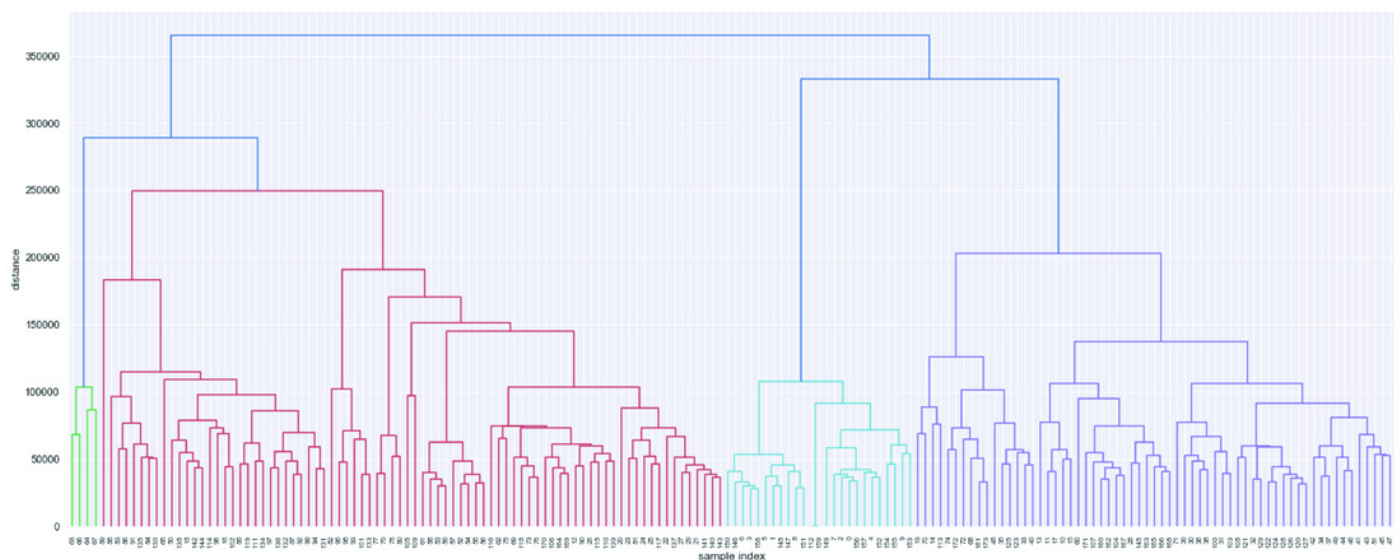



Figure 4

Hierarchical and heat map analysis utilizing A) raw data and B) Data processed by PCA.

These heat maps show how similar (near zero) or different (about 200,000) the individuals in the clusters are. A cluster is interesting when its members are very similar and are very different from individuals in other groups.

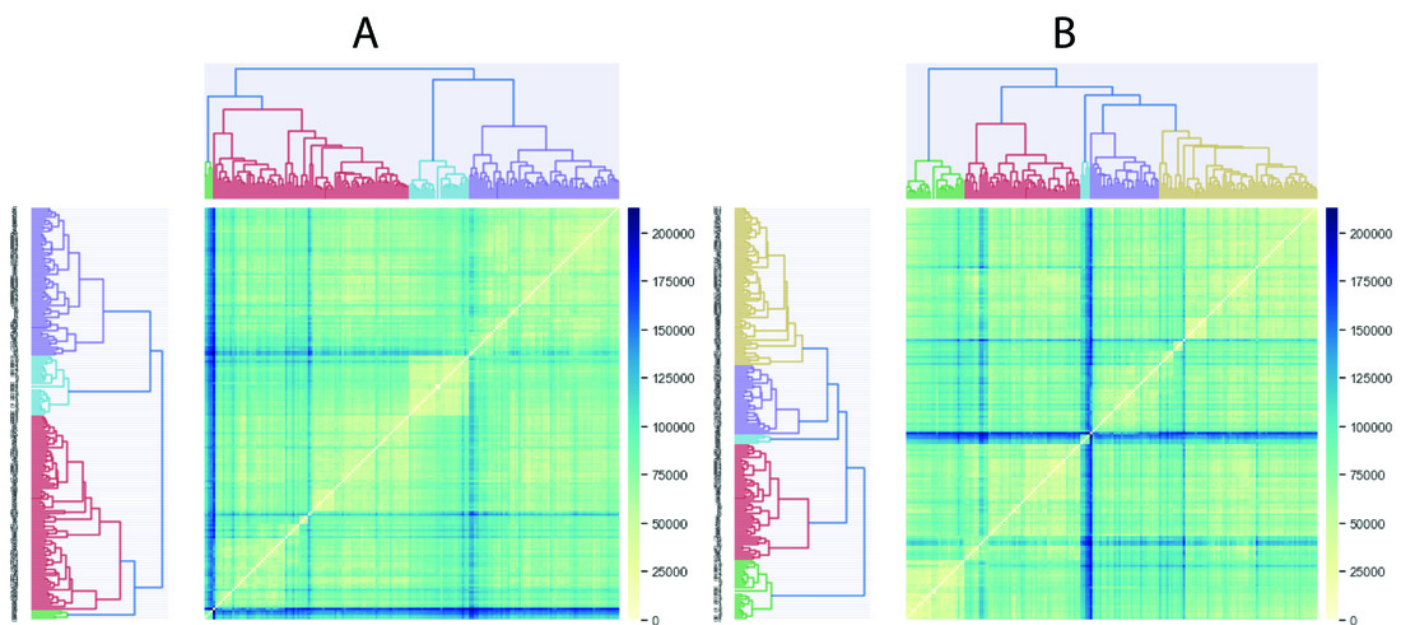



Figure 5

Clusters composition using A) raw data and B) Data processed by PCA.

Clustering was performed using Ward as distance algorithm. Label correspond to the cluster number predicted by the algorithm and may not correspond to labels of Table 1.

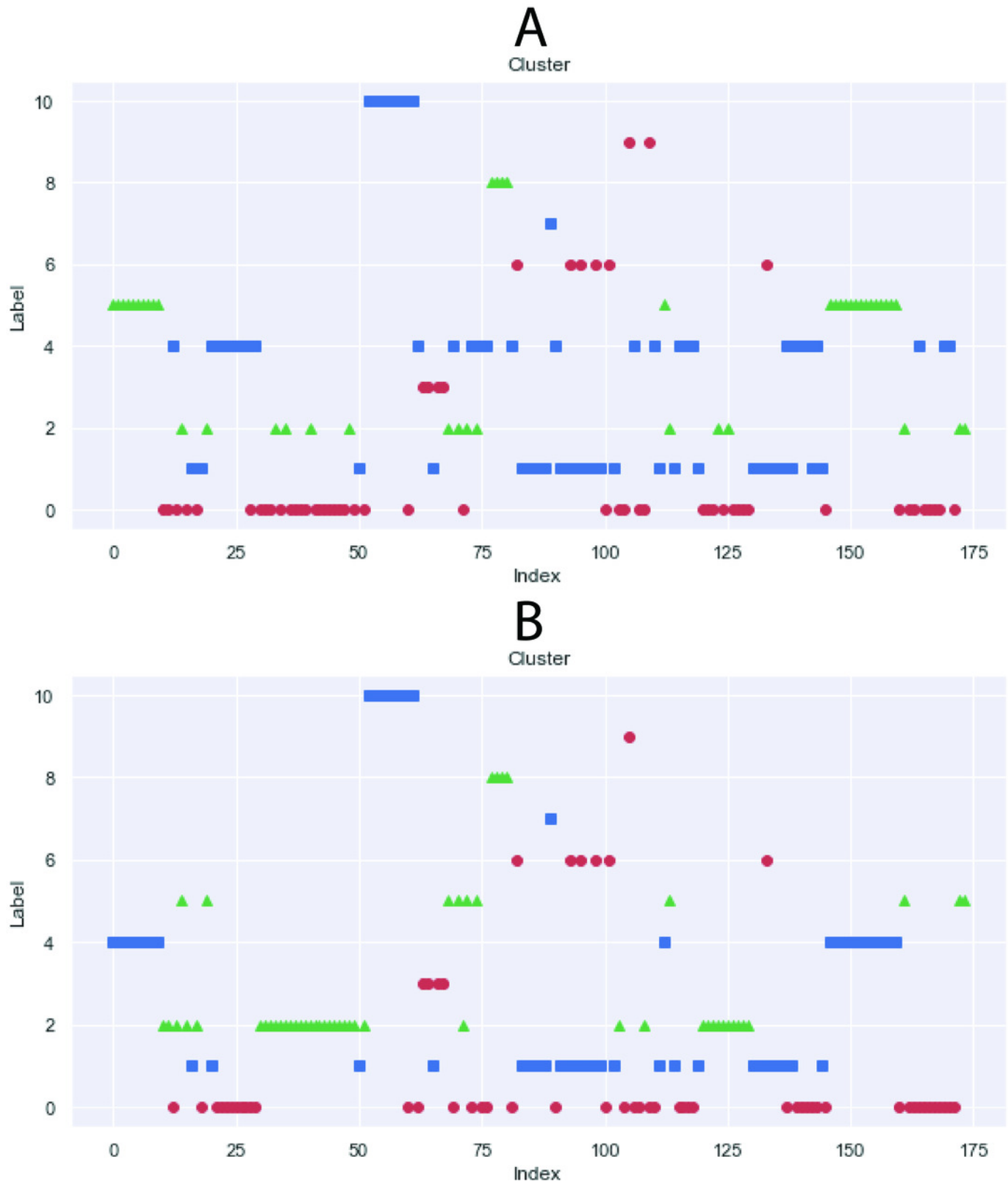



Figure 6

A) Behavior of Accuracy in terms of number of clusters and B) Confusion matrix with best results (clusters = 11) using K-means algorithm.

Results showed in 6 A are the accuracy using validation data set which correspond to 20% of whole data.

Figure 7

Comparison of training and validation accuracy between parameters using all data set and A) KNN, B) SVC, C) LG, D) MLP, and E) K-Means.

Algorithm do not present in this Figure, appear in Table 5 as default in column "Tuned Parameters".

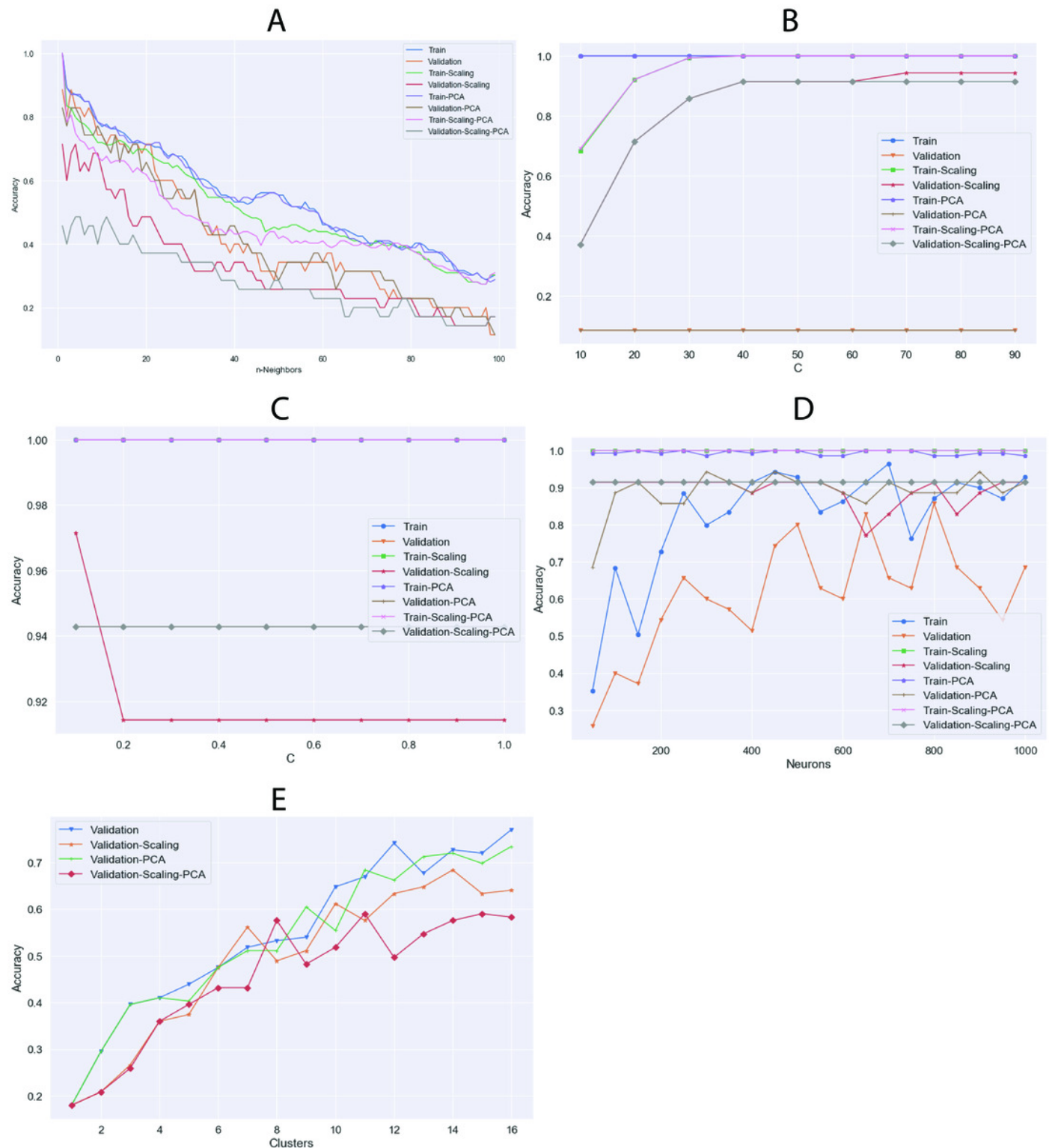



Figure 8

Confusion matrix of LG algorithm results.

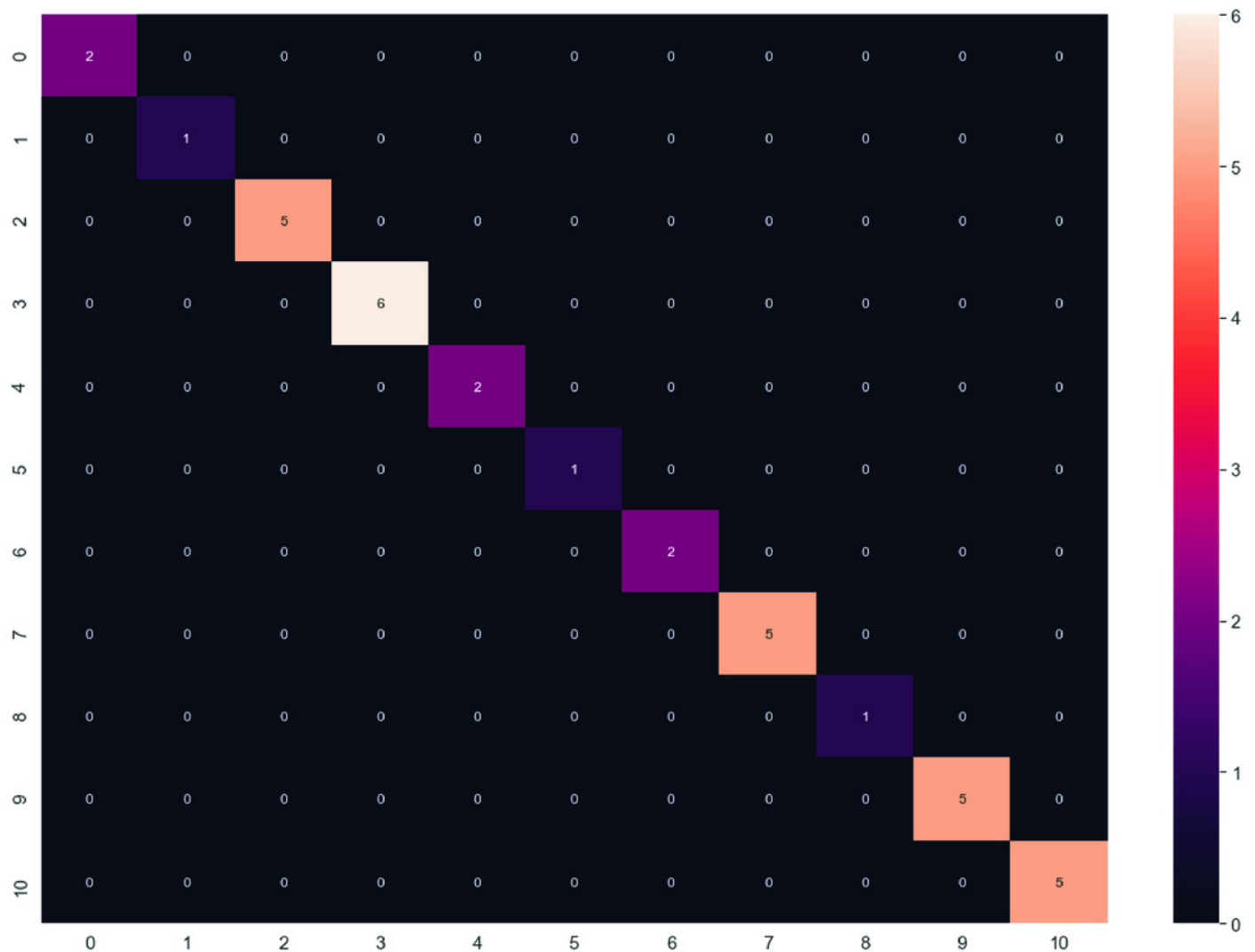



Figure 9

Results obtained by FNN architecture in training using 100 epochs.

A) Lost value and B) Accuracy. Lost function and accuracy is plotted on both training and validation data sets in order to observe behavior. When both data sets show very distant results, the architecture may be overfitting.

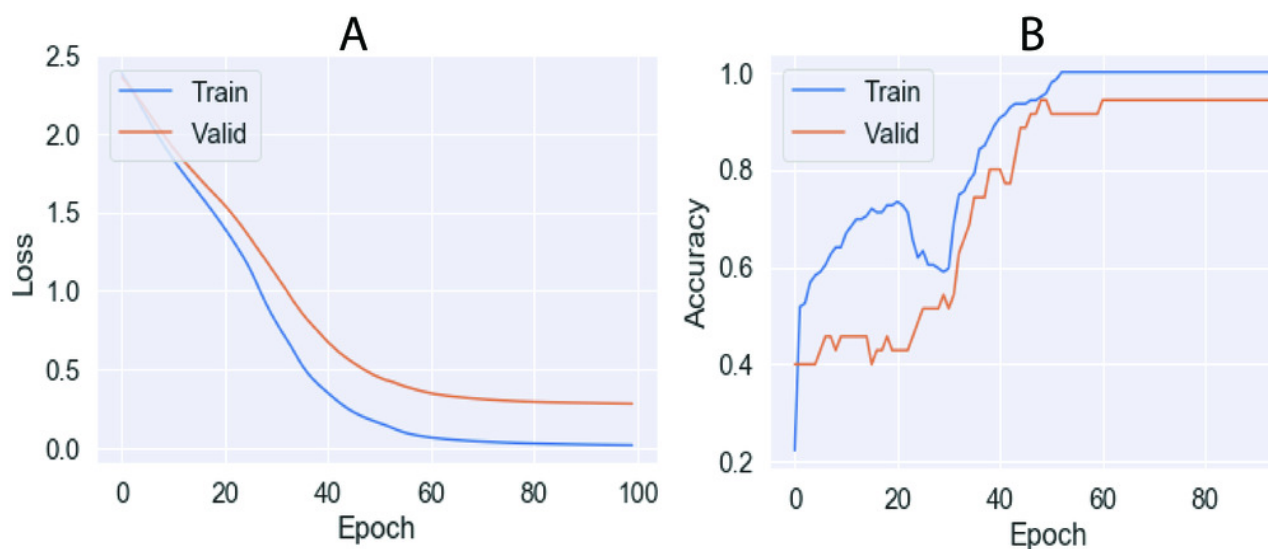
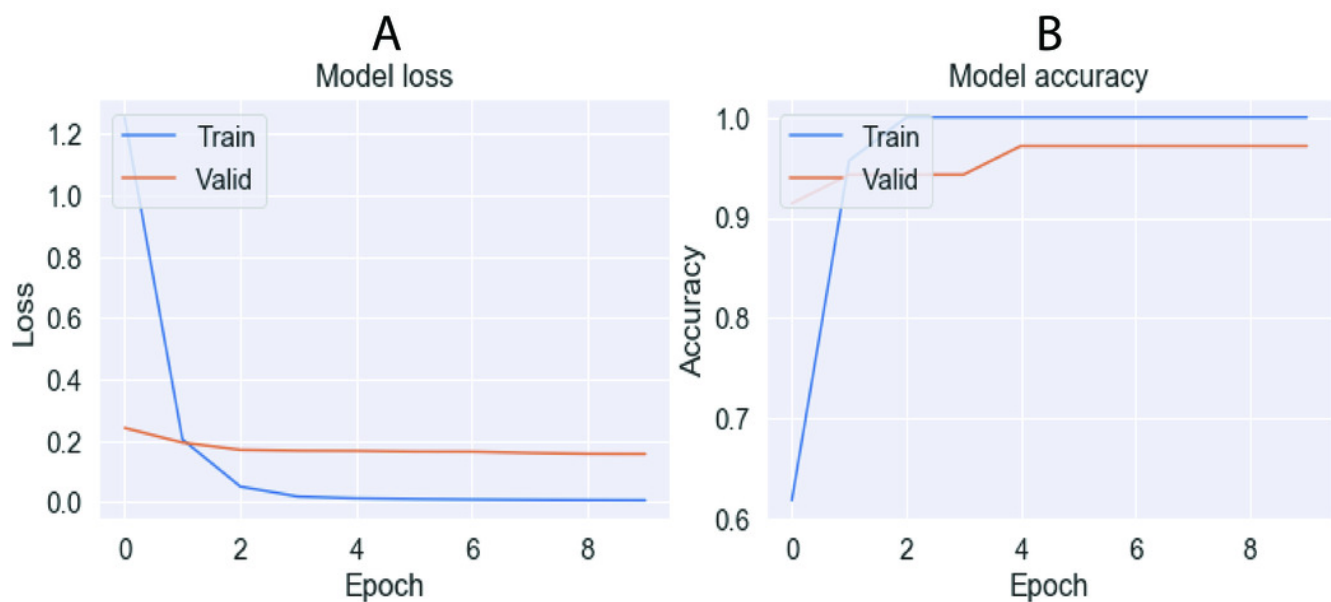



Figure 10

Results obtained by CNN architecture in training using 10 epochs.

A) Lost value and B) Accuracy. Lost function and accuracy is plotted on both training and validation data sets in order to observe behavior. When both data sets show very distant results, the architecture may be overfitting.

Cancer type classification in the 11_tumor data base.

Class	Cancer type	number of patients
0	Ovary	27
1	Bladder/Ureter	8
2	Breast	26
3	Colorectal	23
4	gastroesophagus	12
5	Kidney	11
6	Liver	7
7	Prostate	26
8	Pancreas	6
9	adenocarcinoma	14
10	lung squamous cell carcinoma	14

Tested Algorithm Parameters.

Algorithm	Parameter	Range	Step	Description
KNN	n_neighbors	1-99	1	Number of neighbors
SVC	C, gamma	C: 10-100, gamma: 1e-9 – 1e-4	C:10, gamma: 10	Penalty parameter C of the error term. Gamma is the free parameter of the Gaussian radial basis function
LG	C	0.1-1	0.1	Inverse of regularization strength
LDA	N/A	N/A	N/A	N/A
NB	N/A	N/A	N/A	N/A
MLP	solver='lbfgs', alpha=0.5, hidden_layer_sizes	50-1050	50	Number of neurons in hidden layers. In this study we used solver lbfgs and alpha 0.5
RF	n_estimators, max_depth, min_samples_split, max_features	n_estimators: 1 - 91, max_depth: 1 - 91, min_samples_split: 10 - 100, max_features: 10 - 90	10 for all parameters	N/A
DT	max_depth, min_samples_split, max_features	max_depth: 1 - 91, min_samples_split: 10 - 100, max_features: 10 - 90	10 for all parameters	N/A
K-Means	n_clusters, random_state=0	1-17	1	Number of clusters. In this study we used random state equals to zero.

Parameters tuned in DNNs.

Parameter	Values	Description
Batch size	10, 20,30, 40,50, 60,70, 80,90, 100	Number of training examples utilized in one iteration
Epochs	10, 50, 100, 200	Number of times that the learning algorithm will work through the entire training
Training Optimization Algorithm	SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam	tools that update model parameters and minimize the value of the loss function, as evaluated on the training set
Learning Rate	0.001, 0.01, 0.1, 0.2, 0.3	Hyper-parameter that controls how much the weights are being adjusting with respect to the loss gradient
Momentum	0.0, 0.2, 0.4, 0.6, 0.8, 0.9	Value between 0 and 1 that increases the size of the steps taken towards the minimum by trying to jump from a local minima
Network Weight Initialization	uniform, lecun_uniform, normal, zero, glorot_normal, glorot_uniform, he_normal, he_uniform	Initialization of weights into hidden layers of the network.
Neuron Activation Function	softmax, softplus, softsign, relu, tanh, sigmoid, hard_sigmoid, linear	How the neuron output is activated based on its inputs
Dropout Regularization	0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9	Process of randomly dropping out nodes during training
Weight constraint	1, 2, 3, 4, 5	Value that introduces a penalty to the loss function when training a neural network to encourage the network to use small weights
Number of Neurons in the Hidden Layers	1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100	Amount of neurons that composed each hidden layers of the network.

Cluster composition and original number of individuals from each class of cancer.

Class	Original number	Clustering using raw data	Clustering using data processed by PCA
0	27	47	47
1	8	29	28
2	26	16	39
3	23	4	4
4	12	31	25
5	11	25	10
6	7	6	6
7	26	1	1
8	6	4	4
9	14	2	1
10	14	9	9

Metrics obtained by K-Means for each cancer type.

Class	Precision	Recall	F1-Score
0	0.74	0.68	0.71
1	0	0	0
2	0.45	0.9	0.6
3	0.68	1	0.81
4	0	0	0
5	0.91	1	0.95
6	1	0.4	0.57
7	1	0.95	0.98
8	0	0	0
9	1	0.11	0.2
10	0.53	0.89	0.67

Tuning hyperparameters of best results of algorithms tested.

Results on validation data (The best result)			
Algorithm	**Conditions on the data set**	**Tuning parameters**	**% Accuracy**
KNN	Any	Neighbors=1	88.57
	Scaling	Neighbors: 1	71.43
	PCA	Neighbors: 1	82.86
	Scaling + PCA	Neighbors: 4	48.57
SVC	Any	C=10	8.57
	Scaling	C=70	94.29
	PCA	C=10	8.57
	Scaling + PCA	C=40	91.43
Logistic Regression	Any	C=0,1	100.00
	Scaling	C=0,1	97.14
	PCA	C=0,1	94.29
	Scaling + PCA	C=0,1	94.29
Linear Discriminant Analysis	Any	Default	91.43
	Scaling	Default	91.43
	PCA	Default	97.14
	Scaling + PCA	Default	82.86
Gaussian NB	Any	Default	85.71
	Scaling	Default	85.71
	PCA	Default	80.00
	Scaling + PCA	Default	71.43
Random Forest	Any	n_estimators=81, max_depth=91, min_samples_split= 10, max_features=50	97.14
	Scaling	n_estimators=91, max_depth=81, min_samples_split= 10, max_features=60	97.14
	PCA	n_estimators=91, max_depth=21, min_samples_split= 10, max_features=30	94.28
	Scaling + PCA	n_estimators=61, max_depth=11, min_samples_split= 10, max_features=20	85.71
Decision Tree	Any	max_depth=71, min_samples_split= 10, max_features=40	68.57
	Scaling	max_depth=51, min_samples_split= 10, max_features=60	68.57

	PCA	max_depth=81, min_samples_split= 10, max_features=30	82.85
	Scaling + PCA	max_depth=51, min_samples_split= 20, max_features=60	74.28
Multi-Layer Perceptron	Any	Neurons=800	85.71
	Scaling	Neurons=50	91.43
	PCA	Neurons=300	97.14
	Scaling + PCA	Neurons=50	91.43
K-Means	Any	Clusters=16	76.97
	Scaling	Clusters=14	68.34
	PCA	Clusters=16	73.38
	Scaling + PCA	Clusters=11	58.99

Cross validation of KNN, SVC, LG, MLP, K-Means, LDA, NB, RF and DT before and after of tuning process.

Results of Cross Validation (10 splits)					
	Before Tuning		After Tuning		Significance difference
Algorithm	% accuracy	Standard Deviation	% accuracy	Standard Deviation	
KNN	78.3	12.71	82.03	10.19	NO
SVC	10.82	6.65	81.98	13.7	YES
Logistic Regression	90.6	7.93	90.6	5.94	NO
Multi-Layer Perceptron	79.89	20.62	83.40	13.64	NO
K-Means	10.16	9.36	68.34	9.26	YES
Linear Discriminant Analysis	83.4	11.62	N/A	N/A	N/A
Gaussian NB	84.12	12.78	N/A	N/A	N/A
Random Forest	66.75	13.79	72.69	15.85	NO
Decision Tree	69.78	14.9	66.04	15.45	NO

Best value of hyperparameters tuned in deep neural networks.

	Best Value	
Parameter	**FNN**	**CNN**
Batch size	20	10
Epochs	100	10
Training optimization algorithm	Adagrad	SGD
Learn rate	0.2	0.1
Momentum	0	0
Network Weight Initialization	normal	glorot_normal
Neuron Activation Function	softsign	linear
Weight constraint	3	1
Dropout Regularization	0	0.4