Computer Methods and Programs in Biomedicine 210 (2021) 106367

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

On computational classification of genetic cardiac diseases applying iPSC cardiomyocytes

Martti Juhola[a,*], Henry Joutsijoki[a], Kirsi Penttinen[b], Disheet Shah[b], Katriina Aalto-Setälä[b,c]

[a] Faculty of Information Technology and Communication Sciences, Tampere University, 33014 Finland
[b] Faculty of Medicine and Health Technology, Tampere University, Finland
[c] Heart Center, Tampere University Hospital, 33520 Tampere, Finland

ARTICLE INFO

Article history:
Received 25 January 2021
Accepted 17 August 2021

Keywords:
Genetic cardiac diseases
Induced pluripotent stem cells
Cardiomyocytes
Transient profiles
Machine learning
Classification
Leave-one-out
K-fold cross-validation

ABSTRACT

Background: Cardiomyocytes differentiated from human induced pluripotent stem cells (iPSC-CMs) can be used to study genetic cardiac diseases. In patients these diseases are manifested e.g. with impaired contractility and fatal cardiac arrhythmias, and both of these can be due to abnormal calcium transients in cardiomyocytes. Here we classify different genetic cardiac diseases using Ca^{2+} transient data and different machine learning algorithms.

Methods: By studying calcium cycling of disease-specific iPSC-CMs and by using calcium transients measured from these cells it is possible to classify diseases from each other and also from healthy controls by applying machine learning computation on the basis of peak attributes detected from calcium transient signals.

Results: In the current research we extend our previous study having Ca-transient data from four different genetic diseases by adding data from two additional diseases (dilated cardiomyopathy and long QT Syndrome 2). We also study, in the light of the current data, possible differences and relations when machine learning modelling and classification accuracies were computed by using either leave-one-out test or 10-fold cross-validation.

Conclusions: Despite more complex classification tasks compared to our earlier research and having more different genetic cardiac diseases in the analysis, it is still possible to attain good disease classification results. As excepted, leave-one-out test and 10-fold cross-validation achieved virtually equal results.

1. Introduction

Since their discovery induced pluripotent stem cells (iPSC) have been widely utilized for scientific research purposes and they hold great promise for use in biomedical research and development [1]. Patient-specific iPSC-derived cardiomyocytes (iPSC-CMs) offer an attractive experimental platform to model cardiac functionality and diseases.

Calcium cycling has an important role in extraction-contraction coupling of cardiomyocytes since it is the central regulator of in cardiac contraction and relaxation. Cardiac diseases often cause variability and distortions in calcium cycling of cardiomyocytes and affect their functionality. Such distortion abnormalities in calcium transients can represent a patient's cardiac phenotype [2]. Detection and characterization of these distortions are important, especially, because they can be used in order to develop recognition and diagnostics of cardiac diseases. Thus far, machine learning methods have rarely been used for data associated with iPSC-derived cardiomyocytes. Machine learning has been applied at least to analyze electrophysiological effects made by chronotropic drugs [3] and mechanistic action of drugs in cardiology [4].

We started our preliminary research of calcium transients measured from iPSC-CMs generated from genetic cardiac disease patients in order to recognize calcium transient peaks from calcium signals and classified regularly i.e. normally cycling from abnormally cycling calcium peaks by means of signal analysis and machine learning algorithms [5]. After this we discovered that it is possible to separate three different genetic cardiac diseases from each other and from controls (wild type, WT) [6]. These diseases were long QT syndrome 1 (LQT1), an electric disorder of the heart that predisposes patients to arrhythmias and sudden cardiac death [7], hypertrophic cardiomyopathy (HCM), disorder that affects the structure of heart muscle tissue leading to arrhythmias and progressive heart failure [8] with a myosin -binding protein C gene

* Corresponding author at: Faculty of Information Technology and Communication Sciences, 33014 Tampere University, Finland.
E-mail address: Martti.Juhola@tuni.fi (M. Juhola).

https://doi.org/10.1016/j.cmpb.2021.106367

mutation (HCMM), and catecholaminergic polymorphic tachycardia (CPVT), an exercise-induced malignant arrhythmogenic disorder [9]. We also found that in order to separate between those diseases and healthy controls it was not necessary to differ calcium signals that were either entirely normal cycling from abnormally cycling signals. Next, we extended our existing signal data and added another HCM disease mutation [10], an α-tropomyosin of the β-myosin heavy chain mutation (HCMT). Our newest research described that is possible to study effects of a drug on calcium transients and to classify and separate drug effects with machine learning method [11].

In this research we extend our data set by adding transient signals of controls and cardiac diseases with two additional genetic cardiac diseases: dilated cardiomyopathy (DCM), a disease of the heart muscle, and long QT syndrome 2 (LQT2), a disease with electrical problems in cardiomyocytes. Thus, there are seven different classes of genetic cardiac diseases with two mutation for HCM, two types of LQTS (LQT1 and LQT2), one mutation for DCM and several mutations for CPVT and then healthy cardiomyocytes from control individuals which makes their computational modelling more complex than in our previous research [5,6,10,11]. We also compare and ponder the use of leave-one-out and 10-fold cross-validation and their possible effects of classification accuracy while building models with several different machine learning methods.

2. Materials

Cell studies and data collection was conducted in Tampere University and was approved by the Ethics Committee of Pirkanmaa Hospital District in order to culture and differ human induced pluripotent stem cell lines (permit R08070). Patient-specific iPSC lines were established and described as previously [6]. iPSC lines were generated from two LQT1 and two LQT2 patients, two HCMT and two HCMM patients, six CPVT patients, two DCM patients, and two healthy control individuals (WT). The studied iPSC lines were UTA.05605.CPVT, UTA.05208.CPVT, UTA.07001.CPVT, UTA.03701.CPVT, UTA.05503.CPVT, and UTA.05404.CPVT generated from CPVT patients carrying cardiac ryanodine receptor (RyR2) mutations; UTA.07801.HCMM, and UTA.06108.HCMM generated from HCM patients carrying myosin-binding protein C (MYBPC3) mutations and UTA.02912.HCMT and UTA.13602.HCMT generated from HCM patients carrying α-tropomyosin (TPM1); and UTA.00208.LQT1 and UTA.00118.LQT1 generated from LQT1 patients carrying potassium voltage-gated channel subfamily Q member1 (KCNQ1) mutation; UTA.03412.LQT2, UTA 03417.LQT2, UTA.03809.LQT2 and UTA.03810.LQT2 generated from LQT2 patients carrying the human ether-a-go-go-related gene (HERG) mutation; UTA.12619.LMNA and UTA.12704.LMNA generated from DCM patients with lamin A and lamin C (LMNA) mutations and UTA.04602.WT and UTA.04511.WT generated from healthy control individuals. iPSCs were differentiated into spontaneously beating cardiomyocytes and dissociated into coverslips for calcium imaging studies, which were conducted in spontaneously beating Fura-2 AM (Invitrogen, Molecular Probes) or Fluo-4 AM (Thermo Fisher Scientific) loaded cardiomyocytes as described earlier [7,8,9,12,13] For calcium analysis, regions of interest were selected for spontaneously beating cells, and background noise was subtracted before further processing. Every calcium transient signal was a recording from one cardiomyocyte. There were 90 LQT1, 270 HCMM, 149 HCMT, 233 CPVT, 138 LQT2, 67 DCM and 226 WT signals. All in all, there were 1173 calcium transient signals. The approximate sampling frequencies were 8 Hz for LQT1, 23 Hz for HCMM and WT, 13 Hz for CPVT, 14 Hz for 54 HCMT and 23 Hz for 95 HCMT signals, 33 Hz for LQT2 and 38 Hz for DCM signals. The sampling frequency was improved (increased) in the long run while updating the measuring device.

3. Peak attributes computed from calcium transient signals

Categorization of iPSC-CM calcium transients to normal and abnormal signals was determined by an expert biotechnologist. In LQT1 transient signals the share of abnormally cycling signals was 69%, in HCMM it was 37%, in CPVT 53%, in HCMT 44%, in LQT2 72%, in DCM 62% and in WT 14%. Abnormality was defined according to remarkably irregular amplitude or duration of calcium peaks whereas normally cycling peaks had regular amplitude and duration of peaks. However, as said above and shown by our earlier research [6,7], this property, either normal or abnormal transient signals, did not affect how well these transient signals could be classified into different classes. Therefore, we used them as such, without computing separately abnormal and normal transient signals. Fig.1 exemplifies a normal LQT2 signal and abnormal LQT2 signal. Correspondingly, Fig. 1 also shows those of DCM signals. Examples of calcium transient signals of other diseases and controls can be found from the figures in our two earlier articles [6,10]. Often the forms and sizes of peaks may vary noticeably, especially in abnormal transient signals.

After recognizing all acceptable peaks in a signal, attribute values of those peaks were computed. In our first research [6] for the machine learning classification of three genetic cardiac diseases and controls, we applied the first ten attributes to be given in the following. Later, up to our most recent research [11] and the current research we designed additional four attributes to obtain more information about peaks of calcium transient signals. Thus, we applied 14 peak attributes illustrated in Fig. 2. Left amplitude is equal to the difference of the peak maximum and the amplitude value of the peak beginning. Right amplitude was computed from the peak maximum and end. Left duration is equal to the time difference from the peak beginning to the maximum, and right duration is the time difference from the peak maximum to the peak end. Next, the maximum of the approximated first derivative values from the peak beginning to the peak maximum (from the left peak side) was computed. Then the minimum of the first derivative values from the peak maximum to the peak end was evaluated. To apply this as a positive value its absolute value was taken. The maximum of the second derivative as well as its absolute minimum were computed from the right peak side only, since sometimes the left side of rather small peaks could be so low containing only a few samples that approximating the maximum and absolute minimum of the second derivative values would not succeed well. The surface area of a peak was computed as the sum of amplitude differences of peak curve values and values from a line calculated from the peak beginning to its end. Peak-to-peak interval was formed as the time difference from the maximum of the current peak and that of the preceding peak. In the case of the first peak in a signal it was estimated from the signal beginning. Next, time difference was computed from the location of the first derivative maximum of the peak left side to the peak beginning. Time difference was also computed from the location of the absolute minimum of the peak right side to the peak maximum. Next, an attribute called mean peak duration was computed in the following way. First, running along peak left and right amplitudes their amplitude halves were approximated, second the mean of those two half amplitudes was calculated, and then two intersection locations were computed for the horizontal line of this mean and peak curve. If rarely a peak was very asymmetric as to its left and right side amplitudes so that either the left side amplitude was smaller than the half of the right side amplitude or vice versa, the half of the smaller side amplitude was not used, but its whole amplitude. In other words, the mean of the smaller side amplitude and the half of the greater side amplitude was computed. Time difference of those two intersection locations is the attribute value called mean peak duration. Ultimately, peak curve length was ap-

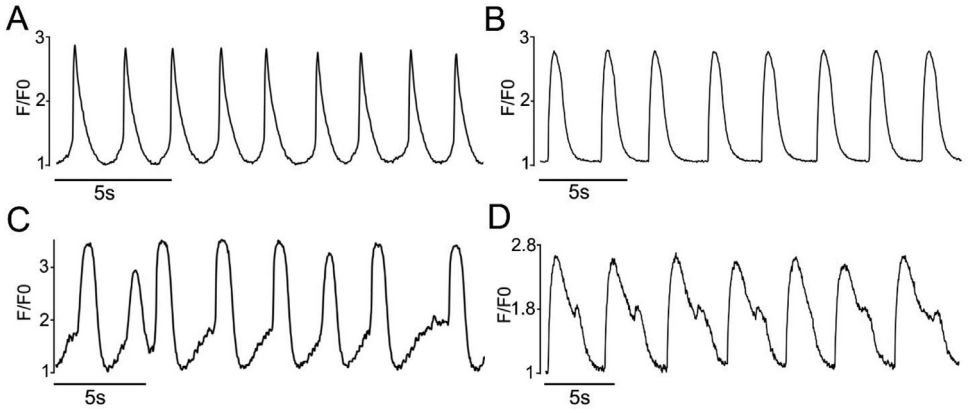

Fig. 1. (A) Approximately 18 s segment of a normal LQT2 calcium transient signal with regular peak shapes and sizes. (B) Around 23 s segment of a normal DCM calcium transient signal with regular peak forms. (C) 23 s of an abnormal LQT2 transient signal containing irregular peak forms and delayed calcium rise. (D) 31 s of an abnormal DCM transient signal including rather irregular peak forms with delayed calcium decay.

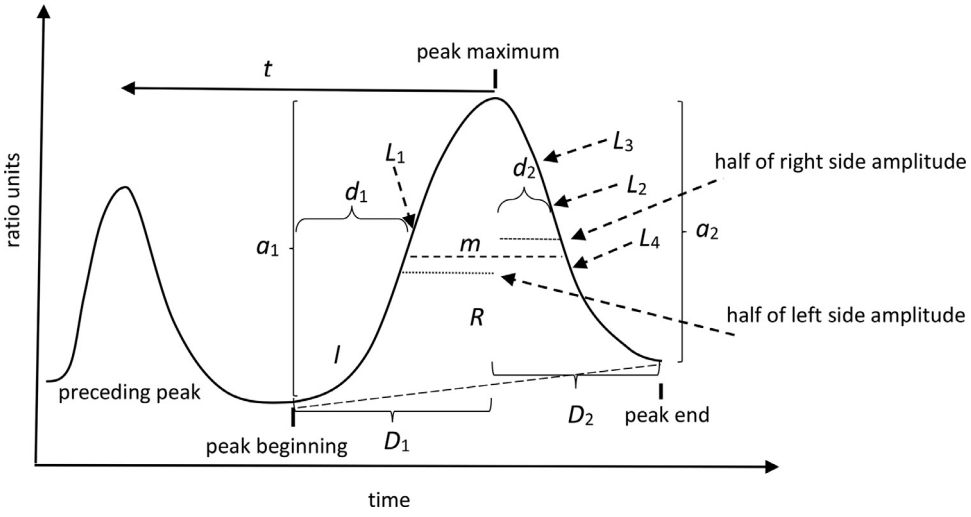

Fig. 2. Peak attributes computed: (1) left a_1 and (2) right a_2 amplitudes, (3) left D_1 and (4) right D_2 durations, approximate location L_1 of the computation of (5) first derivative max(s') for a calcium transient signal s, approximate location L_2 for (6) absolute first derivative |min(s')|, location L_3 for (7) second derivative |min(s")| and location L_4 for (8) second derivative max(s"), (9) surface area R, (10) time interval t from the maximum of the preceding peak, (11) duration d_1 from the peak beginning to L_1, (12) duration d_2 from the peak maximum to L_2, (13) mean peak duration m, and (14) approximated peak curve length l.

Table 1
Numbers separately for abnormal and normal calcium transient signals, number of peaks per disease or controls, number of cell lines and signals per cell line.

Disease or controls	Number of abnormal signals	Number of normal signals	Total number of peaks	Number of cell lines	Signal numbers of cell lines
LQT1	62	28	1635	2	45, 41
HCMM	100	170	4413	2	66, 204
CPVT	119	114	2311	6	42, 25, 55, 20, 59, 32
WT	31	195	2850	2	27, 199
HCMT	65	84	2136	2	106, 43
LQT2	99	39	3870	4	35, 49, 28, 26
DCM	40	27	1172	2	35, 32

proximated by running signal value by value from the following sample location after the peak beginning to the peak end, estimating the Euclidean distance from the current location to the preceding one and summing up all these distances.

The minimum, average and maximum lengths of all 1173 calcium transient signals were 7.7 s, 22.7 s and 46.5 s. The minimum, average and maximum numbers of peaks per signal were 1, 15.7 and 123. Altogether, 18387 peaks were recognized in six diseases classes and controls. See Table 1. After the recognition of peaks in all signals, values of the above-mentioned 14 attribute were computed. In order to study the importance of the current attributes for disease classification, we computed with ReliefF algorithm (in MATLAB as all our data analysis and classification) importance values as depicted in Fig. 3.

In Table 2 the means and standard deviations of 14 attributes are presented. The means of different diseases mostly differ from each other. From the attribute peak-to-peak interval t in Table 2 average peak frequencies can be calculated as its inverse values for different classes: 0.85 Hz for LQT1, 0.69 Hz for LQT2, 1.25 Hz for HCMM, 0.98 Hz for HCMT, 0.88 Hz for CPVT, 0.50 for DCM and 0.52 Hz for WT.

Naturally, different sampling frequencies used are no ideal situation, but this was needed when collecting them took time and the software used was developed along with time. The difference

Table 2
Means and standard deviations for 20 cell lines: left and right side amplitudes a_1 and a_2, left and right side durations D_1 and D_2, left side maximum of first derivative max(s'), right side absolute minimum |min(s')|, right side maximum max(s'') and absolute minimum |min(s'')|of second derivative, peak surface area R, peak-to-peak time interval t, duration d_1 from the peak beginning to the maximum of left side first derivative, d_2 duration from the peak maximum to the first derivative absolute minimum, mean peak duration m and peak curve length l. Two lowermost cell lines are from disease DCM.

Cell lines														
UTA.00208.LQT1	189±81	190±82	0.311±0.180	0.750±0.437	910±477	523±269	1526±1422	1114±1538	68±45	1.245±1.064	0.204±0.143	0.158±0.078	0.417±0.159	381±162
UTA.00118.LQT1	151±71	153±72	0.340±0.174	0.617±0.353	722±445	494±247	1706±1210	1304±1307	47±36	1.090±0.746	0.228±0.147	0.149±0.075	0.382±0.142	305±142
	192±108	194±102	0.326±0.165	0.553±0.345	1741±852	893±342	4720±2242	3449±2941	58±51	0.986±0.620	0.230±0.139	0.121±0.067	0.286±0.103	402±207
UTA.06108.HCMM														
	199±90	202±91	0.258±0.143	0.455±0.233	2007±889	1066±461	6341±3314	3429±3140	49±41	0.753±0.451	0.180±0.115	0.114±0.073	0.250±0.116	419±185
UTA.07801.HCMM														
UTA.05605.CPVT	231±138	233±137	0.469±0.181	1.001±0.476	1696±880	746±394	3012±1944	3410±2824	119±106	1.970±1.224	0.269±0.120	0.152±0.083	0.460±0.179	498±299
UTA.05404.CPVT	136±88	139±86	0.280±0.153	0.435±0.296	746±459	539±233	1857±1358	1491±1625	46±46	0.814±0.719	0.161±0.116	0.138±0.060	0.358±0.118	277±170
UTA.07001.CPVT	285±179	289±187	0.266±0.127	0.560±0.374	1872±1316	1144±657	4414±3330	2583±3586	80±71	0.911±0.637	0.163±0.100	0.111±0.044	0.315±0.108	588±362
UTA.03701.CPVT	291±229	292±226	0.454±0.250	0.762±0.464	1178±982	820±598	2098±1735	1267±1574	122±127	1.382±1.019	0.296±0.209	0.179±0.082	0.463±0.175	586±451
UTA.05208.CPVT	293±173	296±175	0.338±0.170	0.679±0.324	1716±930	1037±600	3699±2720	1679±2492	89±64	1.047±0.573	0.235±0.153	0.151±0.093	0.361±0.115	603±338
UTA.05503.CPVT	276±203	278±202	0.351±0.239	0.692±0.552	2183±1411	1029±547	4566±3507	4452±4099	128±219	1.306±1.087	0.242±0.153	0.145±0.069	0.369±0.208	595±436
UTA.04602.WT	272±170	275±172	0.492±0.263	1.039±0.601	2131±1276	927±635	4465±3386	3938±4359	132±115	1.944±1.580	0.312±0.198	0.156±0.145	0.471±0.259	575±341
UTA.04511.WT	159±69	159±72	0.479±0.279	0.951±0.658	1609±680	561±200	3316±1561	4305±2888	90±89	1.927±2.285	0.257±0.162	0.150±0.140	0.491±0.291	354±149
UTA.02912.HCMT	213±148	217±152	0.373±0.193	0.501±0.348	1529±1153	952±514	4503±3299	3795±3979	67±77	1.003±0.741	0.272±0.177	0.110±0.054	0.287±0.100	452±306
UTA.13602.HCMT	157±68	163±70	0.418±0.167	0.526±0.277	1288±568	757±315	3952±2109	2261±2019	45±30	1.079±0.609	0.343±0.162	0.097±0.045	0.260±0.073	331±144
UTA.03412.LQT2	137±75	135±78	0.285±0.226	0.714±0.627	1656±950	577±263	3353±3426	3207±3483	55±68	1.150±1.335	0.108±0.093	0.132±0.128	0.388±0.298	292±145
UTA.03417.LQT2	129±69	130±72	0.285±0.220	0.724±0.436	1749±916	566±255	3577±2341	2071±2337	42±57	1.330±1.973	0.189±0.157	0.081±0.097	0.280±0.171	273±138
UTA.03809.LQT2	158±132	158±147	0.427±0.255	0.765±0.636	1170±870	549±420	3477±2567	3721±3327	96±115	1.803±2.342	0.227±0.161	0.236±0.233	0.489±0.312	349±282
UTA.03810.LQT2	147±118	153±135	0.463±0.261	1.074±0.635	1475±1045	453±295	2516±1124	2976±2248	82±110	2.309±1.917	0.273±0.190	0.221±0.238	0.515±0.243	334±251
UTA.12619.LMNA	62±53	59±59	0.387±0.238	1.127±0.907	611±569	195±148	1647±1281	1419±1229	39±46	1.994±1.849	0.140±0.083	0.314±0.311	0.625±0.450	135±102
UTA.12704.LMNA	77±48	76±49	0.404±0.240	0.935±0.683	864±568	263±160	2398±2012	2578±2417	44±55	2.018±2.131	0.218±0.135	0.263±0.328	0.524±0.434	173±99

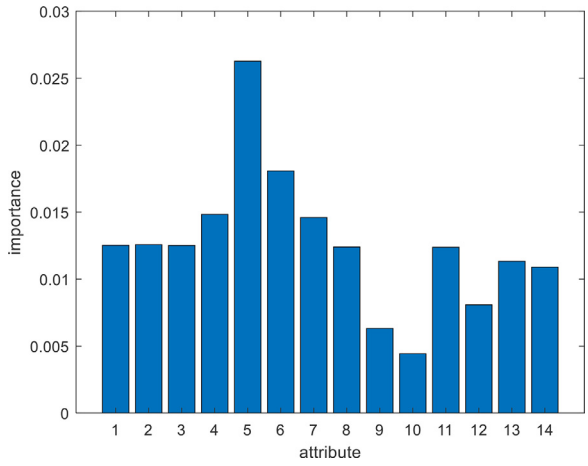

Fig. 3. Importance values for 14 attributes in the same order as described in the caption of Fig. 2 when the averages of 9 test runs were computed applying 9 nearest neighbor numbers of 3, 5, 7, 11, 15, 23, 31, 41, 51 to searching in ReliefF algorithm. The fifth attribute of first derivative max(s') obtained the greatest value indicating it to be more important than the other for classification. Since no attribute importance value was very close to 0, all these attributes are useful for classification of peaks.

between the lowest frequency (f) 8 Hz and the highest 38 Hz is – by time interval T=1/f - approximately 0.125 s – 0.026 s = 0.099 s. Thinking that this difference would be distributed for measured values, e.g., the beginning of a peak in a signal, the difference given by these two frequencies would be from 0 to 0.099 s. When this type of random measuring value follows a uniform distribution (any value between 0 and 0.099 equally probable), its mean is 0.99 s/2. When it is similar for measuring the peak maximum, we would obtain mean 2 • 0.099 s/2= 0.099 s while measuring the beginning and end for the duration D_1 of the left side of a peak. D_1 was one of six attributes directly depending on time. For other five duration attributes D_2, t, d_1, d_2, and m, the differences of their means in Table 2 are, however, mostly greater than those of D_1. Therefore, in principle D_1 would be more critical than those other.

To be more exact, the distribution of the different sampling frequencies have to be taken into account: 90 signals for 8 Hz, 233 signals for 13 Hz and 54 signals for 14 Hz (these two close to each other), 601 signals for 23 Hz, 138 signals for 33 Hz and 67 signals for 38 Hz. Thus, only 90/1173 (total number of signals) or 7.5 % and 67/1173 or 5.7 % originated from the lowest and highest sampling frequencies. This means that the difference of 0.099 s could only occur in a small minority of all possible classification computations. More than half, 51.2 % of all signals were sampled at 23 Hz as WT, HCMM and HCMT data. The approximate frequency 23 Hz is around 15 Hz greater than 8 Hz and 15 Hz less than 38 Hz, i.e., approximately in the middle of minimum 8 Hz and maximum 38 Hz giving approximate symmetry in the frequency distribution around the most used frequency of 23 Hz. Mean random time difference between 23 Hz and 38 Hz would be around 0.017 s according to 1/23 Hz - 1/38 Hz. All these mean that any possible random difference caused by these different sampling frequencies used would be essentially smaller than the maximum 0.099 s for the great majority (around 87 %) of signals and, thus, their classification. Since 51.2 % originated from “the middle” frequency of 23 %, this means that inside this largest part of the signals there is no difference between sampling frequencies. To conclude, any actual average random effect for time attributes of all signals is much smaller than that largest of 0.099 s.

The attributes depending on amplitude (all the rest in addition to the five mentioned) are more complicated. However, all of them also depend on time. Therefore, they also partly follow the preceding analysis.

As an example, two different cell lines of DCM disease are visualized in Fig. 4(A). Subject to the numbers of signals and their peaks this was the smallest class. There were 639 peaks in 35 signals of cell line 12619.DCM and 533 peaks in 32 signals of cell line 12704.DCM. In Fig. 4(B) there are 3870 peaks of cell lines 0341.LQT2, 03417.LQT2, 03809.LQT2 and 03810.LQT2. When the cell lines cover mostly the same areas in each visualization, this reflects such distributions that possible differences of the peaks between cell lines of each disease class are minor in the current data. In addition, Fig. 5 shows the presentation computed for the data by all cell lines.

4. Technical specifications for classification

The genetic cardiac disease data examined in this study is challenging. We have seven classes included (wild type and six genetic cardiac diseases) in our dataset and the total number of signals is 1173. Since the amount of data is still relatively small when taking into account the number of classes, we cannot talk about big data yet in this context compared to many image classification tasks where there are tens of thousands of annotated images available for research. The data collection procedure is a laborious task to do in practice and begins from finding the voluntary patients and controls for the research. Then the actual iPSC reprogramming process is made and the iPSCs are differentiated into cardiomyocytes. From cardiomyocytes the calcium transient signals are measured which is final the stage of data collection process. After that preprocessing of signals is made including tasks such as finding peaks from signals and extracting peak-based variables. When the preprocessing of the signals has been made, classification by means of machine learning methods can be performed. When considering the practical limitations behind the classification of genetic cardiac diseases, the classification method must be selected and fine-tuned carefully. Nowadays, deep learning solutions are popular in different domains and have become a standard approach to use. However, deep learning methods require large amounts of training data, which is a hard constraint to overcome. For genetic cardiac disease classification, interesting deep learning solutions would be to use, for example, RNN networks [14] and LSTM networks [15]. Nevertheless, we have omitted deep learning solutions in this paper due to limited dataset and concentrated on methods, which can handle small datasets well.

Majority of the methods used were applied also in our earlier studies [5,6,10,11,16,17,18], but we have included new method which has not been used in aforementioned studies. Furthermore, we examined two different test set-ups (leave-one-signal data-out, 10-fold cross-validation) giving new perspective to classification of cardiac diseases compared to earlier studies and these are explained in subsequent subsections in detail. All experimental tests were performed using Dell Precision Tower 7810 workstation having dual Intel Xeon E5-2640 v4 @ 2.40GHz processors, 128 GB RAM and Win10 Pro operating system. Tests were made using MATLAB 2019b with Parallel Computing Toolbox and Statistics and Machine Learning Toolbox.

As a first method, we applied k-Nearest Neighbor (kNN) classifier [19,20] with different (k value, distance measure, distance weighting) triplets. We tested altogether 19 different odd values of k ($k \in \{1, 3, 5, \ldots, 37\}$). Justification behind testing only odd k values is to decrease the possibility of ending up to a tie, which would again need special attention. If a tie, however, occurred in a classification, it was solved by random choice. In the case of distance measures, we examined eight alternatives. These were Chebychev, Manhattan, correlation, cosine, Euclidean, standardized Euclidean, Mahalanobis and Spearman distance measures. Finally,

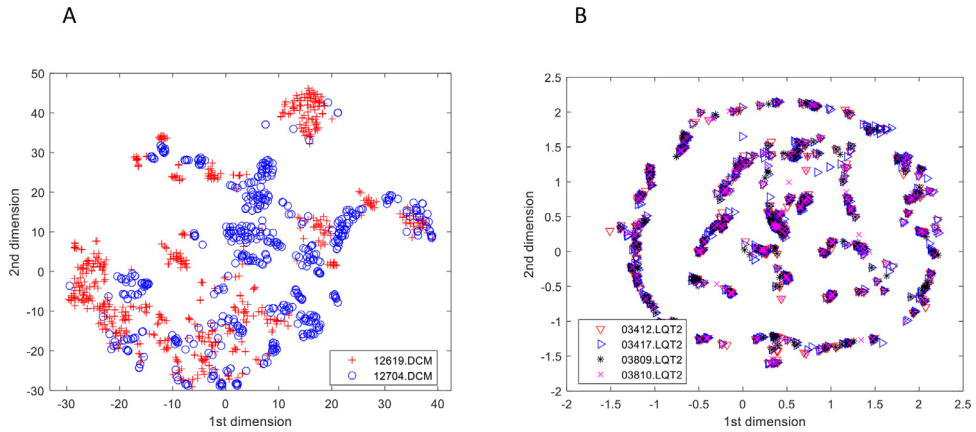

Fig. 4. (A) Peaks of cell lines 12619.DCM and 12704.DCM follow principally the same graphical distribution. (B) Peaks of cell lines 0341.LQT2, 03417.LQT2, 03809.LQT2 and 03810.LQT2 are considerably concentrated in the same locations indicating their similar properties. These were computed with t-SNE algorithm (t-Distributed Stochastic Neighbor Embedding).

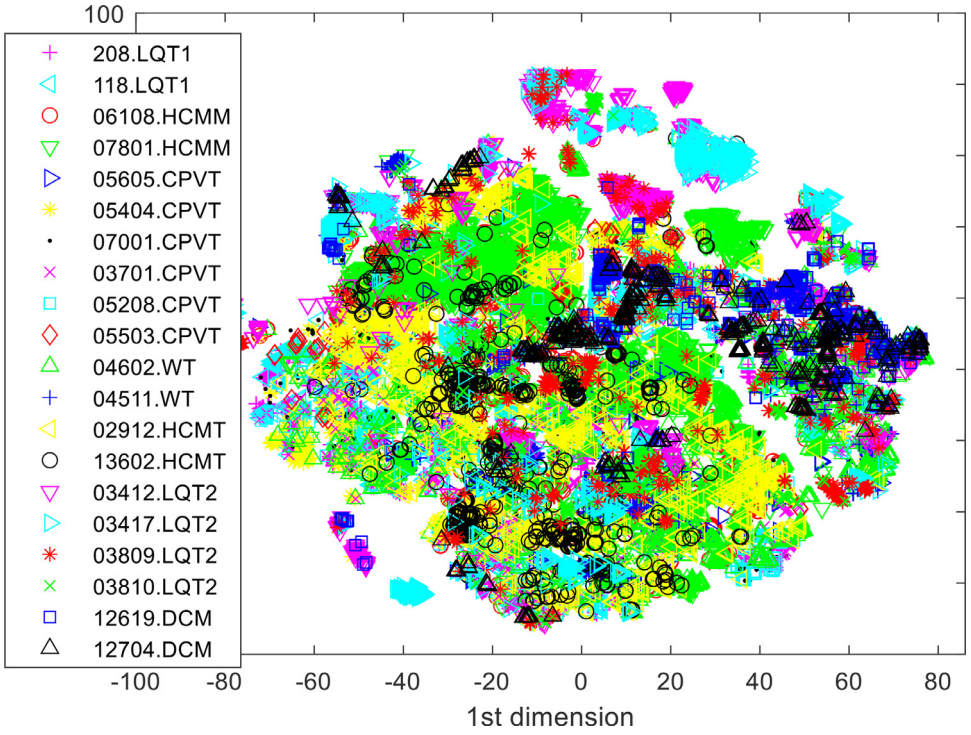

Fig. 5. The whole peak data of 20 cell lines computed with t-SNE algorithm. This is a mapping of the 14-dimensional attribute space to the 2-dimensional space. Thus, even if some cell lines are somewhat overlapping each other in this planar space, their dispersion can be much greater in the attribute space where the classification tests were made.

we also tested three distance weighting schemes called equal, inverse weighting and squared inverse weighting. Overall, with the *k*NN classifier 456 different triplets were tested and all of them were examined with both test set-up approaches.

Decision trees have performed well in our earlier studies and in this paper we applied two tree-based algorithms. Firstly, we used classical CART [21,22] decision tree algorithm with default parameter values. Secondly, we examined Random Forests classifier [23,24] where we tested 100 different values for the number of trees in a forest ($\#trees \in \{1, 2, \ldots, 100\}$). Both Random Forests and CART classifiers were tested with similar parameter settings on both test set-ups.

Discriminant analysis is a famous and traditional method for classification tasks. We investigated the suitability of three discriminant analysis methods for the genetic cardiac disease classification. These methods were linear discriminant analysis [25], quadratic discriminant analysis [25,26] and Mahalanobis discriminant analysis also known as minimum Mahalanobis distance classifier [27,28]. We used the default parameter settings in all discriminant analysis experiments.

Multinomial logistic regression (MLR) [29,30] is an extension of logistic regression applied in binary classification tasks. We used MLR method with default parameter settings with both test set-ups. Next classification method was Naïve Bayes classifier [31] with and without kernel smoothing density estimate (KDE). Without KDE, Naïve Bayes has assumption of Gaussian distribution. When KDE is applied with Naïve Bayes classifier, there are several alternatives for kernel selection. In this study, we tested four kernels namely Epanechnikov, normal (Gaussian), triangle and box kernels. These four kernels have also been examined in our previous stud-

ies [6,10,11,16,17,18]. Otherwise, with Naïve Bayes classifier we applied the default parameter settings.

Support Vector Machine (SVM) is a well-known classification algorithm. The original purpose was to use it in two-class classification problems, but it has been extended to cover also multiclass cases where the number of classes is larger than two. We have used binary and different multiclass extensions of Least-Squares Support Vector Machines (LS-SVMs) [32,33] in our previous studies [5,6,10,11,16,17,18]. However, in this research we have done several modifications compared to previous researches. Firstly, we apply now the SVM algorithm [34] instead of LS-SVMs. Secondly, earlier our LS-SVMs approaches included the use of either binary LS-SVM classifier or tree-based multiclass LS-SVM. However, now we utilize one-vs.-one (OVO) [35] approach where $\frac{M(M-1)}{2}$ individual binary SVM classifier are constructed when M is the number of classes. We are using error-correcting output codes (ECOC) framework [36] to model OVO approach and due to the properties of OVO, ECOC uses ternary coding [37,38] and loss-weighted decoding scheme is applied. When training an individual binary SVM classifier, we utilize Iterative Single Data Algorithm (ISDA) [39] in hyperplane optimization. ISDA algorithm is designed, especially, to be used with large datasets. Moreover, with the SVM classifier Hinge loss function is applied. When dealing with SVMs, parameters and kernel selection are important issues. We tested four kernels (the linear, quadratic, cubic and the RBF) in this paper. A common parameter for all kernels is boxconstraint (C) and we tested 21 values for this hyperparameter ($C \in \{2^{-8}, 2^{-7}, \ldots, 2^{12}\}$). In the case of the RBF kernel we also examined the impact of kernel scale parameter (σ) and we had the same parameter value space for it as for boxconstraint. Thus, all the polynomial kernels were tested with 21 parameter values and the RBF kernel with 21^2 parameter combinations.

The first test set-up was leave-one-signal data-out (LOSDO), which is a variant from leave-one-out method. The classification process with LOSDO technique can be described as follows:

1 Let the number of signals be N in the data.
2 Extract data from the ith signal for test set. Notice that data from one signal includes several rows of data and each one of the rows represents data gained from one peak.
3 Leave the data from *N-1* signals to training set.
4 Perform z-score standardization (columns to zero mean and unit variance) for the training set.
5 Scale the columns of test set with the scaling parameters gained from the training set in step 4.
6 Train classifier with the training set and with the given parameter settings.
7 Predict class labels for the test set data. Now, prediction is made at peak level.
8 Take the mode of predictions in order to get a signal level prediction. If mode is not unambiguously determined, take the smallest value occurred in a tie.
9 Repeat steps 2-8 for all signals in a dataset.
10 When signal level prediction for all signals has been made, evaluate confusion matrix.
11 From confusion matrix (CM) evaluate accuracy and TPR (true positive rate) for each class.
12 If a classification method requires parameter tuning, repeat the process with all parameter settings and select that parameter combination as a final result, which receives the highest accuracy.

The other test set-up was to use 10-fold cross-validation in classification. 10-fold- cross-validation is a standard way to perform classification in machine learning, but in the case of signal classification, there are some special issues that need to be taken into account. We used stratified sampling at signal level when doing the 10-fold cross-validation division. By this means, we ensured that all classes in a data are represented in each fold. Furthermore, the data from one signal is included only to one-fold. In other words, data from one signal is not scattered to several folds simultaneously. The same cross-validation division was used with all parameter settings tested and/or classification methods in order to ensure fair comparison of the results. Classification procedure goes as follows in detail:

1 Extract the ith fold from the data to test set.
2 Leave the rest of the folds to training set.
3 Perform z-score standardization (columns to zero mean and unit variance) for the training set.
4 Scale the columns of the test set with the scaling parameters gained from the training set in step 3.
5 Train classifier with the training set and with the given parameter settings.
6 Predict class labels for the test set (ith fold). Now the predictions are at peak-level.
7 Obtain a signal level prediction for each signal in a test set by taking the mode separately from the predicted class labels for each signal data. If mode is not unambiguously determined, take the smallest value occurred in a tie.
8 When signal level prediction for all signals within the ith fold has been made, evaluate the confusion matrix (CM_i).
9 Repeat steps 1-8 with all folds.
10 Calculate a combined confusion matrix (CCM) by summing up all confusion matrices together. In other words, $CCM = \sum_{i=1}^{10} CM_i$.
11 Evaluate accuracy and TPR for each class from CCM.
12 If a classification method requires parameter tuning, repeat the process with all parameter settings and select that parameter combination as a final result, which receives the highest accuracy.

Since accuracy and TPRs are calculated from CCM, we do not get standard deviation for accuracy and/or TPRs. If we would calculate accuracy and/or TPRs from each fold separately, the standard deviation obtained would not be comparable with accuracy/TPRs obtained from CCM, since CCM represents evaluation measures obtained from the whole data whereas standard deviation would be calculated from folds (each fold is around 10% of the data). LOSDO procedure presents performance measures from the whole data and, thus, performance measures gained from the CCM are more natural choice compared to traditional 10-fold cross-validation process where performance measures are presented as a mean of ten folds. Accuracy is defined in this study as follows $accuracy = \frac{\sum_{k=1}^{7} CM_{kk}}{\sum_{k=1}^{7}\sum_{m=1}^{7} CM_{km}} 100\%$ whereas TPR for class i is defined with the following way $TPR_i = \frac{CM_{ii}}{\sum_{j=1}^{7} CM_{ij}} 100\%,\ i = 1, 2, \ldots, 7.$

5. Comparing model building with cross-validation or leave-one-out techniques

Obviously, the hold-out procedure is the simplest technique for computing machine learning models and to divide an available data set into two parts, a training set and a test set typically with the equal size or sometimes a training set is larger, e.g., 2/3 from an original data set. If a data set is small, a larger training set is used in order to attempt to build a model or classifier by applying as representative training set as possible. Sampling for training and test sets is, of course, made randomly. However, the stratified hold-out is suggested [40], since then the random sampling is executed so that every class of a data set is represented in both training and test sets approximately according to the class distribution of the original whole data set. Cross-validation [40] can be seen to

form a more versatile division, when a data set is divided or randomly sampled into several (K) separate parts called folds of the equal size or as equal as possible if the number of data instances n is not divisible by K. Usually, K is equal to 10, i.e., 10-fold cross-validation is applied, but also 5- or 20-fold cross-validation may be utilized. Thus, the choice of 10 is chiefly an established practice. Then one by one, K folds or subsets are used as a test set when the corresponding training set is formed based on other K-1 subsets. This way, a test set is around 10% of the original data set and its training set 90%, respectively. Leave-one-out technique is equal to the special cross-validation procedure in which every test set includes only one instance and its training set all other n-1 instances [40]. In this sense, a maximal number of data instances is used for building a model. The number of models or classifiers built is equal to n (the size of the data set) which may cause high running times. Leave-one-out is mainly applied only to small data sets.

When a classification or other machine learning problem cannot be fully solved with finite (and not even infinite in principle) data sets, because this also depends on the capacity of an algorithm applied to solve the learning problem, there may appear more or less errors that are called bias that cannot be fully eliminated and not calculated precisely in practice, but can be approximated. The other error source stems from a practical limitation, training set applied that is always finite and limited in reality. Thus, such a training set is not able very well to represent the actual population of data instances. This error over all potential training sets of the same size as well as to test sets is variance of a machine learning technique for a problem given. The expected error of a classifier is the sum of bias and variance [40].

Leave-one-out is seen virtually unbiased and, on the other hand, it has high variance [41]. Classification accuracy is used to express how well a classifier can classify test data instances on average. In a way, it is opposite to prediction error. After assuming independent and identically distributed data instances correct and incorrect classifications were expressed with loss function values 1 and 0 [41]. Then accuracy value A for classification could be computed as follows where T is the test set, x_i its instance, C a classifier and y_i the actual (known) class of x_i. Then $\delta(C(x_i),y_i))$ is equal to 1 if $C(x_i)$ is equal to y_i and otherwise it is 0 [41].

$$A = \frac{1}{|T|}\sum_{x_i \in T} \delta(C(x_i), y_i)$$

K-fold cross-validation was considered by using prediction error and its expected value over training sets [42]. They studied variance in the context of cross-validation starting from identically distributed (dependent) attributes ($\hat{H}$ their average) containing the property asymptotically converging to a normally distributed attribute which is characterized with its expectation $\mathrm{E}(\hat{H})$ and variance as follows.

$$\mathrm{Var}\left(\hat{H}\right) = \mathrm{E}\left(\hat{H}^2\right) - \mathrm{E}\left(\hat{H}\right)^2$$

Further, they used the covariance matrix of cross-validation errors and found that there are much similarity in its contents shown in covariance matrix blocks [42] which was natural, since it was based on cross-validation using the same data set divided into folds. They formed a linear combination of three moments derived from the covariance matrix. By using the moments derived it was possible to show that no unbiased estimator of variance $\mathrm{Var}(\hat{H})$ exists. This was also proved for leave-one-out [42].

Independence assumptions for K-fold cross-validation and leave-one-out were introduced and the assumptions were utilized to derive sampling distributions for their estimators of cross-validation and leave-one-out [43]. It was presented that K-fold cross-validation should not be executed repeatedly, since this could not give a more reliable estimate since mean accuracies produced by any two repetition are dependent. Also, the sample variance of leave-one-out is constant because of the characterization of it. This was interpreted as another explanation not to execute K-fold cross-validation repeatedly [43].

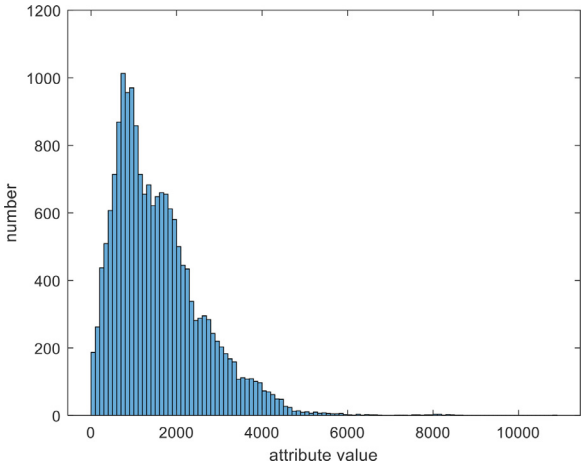

Fig. 6. The distribution of attribute max(s') computed from the whole peak data.

Our current data contained 1173 signals originating from seven classes and comprising different numbers of peaks from 1 to 123. Peak sizes and shapes varied much. A signal classification was made on the basis of its peaks that were first classified one by one with a classifier and the class of that signal was decided according to the majority of its peak classifications. When the actual classifiers tested were applied to peak data instances, the main process focused on them. On the other hand, the results gained are given as classification accuracies stemming from entire signals of its own peaks. Accordingly, we examined their common influence.

Peak attribute distributions are right-skewed, e.g., as in Fig. 6. All 14 attributes were right-skewed. It is natural when all these are biophysical attributes of non-negative real values. None was normally distributed according to Kolmogorov-Smirnov test. They are not ideal such as independently and identically distributed.

Although 10-fold cross-validation and leave-one-out were seen rather different from each other [40-43], at least as different methods, they resemble much each other in the sense that the majority, approximately 90%, of every training set of 10-fold cross-validation contains the same data instances than those in any training set of leave-one-out. Because sampling is made randomly, any test set of size 10% from the whole data set in 10-fold cross-validation ought to mainly include data following the same distribution as in the corresponding training set. For these reasons, at the beginning we assumed that differences of each classification technique between classification results (accuracies) computed with either 10-fold cross-validation or leave-one-out would mostly be minor. Those theoretical considerations above suggest that variance associated with leave-one-out would be extensive. Usually, all literature sources mention that leave-one-out should be used for small data sets only. Instead, it is rather indefinite what size of a data set may or may not be small.

In the present research we had only one data set. Usually more data sets are used, for example, six [41] or more while comparing machine learning methods. Often tens of them are recommended to obtain somewhat wide and general understanding about classification results gained. If there are several data sets, they may be rather small to restrict execution times of classifications to be reasonable. Nevertheless, we used one data set only, but made classifications with numerous classification algorithms and their variations to obtain generality but, at the same time, restricting execution times within quite reasonable durations. This restriction

Table 3
Leave-one-out: Classification results as sensitivities and accuracies for seven disease classes when C and σ are the control parameters of radial basis function support vector SVM RBF. The three highest accuracies are in Bold.

Method	Sensitivity %							Accuracy %
Class	LQT1	LQT2	HCMM	HCMT	CPVT	DCM	WT	
Random forests, 61 trees	94.4	74.6	86.3	59.1	70.8	71.6	62.8	**73.7**
ECOC-SVM linear, C=4	83.3	73.9	71.1	46.3	38.6	73.1	44.2	57.7
ECOC-SVM polynomial 2, C=2^{-3}	86.7	75.4	84.4	61.1	57.9	74.6	45.1	67.2
ECOC-SVM polynomial 3, C=2^{-5}	93.3	76.8	79.6	59.7	64.8	70.1	44.7	67.6
ECOC-SVM RBF, C=1024, σ=4	87.8	76.8	87.4	67.1	65.2	71.6	55.8	**72.2**
Multinomial logistic regression	83.3	71.7	66.3	43.0	39.5	62.9	46.5	55.9
Linear discriminant analysis	63.3	54.3	61.9	36.2	39.9	62.7	48.2	50.9
Quadratic discriminant analysis	90.0	42.0	78.5	45.6	21.9	79.1	35.0	51.3
Mahalanobis discriminant analysis	7.8	52.9	23.3	17.4	40.8	74.6	80.5	42.3
CART (decision tree)	87.8	73.2	86.3	57.7	67.0	74.6	59.3	**71.5**
Naïve Bayes	57.8	8.7	60.0	9.4	9.4	88.1	47.8	36.6
Naïve Bayes normal kernel	56.7	31.2	59.6	28.9	11.2	79.1	71.2	45.9
Naïve Bayes Epanechnikov kernel	45.6	25.4	57.0	24.8	9.4	77.6	71.7	42.9
Naïve Bayes box kernel KNN Chebysev equal weighting	51.1	29.7	59.6	22.1	9.9	77.6	70.8	44.0
Naïve Bayes triangle kernel	56.7	29.0	60.4	28.9	11.2	77.6	70.4	45.5

Table 4
Leave-one-out: Classification results as sensitivities and accuracies in which k is the number of nearest neighbor search that produced the best results. The highest accuracy is in Bold.

Method	Sensitivity %							Accuracy %
Class	LQT1	LQT2	HCMM	HCMT	CPVT	DCM	WT	
KNN Chebysev equal weighting, k=1	73.3	68.1	84.1	46.3	59.7	50.7	49.1	63.1
KNN Chebysev inverse weighting, k=1	73.3	68.1	84.1	46.3	59.7	50.7	49.1	63.1
KNN Chebysev squared inverse weighting, k=1	73.3	67.4	83.0	51.0	55.8	53.7	54.0	63.7
KNN cityblock equal weighting, k=1	86.7	70.3	86.3	59.1	62.7	58.2	53.5	68.4
KNN cityblock inverse weighting, k=1	86.7	70.3	86.3	59.1	62.7	58.2	53.5	68.4
KNN cityblock squared inverse weighting, k=1	86.7	70.3	86.3	59.1	62.7	58.2	53.5	68.4
KNN correlation equal weighting, k=1	83.3	73.2	81.1	56.4	63.1	55.2	51.3	66.4
KNN correlation inverse weighting, k=1	83.3	73.2	81.1	56.4	63.1	55.2	51.3	66.4
KNN correlation squared inverse weighting, k=3	85.5	71.7	79.3	53.0	64.4	56.7	54.9	66.4
KNN cosine equal weighting, k=1	81.1	68.1	82.6	53.0	61.4	65.7	55.8	66.7
KNN cosine inverse weighting, k=7	84.4	68.8	79.3	54.4	60.5	65.7	58.4	66.8
KNN cosine squared inverse weighting, k=7	84.4	68.8	79.3	54.4	63.5	65.7	59.7	67.6
KNN Euclidean equal weighting, k=1	81.1	68.1	87.8	51.0	62.7	53.7	53.1	66.7
KNN Euclidean inverse weighting, k=3	82.2	71.0	87.0	57.0	62.7	53.7	56.2	68.3
KNN Euclidean squared inverse weighting, k=3	81.1	71.7	87.8	56.4	61.4	52.2	56.6	68.1
KNN Mahalanobis equal weighting, k=1	90.0	71.0	86.7	58.4	63.1	53.7	52.2	68.3
KNN Mahalanobis inverse weighting, k=3	90.0	71.0	86.7	56.4	62.7	55.2	54.9	**68.5**
KNN Mahalanobis squared inverse weighting, k=3	90.0	71.0	86.7	58.4	63.0	53.7	52.2	68.3
KNN standardized Euclidean equal weighting, k=1	81.1	68.1	87.8	51.0	62.7	53.7	53.1	66.7
KNN standardized Euclidean inverse weighting, k=3	82.2	71.0	87.0	57.0	62.7	53.7	56.2	68.3
KNN standardized Euclidean squared inverse weighting, k=3	81.1	71.7	87.8	56.4	61.4	52.2	56.6	68.1
KNN Spearman equal weighting, k=1	88.9	64.5	78.1	55.7	57.9	52.2	50.9	63.8
KNN Spearman inverse weighting, k=3	85.6	69.6	78.1	56.4	57.5	62.7	54.9	65.5
KNN Spearman squared inverse weighting, k=3	87.8	67.4	79.6	57-0	60.1	62.7	54.9	66.3

was made because execution times for radial basis function support vector machines (RBF-SVM) took around eight weeks for 441 control parameter value combinations used in leave-one-out. In addition, other SVM methods, multinomial logistic regression and random forests were relatively slow and took hours or even days. Other, simpler methods took short times such as minutes, but being simple methods meant that those more complicated gave better results being able to model data better and were so necessary to be included in the tests.

6. Results

Many classification methods based on machine learning were run for the current data to analyze how efficiently the six diseases and controls were possible to differentiate from each other by applying peaks recognized from calcium transient data. Sensitivities or true positive ratios and accuracies gained with the leave-one-out technique are shown in Tables 3 and 4. In Table 3, random forests generated the highest classification accuracy of 73.7 %. SVM with radial basis function (RBF) of 72.2 % and decision trees (CART) of 71.5 % were almost equally efficient. In Table 4, kNN with Mahalanobis equal weighting produced the highest accuracy of 68.5 %, and several other were virtually equally good.

Sensitivities and accuracies produced by applying 10-fold cross-validation are presented in Tables 5 and 6 after having used the same classification methods similarly to the tests made with leave-one-out. Cross-validation test series was executed only once so that every instance was run once for testing, i.e., as for leave-one-out test. In Table 5 the best results are 72.5 % of random forests, 70.3 % of ECOC-SVM RBF and 69.4 % of CART. In Table 6 the best is 67.9 % of kNN Mahalanobis squared inverse weighting.

As compared to our earlier results the current accuracies of seven classes, that is, 6 diseases or mutations and controls (wild type) the classification accuracies obtained are smaller than those with 527 calcium transit signals for four classes (three diseases and controls) of 78.6 % [6] and those with 941 signals for five classes

Table 5
10-fold cross-validation: Classification results as sensitivities and accuracies for seven. The three highest accuracies are in Bold.

Method	Sensitivity %							Accuracy %
Class	LQT1	LQT2	HCMM	HCMT	CPVT	DCM	WT	
Random forests, 40 trees	91.1	71.0	87.0	59.7	70.0	67.2	61.1	**72.5**
ECOC-SVM linear, C=8	83.3	70.3	71.5	47.0	40.3	74.6	44.2	57.9
ECOC-SVM polynomial 2, C=2^{-3}	90.0	72.5	82.2	58.4	57.5	77.6	47.3	66.8
ECOC-SVM polynomial 3, C=2^{-8}	92.2	74.6	84.8	53.7	58.4	66.7	46.9	66.8
ECOC-SVM RBF, C=2^{11}, σ=4	91.1	75.4	85.2	63.8	63.9	67.2	53.1	**70.3**
Multinomial logistic regression	90.0	69.6	67.8	40.9	37.3	65.7	45.6	55.8
Linear discriminant analysis	63.3	53.6	62.6	35.6	40.8	62.7	47.8	51.0
Quadratic discriminant analysis	90.0	44.2	77.8	47.6	22.7	77.6	33.2	51.4
Mahalanobis discriminant analysis	7.8	52.9	23.3	18.1	42.1	71.6	79.2	42.2
CART (decision tree)	83.3	68.1	82.3	55.7	66.1	67.2	57.5	**69.4**
Naïve Bayes	58.9	9.4	61.1	8.7	9.9	88.1	49.1	37.3
Naïve Bayes normal kernel	51.1	27.5	61.1	28.9	12.0	79.1	70.4	45.4
Naïve Bayes Epanechnikov kernel	45.6	22.5	56.3	23.5	9.0	76.1	70.0	41.7
Naïve Bayes box kernel KNN Chebysev equal weighting	47.8	27.5	58.1	20.1	7.7	77.6	70.8	42.5
Naïve Bayes triangle kernel	52.2	26.8	61.9	28.9	11.6	76.1	70.0	45.2

Table 6
10-fold cross-validation: Classification results as sensitivities and accuracies in which k is the number of nearest neighbor search that produced the best results. The highest accuracy is in Bold.

Method	Sensitivity %							Accuracy %
Class	LQT1	LQT2	HCMM	HCMT	CPVT	DCM	WT	
KNN Chebysev equal weighting, k=1	73.3	68.8	83.3	47.0	58.8	44.8	49.6	62.7
KNN Chebysev inverse weighting, k=1	73.3	68.8	83.3	47.0	58.8	44.8	49.6	62.7
KNN Chebysev squared inverse weighting, K=3	71.1	66.7	83.7	49.7	56.2	49.3	52.7	63.0
KNN cityblock equal weighting, k=3	90.0	67.3	85.6	52.3	61.4	55.2	54.3	67.0
KNN cityblock inverse weighting, k=5	87.8	66.7	86.3	54.4	61.8	58.2	54.0	67.3
KNN cityblock squared inverse weighting, K=5	85.6	66.7	87.0	54.4	62.7	59.7	53.1	67.4
KNN correlation equal weighting, k=1	82.2	71.7	80.0	56.4	65.2	52.2	51.3	66.2
KNN correlation inverse weighting, k=1	82.2	71.7	80.0	56.4	65.2	52.2	51.3	66.2
KNN correlation squared inverse weighting, k=1	82.2	71.7	80.0	56.4	65.2	52.2	51.3	66.2
KNN cosine equal weighting, k=1	80.0	65.9	82,2	54.4	64.4	58.2	55.8	66.6
KNN cosine inverse weighting, k=1	80.0	65.9	82.2	54.4	64.4	58.2	55.8	66.6
KNN cosine squared inverse weighting, k=1	80.0	65.9	82.2	54.4	64.4	58.2	55.8	66.6
KNN Euclidean equal weighting, k=1	77.8	66.7	87.0	51.0	62.7	44.8	49.6	64.9
KNN Euclidean inverse weighting, k=3	81.1	67.4	85.9	53.7	61.8	47.8	51.8	65.7
KNN Euclidean squared inverse weighting, k=3	80.0	68.1	86.3	53.0	62.7	47.8	53.0	66.0
KNN Mahalanobis equal weighting, k=3	88.9	70.3	87.0	57.7	60.1	49.3	48.7	66.0
KNN Mahalanobis inverse weighting, k=3	91.1	70.3	85.9	56.4	62.7	49.3	53.5	67.8
KNN Mahalanobis squared inverse weighting, k=7	93.3	72.5	86.3	57.0	62.2	55.2	50.0	**67.9**
KNN standardized Euclidean equal weighting, k=1	77.8	66.7	87.0	51.0	62.7	44.8	49.6	64.9
KNN standardized Euclidean inverse weighting, k=3	81.1	67.4	85.9	53.7	61.8	47.8	51.8	65.7
KNN standardized Euclidean squared inverse weighting, k=3	80.0	68.1	86.3	53.0	62.7	47.8	52.2	66.0
KNN Spearman equal weighting, k=1	87.7	67.4	75.9	54.4	57.9	44.8	52.2	63.2
KNN Spearman inverse weighting, k=3	85.6	63.0	78.9	53.0	60.5	50.7	50.4	63.5
KNN Spearman squared inverse weighting, k=3	87.8	63.0	78.1	53.0	60.5	53.7	52.2	64.0

(4 diseases or mutations and controls) of 77.8 % [10] as their best results both with random forests. Now the seven classes classification was naturally more complicated, which affected the results.

On the basis of the results in Tables 3-6, LQT1 and HCMM are among the best separated and the controls (WT) often had the poorest sensitivities (true positive ratios). LQT1 and LQT2 could be separated very accurately from each other. HCMM and HCMT could also be separated even if these are two mutations of the same disease.

There are the accuracies of 40 different methods together in Tables 3 and 4 as well as in Tables 5 and 6. For 36 methods in Tables 3 and 5, the accuracy results of leave-one-out are slightly higher than those corresponding of cross-validation in Tables 5 and 6. For these 36 the differences are from 0.1 % to 2.6 %. For the rest four methods cross-validation gave very slightly better accuracies, when the differences of leave-one-out and cross-validation are small being from 0.1 % to 0.7 %. Altogether, the minimum difference is 0.1 %, maximum 2.6 % and mean 1.0 %, in other words, leave-one-out generated better accuracies, but the mean difference is rather insignificant.

Specificity values resembled roughly sensitivity values. For example, in association with the best accuracy result of 73.7 % in Table 3 for random forests specificities were 79.8 % for LQT1, 73.5 % for LQT2, 68.0 % for HCMM, 75.8 % for HCMT, 74.4 % for CPVT, 73.8 % for DCM and 76.2 % for WT.

For the sake of comparison, we still computed with clustering how the peak data were distributed while applying unsupervised machine learning. K-means++ clustering method has been applied to the whole dataset with different parameter values. As a preprocessing stage, we z-score standardized the features in a dataset. Clustering has been repeated with 7, 8, 9 and 12 clusters. Squared Euclidean distance measure has been used in clustering. The number of iterations was 1000 and 100 different initializations were tested with all clusterings.

Clustering was performed on a peak level and this means that each peak in a signal is assigned to a certain cluster. In order to have a signal level clustering for specific signal, we take a mode (majority vote) from the peak level cluster assignments of a specific signal and this determines to which cluster a signal belongs. In Table 7 there are the best results given by 9 clusters,

Table 7
K-means++ results computed with 9 clusters. The most frequent class/classes in a cluster have been emphasized with bold font and the proportion of those classes in a cluster in percentages have also been represented.

Class/Cluster	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
LQT1	24	15	16	28	0	0	0	1	6
HCMM	50	6	**106 (52.0)**	39	**48 (38.7)**	4	0	0	17
CPVT	47	11	24	**66 (28.3)**	25	**22 (37.9)**	0	2	36
WT	35	33	15	16	33	15	5	**9 (52.9)**	**65 (43.3)**
HCMT	**52 (22.7)**	0	21	35	16	14	0	0	11
LQT2	19	**36 (27.1)**	21	36	2	3	5	1	15
DCM	2	32	1	13	0	0	**15 (60.0)**	4	0

where the rows represent classes and columns clusters. Numbers in Table 7 represent the number of signals in a specific cluster from a specific class.

These results show that also unsupervised machine learning produced clustering to different groups of data peaks where the majority classes were found for clusters except for LQT1. Note that more than one cluster may represent the same class. In Table 7 there are two such clusters for HCMM, CPVT and WT. Class LQT1 did not reach the signal majority in any cluster in Table 7. Possibly, this came partially from its relatively small number of peaks from among all classes (Table 1). Poorer results were obtained compared with the results of the classification methods which is natural and typical, since unsupervised machine learning methods of clustering do not utilize the class information which supervised classification methods do. These clustering results support our preceding classification results that the classification of the different genetic diseases included can be possible.

7. Conclusion

The classification accuracies are smaller than in our earlier studies so that, for example, the best accuracy in Table 3 was approximately 5 % smaller than the best one of three disease classes and controls [6] or 4 % smaller than the best of four disease classes and controls [10]. This is rational because the current task of six disease classes and controls is more complicated. Nevertheless, the best accuracy of above 70 % gained individual sensitivities above 59 % of random forests in Table 3 show that the six diseases classes and controls are possible to separate from each other by machine learning techniques.

The comparison of leave-one-out and cross-validation indicated that leave-one-out produced 1 % better accuracies on an average for the current data and test set-ups. This is virtually insignificant. Obviously, it was caused by the essential difference of the two techniques, the larger training set of leave-one-out and from its almost constant character. While comparing any two different training sets and the corresponding test sets in the leave-one-out training process, two instances only, their test instances (one present and the other absent or the opposite as to two training sets) are exceptional and otherwise the training sets are identical. Instead, 10 training sets in cross-validation are slightly more different, which may produce a little more variation for machine learning models computed with the same method. For 10-fold cross-validation 80 % of instances are the same for the training sets of two different folds. To summarize, the results show that in principle both techniques could be applied as well for the current data with these numbers: 1173 signals containing 18387 peaks used for training from seven classes. Of course, applying leave-one-out for large datasets would require long running times, at least for such time-consuming methods as support vector machines with radial basis function (RBF) and random forests being also typically among the best, and therefore their use would not then be practical.

This relatively large dataset utilized in this study included two to six hiPSC cell lines from each seven conditions (control or disease) and by using complex classification tasks we gained the classification accuracy of above 70 %. We see this as an indicator that these diseases could be separated from each other by machine learning techniques. Nevertheless, even larger datasets could strengthen our finding. Abnormalities in calcium transients in many cases represent a patient's cardiac phenotype [7,8,9,12,13]. However, a known problem with hiPSC-CMs is their immature structural and functional characteristics like immature calcium handling that differs from those of adult cardiomyocytes. In the literature variation between hiPSC lines and in their phenotypic responses, e.g. in baseline action potential characteristics and drug responses, have been reported even between control cell lines [44-46]. This variability could exceed differences between parameters of patient and control hiPSC-CMs. Since these cells can be phenotypically immature and culture and assay methods are not standardized, it can be a disadvantage to the development of predictive computational models [45]. One option to reduce variability between cell lines, in addition to increase cell maturation with certain techniques, is recent advances in genome-editing techniques, like CRISPR/Cas9 method [47]. This allows modification of the stem cell genome when isogenically matched controls are generated for diseased hiPSC lines [45]. In the future, experimental datasets for machine learning should be broadened with isogenic controls and larger datasets of hiPSC lines to better understand the variation between healthy and diseased lines.

The classification tests performed here strengthens our previous results showing that it is achievable to get a good classification accuracy with disease-specific iPSC-CM calcium transient data, even though the number of test signals here was increased with signals of two additional diseases. The result strengthens our previous finding that the machine learning method could be utilized in identification of several genetic cardiac diseases, but may also separate mutations in different genes resulting in the same clinical phenotype. Machine learning classification of disease-specific CM calcium transients could be exploited to diagnose genetic cardiac disease and could even predict the type of disease mutation. For this more advanced cardiac differentiation methods would be needed in the future, e.g. direct differentiation of blood cells into CMs, to achieve a realistic additional tool for diagnosing genetic cardiac diseases.

Declaration of Competing Interests

As to the manuscript of 'On computational classification of genetic cardiac diseases applying iPSC cardiomyocytes' submitted to Computer Methods and Programs in Biomedicine, there is no conflict of interest.

Acknowledgements

We would like to thank Academy of Finland Centre of Excellence in Body-on-Chip Research, Finnish Cardiovascular Research Foundation, Sigrid Juselius Foundation and Pirkanmaa Hospital district funding.

References

[1] K. Takahashi, K. Tanabe, M. Ohnuki, M. Narita, T. Ichisaka, K. Tomoda, S. Yamanaka, Induction of pluripotent stem cells from adult human fibroblasts by defined factors, Cell 131 (2007) 861–872.

[2] A. Moretti, M. Bellin, A. Welling, C.B. Jung, J.T. Lam, L. Bott-Flügel, et al., Patient-specific induced pluripotent stem-cell models for long-QT syndrome, New Engl. J. Med. 363 (2010) 1397–1409.

[3] C. Heylman, R. Datta, A. Sobrino, S. George, E. Gratton, Supervised machine learning for classification of the electrophysiological effects of chronotropic drugs on human induced pluripotent stem cell-derived cardiomyocytes, Plos One 10 (2015) e0144572.

[4] E.K. Lee, D.D. Tran, W. Keung, P. Chan, G. Wong, C.W. Chan, et al., Machine learning of human pluripotent stem cell-derived engineered cardiac tissue contractility for automated drug classification, Stem Cell Reports 9 (2017) 1560–1572.

[5] M. Juhola, K. Penttinen, H. Joutsijoki, K. Varpa, J. Saarikoski, J. Rasku, H. Siirtola, K. Iltanen, J. Laurikkala, H. Hyyrö, J. Hyttinen, K. Aalto-Setälä, Signal analysis and classification methods for calcium transient data of stem cell derived cardiomyocytes, Comput. Biol. Med. 61 (2015) 1–7, doi:10.1016/j.compbiomed.2015.03.016.

[6] M. Juhola, H. Joutsijoki, K. Penttinen, K. Aalto-Setälä, Detection of genetic cardiac diseases by Ca2+ transient profiles using machine learning methods, Sci. Rep. 8 (2018) 9355 www.nature.com/articles/s41598-018-27695-5.

[7] A.L. Kiviaho, A. Ahola, K. Larsson, K. Penttinen, H. Swan, M. Pekkanen-Mattila, H. Venäläinen, K. Paavola, J. Hyttinen, K. Aalto-Setälä, Distinct electrophysiological and mechanical beating phenotypes of long QT syndrome type 1-specific cardiomyocytes carrying different mutations, Int. J. Cardiol. Heart Vasc. 25 (8) (2015) 19–31.

[8] M. Ojala, C. Prajapati, R.P. Pölönen, K. Rajala, M. Pekkanen-Mattila, J. Rasku, K. Larsson, K. Aalto-Setälä, Mutation-specific phenotypes in hiPSC-derived cardiomyocytes carrying either myosin-binding protein C or α-tropomyosin mutation for hypertrophic cardiomyopathy, Stem Cells Int (2016).

[9] K. Penttinen, H. Swan, S. Vanninen, J. Paavola, A.M. Lahtinen, Kimmo Kontula, K. Aalto-Setälä, Antiarrhythmic effects of dantrolene in patients with catecholaminergic polymorphic ventricular tachycardia and replication of the responses using iPSC models, Plos One 10 (5) (2015) e0125366.

[10] M. Juhola, H. Joutsijoki, K. Penttinen, K. Aalto-Setälä, Differentiation of genetic cardiac diseases on the basis of artificial intelligence, Eur. J. Biomed. Inform. 15 (3) (2019) 43–52.

[11] M. Juhola, K. Penttinen, H. Joutsijoki, K. Aalto-Setälä, Analysis of Drug Effects on iPSC Cardiomyocytes with Machine Learning, Annals Biomed. Eng. 49 (2021) (2021) 129–138, doi:10.1007/s10439-020-02521-0.

[12] D. Shah, L. Virtanen, C. Prajapati, M. Kiamehr, J. Gullmets, G. West, J. Kreutzer, M. Pekkanen-Mattila, T. Heliö, P. Kallio, P. Taimen, K. Aalto-Setälä, Modeling of LMNA-related dilated cardiomyopathy using human induced pluripotent stem cells, Cells 8 (6) (2019) 594.

[13] D. Shah, C. Prajapati, K. Penttinen, R.M. Cherian, J.T. Koivumäki, A. Alexanova, J. Hyttinen, K. Aalto-Setälä, hiPSC-derived cardiomyocyte model of LQT2 syndrome derived from asymptomatic and symptomatic mutation carriers reproduces clinical differences in aggregates but not in single cells, Cells 7;9 (5) (2020) 1153.

[14] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[15] R.C. Staudemayer, E.R. Morris, Understanding LSTM – a tutorial into long short-term memory recurrent neural networks, arXiv (2019) 1–42 1909.09586.

[16] H. Joutsijoki, K. Penttinen, M. Juhola, K. Aalto-Setälä, Separation of HCM and LQT cardiac diseases with machine learning of Ca2+ transient profiles, Methods Inf. Med. 58 (04/05) (2019) 167–178, doi:10.1055/s-0040-1701484.

[17] M. Juhola, K. Penttinen, H. Joutsijoki, K. Varpa, K. Aalto-Setälä, On the classification of stem cell-derived cardiomyocytes' calcium transient signals, Int. J. Extreme Autom. Connect. Healthcare 2 (2) (2019) 22–37 https://www.igi-global.com/article/on-the-separation-of-normal-and-abnormal-stem-cell-derived-cardiomyocytes-calcium-transient-signals/232330.

[18] M. Juhola, H. Joutsijoki, K. Penttinen, K. Aalto-Setälä, Machine learning to differentiate diseased cardiomyocytes from healthy control cells, Inform. Med. Unlocked 14 (2019) 15–22, doi:10.1016/j.imu.2019.01.006.

[19] L. Jiang, Z. Cai, D. Wang, S. Jiang, Survey of improving K-Nearest-Neighbor for classification, in: Proc. Fourth Int. Conf. Fuzzy Systems and Knowledge Discovery, 2007, pp. 1–5.

[20] S.A. Dudani, The distance weighted k-nearest neighbor rule, IEEE Trans. Systems, Man, Cybern. 6 (4) (1976) 325–327.

[21] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd Ed., John Wiley & Sons, New York, USA, 2001.

[22] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowl. Inf. Systems 14 (2008) 1–37.

[23] G. Biau, E. Scornet, A random forest guided tour, TEST 25 (2) (2016) 197–227.

[24] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[25] A.J. Izenman, Modern Multivariate Statistical Techniques – Regression, Classification, and Manifold Learning, Springer, 2008.

[26] A. Tharwat, Linear vs. quadratic discriminant analysis classifier: a tutorial, Int. J. Applied Pattern Recogn. 3 (2) (2016) 145–180.

[27] G. Bohling, Classical normal-based discriminant analysis, Technical Report EECS 833, Kansas Geol. Survey (2006) 1–24.

[28] K.J. Cios, W. Pedrycz, R.W. Swiniarski, L.A. Kurgan, Data Mining: A Knowledge Discovery Approach, Springer-Verlag, New York, 2007.

[29] C. Kwak, A. Clayton-Matthews, Multinomial logistic regression, Nursing Res 51 (6) (2002) 404–410, doi:10.1097/00006199-200211000-00009.

[30] C.J. Petrucci, A primer for social worker researchers on how to conduct a multinomial logistic regression, J. Social Service Res. 35 (2) (2009) 193–205.

[31] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed., Springer, New York, 2008.

[32] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters 9 (3) (1999) 293–300.

[33] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, 2002.

[34] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[35] H. Joutsijoki, M. Haponen, J. Rasku, K. Aalto-Setälä, M. Juhola, Machine learning approach to automated quality identification of human induced pluripotent stem cell colony images, Comp. Math. Methods Med. 3091039 (2016) 1–15.

[36] H. Joutsijoki, M. Haponen, J. Rasku, K. Aalto-Setälä, M. Juhola, Error-correcting output codes in classification of human induced pluripotent stem cell colony images, BioMed Res. Int. 3025057 (2016) 1–13.

[37] S. Escalera, O. Pujol, P. Radeva, Separability of ternary codes for sparse designs of error-correcting output codes, Pattern Recogn. Letters 30 (3) (2009) 285–297.

[38] S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, IEEE Trans. Pattern Analysis Machine Intell. 32 (1) (2010) 120–134.

[39] V. Kecman, T.-M. Huang, M. Vogt, Iterative single data algorithm for training kernel machines from huge data sets: Theory and performance, in: Support Vector Machines: Theory and Applications, Springer, 2005, pp. 255-274.

[40] I.H Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, third,ed., Morgan Kaufmann, Massachusetts, USA, 2011.

[41] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, https://www.researchgate.net/publication/2352264_A_Study_of_Cross-Validation_and_Bootstrap_for_Accuracy_Estimation_and_Model_Selection

[42] Y. Bengio, Y. Grandvalet, No unbiased estimator of the variance of k-fold cross–validation, J. Mach. Learn. Res. 5 (2004) 1089–1105.

[43] T.-T. Wong, Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation, Pattern Recogn 48 (2015) 2839–2846.

[44] P. Machiraju, S.C. Greenway, World J. Stem Cells 11 (1) (2019) 33–43 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6354100/.

[45] L. Sala, M. Bellin, C.L. Mummery, Integrating cardiomyocytes from human pluripotent stem cells in safety pharmacology: has the time come? British J. Pharmacology 174(21) 3749-3765. https://bpspubs.onlinelibrary.wiley.com/doi/full/10.1111/bph.13577.

[46] I. Mannhardt, U. Saleem, D. Mosqueira, M.F. Loos, B.M. Ulmer, M.D. Lemoine, C. Larsson, C. Améen, T. de Korte, M.L.H. Vlaming, K. Harris, P. Clements, C. Denning, A. Hansen, T. Eschenhagen, Comparison of 10 control hPSC lines for drug screening in an engineering heart tissue format, Stem Cell Reports 15 (2020) 983–998 https://www.sciencedirect.com/science/article/pii/S2213671120303775?dgcid=rss_sd_all.

[47] J.D. Sander, J.K. Joung, CRISP-cas systems for genome editing, regulation and targeting, Nat. Biotechnol. 32 (4) (2014) 347–355 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4022601/.