Unsupervised Learning to Analysis of Population of Models in Computational Electrophysiological Studies

Konstantin Ushenin
Institute of Natural Sciences
Ural Federal University
Ekaterinburg, Russia
konstantin.ushenin@urfu.ru

Nikita Detkov
Institute of Natural Sciences
Ural Federal University
Ekaterinburg, Russia
ndsgfk@gmail.com

Tatyana Nesterova
Laboratory of Mathematical Physiology
Institute of Immunology and Physiology
Ekaterinburg, Russia
tatiannesterova@gmail.com

Dmitry Smarko
Institute of Physics and Technology
Ural Federal University
Ekaterinburg, Russia
d.shmarko@yandex.ru

Artem Razumov
Institute of Natural Sciences
Ural Federal University
Ekaterinburg, Russia
airplaneless@yandex.ru

Olga Solovyova
Institute of Natural Sciences
Ural Federal University
Ekaterinburg, Russia
o-solovey@mail.ru

***Abstract*—Unsupervised learning is a helpful tool for processing experimental data. These types of machine learning algorithms can address problems of dimensionality reduction, clusterization, anomaly detection, visualization, and many others. We propose that unsupervised learning may also be useful for the analysis of the results of computational experiments. This proof-of-concept paper is about the application of several unsupervised learning algorithms for computational studies of the electrical activity in human atrial cardiomyocytes. A dataset for analysis was generated with a population modeling approach, which is widely used in cardiac modeling. At first, the signals generated by a model population were processed with the manifold learning and clusterization methods. Then, the dataset was processed with the deep autoencoder neural networks, and the latent space of the autoencoder was used as a tool for comparison of the two ionic models of the human atrial cardiomyocyte. Applied methods allowed us to classify the model solutions according to their physiological meaning, to visualize the model behavior, and to find special types of model solutions which would be almost impossible to notice via manual analysis.**

***Index Terms*—unsupervised learning, visualization, autoencoder, computational cardiac electrophysiology, population of models**

I. Introduction

Machine learning shows high efficiency in a wide range of applications: image classification, image segmentation, image reconstruction, signal classification, text classification, text translation, and others. Several recent works report on the successful application of unsupervised learning for the extraction of new knowledge from experimental data [1].

We propose that unsupervised learning may also be useful for obtaining new knowledge from computer simulations of biophysical processes. Our study combines recent approaches in the study of single cardiomyocyte behavior with using *populations of models* [2] [3] and methods of unsupervised learning as tools for the analysis of the results of numerical experiments.

Modern mathematical models of cardiomyocyte electrophysiology are presented as a system of ordinary differential equations (ODE) that describe biophysical processes underlying the electrical excitation of cardiac cells and realistically simulate the time course of action potential generation (i.e., cellular transmembrane potential, V_m), activation-inactivation of the transmembrane ionic currents [where $\mathbf{I} = (I_1, I_2, .., I_n)$], and the dynamics of intracellular ionic concentrations (e.g., intracellular Ca^{2+}) [4], as follows:

$$C_m \frac{dV}{dt} = \sum_k I_k + I_{stim}, \tag{1}$$

$$\frac{d\mathbf{w}}{dt} = f(\mathbf{w}, V_m, \mathbf{I}) \tag{2}$$

Such ODE systems usually include from 20 to 40 differential equations and hundreds of parameters. The main part of these mathematical constructs have a biophysical meaning and are fitted accordingly to data from biophysical experiments with living cells or tissues.

The ionic models can reproduce the behavior of real cardiomyocytes in a wide range of experimental conditions. Particularly, the models are able to reproduce the effects of drugs and the consequences of pathological conditions [2]. In some cases, a change in the model parameters causes pro-arrhythmic effects in the cardiomyocytes, i.e., abnormalities in the cellular depolarization and repolarisation, such as abnormal elongation of the action potential duration, early afterdepolarizations, and delayed afterdepolarizations.

This work was supported by RFBR 18-015-00368, IIP UrB RAS theme AAAA-A18-118020590031-8, RF Government Act 211 of March 16, 2013, the Program of the Presidium RAS.

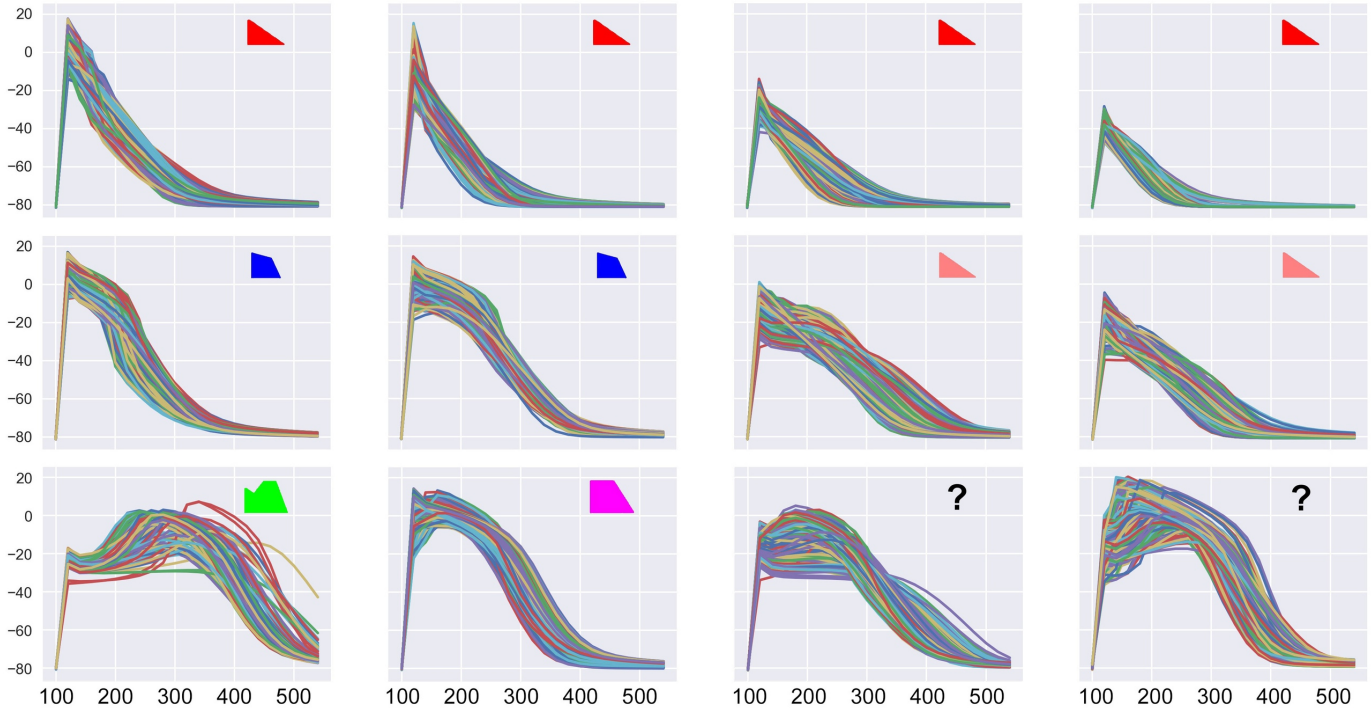

Fig. 1. Results of the first approach. Clusters have been rearranged for clarity. The types of the action potential are annotated visually in the top-right corner of each image. The first row shows four groupings of the triangular action potentials. The second row shows three clusters of the action potentials that were close to the Countermanch98 model results under properties from the original paper and one group of the action potentials with poor clusterization. The last row shows one cluster of the triangular action potentials, one cluster of the action potentials with plateaus and two clusters of the action potentials with early afterdepolarization.

Using populations of models is an advanced approach to the analysis of cardiomyocyte behavior [2] [3]. Similar to biophysical experiments where experimental data are registered in a number of subjects with natural variability in their behavior, the study of populations of models also accounts for the variability in the individual behavior of cardiac cells under the same conditions. Such model populations are formed by the variation in some model parameters reflecting natural uncertainty in the cellular characteristics. Usually, the input space consists of 3 to 12 parameters that are varied within a certain range via Latin hypercube sampling. For each sample of the model parameter vector, a model from the population is computed with a stimulation frequency used to pace real cardiomyocytes in experiments and *steady-state* excitation-relaxation cycling is achieved where the action potential signal is the same between pacing cycles. The steady-state action potential signals that are generated by all the models from the population form the *population output space*.

The study of the output space using statistical approaches allows researchers to predict cardiomyocyte responses to different interventions or conditions. For example, this approach demonstrated high efficiency in the prediction of arrhythmogenic drug effects [2] and in fitting the model behavior to experimental data [3].

In this study, we aimed to test the unsupervised learning methods for the classification of the outputs from the population of ionic models. These methods could help to reveal the physiologically important differences between the models. At first, we used dimensionality reduction and clusterization methods from classical non-deep machine learning for the decomposition of the entire output space generated by the model population (as described in Section II-B). That design was inspired by [1], where such approaches were applied to classify transcriptomics data. Then, we used deep autoencoders to transform the model data in the latent space and create the clusterization (as described in Section II-C). This approach was inspired by the content-based image retrieval problem [5]. In the third part of this paper, we compared the latent spaces for the two model populations generated by the different ionic cellular models and showed where the model results did not overlap (as described in Section III).

II. Methods and Results

A. Generation of Dataset

We chose two widely used ionic models of the human atrial cardiomyocyte: *Courtemanche98* [6], and *Maleckar09* [7]. The input space consisted of five model parameters (g_{Na}, g_{CaL}, g_{to}, τ_{to}, $V_{to,0.5}$) varied over a wide range (from 0% to 200%) with regard to the reference values (100%) set in the original articles. The choice of the varied parameters stems from our study of the effects of age-related ionic remodelling in cardiomyocytes [8].

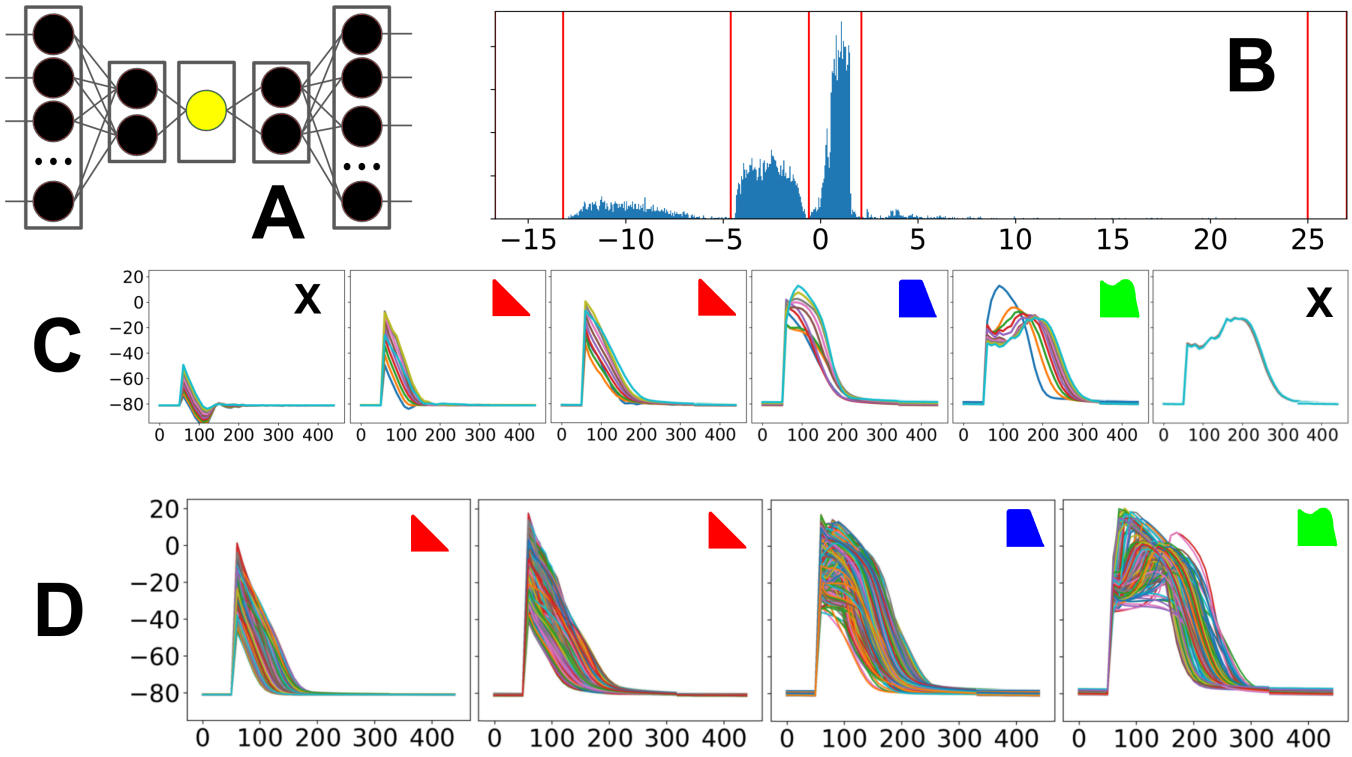

Fig. 2. (A) The deep autoencoder artificial neural network. (B) Representation of the dataset as a vector in latent space. The best result was obtained with one neuron in the narrow layer. The red lines represented the classification of the latent space and were obtained with Otsus multi-label method. (C) The vectors of the output space produced by the decoder according to the vectors in the latent space. Some of the presented action potentials were related to certain electrophysiological effects. Left to right: hyperpolarization (C1), activation faults (C2), normal atrial action potentials with triangular shape (C3), action potentials with a plateau phase (C4), action potentials with early afterdepolarization (C5), and physiologically meaningless artifacts (C6). (D) All action potentials of the output space that were related to automatically proposed classifications.

A sampling of the input space was performed using the Latin hypercube algorithm. Each individual model with a sampled input vector of parameters was computed by means of the forward Euler method with a time step of 0.001 ms. Every model was paced at 1Hz for 100 cycles to reach a steady-state solution, then the last output signal of the action potential generated by the model was stored as a sample vector of the output space. Unconverted solutions with NaN values were not included in the output space. To simplify further data processing, all the vectors from the output space were resized to 55 time-dependent elements by means of linear interpolation. Model computation was performed on a computational accelerator (NVIDIA Tesla K80) using the Myokit software package [9] with an original in-house extension.

Finally, the output spaces produced by the populations of either the *Courtemanche98* or the *Maleckar09* ionic model of the cardiomyocyte consisted of 10000 output vectors of the time-dependent signals of the action potentials.

B. Clusterization Approach with Dimensionality Reduction

To organize the data into clusters, we sequentially applied methods of dimensionality reduction and clusterization that were not based on neural networks.

The data were rescaled 12 times using a variety of options implemented in the *scikit learn* package [10] for the Python language: *MinMaxScaler*, *MaxAbsScaler*, *StandardScaler*, *Normalizer* (with the l_1 or l_2 norm), *RobustScaler* (with scaling centering, without scaling centering, with scaling but without centering, with centering but without scaling), *PowerTransformer*, and *QuantileTransformer* (with normal or uniform output distribution).

The last option was an absence of any normalization at all. In this case, further processing was performed on the original data.

Dimensionality reduction was performed with the t-SNE [11] and UMAP [12] methods. The number of components was equal to two. As in the previous stage, the absence of any dimensionality reduction was also an option for this data processing step.

Clusterization was performed with agglomerative clustering, K-means clustering, and Gaussian mixture clustering with 8, 12, and 16 clusters [13].

In total, three data processing pipelines were used in testing a number of approaches. The quality of the results was analyzed manually based on expert opinion about the clusterization relevance for data analysis.

The most notable results are presented in Fig. 1. They were obtained for the data processing pipeline with no dimensionality reduction and combining *MaxAbsScaler* and agglomerative

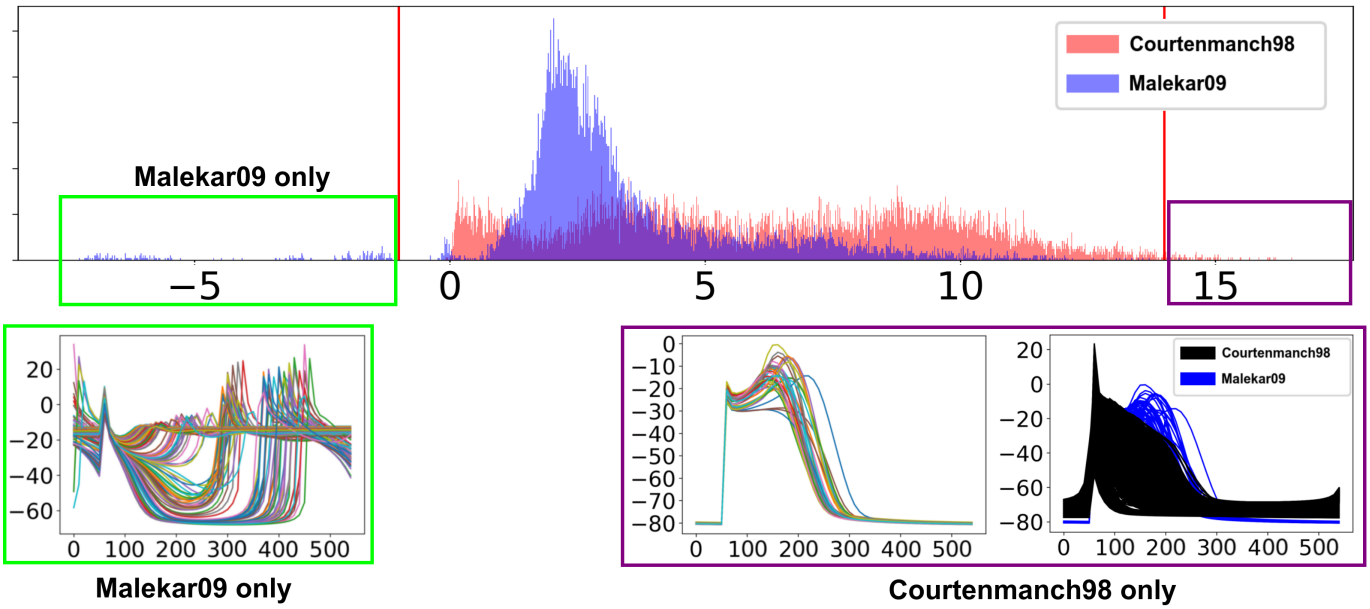

Fig. 3. Comparison of the model populations in the latent space. The non-intersected region of the histogram corresponds to the action potential shapes which were included in only one type of models. The left image includes the delayed afterdepolarizations, the early afterdepolarization-induced sustained triggered activity, and some "broken" models that were provided by the *Maleckar09* model, but not the *Courtenmanch98* model. The right image shows delayed auto-depolarization with action potential shapes that were not observed in the *Maleckar09* model. The additional image shows the intersection of the *Maleckar09* model without delayed afterdepolarizations and the specific shapes of the *Courtenmanch98* model solutions.

clustering with Euclidean distance. The Silhouette value was equal to 0.2977, and the Calinski-Harabasz score was equal to 11738.20.

Clusters that were determined by means of the pipeline were related to the shapes of action potentials that were identified in the electrophysiological studies. There were action potentials of a triangular shape, action potentials of trapezoidal shape with a plateau phase during repolarisation, and action potentials with abnormal repolarization of an early afterdepolarization type.

After processing, some classes may have been merged together according to expert opinion to improve clusterization in terms of the physiological meaning of each class. For example, in Fig. 1, the action potentials of triangular shape may be annotated as one class. However, two classes contained action potentials with different shapes and physiological behaviors (marked by '?' in Fig. 1). Thus, clusterization could not correctly separate all the shapes.

C. Clusterization Approach with Deep Autoencoder

In the second approach, we used deep neural networks for dimensionality reduction and the analysis of variables in the latent space for the clusterization of vectors in the output space.

We examined six approaches to the normalization of the output space: 1) an entire vector normalization to the [0,1] range according to the min-max values of each vector, 2) an entire vector normalization to the [0,1] range according to the min and max absolute values of the vector, 3) an entire vector normalization to the [0,1] range according to the mean and standard deviation values of the vector, 4) a component-wise vector normalization to the [0,1] range according to the min-max values of the components across all vectors, 5) a component-wise vector normalization to the [0,1] range according to the min and max absolute values of the components across all vectors, and 6) a component-wise vector normalization to the [0,1] range according to the mean and standard deviation values of the components across all vectors.

An autoencoder solved the problem $\mathbf{y} = h(g(\mathbf{y}))$, where the transformation from the output space to the latent space was performed with the encoder, $\mathbf{z} = g(\mathbf{y})$, and the reconstruction from the latent space to the output space was performed by the decoder, $\mathbf{y} = h(\mathbf{z})$. The general concept of this artificial neural network is presented in Fig. 2A.

We used an autoencoder with three hyperparameters: number of neurons in the narrow layer, depth of the autoencoder, and function of activation in all layers. The latent space provided from 1 to 10 neurons. The encoder consisted of 55 neurons in the input layer and twice fewer neurons in each following fully-connected layer. Layers of the decoder consisted of the same number of neurons as the encoder. Thus, the autoencoder depth varied from one layer to four layers in the encoder and the decoder. Because the vectors in the output space were normalized to the range [0,1], we also varied the activation function in each layer: sigmoid, softmax, and ReLU (rectified linear unit). In the last layer of the decoder, we used a linear activation function.

In this way, the described approach provided us with a family of autoencoders of 120 different configurations. Clus-

terization of the vectors in the latent space was performed with the DBSCAN clusterization method and Otsus multi-label thresholding in the case of one component vectors used after encoding.

The most notable results are presented in Fig. 2. They were obtained for a deep autoencoder with one component in vectors of the latent space, four layers in the decoder and encoder, a ReLU activation function, and Otsus multi-label thresholding approach for the analysis of the latent space.

Because of the lack of suitable metrics, the choice of the best processing approach was done manually according to a visual analysis of the results, see Figs. 2B-2D. The proposed method allowed us to classify the data into physiologically meaningful groupings, such as the signals with hyperpolarization and activation faults, the normal atrial action potentials of a triangular shape, the action potentials with a plateau phase, and the action potentials with early afterdepolarization.

The best clusterization approach had a silhouette value equal to 0.7521, and a Calinski-Harabasz score equal to 8506.84 for the vectors in the output space. For values in the latent space, the silhouette value was equal to 0.3244, and the Calinski-Harabasz score was equal to 33238.23. Intra-group dispersion for values in the latent shape was equal to 12.29, 17.02, 26.71, and 28.39.

D. Comparison of Populations in Latent Space

We used the latent space approach for the comparison of the model populations generated by different ionic cellular models.

The populations of *Maleckar09* and *Courtenmanch98* models were processed using the autoencoder configuration described in the previous section. The representation of the latent space as a histogram for each of the model populations revealed some ranges of the space where the models did not overlap (see Fig 3). This analysis allows one to reveal specific model samples where the action potential shapes specific to one ionic model can not be generated by another model for a certain range of model parameters.

The most notable results are presented in Fig 3. The *Maleckar09* model population provided action potential shapes with the delayed afterdepolarizations; the early afterdepolarization-induced sustained triggered activity and some "broken" action potentials. The *Courtenmanch98* model did not provide all of these. On the other hand, the *Courtenmanch98* model provided a sub-population with early afterdepolarization and specific action potential shapes that have not consisted of the *Maleckar09* model. That sub-population is presented in Fig. 3. This type of model behavior would be almost impossible to identify if the analysis was based only on the typical action potential biomarkers without the application of a deep-learning approach.

III. Discussion and Conclusion

As we have shown, the model population may be classified according to the specific features of the action potential morphology and visualized as the latent space of the autoencoders. Here, we will focus on the main benefits of the proposed approaches.

The transformation of the data set to the latent space may be used for visualization of the model population, as shown in Fig. 2. This visualization allows the action potentials that can be produced by the model to be distinguished by their phenotypes.

Classification using artificial intelligence significantly simplified the analysis of the model population. Expert classification of the action potential shape of more than 10,000 images is hardly possible. At the same time, the final choice of proper classification from many possible configurations suggested by different analysis procedures required expert evaluation but from the smaller set of grouped images (see Fig. 3). Some clusters may be easily combined for improved results of the classification following expert opinion.

As shown in Fig. 3, some specific hidden features of the model behavior may be revealed via comparison of the models within the entire latent space.

References

[1] K. Shekhar, S. W. Lapan, I. E. Whitney, N. M. Tran, E. Z. Macosko, M. Kowalczyk, X. Adiconis, J. Z. Levin, J. Nemesh, M. Goldman *et al.*, "Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics," *Cell*, vol. 166, no. 5, pp. 1308–1323, 2016.

[2] E. Passini, O. J. Britton, H. R. Lu, J. Rohrbacher, A. N. Hermans, D. J. Gallacher, R. J. Greig, A. Bueno-Orovio, and B. Rodriguez, "Human in silico drug trials demonstrate higher accuracy than animal models in predicting clinical pro-arrhythmic cardiotoxicity," *Frontiers in physiology*, vol. 8, p. 668, 2017.

[3] S. Coveney and R. H. Clayton, "Fitting two human atrial cell models to experimental data using bayesian history matching," *Progress in biophysics and molecular biology*, vol. 139, pp. 43–58, 2018.

[4] J. P. Keener and J. Sneyd, *Mathematical physiology*. Springer, 1998, vol. 1.

[5] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval." in *ESANN*, vol. 1, 2011, p. 2.

[6] M. Courtemanche, R. J. Ramirez, and S. Nattel, "Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 275, no. 1, pp. H301–H321, 1998.

[7] M. M. Maleckar, J. L. Greenstein, W. R. Giles, and N. A. Trayanova, "K+ current changes account for the rate dependence of the action potential in the human atrial myocyte," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 297, no. 4, pp. H1398–H1410, 2009.

[8] T. Nesterova, D. Shmarko, and K. Ushenin, "In silico study of the aging of cardiomyocytes in human and canine atriums," in *AIP Conference Proceedings*, 2019, [in print].

[9] M. Clerx, P. Collins, E. de Lange, and P. G. A. Volders, "Myokit: A simple interface to cardiac cellular electrophysiology," *Progress in Biophysics and Molecular Biology*, vol. 120, no. 1–3, pp. 100–114, 2016.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[12] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[13] S. K. Popat and M. Emmanuel, "Review and comparative study of clustering techniques," *International journal of computer science and information technologies*, vol. 5, no. 1, pp. 805–812, 2014.