Acoustic Identification of Cardiomyocytes
Alex Lemon

1 Introduction

Pluripotent stem cells can be made to differentiate into cardiomyocytes (heart-muscle cells). However, there are three different types of cardiomyocytes: atrial, ventricular, and nodal, and it is difficult to force cells to differentiate into a particular type. In therapeutic applications it is extremely important to use the correct type of cardiomyocytes – implanting the wrong type of cell will not restore function as desired, and may lead to cancer. Thus, it is extremely important to be able to accurately identify different types of cardiomyocytes. Moreover, the identification process must be minimally invasive if the cells are to be subsequently used for therapy. In this paper we use noninvasive acoustic measurements of heart-muscle cells to classify the cells as atrial, ventricular, or nodal; we apply several classification methods to the data, including K-nearest-neighbors, multinomial regression, discriminant analysis, support-vector machines, and tree-based classifiers.

2 Related work

Junyi et al showed that the different types of cardiomyocytes can be identified using the electrical signals associated with action potentials [4]. These measurements are obtained using an invasive technique called patch clamping, which renders the cells unfit for subsequent therapeutic use. More recently a team of researchers at Stanford has been using a photonic-crystal hydrophone to take acoustic measurements of murine cardiomyocytes. (This work is currently unpublished, although technical details of the measurement device are available [3].) Catherine Jan (a doctoral student in the Department of Electrical Engineering, Stanford University) and Sally Kim (a postdoctoral researcher in the Department of Psychiatry, Stanford University) provided their acoustic measurements for our analysis.

3 Data and preprocessing

The data set consists of thirty-four time traces ranging in length from thirty seconds to two minutes. The original data was sampled at $10\,\text{kHz}$, while the features of interest occur in a frequency range that is an order of magnitude slower. I used $5\times$ downsampling with averaging to reduce the size of the data, while preserving the features of interest.

After downsampling the data, I developed a method for extracting the individual pulses. This method was based on three assumptions about the nature of the data.

(1) Each pulse can be approximated as a triangle wave of width $w = 10\,\text{ms}$; we let ϕ_t denote the triangle wave that starts at time t (see figure 1).

(2) The pulses corresponding to different cells are added to give the observed signal. (In particular, different pulses cannot cancel each other out.)

(3) Pulses begin at relatively few of the time sample points.

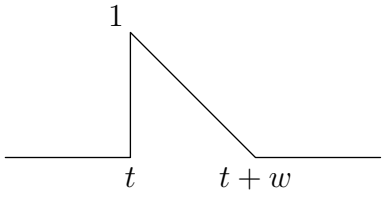


Figure 1 – a triangle wave

Based on these assumptions, we propose extracting the individual pulses by solving the following optimization problem:

$$\begin{aligned} &\underset{\alpha}{\text{minimize}} : \|x - \textstyle\sum_\tau \alpha_\tau \phi_\tau\|_2^2 + \rho\|\alpha\|_1 \\ &\text{subject to} : \alpha \geq 0, \end{aligned}$$

where $\rho > 0$ is a regularization parameter. Intuitively, we approximate the time trace as a weighted sum of triangle waves. The first term in the objective is the approximation error, which we want to be small.

The penalty term $\|\alpha\|_1$ is used to promote sparsity, corresponding to the assumption that pulses start at relatively few of the time sample points. The constraint $\alpha \geq 0$ models the assumption that the measured signal is obtained by adding the pulses corresponding to different cells, and there is no cancellation. Thus, the optimization problem above is a form of sparse, nonnegative regression. An example of the results are shown in figure 2, where we see that the method is effective at extracting the pulses.

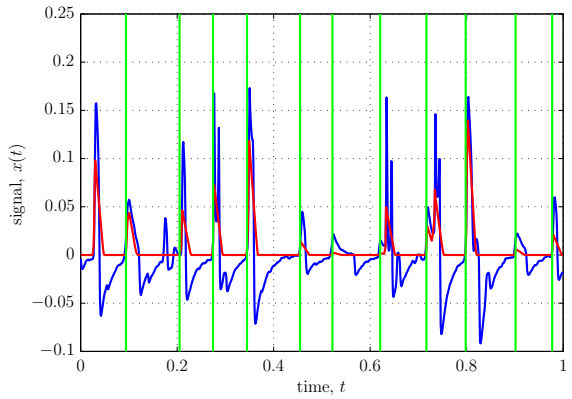
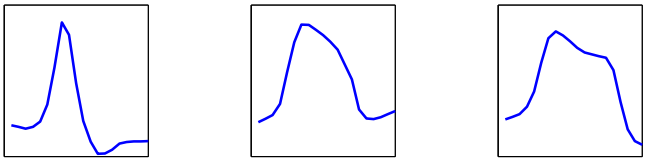


Figure 2 – approximating the measured signal using triangle waves

Having extracted the pulses from the measurements, we manually labeled 1220 observations. Representative examples of the pulses corresponding to the three different types of cells are given in figure 3. We labeled all of the observations twice in order to assess the quality of our labels. The two labels differed for approximately one quarter of the examples, indicating that the problem is difficult, and suggesting that an error rate of about 25 % should be considered a success.



(a) atrial **(b)** nodal **(c)** ventricular

Figure 3 – pulses corresponding to the three different types of cardiomyocytes

4 Classification methods

The following classification methods were applied to the data. More detailed explanations of these methods are given in James et al [2] and Hastie et al [1].

(a) *K-nearest neighbors.* In this method we classify an observation with predictor vector x_0 by identifying the K training examples whose predictor vectors are closest to x_0, and using a majority vote among these neighboring training examples. (We used the Euclidean distance to measure closeness, but it is also possible to use other metrics.) The parameter K must be chosen to balance bias (which increases with K) and variance (which decreases with K).

(b) *Multinomial logistic regression.* An extension of the usual two-class logistic regression, K-class multinomial logistic regression assumes that the conditional class probabilities are

$$\mathbb{P}(Y = k \,|\, X) = \frac{\exp(\beta_1^{\mathsf{T}} X)}{1 + \sum_{k=1}^{K-1} \exp(\beta_k^{\mathsf{T}} X)}, \qquad k = 1, \ldots K-1,$$

$$\mathbb{P}(Y = K \,|\, X) = \frac{1}{1 + \sum_{k=1}^{K-1} \exp(\beta_k^{\mathsf{T}} X)},$$

where $\beta_1, \ldots, \beta_{K-1}$ are parameters, which we can fit using the method of maximum likelihood.

(c) *Support-vector machine (SVM).* A support-vector machine partitions the predictor space in order to maximize the distance of the observations from the decision boundaries, while keeping the classification errors small. More concretely, we fit an SVM by solving the following optimization problem:

$$\begin{aligned} &\underset{\beta_j,\, \epsilon_i,\, M}{\text{maximize}} : M \\ &\text{subject to} : \quad \sum_{j=1}^{p} \beta_j^2 = 1 \\ &\qquad\qquad y_i(\beta_0 + \textstyle\sum_{j=1}^{p} \beta_j x_j) \geq M(1 - \epsilon_i) \\ &\qquad\qquad \epsilon_i \geq 0 \\ &\qquad\qquad \textstyle\sum_{i=1}^{n} \epsilon_i \leq C. \end{aligned}$$

Important considerations for SVMs are the kernel (that is, the measure of similarity between predictors) and the cost parameter C. In addition, because our classification problem has more than two classes, we need to decide whether to use one-versus-one comparisons or one-versus-all comparisons. (In a one-versus-one comparison, we fit an SVM for all pairs of classes, and predict the class that appeared most in the pairwise classifications; in a one-versus-all comparison, we fit an SVM for each class against all of the other classes, and predict the class with the highest confidence.) We used the standard radial kernel and one-versus-alls comparisons in this project.

(d) *Discriminant analysis.* In discriminant analysis we assume that the predictors from each class come from a multivariate normal distribution, with possibly different parameters for each class. We estimate the distribution parameters using their sample analogs (that is, the sample mean and sample variance), and we classify a new observation to the class with the highest probability density at the observed value of the predictor vector.

(i) *Linear discriminant analysis (LDA).* If we assume that the variance matrix is the same for all classes, then we obtain linear decision boundaries.

(ii) *Quadratic discriminant analysis (QDA).* If we assume that the variance matrix is different for different classes, then we obtain quadratic decision boundaries. We tend to obtain better results with QDA than LDA if the variance matrix differs substantially across classes, and worse results otherwise.

(e) *Tree-based methods.*

(i) *Classification trees.* In a classification tree, we recursively partition the training set using binary decisions based on the predictors. In order to classify a new observation, we apply the same sequence of binary decisions to the predictors of the new observation, and label the observation using majority vote when we reach a terminal node. Trees are very prone to overfitting, which can be ameliorated by tuning the number of terminal nodes.

(ii) *Random forests.* A random forest is an ensemble of classification trees. The trees in the ensemble are decorrelated by bootstrapping the data used to generate the trees, and randomly selecting the predictors that can be used for decisions at each node. Classification is performed using a majority vote of the trees in the ensemble. Important parameters are the number of trees in the forest, and the number of predictors selected at each node.

(iii) *Boosting.* Like random forests, boosting also uses an ensemble of trees. However, instead of producing a decorrelated ensemble, boosting instead uses subsequent trees to fit the errors in previous trees. The key parameters are the number of trees and the learning rate.

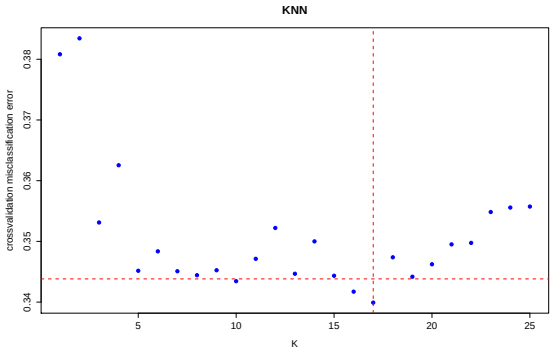
5 Results

All algorithm parameters were chosen using ten-fold cross-validation repeated ten times (that is, with ten different random groupings for cross-validation; repeating cross-validation gives better estimates of the error, and decreases the standard error of these estimates). In particular, we chose the number of neighbors for K-nearest neighbors, the number of terminal nodes in the classification tree, the number of predictors

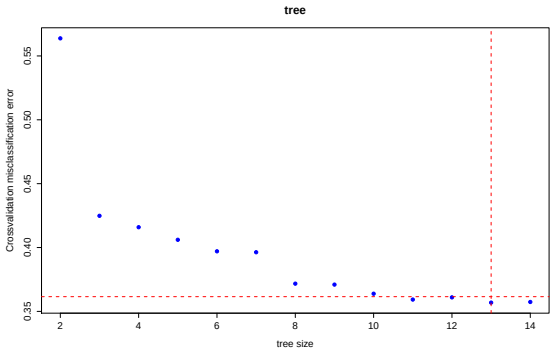
chosen for each node in the random forest, and the number of trees in the boosting ensemble. The cross-validation curves are shown in figure 4. The vertical red lines in these plots mark the minimum error, while the horizontal red lines are one standard error above the minimum. The final fitted models were chosen using the one-standard-error rule: that is, we selected the simplest model that was within one standard error of the minimum cross-validation error.

Performance metrics for the different classification algorithms are given in table 1. Many of the methods exhibit substantial overfitting (that is, a training error much lower than the cross-validation error). I attempted to reduce overfitting by using the one-standard-error rule to select less complex models, and this seemed particularly effective for K-nearest neighbors, where we see a U-shaped cross-validation curve. Nevertheless, substantial overfitting remains, particularly for the random forest. I think much of this overfitting can be attributed to the noisy labels in the data. As mentioned above, I labeled all of the data twice, and the labels were inconsistent for about one quarter of the observations. With such a relatively low signal-to-noise ratio in the training data, overfitting is very difficult to avoid.

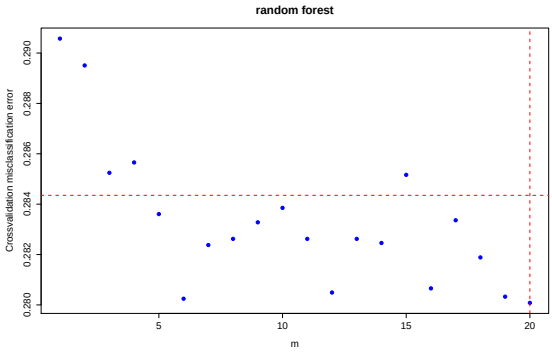
Several algorithms came close to the $25\,\%$ error rate that is a lower bound on the accuracy due to the noisy labels. In particular, the random forest achieved an error rate of $28\,\%$. The multinomial logistic regression did not perform well, indicating that linear decision boundaries are probably not appropriate. Both forms of discriminant analysis performed especially poorly, which is likely a result of the fact that the assumptions of these generative models are not even approximately correct. (Recall that these models assume that the predictor vector for each class has a multivariate normal distribution; there is no reason to think our data satisfy this assumption.)



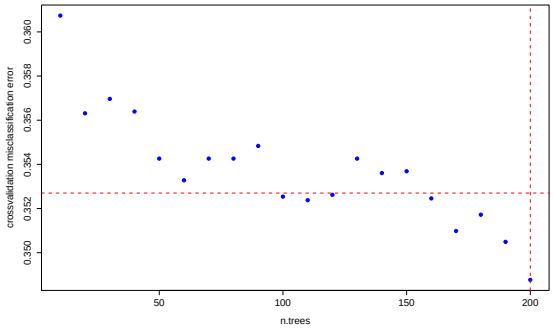
(a) KNN



(b) tree



(c) random forest



(d) boosted tree

Figure 4 – CV curves for choosing algorithm parameters

6 Conclusion

We used acoustic measurements to classify cardiomyocytes as atrial, nodal, or ventricular; such a noninvasive classification procedure is essential for practical cardiac stem-cell therapy. The classification problem is very difficult – a human only managed to achieve an error rate of about $25\,\%$. We obtained an error rate

algorithm	training error	training confusion matrix	cross-validation error	cross-validation confusion matrix
KNN	0.2639	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 392 & 71 & 18 \\ 113 & 241 & 51 \\ 35 & 34 & 265 \end{bmatrix} \end{array}$	0.3434	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 355.5 & 100.5 & 25.0 \\ 145.5 & 197.0 & 62.5 \\ 39.5 & 46.9 & 247.6 \end{bmatrix} \end{array}$
multinomial	0.3770	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 362 & 75 & 44 \\ 153 & 186 & 66 \\ 65 & 57 & 212 \end{bmatrix} \end{array}$	0.3939	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 355.5 & 80.7 & 44.8 \\ 156.7 & 179.9 & 68.4 \\ 66.6 & 63.4 & 204.0 \end{bmatrix} \end{array}$
SVM	0.2902	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 417 & 42 & 22 \\ 139 & 183 & 83 \\ 34 & 34 & 266 \end{bmatrix} \end{array}$	0.3301	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 399.6 & 50.3 & 31.1 \\ 160.4 & 160.7 & 83.9 \\ 42.3 & 34.7 & 257.0 \end{bmatrix} \end{array}$
LDA	0.3852	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 354 & 85 & 42 \\ 150 & 189 & 66 \\ 62 & 65 & 207 \end{bmatrix} \end{array}$	0.4008	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 343.0 & 96.0 & 42.0 \\ 152.2 & 186.4 & 66.4 \\ 65.0 & 67.4 & 201.6 \end{bmatrix} \end{array}$
QDA	0.4115	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 224 & 236 & 21 \\ 57 & 321 & 27 \\ 26 & 135 & 173 \end{bmatrix} \end{array}$	0.4447	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 215.5 & 240.0 & 25.5 \\ 73.5 & 298.3 & 33.2 \\ 31.5 & 138.8 & 163.7 \end{bmatrix} \end{array}$
tree	0.3139	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 372 & 65 & 44 \\ 124 & 23 & 48 \\ 31 & 71 & 232 \end{bmatrix} \end{array}$	0.3610	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 366.2 & 70.9 & 43.9 \\ 145.9 & 194.2 & 64.9 \\ 53.4 & 65.8 & 214.8 \end{bmatrix} \end{array}$
random forest	0.0008	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 481 & 0 & 0 \\ 0 & 405 & 0 \\ 0 & 1 & 333 \end{bmatrix} \end{array}$	0.2801	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 386.5 & 73.1 & 21.4 \\ 116.0 & 235.4 & 53.6 \\ 33.4 & 45.9 & 254.7 \end{bmatrix} \end{array}$
boosted tree	0.3262	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 421 & 44 & 16 \\ 186 & 167 & 52 \\ 70 & 30 & 234 \end{bmatrix} \end{array}$	0.3525	$\begin{array}{c} \\ A \\ N \\ V \end{array}\begin{array}{c} \begin{array}{ccc} A & N & V \end{array} \\ \begin{bmatrix} 414.2 & 45.8 & 21.0 \\ 189.4 & 158.8 & 56.8 \\ 75.5 & 41.2 & 217.3 \end{bmatrix} \end{array}$

Table 1 – performance of classification algorithms

of 28 % using a random forest, and also had success with K-nearest-neighbors, support-vector machines, classification trees, and boosted trees; multinomial regression and discriminant analysis performed poorly. The largest source of error for most methods was classifying nodal cells as atrial.

The most important step for future work is obtaining more accurate training data. The labels used in this project were not assigned by a biologist; if a domain expert were to assign the labels, then we could obtain more accurate training data, which should reduce the overfitting problems that we observed. Many of the methods can probably be improved by further tuning. For example, we did not use cross-validation to adjust the cost parameter in the SVM, or the number of trees in the boosting algorithm. It may also be worth investigating specific methods for differentiating between these two types of cells, perhaps adding special features that can identify this difference. Finally, another interesting direction of research is using multiple sensors simultaneously. If we have one sensor for each clump of cells, then we can use independent-components analysis (ICA) to separate the signals corresponding to the different clumps.

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.

[2] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.

[3] Catherine Jan, Wonuk Jo, Michel J. F. Digonnet, and Olav Solgaard. Photonic-crystal-based fiber hydrophone with sub-100 µPa/$\sqrt{Hz}$ pressure resolution. *IEEE Photonics Technology Letters*, 28(2):123 – 126, 2016.

[4] Junyi Ma, Liang Guo, Steve J. Fiene, Blake D. Anson, James A. Thomson, Timothy J. Kamp, Kyle L. Kolaja, Bradley J. Swanson, and Craig T. January. High purity human-induced pluripotent stem cell-derived cardiomyocytes: electrophysiological properties of action potentials and ionic currents. *American Journal of Physiology: Hearty and Circulatory Physiology*, 301(5):H2006 – H2017, 2011.