CompBioMed Laboratory Report: A multidisciplinary Pipeline for Ventricular Arrhythmia Localization

Al Hakioui Z., Boncompte P., Cardona A., Roca N.

ABSTRACT

Ventricular arrhythmia is a pathology generated when the electrical system of the heart is disrupted. This leads to an abnormal heart rhythm that worsens the heart function, and possibly leading to other heart diseases or several consequences for the life of the patients. Hence, the task of detecting if there are electrical re-entrances and their site of origin of the arrhythmias is crucial for a good treatment.

In this paper, we present a multidisciplinary project for simulating an arrhythmic scenario and for detecting the heart regions that lead to a ventricular arrhythmia. This is done by segmenting an MRI cardiac image from a patient in an open-source software, generating a 3D model from it and performing several simulations of the cardiac cycle in different software. Moreover, we also make use of a machine learning model to identify the origin of idiopathic right ventricular tachycardias and show an open-source app for the visualization and study of ECG signals.

The results show that this computational pipeline approach may have the potential to provide realistic and reliable predictions of the presence of electrical re-entries and their site of origin. Moreover, the Machine Learning model seems to be a very useful tool for classifying RVOT, and the open-source app for ECG simulation provides a visual insight into the underlying phenomena causing the different ECG waves.

I. INTRODUCTION

Ventricular arrhythmia refers to an abnormal heart rhythm that can disrupt the normal pumping action of the heart. This can be generated by disorders in the heart's electrical system. Normally, the electrical stimulus is generated in the sinoatrial node, and it travels through the right and left atria. Then, this signal travels to the atrioventricular node, where it goes slower for a short time to allow the atria to contract before the ventricles. After that, the signal travels down the bundle of His and finally extends along the ventricles through the Purkinje fibers, thus causing them to contract in order to pump the blood out of the heart.

When a ventricular arrhythmia occurs, this electric pathway is disordered, and it can lead to ventricular tachycardia. The disruption of the heart's electric system can consist of the re-entrance of the electric signal in the ventricle due to the presence of scar tissue because of a myocardial infarction [1] (scar-related re-entrant ventricular tachycardia).

Radiofrequency ablation has become one of the most effective procedures for treating ventricular arrhythmia [2]. During this intervention, a catheter is inserted into the heart of the patient. There, the regions of the heart where the abnormal electric signals appear are localized and then "ablated". With that, the heart tissue that is causing the arrhythmia is destroyed. One of the most challenging issues of this procedure is the localization of the heart regions that must be ablated.

In this study, we show a multidisciplinary project involving the use of medical imaging, simulation and machine learning techniques to study the case of a patient arrythmia on the left ventricle (LV) and obtain the location of interest for clinicians to perform a radiofrequency ablation as a treatment.

In addition to our main pipeline for the clinical case, we report the use of two models for the study of heart diseases. The first one is ECGSIM [3], an interactive simulation tool intended both for educational and research purposes that allows to study the relationship between the electric activity of the heart and the electric potentials that can be observed in an ECG (the PQRST wave forms).

In this case, we will study a case of Right Bundle Branch Block (RBBB), a pathology where the electrical impulses that control the contraction of the right ventricle of the heart are slowed down or even blocked [4]. We will replicate this condition in the heart model to observe how the ECG is affected in the different derivations.

We also include a Machine Learning-based approach to identify the origin of idiopathic right ventricular tachycardias (RVOT).

II. METHODOLOGY

As said in the previous section, this work will be divided in 3 parts: Arrhythmia detection, ECG study with ECGSIM and RVOT classification with Machine Learning. We are going to introduce and explain the methodology used in each section separately.

A. Arrythmia detection

1) Image Segmentation

We are provided cardiac medical image data of 10 patients. Five of these patients are pro-arrhythmic and five they are not. Our case is patient 2, which we cannot tell yet in which group could be classified. The first task that we have is the segmentation and the post-processing of this data. The goals of this first part are to get familiarized with an open-source software for segmentation such as 3DSlicer [5], the segmentation of the left ventricle geometry and the scar from the medical images, to generate a 3D model based on the result and finally its post-processing.

The data of patient 2 is a cardiac MRI. We know from literature [6] that in a magnetic resonance the regions with blood vessels and scars appear in a brighter color, however, cardiac muscle and air appear in a darker color. We can see an image of the 3DSlicer planes in figure 1.

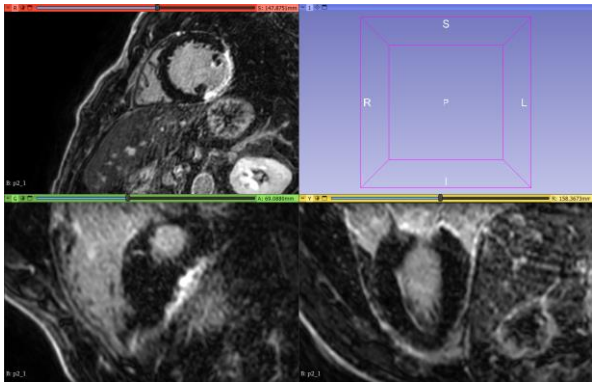

Fig 1. Image from 3DSlicer. Cardiac MRI from patient number 2. Top-left: axial plane, bottom-left: coronal plane, bottom-right: sagittal plane.

In order to segment the left ventricle geometry, we use this open-source software and its tools. We start by importing the data of the patient into the interface. We then used the "Paint" tool with a sphere brush, which makes easier to paint in different cuts of the plane at the same time, and an editable intensity range to ensure we are segmenting our Region of Interest (ROI). This intensity range it has been changing during the time, but it oscillates from 145.27 to 2103. For the scar, a very similar process is followed. Nonetheless, now the range goes from 2103 to the maximum.

After this, we use other features such as "Remove small islands" to keep the largest island and to avoid noise in our form. We also make use of a slight smoothing in the same software; the minimum size of this smoothing is 4000 voxels. This process helped us to get a preliminary segmentation of the desired geometry.

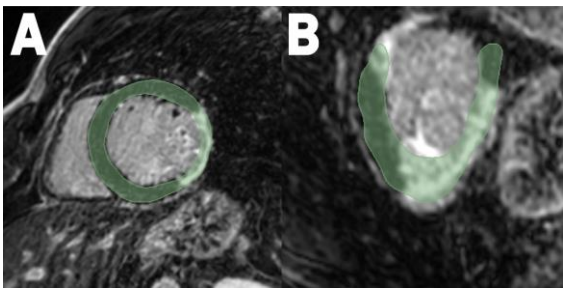

Fig 2. Painted region of LV in the axial plane (A) and in the sagittal plane (B).

After obtaining a preliminary outline, we import it into Meshmixer [7] to improve the mesh and get a more similar result to the one that is provided to us. Meshmixer is a software that allows users to edit, manipulate and repair 3D meshes. First, we show the wireframe of the figure with the "view" options.

Afterwards, we use the mesh reduction and smoothing functionalities of Meshmixer to get a simple figure to work with. These functionalities helped to reduce the number of faces and vertices in the wireframe, which makes it easier to work with in the next steps.

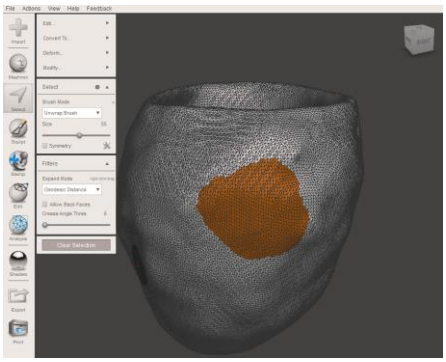

Fig 3. Some tools that Meshmixer offers in order to improve our mesh.

After segmenting and improving the mesh, we loaded the segmentation in Paraview [8] for post-processing purposes. Paraview is an open-source software that provides an interface for visualization and analysis of large datasets. We use some of the features in order to enhance the geometry of the model.

The filters that we use in this platform are "Extract surface", "Tessellate", "Decimate" and "Triangulate". Here is a little introduction of each one of these filters:

-Extract surface: This one extracts the outer surface of our segmentation. Thus, we just preserve the surface geometry and delete the inside of the figure. This is helpful when we want to simplify the model and analyze just the outer information.

-Tessellate: The Tessellate filter transforms a polygonal wireframe into a triangular mesh, this is also for simplicity purposes.

-Decimate: It reduces the number of vertices but preserves the visual information of the geometry. So as the other filters, this filter helps to reduce the complexity.

-Triangulate: Very similar to Tessellate, the triangular filter also converts the different polygonal faces of the mesh into triangles.

We start the post-processing with 115,356 cells and 57,682 points. These tools helped us to simplify the model, smooth the mesh and remove unnecessary features. Finally, we must make sure that our structure has around 20,000 cells and 10,000 points.

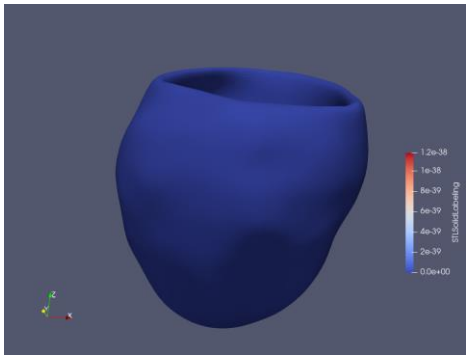

Fig 4. Geometry visualized in Paraview before the post-processing filters.

As we said before, in addition to segmenting the left ventricle geometry, we also segment the scar of the LV wall. This scar is divided in two components: Scar core zone and border zone. The border zone is the region surrounding the injury core and it is the transition between the myocardium and the scar. The core zone is the central region of this cicatrix. To segment these regions, we use the "Grow seeds" feature and the "Smoothing" functionalities of the 3DSlicer. This time it is not necessary to enhance the mesh with Meshmixer and Paraview because the result is very similar to the one provided for the following steps.

2) Simulation

In this part of the labs, we are provided with a pre-existing segmentation like we said previously, and we will not be making use of the segmentation that we performed ourselves. Nonetheless, the geometry that we have now has been processed with ADAS Medical 3D [9]. This software employs advanced algorithms and techniques to precisely segment cardiac structures from three-dimensional medical images, like the MRI in this case. These geometries generated provide an accurate view and analysis of different components within the heart, including ventricles, atria, scars, and other significant anatomical characteristics. Thus, the folder generated by ADAS 3D contains: Border zone, Core surface, endo layer, epi layer, left ventricle, myocardium, scar surface and the full magnetic resonance image.

The goals for this part are to prepare the files for the simulation with help of a Python [10] code, to simulate and analyze a single beat and to perform a parameter sweep to detect all the simulation that contain re-entries.

Additionally, to Paraview, which was used before, now we need Python and the libraries: VTK (9.2.6), NumPy (3.10.2), Os (3.1.0) and Shutil (3.8.0). Moreover, we will make use of Processing [11] with the libraries: Control P5 (2.2.6), ToxicLib (22) and PeasyCam (302).

The first thing we must do is to place the folder that contains the ADAS 3D segmentations next to the Python script. Afterwards, we make in the corresponding changes in the script in order to have our patient as the input. Once we run this code, we will have a subfolder as the output. This folder contains three files:

-*endo.stl*: This file contains the inner part of the ventricle and is our reference for the visualization.

-*params.dat*: This document contains the configuration file for the Arritmic3D.

-*ventricle_tagged.vtk*: This file can be visualized in Paraview and have all the information related to the geometry: ventricle, scar tissue, fibers orientation, etc.

From the mentioned files the most interesting is the last one because it allows us to see all the relevant parameters and the voxelated left ventricle geometry in Paraview. This geometry is a cellular automaton, which is a mathematical and computational model that is based on a grid of cells. The basis of this kind of model and finite elements is different.

Here in figure 5, we have some examples where with the help of the filters of the post-processing platform we can analyze the geometry.

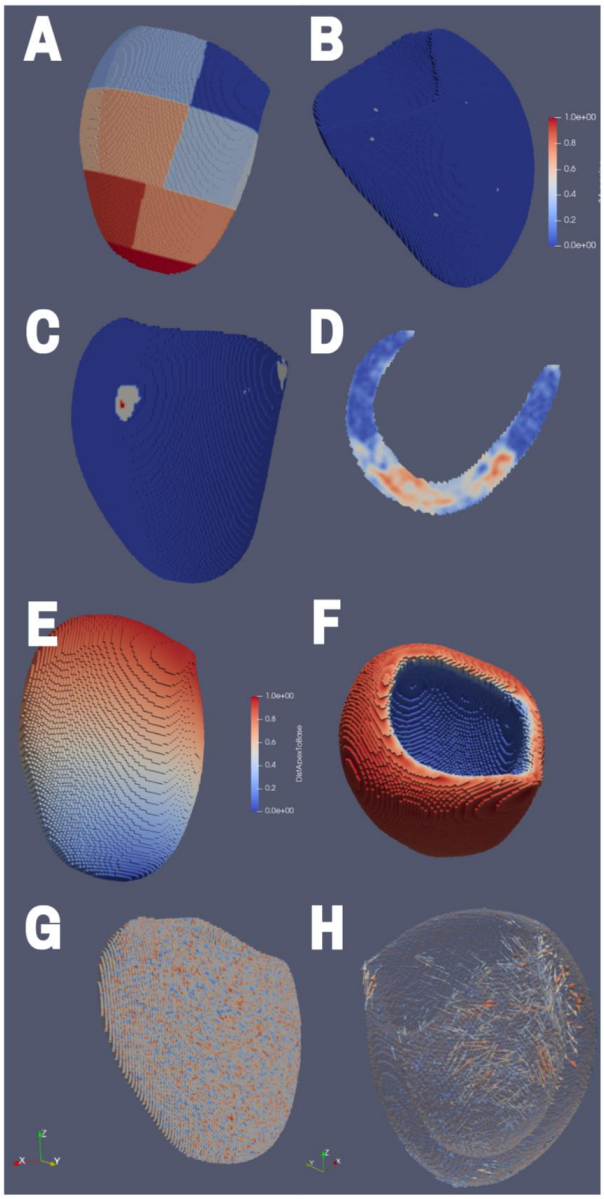

Fig 5. Here we can observe some displays of the geometry. A: 17_AHA. B: 34_pacing. C: Cell_type. D: Cell_type with silce filtering. E: DistApexToBase . F: DistEndoToEpi. G: Fibers_OR. H: Fibers_OR with a Glyph filter.

The feature 17_AHA that Paraview allows us to see it according to the 17 segments AHA standard. The anatomy of the left ventricle can be divided into multiple segments, but this standard model recommends based on autopsy data, functionality analysis and myocardial perfusion 17 regions [12].

In figure 5 “B” we can also observe the 34_pacing information. The Arritmic3D works with a pacing protocol that behaves differently with different stimuli. This shows the 34 possible stimulation points, which are the 17 AHA segments for the endo and epi. The images “C” and “D” we see the type of the cells. We are given a unique information for each voxel depending on the state of the tissue. We can notice easily that most part of the epicardium seems to be healthy, but if we watch a section of this geometry, we observe that there is a scar. For images “E” and “F” the information is about the distance between the apex to the base and from the epicardium to the endocardium, in this order. Finally, we can see the orientation of the fibers in the last two images.

3) Data Visualization

After testing with the Paraview, it is time to visualize this simulation in Processing. We select Arritmic3D and try the default combination of parameters first.

Some of the relevant parameters mentioned before are:

-Id_extraI: The node to stimulate for the re-entrance generation.
-stimFrecsS1: It is the first stimulus; it must behave normally, and it is in ms.
-stimFrecsS2: This is a faster stimulus that would generate a re-entrance, it is also in ms.
-nStimsS1: It is the number of stimuli with S1 frequency before applying the next stimuli.
-nStimsS2: Number of stimuli with S2 frequency before the end of the simulation. Any beat generated after the last S2 it is a re-entrance.

Once we have our simulation, we can use different ways to visualize the beats and analyze the behavior.

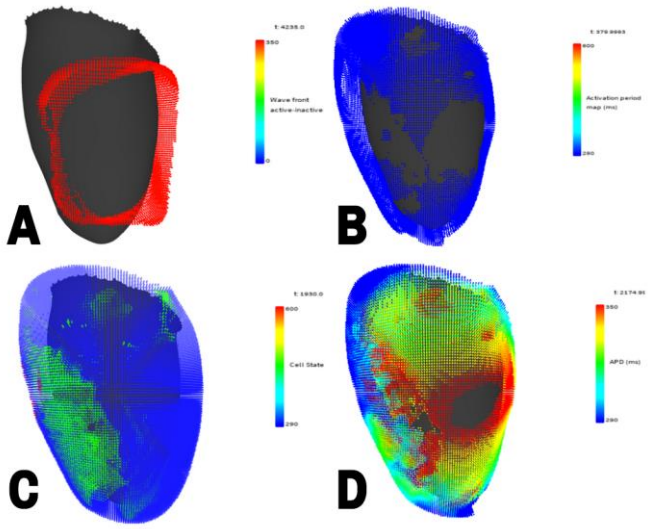

Fig 6. This is the simulation with the default parameters: id_extraI: 100000; stimFrecS1: 600; stimFrecS1: 285; nStimsS1: 6; nStimsS1: 2. Image “A” shows the wave-front activation, image “B” the activation period in ms, image “C” the cell state, where we can see the scar, and image “D” the action potential duration.

We can't repeat this process for every combination of parameters, that's why the *params* file have an option to make a multi-simulation and release an output file which tells all the combinations that contains a re-entrance and in which AHA segment.

From processing and with the desired combination of parameters we can generate a video or return to Paraview and make an analysis with this open-source software.

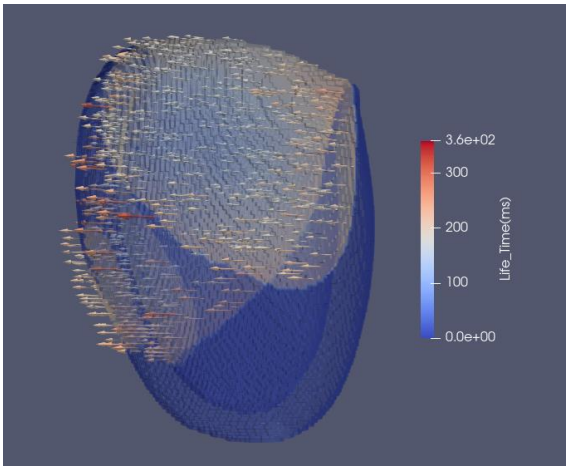

Fig 7. Simulation from Arritmic3D run in Paraview with a Glyph filter.

B. ECG study with ECGSIM

In ECGSIM, we can modify the action potential and activation time in the desired heart areas and observe how these changes affect the ECG recordings on the different derivations. The app contains two 3D heart models by default (normal male and normal young male), out of which we select the first, normal male. ECGsimcase. As we can see in Figure 8, the tool's interface contains a 3D display of the selected model on the top-left window with a surface map that displays the time of depolarization in ms. We can select a region (location and range) to change the depolarization time, the amplitude of the signal and the repolarization time, which can be visualized in the bottom-left window. The ECG signal for each of the 12 derivations is shown on the bottom-right window, with the original signal being marked in white and the signal after altering the electrical activity marked in red. Finally, the top-right window shows the heart model located in its corresponding space in the thoracic cavity, having also the option to show the location of the derivations and the 3D polar moment vector.

We want to simulate a case of Right Bundle Branch Block, which as we have explained consists of the delay or blockage in the conduction of electrical signal along the right bundle branch that originates from the bundle of His and delivers the signal to the right ventricle. Therefore, in order to simulate this disorder, we have delayed the depolarization on the left ventricle 35ms approximately (see the Region selected in the 3D display window in Fig. 8).

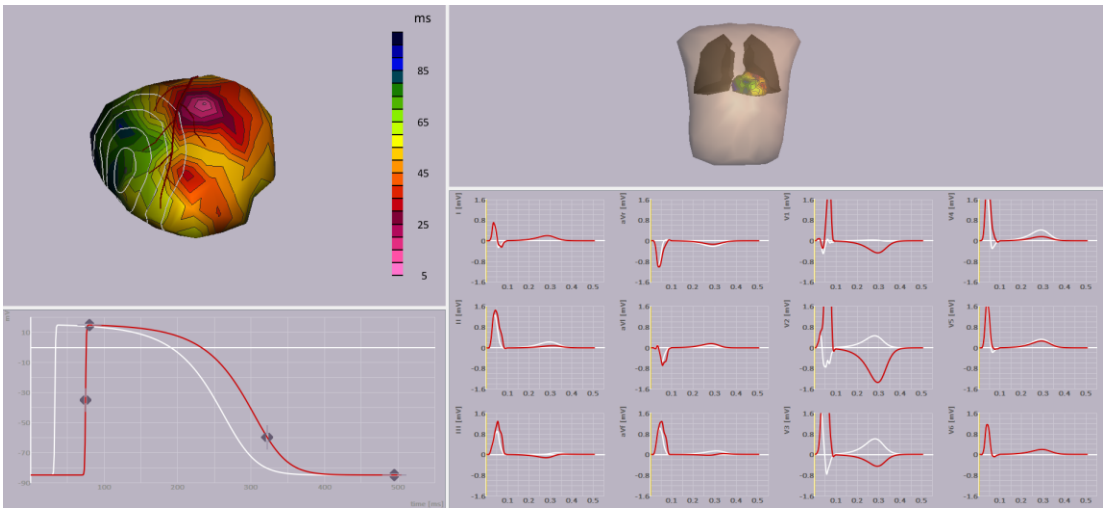

Fig 8. ECGSim interface and resulting ECGs after simulating a Right Bundle Branch Block (retarded depolarization in the right ventricle).

C. RVOT classification with Machine Learning

In order to identify the origin of idiopathic right ventricular tachycardias (RVOT) we will use a Machine Learning model for classification in Python software.

We are provided with four different models: Multilayer Perceptron, Random Forest, Support Vector Machines and XGBoostClassifier. From these models, we must select the one that gives us the best results for classification.

We use two different datasets to train the models. First, we decide to train the models only with the simulated data. In this dataset, the samples are very similar between them since the simulations performed to obtain these data were practically identical. Secondly, we also try to use real data to train them. These are real data corresponding to patients from two different hospitals, one from China and one from Barcelona. To do so, we combine these real data and the previous data obtained from the simulations to generate our whole training set. The resulting dataset is a balanced dataset, with a very similar number of RV than LV, which are the target labels of the prediction. Once that we have the whole dataset generated, we train all the models to see which one gives us the highest accuracy.

After training the models, we want to select the model that gives us the highest accuracy for test the data with it. In this case, we don't want to test the model with data that don't correspond to any patient, so we delete the simulated data from the dataset, and we only test the model with real data.

Once we have tested the model, we want to increase its performance. In order to do that, we try to choose which are the best hyperparameters that could be used with the model. The parameters we will be trying to fit will be the n_estimators, and the max_depth, trying with a lot of combinations.

III. RESULTS AND DISCUSSION

A. Arrythmia detection

1) Image Segmentation

With the open-source software 3DSlicer and the features mentioned in the section of methodology we performed the segmentations of the left ventricle, the border and core zone of the scar and the whole scar.

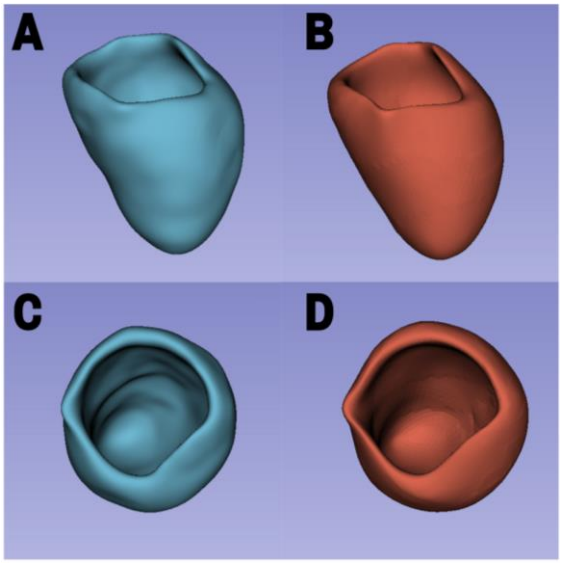

Fig 9. Images from the 3DSlicer platform. "A" and "C" correspond to the segmentation of the LV performed by the authors and "B" and "D" correspond to the ADAS segmentation.

We can observe that there is a big similarity between both geometries. Although both segmentations have been processed with other 3D design software like Meshmixer, these pictures of our geometry are before applying the Paraview filters, hence the small texture differences.

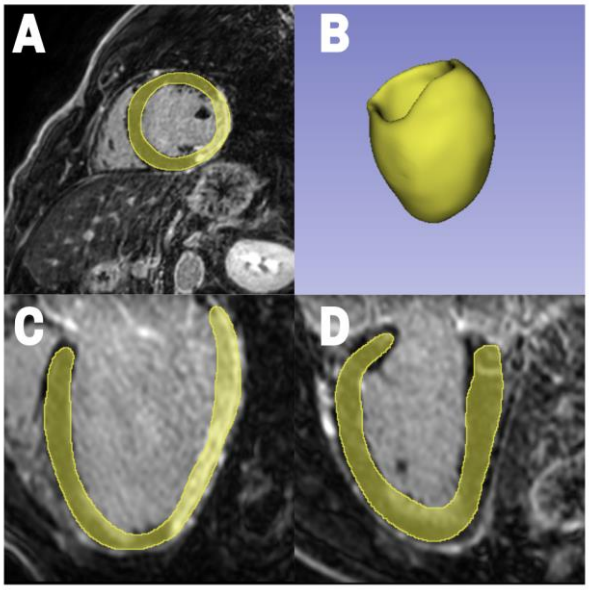

Fig 10. Images from the 3DSlicer platform. All of them correspond to our segmentation of the left ventricle from an MRI. A: Axial plane; B: 3D geometry; C: coronal plane; D: sagittal plane.

We can observe in figure 10 how the 3D volume is placed over the 3 planes. The smoothness of the drawing is due to the modules available.

On the other hand, we also segmented the whole structure of the scar, which is very important in the pipeline of the project. However, we did not process this geometry in any other software but 3DSlicer because we didn't consider it strictly necessary. Again, we can see differences in the texture between both structures due to the filters used.

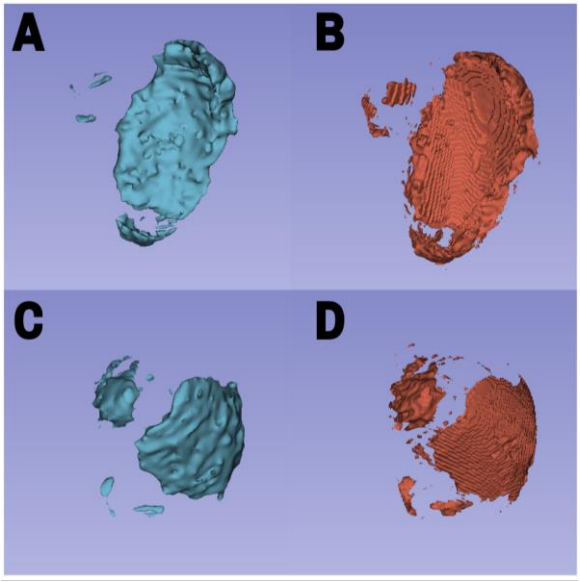

Fig 11. Images from the 3DSlicer platform. "A" and "C" correspond to the segmentation of the scar performed by the authors and "B" and "D" correspond to the ADAS segmentation.

As an extra exercise we segmented the core and the border zone. These two sections refer to different regions of a scar. The core zone is usually the central region, and the border zone is located in the outer region. Compared to the left ventricle, the scar is more difficult to detect and segment.

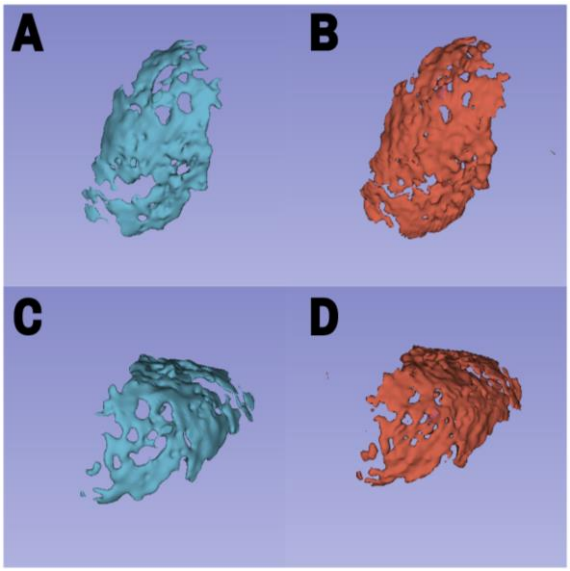

Fig 12. Images from the 3DSlicer platform. "A" and "C" correspond to the segmentation of the core zone of the scar performed by the authors and "B" and "D" correspond to the ADAS segmentation.

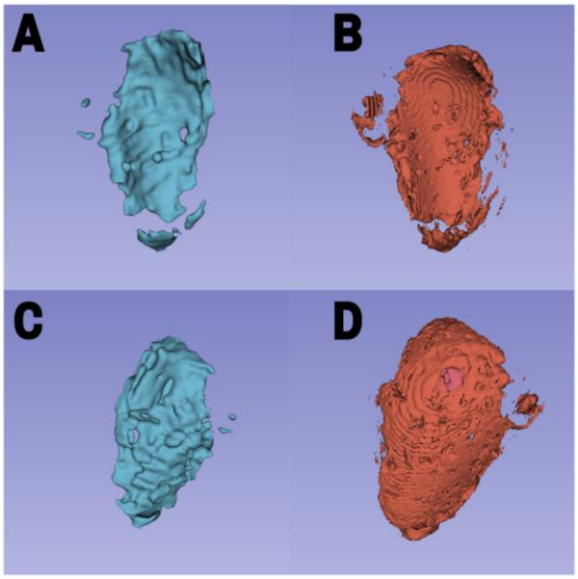

Fig 13. Images from the 3DSlicer platform. "A" and "C" correspond to the segmentation of the border zone of the scar performed by the authors and "B" and "D" correspond to the ADAS segmentation.

In general terms, the segmentations of the case that has been provided and the one performed by the authors have a lot of similarities, specially in size, orientation and shape.

Once we had our cardiac geometry, we used Paraview as our post-processing software as we said in previous sections. We used the filters mentioned above and compared the result to the ideal case.

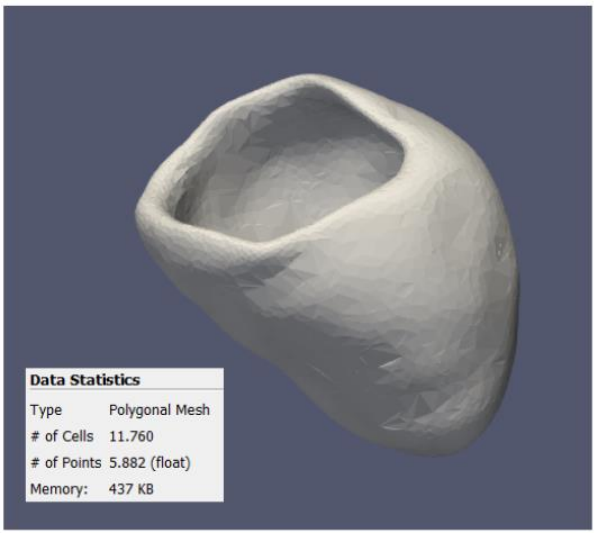

Fig 14. Images from Paraview. Left ventricle segmentation after applying all the required filters. In the left bottom we have the data statistics provided by the open-source software.

The main goal of this filtering was to simplify as much as possible the geometry without losing any relevant detail. That's why we are asked to have less than 20,000 cells and 10,000 points. We can observe in figure 14 that we have 11,760 cells and 5,882 points.

2) Simulation

As we explained, now we work with the ADAS segmentation. We run the code of python and get the output with the information for the Arritmic3D. With processing we run a multi-simulation that lasted more than 24 hours. The output of this multi-simulation was a file with all the parameters and whether there was a re-entrance or not.

The simulation run for S1: 600; S2: 270, 275, 280, 285, 290, 295; Number of stimulus S1: 6; Number of stimulus S2: 1, 2, 3 and Nodes to stimulate for re-entrance: 188430, 191081, 187265, 189328, 186778, 185092, 96899, 88444, 84986, 89634, 90551, 96342, 27260, 19016, 30010, 32481, 3938, 188463, 191104, 187281, 189166, 186750, 185078, 94605, 88466, 85001, 89440, 90536, 94301, 27280, 19002, 31412, 32464, 0; APDR: 0.75, 1, 1.25; CVR: 1, 1.25.

The total number of simulations was 3672. From this number, 22 of those simulations presented a re-entrance and 14 of them were sustained re-entrances. In figure 15 we have a table with all these sustained re-entrances simulations.

Nº	Sg. AHA	S1	Nº S1	S2	Nº S2	APDR	CVR	id
736	7	600	6	285	3	1.0	1.0	96899
737	7	600	6	290	3	1.0	1.0	96899
738	7	600	6	295	3	1.0	1.0	96899
810	11	600	6	295	3	1.0	1.0	90551
898	16	600	6	285	3	1.0	1.0	32481
900	16	600	6	295	3	1.0	1.0	32481
1131	12	600	6	280	3	1.0	1.0	94301
1134	12	600	6	295	3	1.0	1.0	94301
1166	14	600	6	275	3	1.0	1.0	19002
2573	7	600	6	290	3	1.0	1.25	96899
2966	12	600	6	275	3	1.0	1.25	94301
2967	12	600	6	280	3	1.0	1.25	94301
2968	12	600	6	285	3	1.0	1.25	94301
2969	12	600	6	290	3	1.0	1.25	94301

Fig 15. Table with the simulations that presented a sustained re-entrance.

3) Data Visualization

We chose one of those simulations that contained a re-entrance. In our case, we selected a sustained re-entrance, which means that it persists during the time. The one selected is number 737 and its parameters are: stimFrecS1: 600.0; stimFrecS2: 290.0; nStimsS1: 6; nStimsS2: 3; APDR factor: 1.0; CVR factor: 1.0 id_extraI: 96899; Segment AHA: 7.0; Endo/Epi: Endo.

If we attend to the parameters, we can compute how much it must last:

$$t = stimFrecS1 \cdot NºS1 + stimFrecS2 \cdot NºS2 \qquad (1)$$

In this case we have that it will finish after 4455 ms, any other beats after this time it will be considered as a reentrance, and we can detect the region of re-entrance with the segments AHA. Nevertheless, this re-entrance can appear before the last S2 pulse. It could happen with the first or the second if there is three.

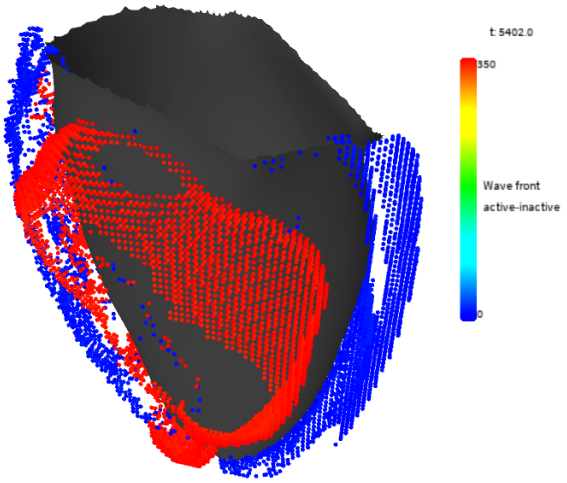

Fig 16. Images from Processing. Left ventricle segmentation with re-entrance simulation. stimFrecS1: 600.0; stimFrecS2: 290.0; nStimsS1: 6; nStimsS2: 3; APDR factor: 1.0; CVR factor: 1.0 id_extraI: 96899; Segment AHA: 7.0; Endo/Epi: Endo.

We can observe in figure 16 that after t=5402 we still have beats, so we can confirm the existence of the re-entrance.

Another way to prove the existence of this re-entrances are the superposition of the waves of activation and deactivation. In other words, the superposition of ventricular systole and diastole. If we can observe a relaxation and a contraction at the same time it means that we are entering in a possible arrhythmia.

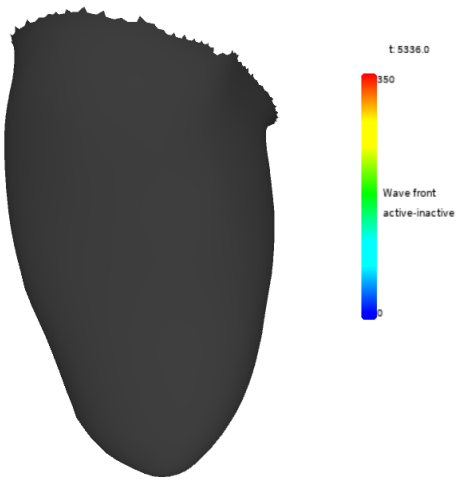

Fig 17. Images from Processing. Left ventricle segmentation with a non-re-entrance simulation. stimFrecS1: 600.0; stimFrecS2: 290.0; nStimsS1: 6; nStimsS2: 3; APDR factor: 1.0; CVR factor: 1.0 id_extraI: 191081; Segment AHA: 2.0; Endo/Epi: Endo.

In figure 18 we observe the same two cases we analyzed previously, but this time in the same step: 3921. Due to the parameters, we know that the simulations must last the same time, thus, they should behave in the same way. Nonetheless, we see that in this exact frame the superposition of waves star in the arrhythmic case, while in the control case doesn't happen.

Afterwards we imported the data of the pathological case to Paraview in order to observe where this problem could come from.

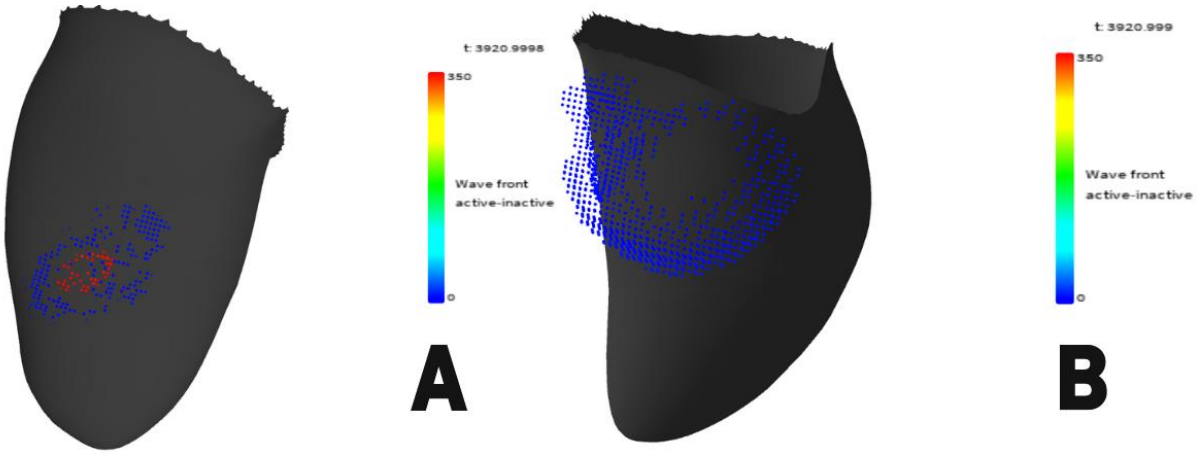

Fig 18. A: Step 3921 of pro-re-entrance case. B: Step 3921 of a non-re-entrance case.

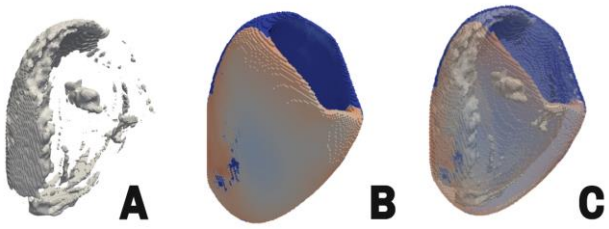

Fig 19. Images from the Paraview software. Step 4090 of the pro-arrhythmic simulation. A: Scar. B: Left ventricle. C: Superposition of scar and LV.

In figure 19 we observe the scar and the left ventricle in the simulation that contained a re-entrance. The wave propagation has some areas that doesn't activate properly. If we superpose this wave propagation with the scar, we can see that these regions belong to the damaged tissue. Our hypothesis is that this damaged tissue causes problems for the electrical propagation, and this fact could cause an asynchronization of the beats, which lead the cardiac geometry to an arrhythmia.

Besides this data, the clinicians could also use this information to get the segment AHA and localize it in the ventricle like we did in the methodology section.

B. ECG study with ECGSIM

After the steps mentioned, we obtain the following ECG graphs (see Fig 20) containing all derivations: the six limb leads (I, II, III, aVR, aVL, aVF) and the six precordial leads (V1,V2,V3,V4,V5,V6).

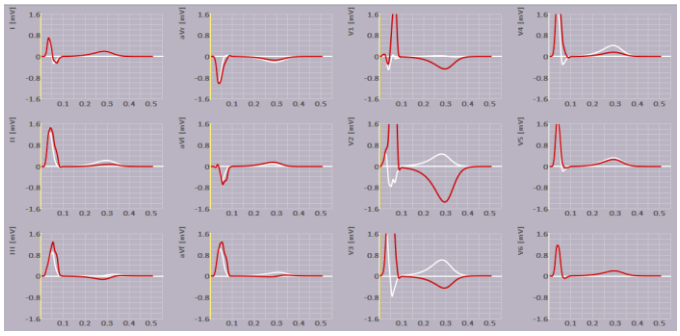

Fig 20. ECG signals obtained in ECGSIM after simulating a case of RBBB.

To visualize the effect this disorder has on the ECG, we focus on the bottom-right window. We can see a major change in the derivations V1-V3, where the T wave has been reduced until reaching negative values. Also, in the lead V1, there is a subtle increase in the voltage before the S segment (this is known as an RSR' pattern). If we observe the limb leads (leads I-III), we can also see some changes, such a small increase in the duration of the S wave in lead I.

The ECG criteria to diagnose RBBB [4] include the inversion of the T wave in the right precordial leads (V1-V3), an RSR' pattern in V1 and V2, as well as an S wave longer than the R wave in I and V6, changes that we have been able to distinguish from the ECG obtained with ECGSIM. Other changes such as a QRS duration higher than 120ms, an R wave peak time greater than 50 milliseconds in V1 and upright T waves in the left precordial leads (V4-V6).

C. RVOT classification with Machine Learning

For the case in which we only trained the models with the simulated data, the accuracy score obtained for all the models is very high since all the data samples are very similar between them.

Nevertheless, when using a more realistic database (the one that combines simulated data with real data), the model that gives us the best accuracy score is XGBClassifier. Because of that, this is the model that we choose for performing the final classification.

We test the selected model XGBClassifier with real data and it gets an accuracy of 82%. The results are shown in figure 20.

0	0.89	0.86	0.88	57
1	0.62	0.68	0.65	19
accuracy			0.82	76
macro avg	0.75	0.77	0.76	76
weighted avg	0.82	0.82	0.82	76

Fig 20. Image from python. Accuracy score, macro average and weighted average of the XGBClassifier performance on the test set.

Finally, we choose the best hyperparameters to increase the model performance. The parameters selected are *max_depth* and *n*_estimators. After trying with a lot of combinations, the results obtained for the parameters that we want to fit in the model can be seen in figure 21.

0	0.91	0.84	0.87	57
1	0.61	0.74	0.67	19
accuracy			0.82	76
macro avg	0.76	0.79	0.77	76
weighted avg	0.83	0.82	0.82	76

Best parameters: {'max_depth': 150, 'n_estimators': 400}

Fig 21. Image from python. Accuracy score, macro average and weighted average of the XGBClassifier performance on the test set. Below, values obtained for the hyperparameters max_depth and n_estimators.

CONCLUSIONS

In this paper we presented a multidisciplinary project in which we wanted to study ventricular arrhythmia and develop a treatment approach using medical imaging, computational simulations, and machine learning techniques. We focused on the segmentation and post-processing of cardiac medical images, simulation of arrhythmia using a computational model, analysis of electrocardiogram (ECG) signals, and classification of idiopathic right ventricular tachycardias (RVOT) using machine learning.

The first part of the project involved the segmentation and post-processing of cardiac MRI images using the open-source software 3D Slicer. In this platform we segmented the left ventricle geometry and scar regions. From these, we generated a 3D model of the heart structure for further analysis.

Next, we performed a simulation using the processed geometry that uses a cellular automaton model to analyze the electrical activity and identify re-entries in the ventricles. Then, we visualized the results using Paraview and Processing software, and we could identify the presence of the re-entrance, meaning that our patient is pro-arrhythmic.

Furthermore, using ECGSIM, we studied the effects of changes in electrical activity on ECG recordings by modifying the action potential and activation time in specific heart areas and observing the resulting alterations in ECG signals. A case of Right Bundle Branch Block was simulated by delaying depolarization in the left ventricle, providing a better understanding of the disorder's impact on ECG waveforms.

Finally, we performed a classification of the origin of RVOT using machine learning models. These models were evaluated based on their classification performance, and the most effective for RVOT classification was the XGBClassifier.

Overall, in this project we want to show the relevance of medical imaging, simulation, and machine learning techniques in the analysis and treatment of ventricular arrhythmias. The proposed methodology has the potential to assist clinicians in accurately localizing abnormal regions in the heart and guiding in radiofrequency ablation procedures. It must be enhanced the performance of the segmentation, which must be as similar as possible to the real heart structure. Moreover, further research and clinical validation are necessary to fully make use of the potential of these tools in real-world scenarios.

References

[1] John RM, Tedrow UB, Koplan BA, Albert CM, Epstein LM, Sweeney MO, et al. Ventricular arrhythmias and sudden cardiac death. Lancet [Internet]. 2012;380(9852):1520–9. Available from: https://pubmed.ncbi.nlm.nih.gov/23101719/

[2] F. Morady. Radio-Frequency ablation as a treatment for cardiac arrhythmias. N Engl J Med 1999; 340:534-544. Available from: https://doi.org/10.1056/NEJM199902183400707

[3] ECGSIM Development Group. (2004). ECGSIM (3.0.1). https://www.ecgsim.org/index.php

[4] Harkness WT, Hicks M. Right Bundle Branch Block. [Updated 2022 Aug 8]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2023 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK507872/

[5] Brigham and Women's Hospital.(2000). 3DSlicer (5.0.3). https://www.slicer.org/

[6] Resonancia magnética. (2021, septiembre 4). Mayoclinic.org. https://www.mayoclinic.org/es-es/tests-procedures/mri/about/pac-20384768

[7] Autodesk123D.(2009). Meshmixer (3.5). https://meshmixer.com/

[8] Kitware, Inc.(2002). Paraview (5.10). https://www.paraview.org/

[9] Adas Medical 3D.(2013). Adas3D. https://www.adas3d.com/

[10] Guido van Rossum.(1991). Python (3.8). https://www.python.org/

[11] Ben Fry and Casey Reas.(2001). Processing (4.2). https://processing.org/

[12] 17-Segment Model (AHA) [Internet]. Pmod.com. [citado el 16 de mayo de 2023]. Disponible en: https://www.pmod.com/files/download/v34/doc/pcardp/3615.htm