Development of a User-Friendly Pipeline for Constructing Atrial Models at Scale: Importance of the End-User for Clinical Uptake

Laura Bevis[1], Semhar B Misghina[1], Elisa Rauseo[1], Carlos E Lopez Barrera[1], Gernot Plank[2], Edward Vigmond[3], Axel Loewe[4], Elias Karabelas[5], Jose Alonso Solis-Lemus[6], Steven Niederer[6], Steffen E Petersen[1], Greg Slabaugh[1], Anthony Mathur[1], Caroline H Roney[1]

[1] Queen Mary University of London, London, United Kingdom
[2] Medical University of Graz, Graz, Steiermark, Austria
[3] LIRYC - University of Bordeaux, Talence, France
[4] Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[5] University of Graz, Graz, Steiermark, Austria
[6] Imperial College London, London, United Kingdom

Abstract

Atrial fibrillation (AF) is a common cardiac arrhythmia with high recurrence rate post-treatment. Personalised computational models can be used to investigate patient-specific mechanisms underlying AF, explore personalised treatment approaches, and for large-scale in silico trials. To ensure accuracy, reproducibility, and realistic uptake in clinical practice, investigations require consistent construction of models from patient-specific data. Clear and user-friendly model pipelines are therefore critical and should be developed with the end-user in mind.

The atrial modelling toolkit (atrialmtk: `https://github.com/pcmlab/atrialmtk`) allows users to produce atrial meshes for electrophysiological simulation, incorporating atrial regions, fibres, and transmural variations across the atrial wall. Multiple workflows allow the use of several input data types that may be available to the user, while mesh type and fibre distribution options allow users to investigate the effect of various factors on fibrillatory dynamics. The pipeline has been tested by multiple users across operating systems to ensure compatibility and usability, and its success at scale demonstrated in a previous study of 1000 atrial geometries. By expanding testing to a wider audience, we aim to increase clinical engagement and the uptake of modelling and digital twins in clinical practice.

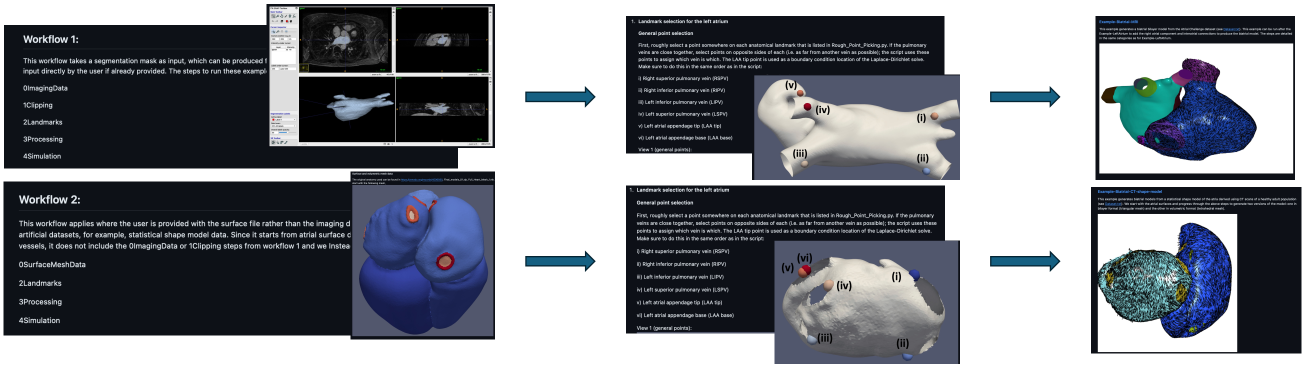
1. Introduction

AF is the most common cardiac arrhythmia in adults worldwide and is associated with increased risk of stroke, heart failure and mortality. Current treatment approaches for AF, such as catheter ablation and anti-arrhythmic drug therapy, have high recurrence rates, potentially due to treatments not being personalised to the individual. Personalised computational models can be used to investigate AF mechanisms and test different treatment approaches *in silico*, both at the individual level and in large-scale *in silico* clinical trials. To ensure the accurate and reproducible construction of models from patient-specific data, which can vary widely between individuals, as well as their fast and consistent construction at scale and realistic uptake in clinical practice, clear and user-friendly model pipelines are critical and must be developed to align with standard clinical workflows.

We have therefore developed the atrial modelling toolkit (atrialmtk: `https://github.com/pcmlab/atrialmtk`) for constructing atrial models [1]. With a few manual inputs from the user, this pipeline generates simulation-grade bilayer or volumetric atrial meshes incorporating atrial regions, fibres and transmural variations across the wall, for electrophysiological simulation. Multiple workflows allow the use of several data types, including MRI images, electroanatomical mapping data, and artificial geometries such as those produced by statistical shape models, maximising usability and allowing for the construction of larger model cohorts. By varying options within the pipeline, the user can investigate the effects of factors such as fibrosis, fibre distribution, model calibration, and bilayer vs volumetric meshes on fibrillatory dynamics and AF outcomes. Clear instructions, computational requirements, required user expertise and training materials are detailed at each stage of the pipeline to ensure ease of use and correct implementation.

In this paper we outline the steps of the pipeline and its

ISSN: 2325-887X DOI: 10.22489/CinC.2024.014



Workflow 1:

This workflow takes a segmentation mask as input, which can be produced f
input directly by the user if already provided. The steps to run these exampl

0ImagingData

1Clipping

2Landmarks

3Processing

4Simulation

Workflow 2:

This workflow applies where the user is provided with the surface file rather than the imaging d
artificial datasets, for example, statistical shape model data. Since it starts from atrial surface d
vessels, it does not include the 0ImagingData or 1Clipping steps from workflow 1 and we Instea

0SurfaceMeshData

2Landmarks

3Processing

4Simulation

1. **Landmark selection for the left atrium**

   General point selection

   First, roughly select a point somewhere on each anatomical landmark that is listed in Rough_Point_Picking.py. If the pulmonary veins are close together, select points on opposite sides of each (i.e. as far from another vein as possible); the script uses these points to assign which vein is which. The LAA tip point is used as a boundary condition location of the Laplace-Dirichlet solve. Make sure to do this in the same order as in the script:

   i) Right superior pulmonary vein (RSPV)

   ii) Right inferior pulmonary vein (RIPV)

   iii) Left inferior pulmonary vein (LIPV)

   iv) Left superior pulmonary vein (LSPV)

   v) Left atrial appendage tip (LAA tip)

   vi) Left atrial appendage base (LAA base)

   View 1 (general points):

1. **Landmark selection for the left atrium**

   General point selection

   First, roughly select a point somewhere on each anatomical landmark that is listed in Rough_Point_Picking.py. If the pulmonary veins are close together, select points on opposite sides of each (i.e. as far from another vein as possible); the script uses these points to assign which vein is which. The LAA tip point is used as a boundary condition location of the Laplace-Dirichlet solve. Make sure to do this in the same order as in the script:

   i) Right superior pulmonary vein (RSPV)

   ii) Right inferior pulmonary vein (RIPV)

   iii) Left inferior pulmonary vein (LIPV)

   iv) Left superior pulmonary vein (LSPV)

   v) Left atrial appendage tip (LAA tip)

   vi) Left atrial appendage base (LAA base)

   View 1 (general points):

Example-Biatrial-MRI

This example generates a biatrial bilayer model from the Atrial Challenge dataset (see Dataset.txt). This example can be run after the Example-LeftAtrium to add the right atrial component and interatrial connections to produce the biatrial model. The steps are detailed in the same categories as for Example-LeftAtrium.

Example-Biatrial-CT-shape-model

This example generates biatrial models from a statistical shape model of the atria derived using CT scans of a healthy adult population (see Dataset.txt). We start with the atrial surfaces and progress through the above steps to generate two versions of the model: one in bilayer format (triangular mesh) and the other in volumetric format (tetrahedral mesh).

Figure 1. atrialmtk workflows showing landmarking instructions, final meshes, and some of the included fibers and regions.

provided examples, as well as its inclusion as a cardiovascular use case for the EDITH-CSA project [2] for the integration and clinical uptake of digital twins in healthcare.

2. Methods

atrialmtk is an open-source user-friendly pipeline for atrial model generation. Source code, installation steps, and computational and user requirements are provided, along with detailed instructions and expected inputs and outputs for several examples.

2.1. Installation & Software Requirements

The pipeline requires the installation of the open-source cardiac electrophysiology simulator, openCARP, as well as meshtool and Python [3,4]. Version requirements and necessary conda environments for the pipeline are provided.

2.2. Workflows

The provided examples follow one of two workflows (Figure 1). Workflow 1 uses a segmentation mask of the atrial blood pool from CT or MRI imaging data as input, to define the endocardial surface and generate a bilayer atrial mesh for simulation. Workflow 2 allows the use of existing atrial meshes, which have both surface and volumetric data but from which the pulmonary veins (PV), left atrial appendage (LAA) and vena cava (VC) have already been clipped, as is often the case when using artificial datasets. This workflow therefore allows for the generation of bilayer or volumetric atrial meshes, and does not include ‘0ImagingData’ or ‘1Clipping’ from workflow 1.

2.3. Workflow 1

Workflow 1 includes two examples; one for the left atrium (LA), and one for the addition and connection of the right atrium (RA) to produce a biatrial model.

Example-LeftAtrium

0ImagingData The user begins with the late gadolinium enhanced MRI (LGE-MRI) image lgemri.nrrd and provided segmentation mask laendo.nrrd of the LA blood pool from the openly-available 2018 Atrial Segmentation Challenge dataset (`https://www.cardiacatlas.org/atriaseg2018-challenge/atria-seg-data/`, Training/06SR5RBREL16DQ6M8LWS) [5]. The segmentation mask is opened and exported as a surface mesh in vtk format using capable software (e.g. 3DSlicer - `https://www.slicer.org`, CemrgApp - `http://cemrgapp.com`, itksnap - `www.itksnap.org`, PyVista - `https://pyvista.org/index.html`).

1Clipping The surface is clipped to open the mesh at the PV and mitral valve using the spherical clipping tool in ParaView (`https://www.paraview.org`), and the clipped surface triangulated and exported in vtk format.

2Landmarks Several key anatomical landmarks are then selected by the user so that the model can correctly separate the surface into different atrial regions in the next step. This step uses two custom Python scripts and interactive PyVista windows, which requires the creation and activation of the ‘pointpicking’ conda environment. First the user selects points roughly on each of the four PVs and at the tip and base of the LAA, in the order listed in the script ‘Rough_Point_Picking.py’, then four more specific points as listed in ‘Refined_Point_Picking.py’. Detailed written and visual instructions are provided to ensure correct placement (Figure 1), and the coordinates saved in the files ‘Regions.txt’ and ‘Landmarks.txt’, respectively.

3Processing Using these coordinates, the atrial regions are identified automatically using a series of Laplace-Dirichlet solves in openCARP [3], fibres mapped to the anatomy, and a bilayer mesh generated. To run this step, the user first creates and activates the ‘uac’ conda environment from the installation instructions, and sets the values ‘PVLabelT’ and ‘LAALabelT’ used to threshold

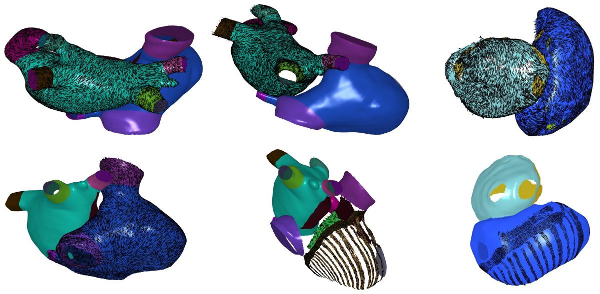


Figure 2. Final biatrial models, showing regions and some included fibres and endocardial structures (SAN, CT, PM).

the Laplace-Dirchlet solves, with values above and below these thresholds assigned as PV/LAA and LA tissue, respectively. The regions are then assigned specific labels depending on their closest landmark. Atrial structures including the crista terminalis (CT), pectinate muscles (PM), sinoatrial node (SAN) and Bachmann's bundle (BB), taken from an atlas bilayer mesh [6], as well as a fibre distribution, chosen by the user from a catalogue of rule-based (Labarthe) and *ex vivo* diffusion tensor MRI (DTMRI) derived fibre atlases (1-7, or average) [7], are mapped to the anatomy using the previously described Universal Atrial Coordinate (UAC) system (Figure 2) [8]. An initial condition, used to start spiral wave re-entries in the simulation step, is also generated for the mesh.

4Simulation Finally, simulations can be run over the resultant mesh using the provided par file and initial condition in openCARP [3], and the results used to investigate wavefront patterns and fibrillatory dynamics in the LA.

Example-Biatrial-MRI

0ImagingData The user starts with the same raw LGE-MRI images, segmenation mask, and the created files and directories from 'Example-LeftAtrium'. Since the RA blood pool segmentation mask raendo.nrrd is not provided with the 2018 dataset, we have segmented it ourselves and provided it to the user.

1Clipping The RA is clipped at the VC, coronary sinus (CS) and tricuspid valve, and exported using the same methods as in 'Example-LeftAtrium'.

2Landmarks The pointpicking scripts and PyVista windows are again used to select the RA landmarks as described in the scripts and provided instructions for the RA.

3Processing The RA processing script separates the RA surface into anatomical components and regions, maps selected fibre information to the RA anatomy, and generates a simulation-ready bilayer mesh and corresponding initial condition for the RA. The thresholds 'PVLabelT' and 'LAALabelT' are again set by the user, but now determine the locations of the junction between the VC/CS and RA tissue, and the right atrial appendage (RAA) and RA tissue. Finally, the biatrial processing script is run to combine the LA and RA meshes as a biatrial model (Figure 2). Interatrial connections are mapped to the meshes at BB, the CS, and at the septal wall, using linear connections between a line of nodes on the LA and RA components of BB, joining nodes of the half of the CS boundary closest to the LA epicardium to nodes on the LA epicardium using linear elements, and joining LA epicardial nodes within a distance threshold of the selected location on the septal wall to their closest RA epicardial nodes using line connections, respectively [8].

4Simulation Electrophysiological simulations can be run over the resultant biatrial mesh to investigate the effect of the RA and factors such as fibres and fibrosis on fibrillatory dynamics and AF outcomes.

2.4. Workflow 2

Example-Biatrial-CT-shape-model

0SurfaceMeshData Here the user starts with a full heart mesh from a CT-derived statistical shape model from a healthy adult population (`https://zenodo.org/records/4506930`, 'Final_models_01.zip/Full_Heart_Mesh_1.vtk') [9]. Meshes are separated into individual surfaces and volumes using the relevant tags and connected component analysis, and assigned as either endocardial or epicardial using meshtool and provided to the user [4].

2Landmarks The user again selects several points using the pointpicking scripts and PyVista windows. To account for the absence of atrial vessels, alternative landmarking instructions are provided for this workflow (Figure 1). The coordinates are again saved for use in the next step.

3Processing Similarly to workflow 1, the anatomy is automatically separated into atrial regions, atrial structures and a chosen fibre distribution mapped to the anatomy using an extended version of the UAC system and the selected coordinates, and an initial condition generated for the mesh (Figure 2) [8]. Scripts are provided to generate bilayer and/or volumetric meshes for this workflow.

4Simulation Finally, simulations can be run over the resultant meshes to investigate the effect of the RA, mesh type (bilayer or volumetric), and factors such as fibres and fibrosis on fibrillatory dynamics.

3. Results & Conclusions

Ease of Use & Compatibility The pipeline is presented in stages, with detailed user instructions, training materials, and example input and output files for comparison. Compatibility, implementation time and user-friendliness have been tested across computer operating systems and

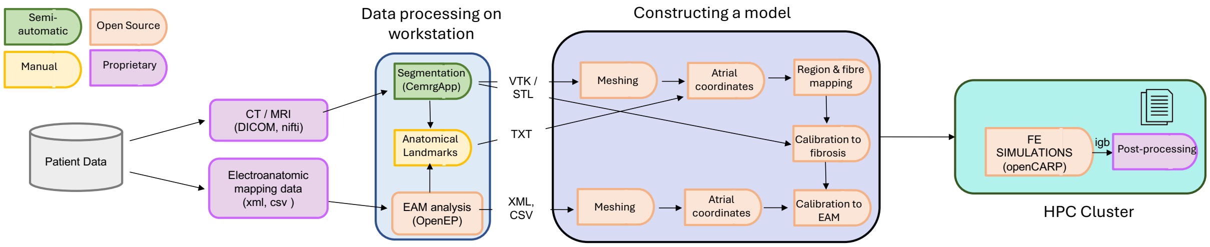


Figure 3. Data flow diagram for atrialmtk, identifying required file formats, resources, software, and manual input.

by multiple users of differing backgrounds and levels of computational skill and understanding of electrophysiology, and improvements made to maximise compatibility, reproducibility and clarity of instruction [10]. Its success at scale has been demonstrated in a previous study of 1000 atrial geometries to study the effects of fibre field, mesh type, and fibrosis on simulated fibrillatory dynamics [8].

Integration & Clinical Uptake The step-by-step approach and expected outputs provided with the pipeline allow users to more easily identify sources of error and integrate the pipeline with their own computational infrastructures. Different workflows allow the use of different clinically available data types, which may vary across hospital trusts and countries. By working closely with clinical collaborators, we aim to build pipelines that consider standard clinical workflows, clinical needs, cost, and transparency of methods for informed decision-making.

Through its inclusion as a cardiovascular use case for the EDITH-CSA project, the pipeline was analysed to identify infrastructure requirements, necessary stakeholders, and barriers to be addressed for the concept of an integrated Virtual Human Twin (VHT). In addition to its successful integration with computational infrastructures within the project, the provision of detailed data flows (Figure 3) and close discussion with clinical collaborators has helped to identify crucial barriers for the clinical adoption of digital twins, such as ethical and legal issues around data use and sharing, economic aspects regarding the maintenance and clinical uptake, and the need for clear standards, validation methods and credibility assessments, which will be included in the final recommendations of the project [2].

Overall, we have developed a user-friendly pipeline for the reproducible construction of atrial models at scale. By extending testing to a wider range of end-users and taking a similar approach in other areas of our research, we hope to increase the likelihood of the uptake of models in clinical practice and increase the engagement of clinicians with other digital twin initiatives.

Acknowledgments

This research was funded by a UKRI Future Leaders Fellowship (MR/W004720/1) and used the ARCHER2 UK National Supercomputing Service (`https://www.archer2.ac.uk`). The authors acknowledge funding from Queen Mary University of London in contribution towards the Digital Europe Programme of the European Commission (grant agreement 101083771) for the EDITH project.

References

[1] Roney CH. Atrial modelling toolkit (atrialmtk), 2023. URL `https://github.com/pcmlab/atrialmtk`.
[2] EDITH consortium. EDITH CSA Deliverable 3.2: first draft of the VHT roadmap, 2023. URL `https://doi.org/10.5281/zenodo.8200955`.
[3] Plank* G, *et al.* The openCARP simulation environment for cardiac electrophysiology. Comput Methods Programs Biomed 2021;208:106223.
[4] Neic A, *et al.* Automating image-based mesh generation and manipulation tasks in cardiac modeling workflows using meshtool. SoftwareX 2020;11:100454.
[5] Xiong Z, *et al.* A global benchmark of algorithms for segmenting the left atrium from late gadolinium-enhanced cardiac magnetic resonance imaging. Med Image Anal 2021; 67:101832.
[6] Labarthe S, *et al.* A bilayer model of human atria: mathematical background, construction, and assessment. Europace 2014;16(suppl_4):iv21–iv29.
[7] Roney CH, *et al.* Constructing a human atrial fibre atlas. Ann Biomed Eng 2021;49:233–250.
[8] Roney CH, *et al.* Constructing bilayer and volumetric atrial models at scale. Interface focus Sep. 2023;13(6):20230038.
[9] Rodero C, *et al.* Virtual cohort of 1000 synthetic heart meshes from adult human healthy population, 2021. URL `https://doi.org/10.5281/zenodo.4506930`.
[10] Solís-Lemus JA, *et al.* Evaluation of an open-source pipeline to create patient-specific left atrial models: A reproducibility study. Comput Biol Med 2023;162:107009. ISSN 0010-4825.

Address for correspondence:

Laura Bevis
Digital Environment Research Institute (DERI), Empire House,
67-75 New Road, London, E1 1HH
l.bevis@qmul.ac.uk