

CEM



Dr José Alonso Solís Lemus

Open-Source Software for Surgical Technologies

26th, June 2023

CemrgApp: A Sustainable and Accessible Platform for Cardiovascular Research

Acknowledgment

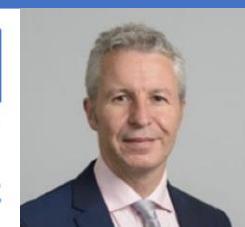
Imperial College London



Imperial College London



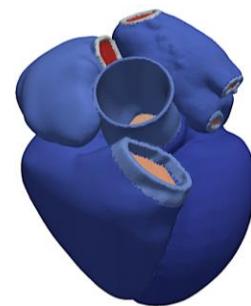
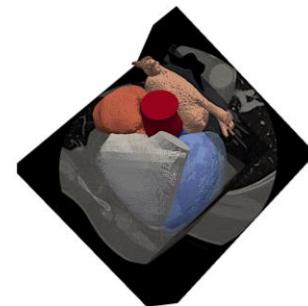
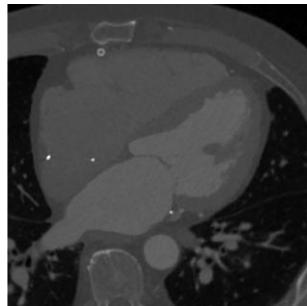
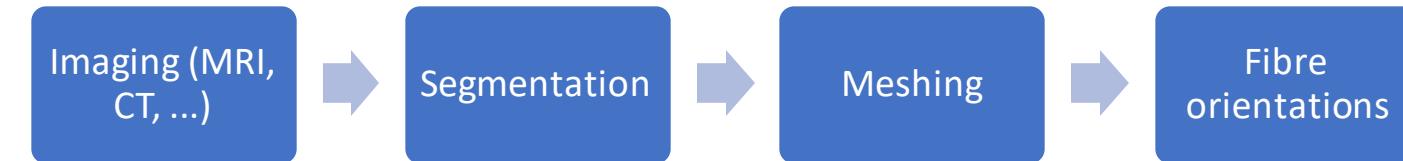
Guy's and St Thomas'
NHS Foundation Trust



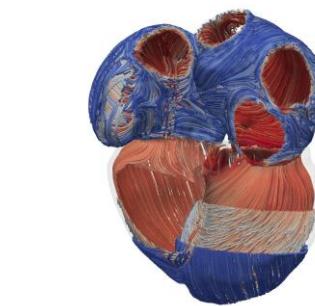


More or less what we do at CEMRG

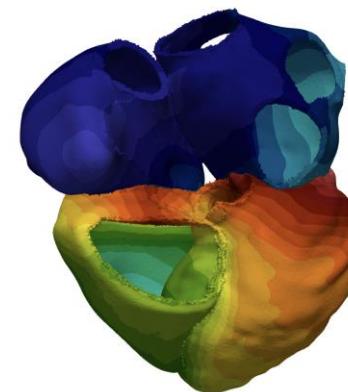
Patient-specific cardiac anatomical models



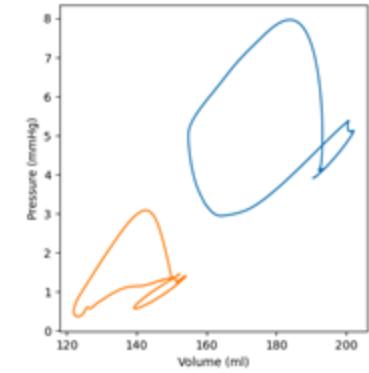
Fibre orientations



Excitation



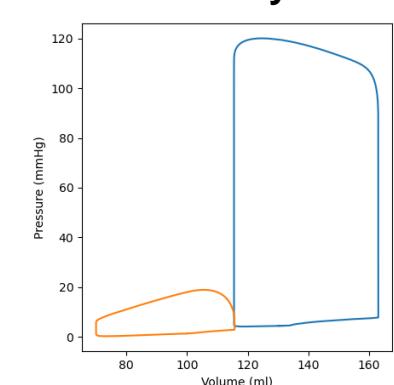
Atrial dynamics



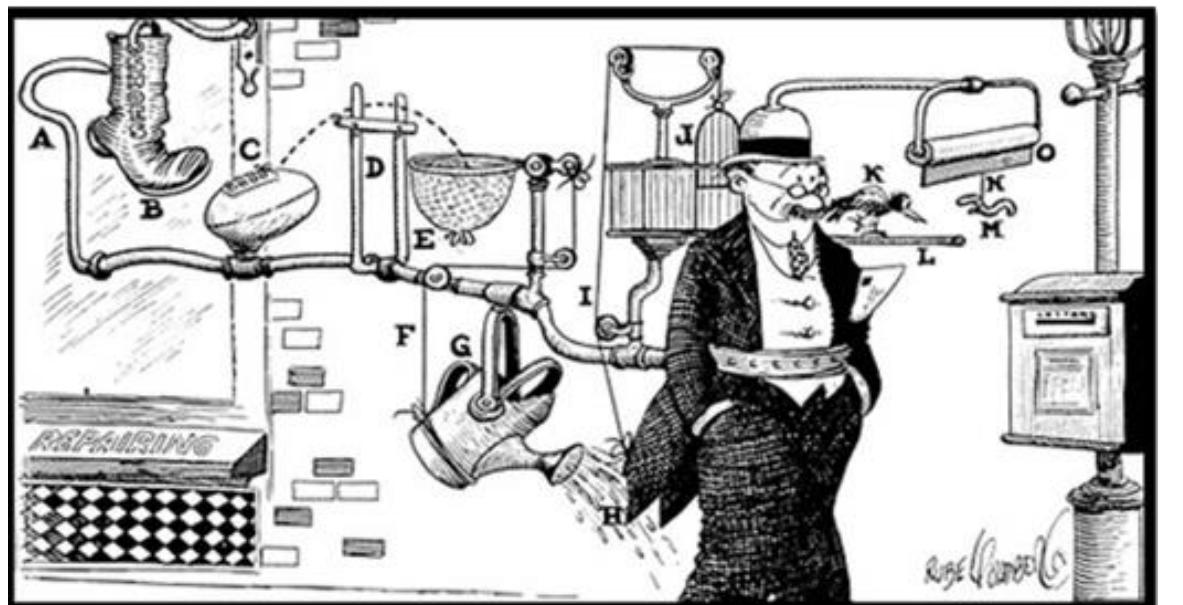
Whole heart motion



Ventricular dynamics



Why CemrgApp?



Paraview
Matlab
Python

CemrgApp

Two tasks:

1. Connected component analysis on automated segmentation shell (vts) (use to start on)
2. UAC codes from labelled mesh needs (use to repair and send)

Print out of landmarks (point selection): 1. RSPV; 2. LSPV; 3. LAA tip; 4. LAA base

Publication plan: Meir FPMB, 22nd Jan https://fbm2021.github.io/abstract_sub/

Conda environment:

```
conda env create -n UAC -f UAC.yaml
conda activate UAC
conda install -c anaconda vtk
conda install -c anaconda scipy
pip install meshio
pip install --upgrade numpy
```

INSTRUCTIONS FOR MODEL CONSTRUCTION

1. Cleaning - To remove artifacts (generated by the coronary sinus catheter) from the electroanatomic mapping (EAM) data

- Navigate to the folder that you are analysing
- Right click on the file select open in terminal - this should open a terminal window
- Type: paraview -Avtit and press enter (this should open the LA.vtk file in paraview). Click Apply - you should see the mesh. Select Filters: Extract Surface. Click Apply.
- Remove abnormal bumps in the mesh due to artifacts. For each bump you remove:
 - Filters: Clip: Either use plane or sphere to remove the abnormal areas (make sure you set to enable clip)
 - Filters: Extract Surface. Click Apply
- File: Save Data: Select legacy VTK File Format (*.vtk) extension (Name file Clipped.vtk). Select ascii (not binary) (Make sure when you save Data, Extract Surface and not Clip is selected)

2. Clipping - To clip the MV again. To clip the end of each PV so that there are 4 PV openings.

- In the same terminal, now type paraview Clipped.vtk and press enter (this should open the Clipped.vtk file in paraview)
- Close meshlab

3. Clipping - To clip the MV again. To clip the end of each PV so that there are 4 PV openings.

- In the same terminal, now type paraview Clipped.vtk and click enter (this should open the Clipped.vtk file in paraview)
- Close meshlab

4. Meshlab - File > Open... (Select the Clipped.vtk file)

- In paraview, right click on the meshlab icon and select the refined mesh. Select Filters: Extract Surface. Click Apply.
- Clip the RSPV: Extract Surface.
- Clip the LSPV: Extract Surface.
- Clip the LAA: Extract Surface.
- Clip the LAA tip: Extract Surface.
- Clip the LAA base: Extract Surface.
- File: Save Data: Select Legacy VTK Files (*.vtk) extension (Name file Clipped.vtk). Select ascii (not binary) (Make sure when you save Data, Extract Surface and not Clip is selected)

Select Clip 2. Untab Inside Out. Click Apply. Select Extract Surface.

- LSPV: Filters: Clip (Select Inside Out), Click Apply, Filters: Extract Surface. Click Apply.
- File: Save Data: Select Legacy VTK Files (*.vtk) extension (Name file LSPV.vtk). Select ascii (not binary) (Make sure when you press Data, Extract Surface and not Clip is selected)
- Select Clip 3. Untab Inside Out. Click Apply. Select Extract Surface.
- RSPV: Filters: Clip (Select Inside Out), Click Apply, Filters: Extract Surface. Click Apply.
- File: Save Data: Select Legacy VTK Files (*.vtk) extension (Name file RSPV.vtk). Select ascii (not binary) (Make sure when you press Data, Extract Surface and not Clip 3 is selected)
- Select Clip 4. Untab Inside Out. Click Apply. Select Extract Surface.
- LAA: Filters: Clip (Select Inside Out), Click Apply, Filters: Extract Surface. Click Apply.
- File: Save Data: Select Legacy VTK Files (*.vtk) extension (Name file LAA.vtk). Select ascii (not binary) (Make sure when you press Data, Extract Surface and not Clip 4 is selected)
- Select Clip 5. Untab Inside Out. Click Apply. Select Extract Surface.
- LAA tip: Filters: Clip (Select Inside Out), Click Apply, Filters: Extract Surface. Click Apply.
- File: Save Data: Select Legacy VTK Files (*.vtk) extension (Name file LAA_tip.vtk). Select ascii (not binary) (Make sure when you press Data, Extract Surface and not Clip 5 is selected)
- LAA base: Filters: Clip (Select Inside Out), Click Apply, Filters: Extract Surface. Click Apply.
- File: Save Data: Select Legacy VTK Files (*.vtk) extension (Name file LAA_base.vtk). Select ascii (not binary) (Make sure when you press Data, Extract Surface and not Clip 6 is selected)

complete steps 4-7: For one of the folders create a high resolution mesh, for the other a low resolution mesh. Steps 5-7 are the same for both folders.

4. UAC code - LabelledMesh_Simplification_Anesthesiology.py: Label the mesh.

- Go to UAC codes folder
- Open the LabelledMesh_Simplification_Anesthesiology.py code and change the path i.e. to the folder you are analysing (Remember to include the final /). Save. Close code.
- Right click UAC codes folder and click open terminal.
- Type: conda activate UAC and click open terminal.
- Type: python LabelledMesh_Simplification_Anesthesiology.py and press enter.
- When the code has finished running, check the mesh has been labelled correctly using meshlab: Open a terminal in the folder that you are analysing and type meshlab
- Move the mesh around by clicking on it and whilst holding the mouse. Check the labelling by selecting: Image: Randomly colour surfaces. Check if the mesh and display elements by selecting: Surface: Outline

- Close meshlab: Close analysed folder terminal.

5. FO: Select a point on the septal wall, close to the mitral valve, follow line from RSPV down to valve.

- LSPV posterior: Select a point to assign the posterior section of the LSPV/LA junction.
- RSPV posterior: Select a point to assign the posterior section of the RSPV/LA junction.

Press q to quit

You can open these landmark point files in paraview.

6. UAC codes: UAC_First_Simplification.py: Create UAC.

- Go to UAC codes folder
- Open the UAC_First_Simplification.py code and change the path i.e. to the folder you are analysing (Remember to include the final /). Save. Close code.
- In the terminal you used to run the LabelledMesh_Simplification_Anesthesiology.py code, now type: python UAC_First_Simplification.py and press enter. The code should now run. Make sure there are no error messages.
- When the code has finished running, check it has run correctly with paraview.
- Open a terminal for the folder that you are analysing and type: meshlab & click enter. This will open paraview.
- Select File: Open: Labelled_Coords_2D_Rescaling_N3.vtk. Click Ok. Click Apply. You will now see the 2D version of the 3D mesh. Check that this version appears normal i.e. you can see LAA, LSPV, RSPV and RPIV - they are whole and well defined.

see LAA, LSPV, RSPV and RPIV - they are whole and well defined.

If normal close paraview and close the UAC_codes terminal

PointSelection.py code - making sure you have selected the correct landmarks

Open a terminal for the folder that you are analysing and type paraview and press enter. This will open paraview.

Select File: Open: Labelled_Coords_2D_Rescaling_v3_C.vtk. Click Ok. Click Apply. You will now see a refined 2D version of the 3D mesh. Check that this version appears normal i.e. you can see LAA, LSPV, RSPV and RPIV - they are whole and well defined.

Continue to next step...

What is CemrgApp?

CemrgApp enables cardiovascular clinical researchers to perform advanced image analysis with limited training



Built on top of MITK (C++)



Custom image processing and computer vision toolkits



Performs: statistical, machine learning, and simulation approaches



Allows the study of physiology, pathology, inform diagnosis and treatment

Disclaimer: CemrgApp is a research tool, not a clinical tool and should not be used in diagnosis or treatment for patients

Open File Save Project Close Project

Undo Redo

DICOM

Image Navigator

View Navigator

Cemrg Scar Quantification

Display

Scar Quantification Pipeline

Step1: Load DICOM

Step2: Process Images

Step3: Scar Analysis

Step4: Segment Images

StepX: Registration

StepY: PV Clipper

StepZ: Create Surface

StepA: MV Clipper

StepB: Scar Map

StepC: Scar Quantification

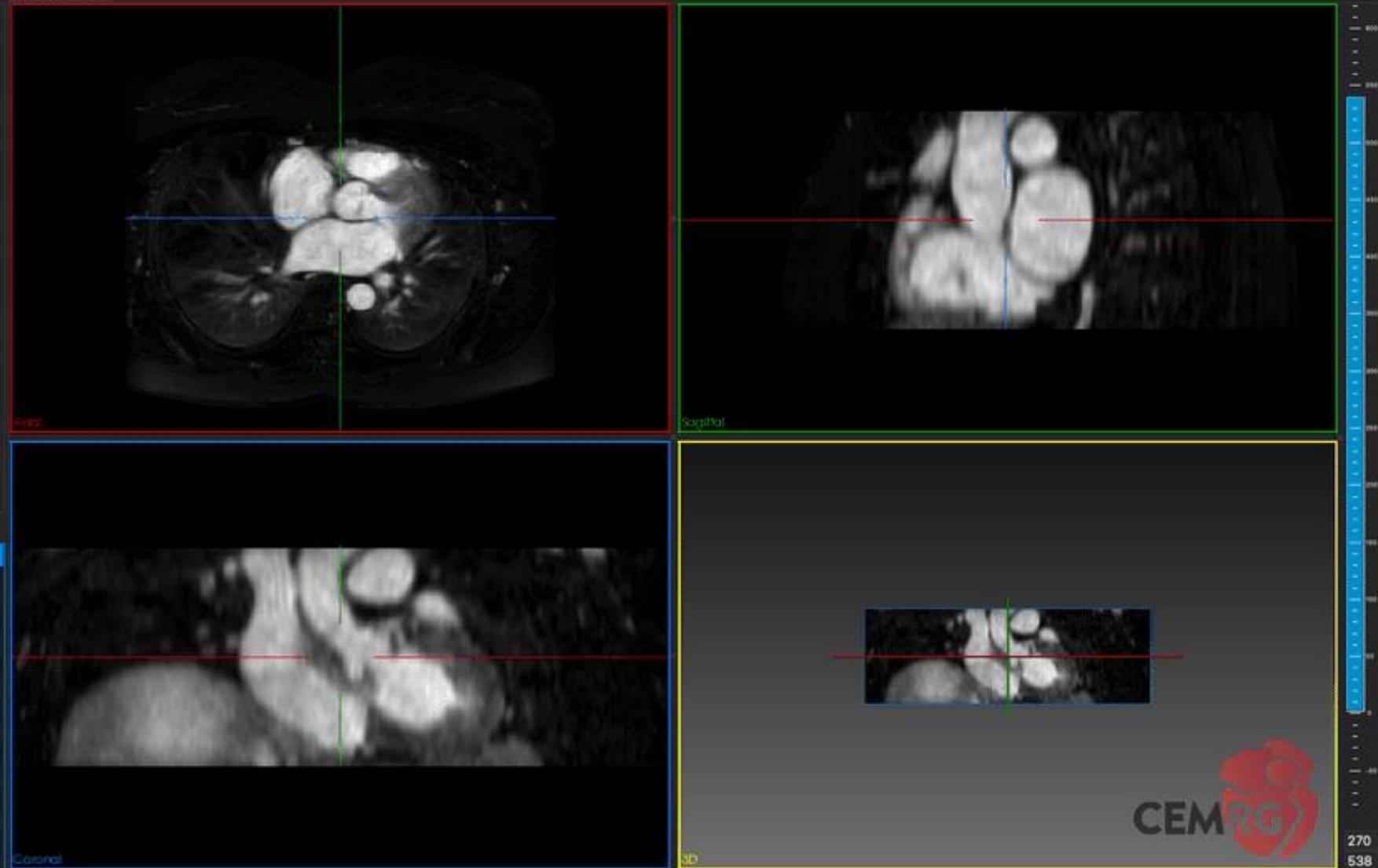
Calculate Sphericity

Advanced Analysis

Reset

Data Manager

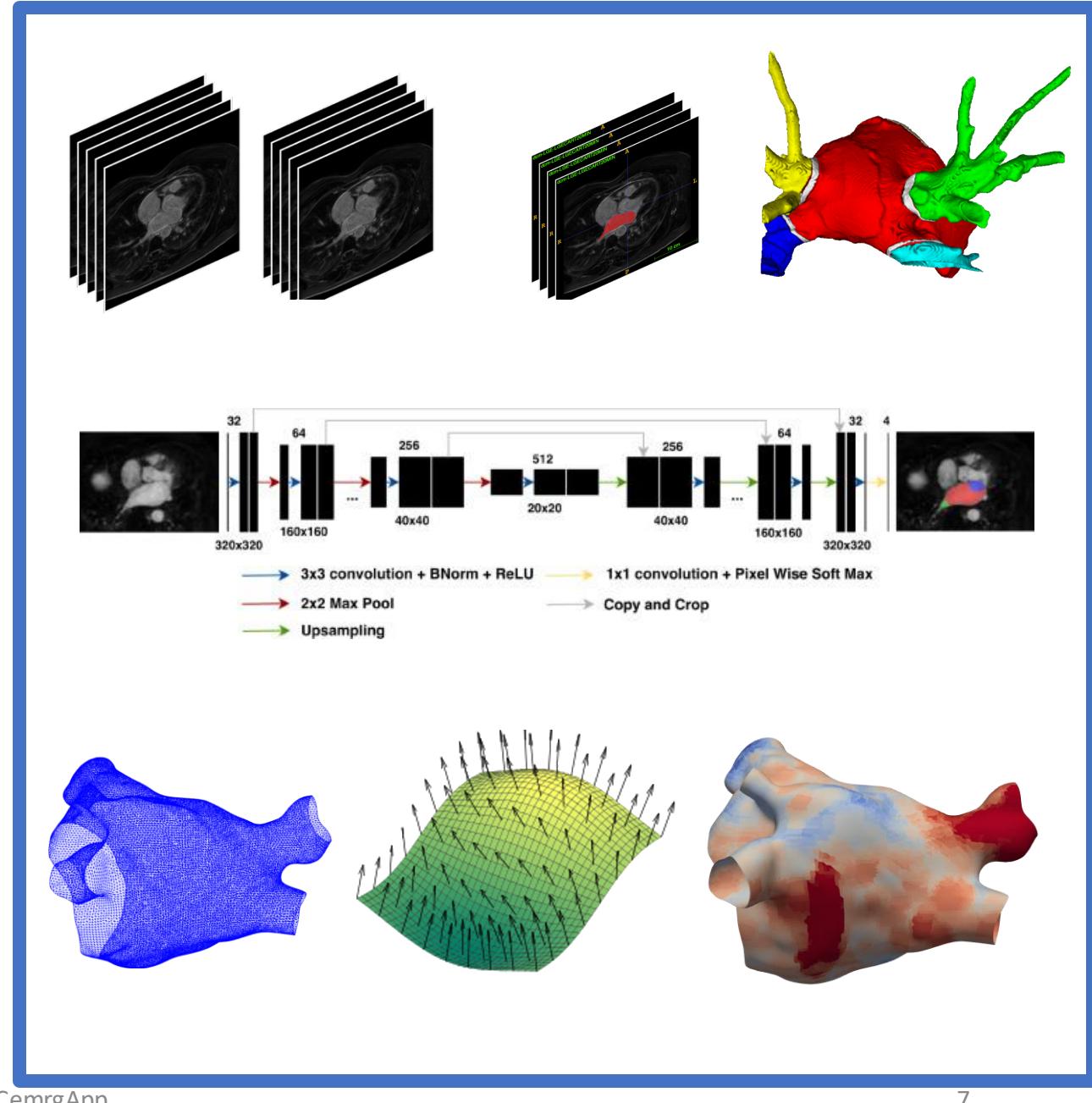
- dcm-MRA-CEMRA90saftergad
- dcm-LGE-LGECART20MIN



Key Features

- Image processing
 - Registration
 - Segmentation
 - Machine Learning tools
- Mesh manipulation
- In-house algorithms
- Collaborations from other groups

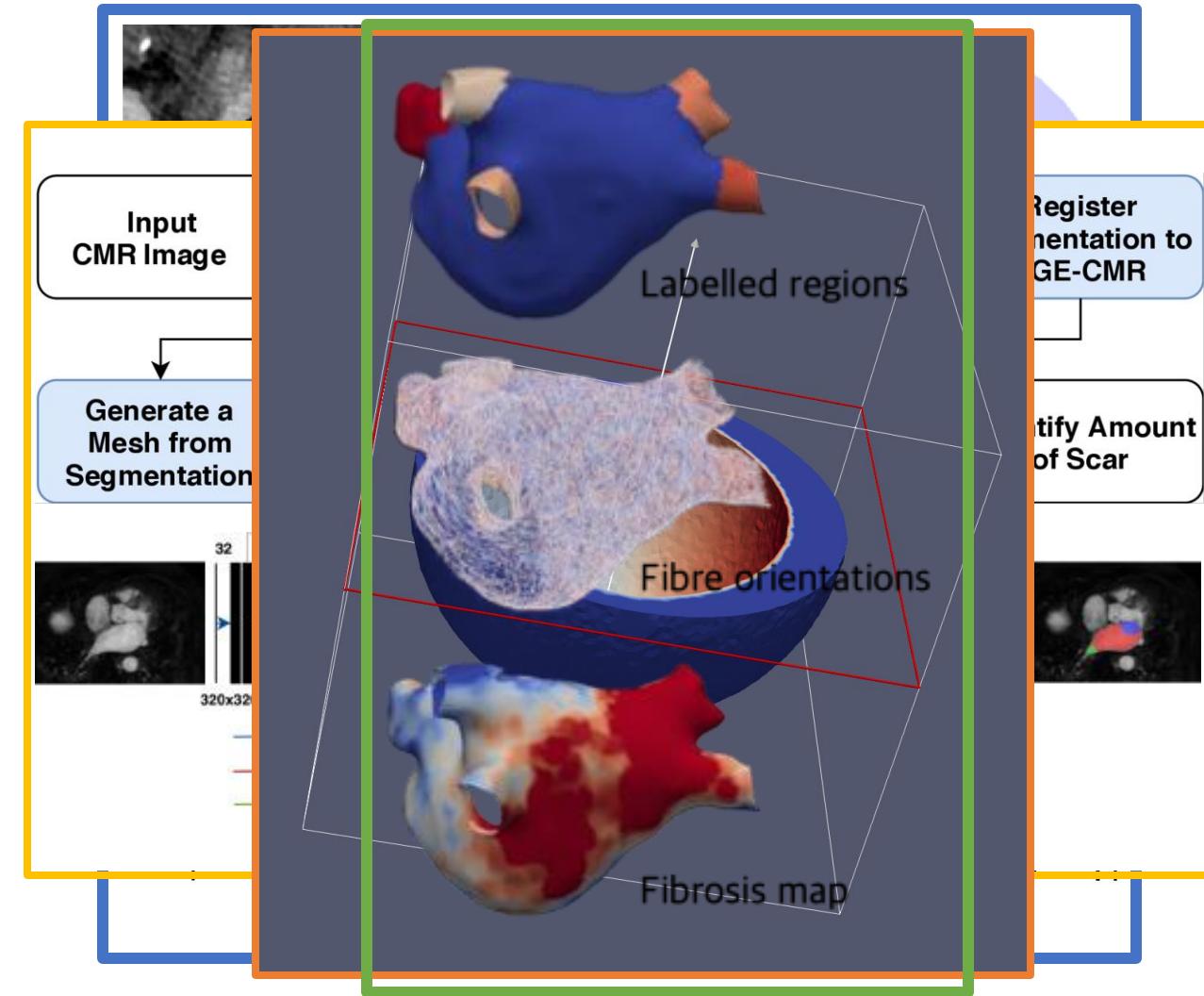
These can be combined into plugins, to address a specific task



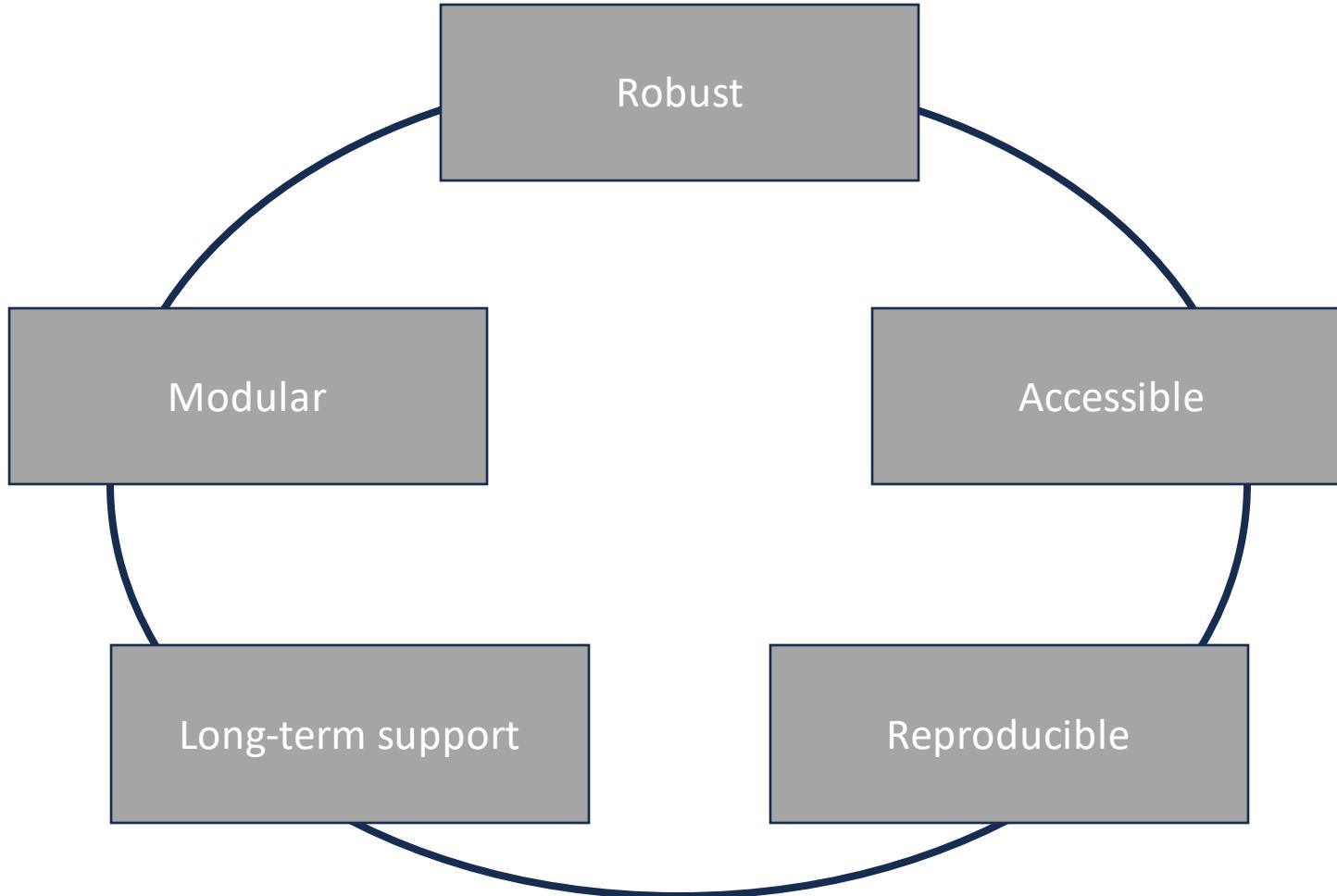
Applications of CemrgApp

Five main workflows (plugins):

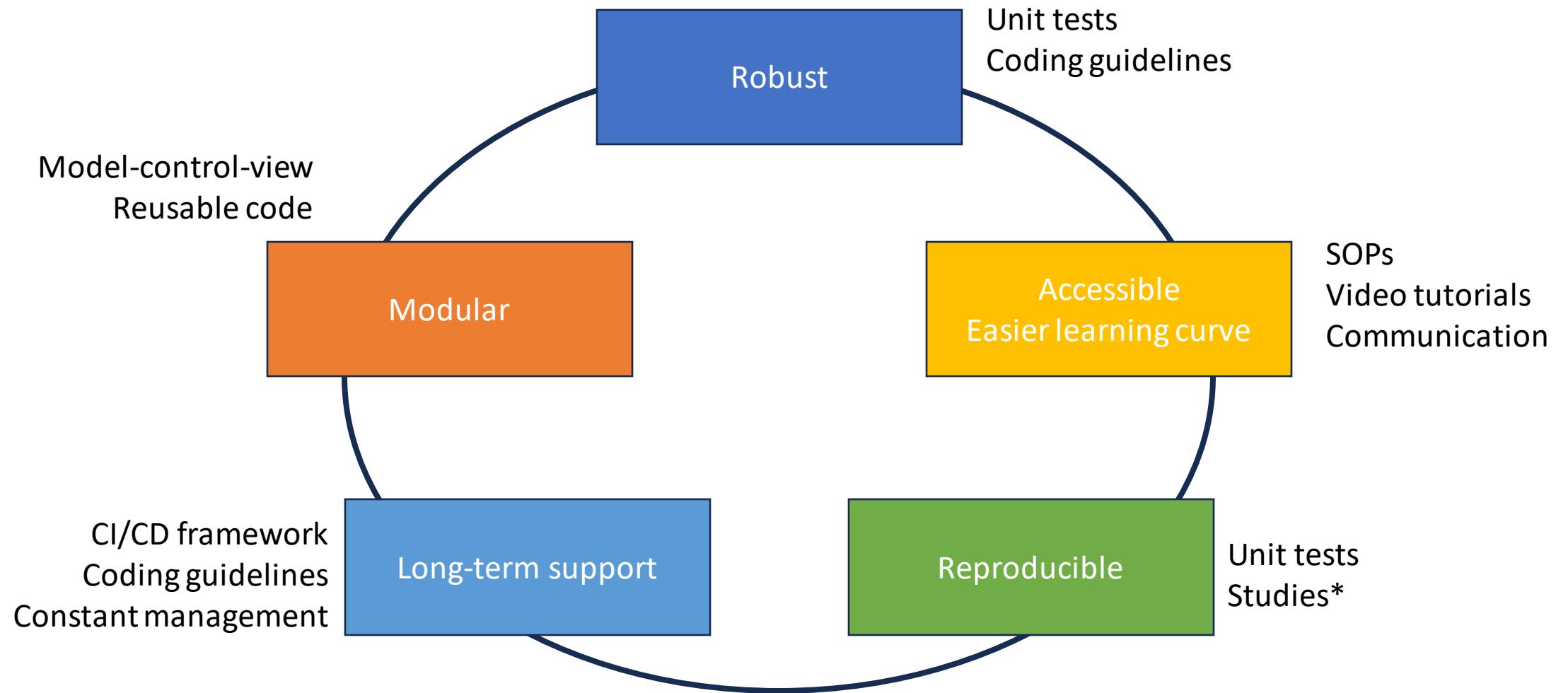
- Motion Quantification
- Anatomical Measurements
- Morphological Measurements
- Scar Quantification
- Atrial Toolkit



Sustainable Development



Sustainable Development



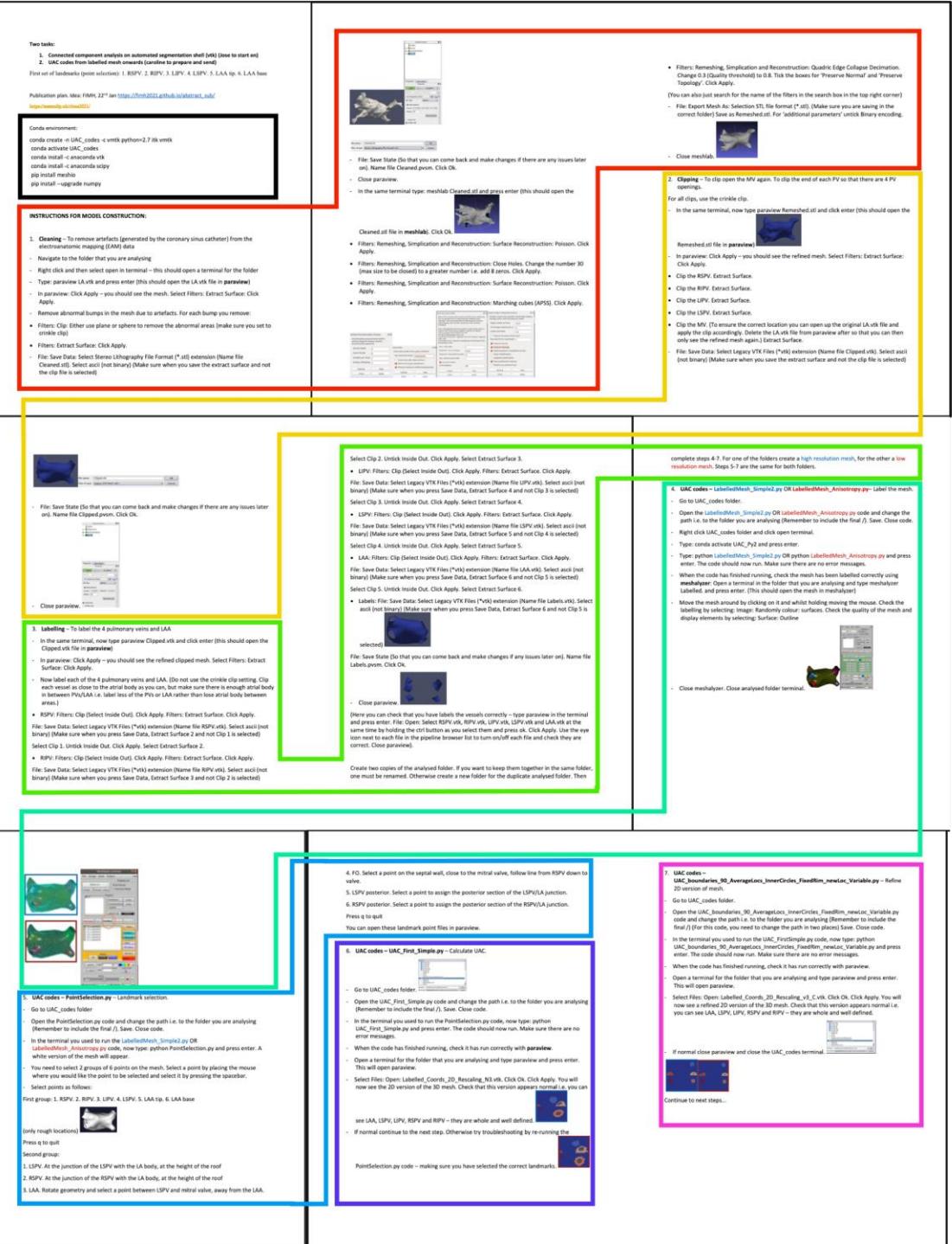
Reproducibility Study

Objective

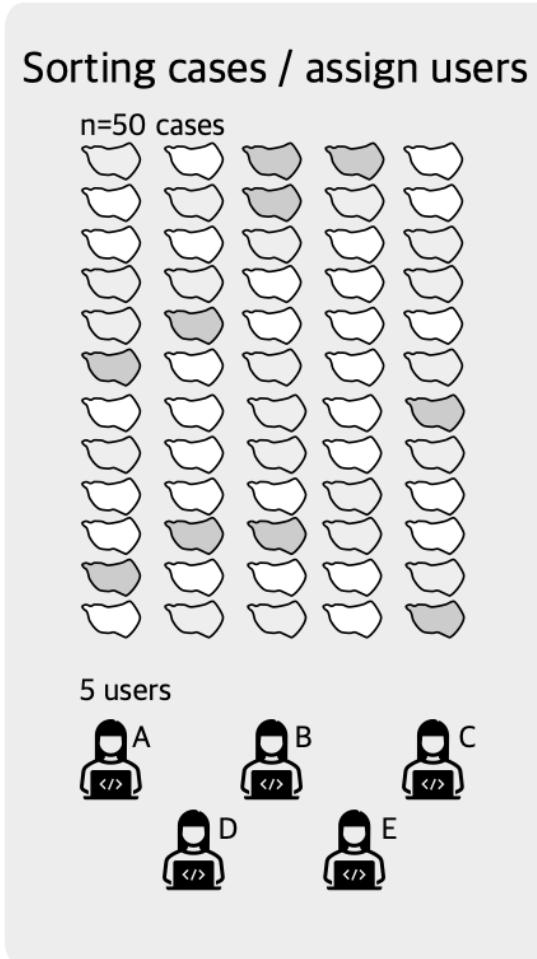
Create a model suitable for electrophysiological simulations from a patient MRI scan

A complicated pipeline...

- Lengthy: 4.5h to complete a single case
 - Manual and involved process
 - Susceptible to user error
 - Not reproducible
 - Not scalable



Reproducibility Study: Left Atrial EP Models



Reproducibility Study: Left Atrial EP Models

 Computers in Biology and Medicine
Volume 162, August 2023, 107009

Evaluation of an open-source pipeline to create patient-specific left atrial models: A reproducibility study

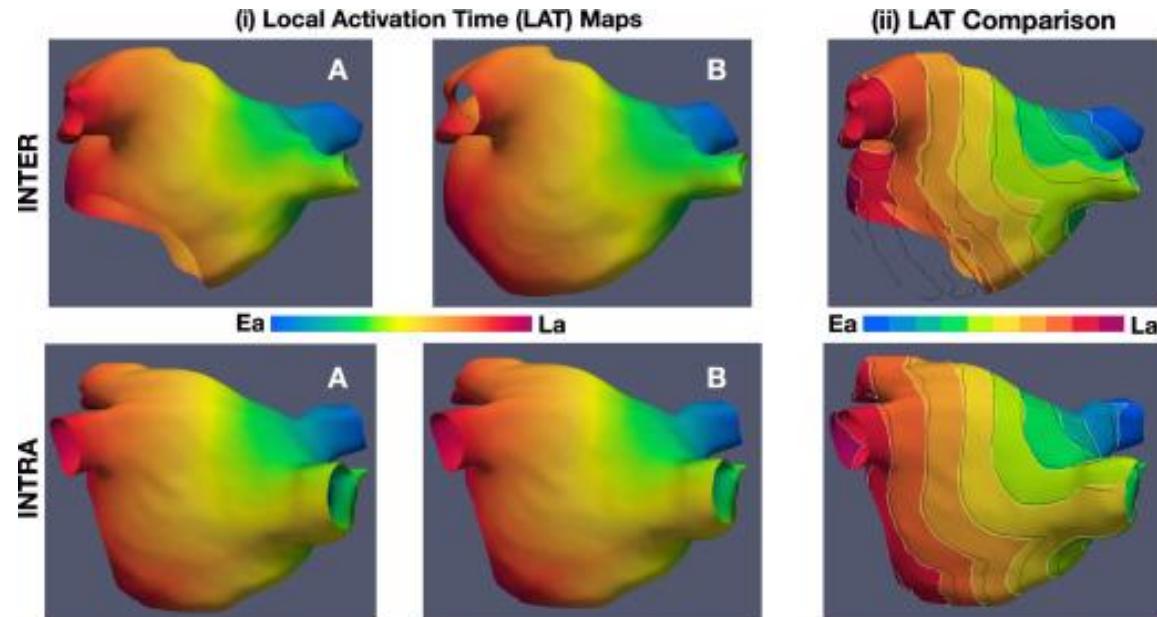
José Alonso Solís-Lemus ^a    , Tiffany Baptiste ^a , Rosie Barrows ^a , Charles Sillett ^a ,
Ali Gharaviri ^{a c} , Giulia Raffaele ^{e a} , Orod Razeghi ^{d a} , Marina Strocchi ^a , Iain Sim ^a , Irum Kotadia ^a ,
Neil Bodagh ^a , Daniel O'Hare ^a , Mark O'Neill ^a , Steven E. Williams ^{a c} , Caroline Roney ^{a b 1} ,
Steven Niederer ^{a f 1}

Show more ▾

 Outline |  Share  Cite

<https://doi.org/10.1016/j.combiomed.2023.107009> 

Under a Creative Commons license 



- Model building time **reduced** to just **16.7 min.**
- **Impact** of operator variability is **similar** to that of image resolution or fibre field

Reproducibility Study: Practical Implications

- To **support** upcoming frameworks for generating patient-specific atrial models (and beyond)
- The **first** simulation-study **reproducibility template**
- CemrgApp is fully open-source, runs on standard laptops, and follows standardised methodologies

Get Involved!

The screenshot shows the GitHub repository page for CemrgDevelopers/CemrgApp. Key features visible include:

- Issues:** 10 open issues.
- Pull requests:** 3 open pull requests.
- Code:** Repository contains 181 commits across 10 branches and 6 tags.
- About:** An interactive Medical Imaging Platform with Image Processing and Computer Vision Toolkit for Cardiovascular Research.
- Badges:** .github, CemrgApp, TestScripts, LICENSE, MITK_LICENSE, README.md.
- Releases:** 5 releases, including "Binary Executables for CemrgApp" (Latest).
- Packages:** No packages published.
- Contributors:** 3 contributors.
- License:** BSD-3-Clause License.
- README.md:** Cardiac Electromechanics Research Group App.
- Build Status:** CemrgApp Build & Deploy: passing, CemrgApp Tests and Code Coverage: passing, CemrgApp Code Analysis: passing, codecov: 13%.
- Description:** CemrgApp is an MITK based interactive medical imaging application with image processing and computer vision toolkits for cardiovascular research.
- Copyright:** Copyright (c) Cardiac Electromechanics Research Group. All rights reserved.

The screenshot shows the GitHub repository page for CemrgDevelopers/CemrgApp, specifically the Issues tab. The list of open issues includes:

- Motion Quantification: Apply transform not working
- Update to MITK v2021
- Reporting bugs and issues easily
- Crop Images Documentation
- Duplication of code in cmdapps
- CI/CD - Move automatic builds dependencies from personal account to a more permanent place
- Investigate storing build and other files under LFS
- Remove Docker Invocation from Module
- Code Review To Identify Dead Code and Document Used Code

The screenshot shows the GitHub repository page for CemrgDevelopers/CemrgApp, specifically the Home tab. The page includes:

- Home:** Orod Razeghi edited this page on Oct 28, 2020 - 4 revisions.
- Cardiac Electromechanics Research Group App:** CemrgApp is an MITK based interactive medical imaging application with image processing and computer vision toolkits for cardiovascular research.
- License:** Copyright (c) Cardiac Electromechanics Research Group. All rights reserved. CemrgApp is available as free open-source software under a 3-clause BSD license. This software is distributed WITHOUT ANY WARRANTY OR SUPPORT. This software SHOULD NOT be used for diagnosis or treatment of patients.
- Current Versions:** Versions of CemrgApp are organised in separate folders in the repository.
- Citation:** Please cite the following article, if you use CemrgApp in your project: Razeghi O, Solis-Lemus J, Lee A et al. CemrgApp: An interactive medical imaging application with image processing, computer vision, and machine learning toolkits for cardiovascular research. SoftwareX. 2020;10:1016/j.softx.2020.100570.
- Clone this wiki locally:** <https://github.com/CemrgDevelopers/CemrgApp>

@CEMRG_
cemrg.com

cemrgapp.com

github/OpenHeartDevelopers

Take home message

CEM

We believe CemrgApp has the potential for:

- the **development** of patient-specific computational heart models,
- facilitating their **adoption into clinical** applications, and
- enable research with **larger cohorts**