

HuBMAP-SPARC Linkage - version 1.0

The scope of this document covers the steps involved in linking anatomical organ scaffolds from the HuBMAP common coordinate framework (CCF) to the SPARC scaffold CCF and vice versa. We also describe progress on defining fat pads around the epicardial surface of the SPARC heart scaffold and plans to map HuBMAP tissue blocks to SPARC scaffolds and to animate the SPARC scaffolds, together with these tissue blocks, such that they can be transferred back to HuBMAP.

HuBMAP organ models

HuBMAP's organ reference objects have been created using data from the Visible Human male and female datasets provided by the National Library of Medicine. These organs can be freely accessed and downloaded in *GLB* file formats from the link below:

<https://hubmapconsortium.github.io/ccf/pages/ccf-3d-reference-library.html>.

These models include 'crosswalk' files which contain anatomical structures (AS) that correspond to the ASCT+B table that is aligned with Uberon/FMA and CL (see the two links below:

- <https://www.biorxiv.org/content/10.1101/2021.05.31.446440v2>
- <https://hubmapconsortium.github.io/ccf-releases/v1.0/docs/ref-organs/asctb-3d-models-mapping.html>

Mapping a HuBMAP model to a SPARC Scaffold

HuBMAP model organs are available in *GLB* format; *GLB* is a binary file format used in virtual reality applications and, in particular, by the Blender software package. To map any HuBMAP CCF to a SPARC scaffold CCF, geometrical and ontological data from HuBMAP models must be extracted into the OpenCMISS-ZINC *EX* file format. This file format contains the spatial coordinate field as well as ontological terms as anatomical annotation groups. To convert a HuBMAP *GLB* model to an *EX* format, a series of steps is involved:

- 1) Load HuBMAP *GLB* model into the Blender desktop application (Blender can be downloaded and installed from <https://www.blender.org/download/>)
- 2) Each annotated anatomical region in the model must be extracted individually. To do that, turn on each anatomical 'node' from the *Scene Collection* pane one by one and export the selected group as *STL* binary object using *File>Export as* in the Blender app.
- 3) Once all anatomical regions are exported into individual *STL* files, they need to be combined into a single *EX* data file. A custom Python script has been written to perform this task and it can be accessed from <https://github.com/mahyar-osn/hubmap2scaffold>.
- 4) The *EX* data file can be used as input into the Scaffold Mapping tools (<https://sparc.science/help/instructions-to-use-the-map-core-scaffold-mapping-tool>) to 'fit' an anatomical Scaffold to the HuBMAP *EX* data.
- 5) Export the fitted Scaffold to *VTK* file format.

Figure 1 shows a general overview of the process described above.

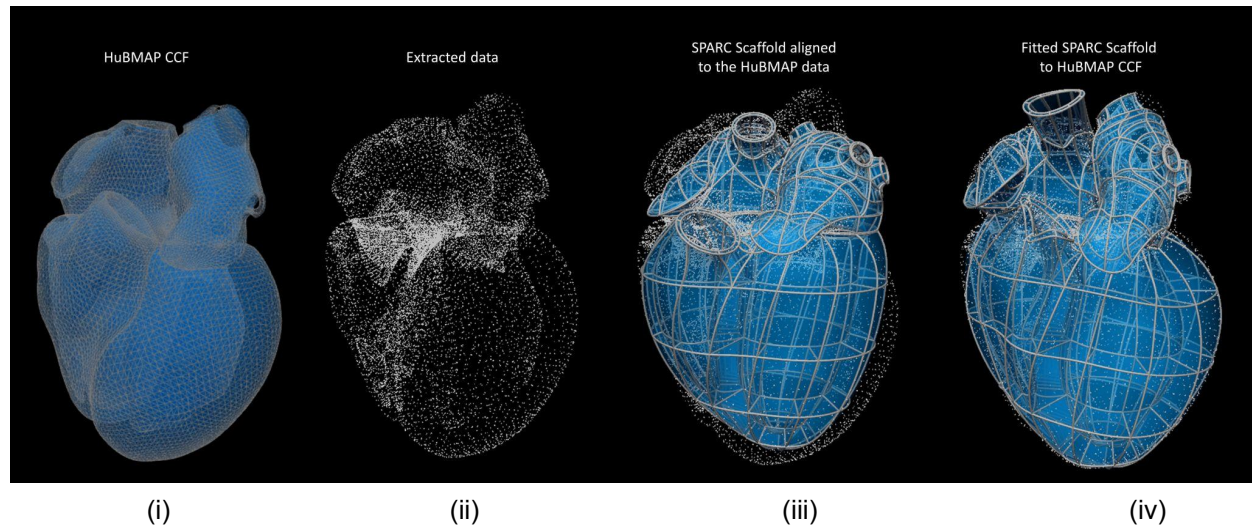


Figure 1. Fitting the SPARC heart Scaffold to HuBMAP heart CCF. The surfaces of the HuBMAP heart model (i) are used to generate point clouds (ii) that are then used to fit the geometry of the SPARC heart scaffold by first automatically orienting and scaling the scaffold (iii), followed by automatically adjusting the finite element nodal parameters to ensure that the surfaces of the scaffold accurately replicate the HuBMAP data.

Transferring fitted SPARC scaffolds to the HuBMAP portal

The fitted SPARC scaffold needs to be represented as a *GLB* file format before it can be integrated into the HuBMAP CCF Portal. The steps below describe the process needed to produce the SPARC-HuBMAP CCF Reference Organ:

- 1) Install the ParaView desktop application (<https://www.paraview.org/>).
- 2) Load the fitted *VTK* Scaffold in ParaView. It is recommended that you edit the colors of the model using the *Properties* tab, depending on which organ you have selected. The GenericAnatomyColors is a useful website to select the appropriate colors (https://www.slicer.org/wiki/Slicer3:2010_GenericAnatomyColors).
- 3) Export the *VTK* file to *GLTF* and save the file.
- 4) Now open the Blender app. Delete the *Cube*, *Camera*, and *Lighting* nodes from the Scene.
- 5) Import the *GLTF* file that you exported in step 3.
- 6) Expand the tree in the *Scene Collection* pane. Rename 'Renderer Node' to 'SPARC<F/M>', and rename both 'mesh0' to 'SPARC<F/M>_<organ>'.
 - a) Note: depending on which sex and organ you have fitted, replace the above placeholders in the object names accordingly.
- 7) Export and save the scene as *GLB* (GLTF Binary). In the export window, expand 'Geometry' and check 'Tangents' (in addition to the default selections), and save the scene to 'human<Organ>_<F/M>.glb'. Again, replace <Organ> and <F/M> with appropriate terms.

- 8) Before closing the Blender app, take note of the dimensions in preparation for the next stage. Dimensions can be found in the *Transform* pane.
- 9) Go to the CCF Registration User Interface (RUI) webpage (<https://hubmap-ccf-ui.netlify.app/rui/>) and create an RUI registration using the tissue dimensions noted in step 8. Place the 'Tissue' relative to the **Skin** reference organ. Use the REVIEW AND DOWNLOAD button to save the RUI location named 'human<Organ>.VH<F/M>Skin.rui_location.json'. Again, replace <Organ> and <F/M> with appropriate terms.
- 10) In order for the Scaffold to be recognized as a reference organ in the HuBMAP CCF Portal, some modifications need to be made in the RUI JSON file. Use this example to make the appropriate changes:
 - a) https://github.com/SPARC-FAIR-Codeathon/hubmap-link/blob/main/api/public/scaffolds/humanStomach/humanStomach_F.ccf.jsonld
- 11) Modify the ccf-eui.html (<https://github.com/SPARC-FAIR-Codeathon/hubmap-link/blob/main/website/src/assets/ccf-eui.html>) file to include the reference organ file in the 'dataSources' array.

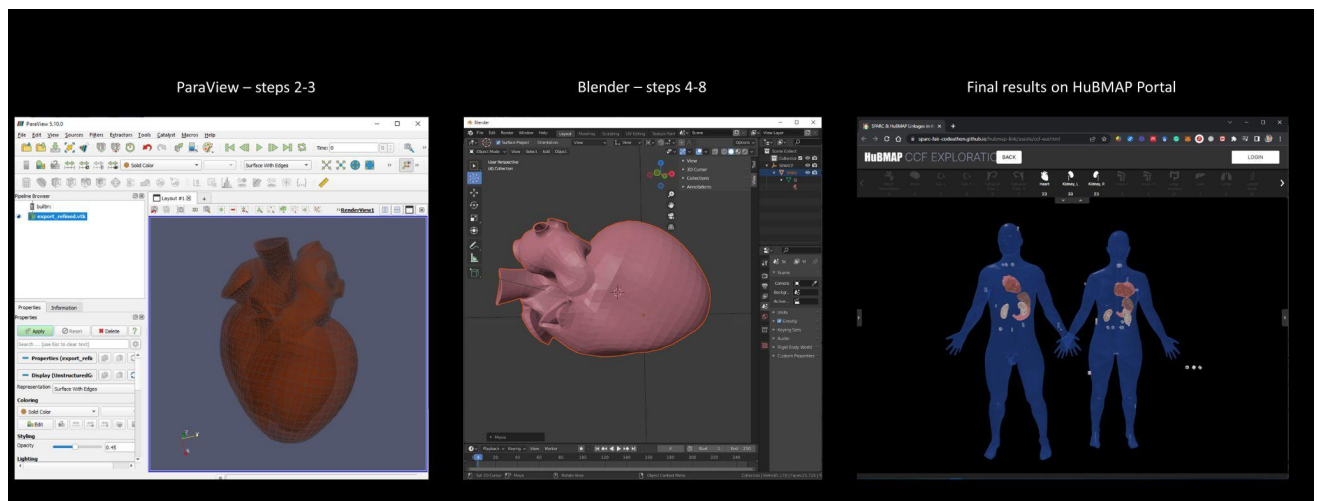


Figure 2. The fitted SPARC scaffold undergoes a series of steps as described [here](#) to produce the appropriate file types for registering the scaffold in the HuBMAP 3D body ("skin organ").

At this stage, we are ready to upload the Scaffold files and also push the changes that we made in the files in steps 10 and 11 in the previous section. The repository to upload and push is <https://github.com/SPARC-FAIR-Codeathon/hubmap-link>. Use the 'Scaffold-to-ccf'.

Defining fat pads around the SPARC heart scaffold

A further requirement for the scaffolds is to be able to embed neuronal data in the epicardial fat pads that typically surround an older and possibly diseased heart. This has now been done and is illustrated in Figure 3.

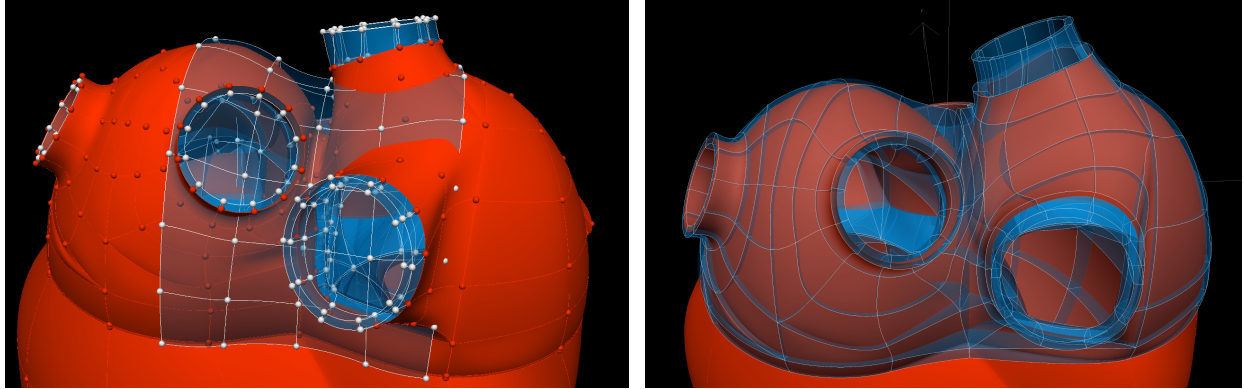


Figure 3. Adding further finite element layers for embedding neurons in epicardial fat pads. The layer shown on the left with white lines and nodes defines the outer boundary of the fat. Elements for the adipose tissue are generated algorithmically to fill the space between this layer and the epicardial surface of the heart model. Only the atrial fat is shown here.

The most urgent next step is automating the mapping between HuBMAP and SPARC scaffolds (i.e., removing any dependence on human intervention). This will require scripts that convert the GLB format to EX format).

Transferring animated SPARC scaffolds to HuBMAP

With the two-way mapping between HuBMAP and SPARC scaffolds now established, we could transfer an animated SPARC scaffold (e.g., based on scaffolds fitted to time varying human MRI data), across to HuBMAP frame by frame. This will require the automated file conversion mentioned in the previous paragraph.

Mapping tissue blocks from HuBMAP to SPARC

The tissue blocks that HuBMAP uses to contain tissue information are defined relative to the HuBMAP CCF. Each block is defined by a central material point (in CCF coords), a 3D orientation, and scaling along the 3 block axes. This information is contained in a JSON file available via the RUI (Registration User Interface).

The codebase now contains relevant codes and tools to map embedded tissue blocks from RUI in the HuBMAP CCF to SPARC Scaffold, given a registered organ scaffold. Each tissue block contains an '@id' and its transformation data associated with it. There are two transformation steps involved in this process: 1) transforming from the organ coordinate space to the body coordinate space, and 2) transforming the block from the origin to the location defined by RUI.

We set up a demo [website](#) to show an example of such mapping for the heart in both the female and the male data, see Figure 4 below.

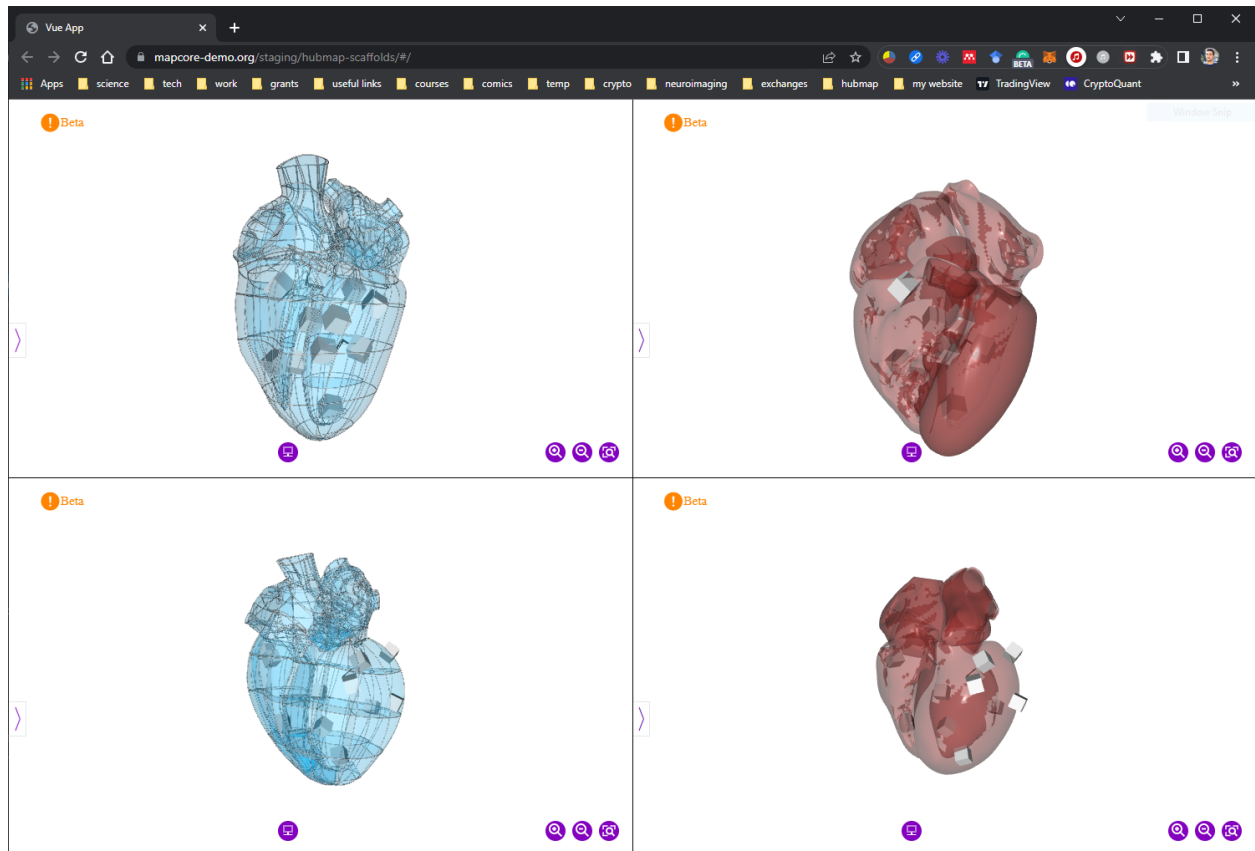


Figure 4. Screenshot of the demo site showing the registered SPARC Scaffolds (left) for the male (top) and female (bottom) hearts, with the corresponding HuBMAP Scaffolds (right) for the male (top) and female (bottom) hearts. For comparison purposes, the registered tissue blocks for each sex are shown.

Mapping tissue blocks from SPARC to HuBMAP

The reverse mapping of tissue blocks from a SPARC scaffold to HuBMAP can also be achieved using the same code repository. The embedded tissue block in the SPARC scaffold contains coordinate information required to compute the transformation to map the tissue block into the HuBMAP CCF organ scaffold. The output of this process is a RUI JSON file. Note that the actual RUI JSON contains a lot more information about the tissue block; since we currently do not have an actual tissue block registered in the SPARC scaffold, we are not able to extract such information and store them in the RUI JSON. Needless to say, the process will remain the same and adding this extra information is trivial once/if the tissue block is available.

Future work

We also need to map other organs, in particular the lungs, stomach, bladder and colon. The mapping between HuBMAP and SPARC CCF needs to be established individually for each

organ. It will never be a simple process to establish the mapping, but once done for each organ of interest it is then fairly straightforward to transfer data between the two. A key difference is that SPARC scaffolds capture spatially continuous fields, whereas HuBMAP defines data within an arbitrary number of discrete tissue blocks. It is important that data are annotated to indicate the region of the organ they lie within (e.g. myocardium or surrounding epicardial fat).