

# Quantitative and At-Scale Electroanatomic Mapping Data Analysis Using OpenEP and EP Workbench

Steven E Williams<sup>1,2,3</sup>, Vinush Vigneswaran<sup>2</sup>, Ali Gharaviri<sup>1</sup>, Kestutis Maciunas<sup>1</sup>, Christopher O'Shea<sup>4</sup>, Irum Kotadia<sup>2</sup>, Iain Sim<sup>2</sup>, Neil Bodagh<sup>2</sup>, Magda Klis<sup>2</sup>, John Whitaker<sup>2</sup>, Nick Linton<sup>6</sup>, Martin Bishop<sup>2</sup> and Mark O'Neill<sup>2,5</sup>

<sup>1</sup>Institute for Neuroscience and Cardiovascular Research, The University of Edinburgh <sup>2</sup>School of Biomedical Engineering and Imaging Sciences, King's College London <sup>3</sup>NHS Lothian, Lothian, UK <sup>4</sup>Institute of Cardiovascular Sciences, University of Birmingham, <sup>5</sup>Department of Cardiology, Guy's and St Thomas' NHS Foundation Trust <sup>6</sup>Department of Bioengineering, Imperial College London

## Abstract

*OpenEP is an open-source library for electrophysiological data analysis, first released in 2020. EP Workbench is the graphical interface for OpenEP which empowers translation by enabling non-technical users to undertake quantitative analysis of electrophysiological data. EP Workbench is cross-platform software, available for Linux, Windows and MacOS operating systems. This paper provides an update on the features and tools added to EP Workbench since initial release. Development is ongoing in the areas of electroanatomic mapping data feature extraction and integration with high performance computing. The addition of an inline code editor ("Work in Progress" workspace) allows implementation of project-specific algorithms. Close integration with core modules allows Work in Progress algorithms to leverage the existing visualization, manipulation and data management functionalities. EP Workbench is supported by an active user and development community. OpenEP and EP Workbench provide the foundations for a forthcoming national UK study (OpenEP|NET) which will create the world's first open-access electrophysiology dataset comprising 2,300 patients with electroanatomic mapping data linked to clinical outcomes. Together, these developments have the potential to kickstart a new era of data-driven diagnosis and therapy for arrhythmia patients.*

## 1. Introduction

Electroanatomic mapping data, collected during routine electrophysiology procedures encapsulates a vast amount of data describing cardiac anatomy, motion, tissue structure and electrophysiological function. Owing to its complexity, clinical studies of electroanatomic data (and to a lesser extent electrophysiology data) frequently apply qualitative or semi-quantitative techniques. To enable quantitative analysis, we released the OpenEP framework for electrophysiology research (<https://openep.io>). This open-source suite of analysis tools provides: (1) interfaces for clinical electroanatomic mapping platforms; (2) a data storage model which reduces data set sizes by several

orders of magnitude, typically to 10-20Mb per patient and (3) a full, publicly available open source validated and reproducible suite of tools for manipulation and analysis of electroanatomic mapping data.[1-13] This paper provides an update on the features added since initial release.

## 2. Methods

### 2.1. Software Development

Software development has been performed in Python. The open-source Python library, `openep-py`, currently supports Python versions 3.8-3.10 and is based on Numpy, Scipy and Numba. This library constitutes the reader and writer for OpenEP files, and allows manipulation of ablation, electric, surface and vector data through the Case object. EP Workbench renders the anatomy's shape and surface data of the Case object and allows a no-code approach to data manipulation through the GUI, built using PySide6, PyVista, Vedo and VTK and extends to support multiprocessing, animation rendering and remote SSH/SFTP access via Paramiko.

For the development of EP Workbench, we have followed best practices in modern software engineering, including unit testing and Behavior Driven Development (BDD) to ensure performance and functionality of the code, and implemented Continuous Development and Continuous Integration through GitHub Actions. The Python version of OpenEP remains hosted on Github (<https://github.com/openep>). The repository for the graphical interface is not yet publicly available. Interested users should contact the OpenEP administrators for access to the beta-testing program.

### 2.2. Communication

The OpenEP website, hosted at <https://openep.io>, provides general information on the project and source code documentation. Through the website, links are provided to the git repositories and discussion forum. A discussion forum is available at <https://openep.discourse.group> and provides a publicly accessible central location for question and answers about

OpenEP. There are actively maintained issue trackers available through Github.

### 3. Results

#### 3.1. User Interface

The EP Workbench user interface is shown in Figure 1. The interface has been designed to be familiar to electrophysiologists used to working with electroanatomic mapping systems. Key visualization components of the user interface include 3-dimensional viewers for map representation, image model and slice viewers for incorporating medical imaging data, and a mapping points list tool for visualizing electrogram data. Additional components include interfaces for feature extraction tools and the Work-in-Progress workspace editor (see below). System preferences and configurations are exposed through a series of options panels. All panels are dockable, to allow users to construct workspaces meeting their individual needs.

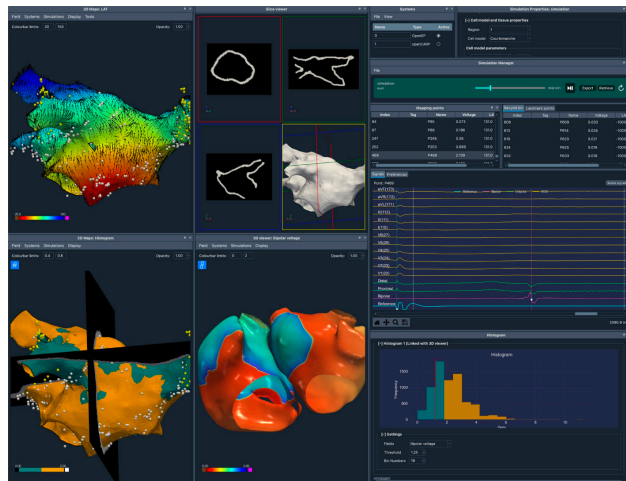


Figure 1: Overview of EP Workbench application for OpenEP. The interface consists of 3D viewer modules, a system list of loaded data sets, an electrogram viewer and analysis tools (shown – histogram analysis tool which also links to the 3D viewer). All panels are dockable and can be repositioned according to the needs of the particular analysis use case.

#### 3.2. Feature Extraction

**Data Cleaning.** Routinely collected electrophysiology data may require data cleaning before feature extraction to ensure that extracted features faithfully represent patient characteristics. Data cleaning usually involves editing the anatomy to remove irrelevant features such as cardiac valve or vein orifices, or reannotating electrogram features to accurately define electrogram reference timings. Further automated data preprocessing may be conducted to remove erroneous values from the original mapping data by (a) automatically removing mapping points with activation time values which are above a threshold (i.e. -10,000) and (b) removing mapping points captured outside of the 3D mesh walls. Examples of anatomic and

electrophysiological data cleaning are illustrated in Figure 2 and Figure 3, respectively.

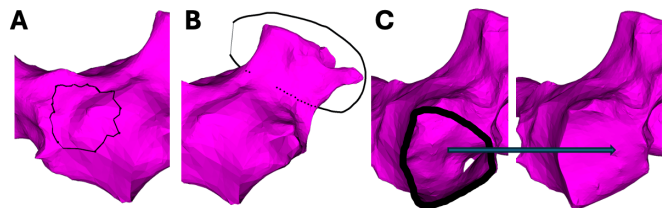


Figure 2: Anatomic data cleaning. A – Geodesic cutting tool for cutting holes in the anatomical data. B – Spline cutting tool for removing structures, shown here the left atrial appendage. C – Hole filling tool for creating watertight anatomies. In each panel, a left atrial geometry is shown with the color map set to pink.



Figure 3: Electrophysiological data cleaning. The electrophysiology reannotation tools include an electrogram points list (upper left panels), a recycle bin listing deleted points (upper right panels) and an electrogram viewer showing surface ECG, unipolar and bipolar electrograms. White arrows indicate manual reannotation of the local activation time based on unipolar and bipolar electrogram morphology.

**Data quantification.** OpenEP and EP Workbench enable customized feature extraction from electrophysiology data. Working across large datasets, openep-py can be used to automate feature extraction in a Python script either external to EP Workbench or within the Work-in-Progress workspace editor (see below). In-built functions are available for quantifying commonly used features across four categories: (1) geometric, (2) voltage, (3) conduction and (4) electrogram morphology. For use with external statistical analysis packages, data can be exported into a variety of formats including both OpenEP's proprietary file format and CSV formats.

#### 3.3. Work-in-Progress Tool

Whilst EP Workbench exposes the spectrum of feature extraction tools available in OpenEP, it is often necessary to implement custom algorithms to extract features from electroanatomic mapping data. However, implementing such features in the full software package for release requires completion of implementation, quality control,

testing and usability activities. To streamline integration of novel algorithms into EP Workbench, we have developed the ‘Work-in-progress’ (WIP) engine (**Figure 4**), consisting of three components: the WIP editor, the WIP manager and the WIP workspace:

**WIP editor:** The WIP editor enables users with programming expertise to write custom Python code within EP Workbench, describing their analysis algorithms. The WIP editor has access to any datasets loaded in the current instance of EP Workbench and can push data fields back to the same instance for visualization. The WIP editor also exposes a JSON field which lets the user customize the appearance of the WIP workspace. This feature is used to enable runtime end user interaction with algorithms, for example specifying which data fields should be processed or setting constants/constraints.

**WIP manager:** The WIP manager (**Figure 5**) is a graphical component which enables non-technical users to run algorithms which implemented as WIP modules. The interface exposes start and pause buttons together with a progress bar to indicate completion of processing. Within the WIP manager, WIPs are identified by their name, which is in turn editable through the WIP editor.

**WIP workspace:** The appearance of the WIP workspace (**Figure 5**) is defined through the WIP editor via a JSON-based syntax. The syntax allows the creation of text, numeric, dropdown list and radio-button user interface components. The contents of the WIP workspace can be edited in real time and refreshed through the WIP editor. User input in the WIP workspace is interpreted at the run time of the relevant WIP.

### 3.4. High Performance Computing

EP Workbench facilitates data processing on remote machines, including high-performance computing. Integration with remote and high-performance computing has been developed for both custom algorithms implemented through the WIP, and for electrophysiology simulations, implemented through openCARP.

Connections with remote machines are established via SFTP for file transfer and SSH for remote access, with all connection options specified through the Preferences pane of EP Workbench (**Figure 6**).

EP Workbench creates a snapshot of the current state and case object (3D model and its attributes) into a temporary compressed file, and sends the data files to the configured remote machine. For supercomputers or computer clusters, EP Workbench builds a SLURM file (workload manager configuration), and triggers the execution command. Upon completion, relevant results and files are retrieved. To improve computational efficiency, caching techniques, vectorization, and parallel processing are implemented for large-scale experiments. Caching has proven particularly effective when repeating multiple experiments on the same dataset, by storing intermediate files shared across experiments, resulting in a

substantial reduction in compute time.

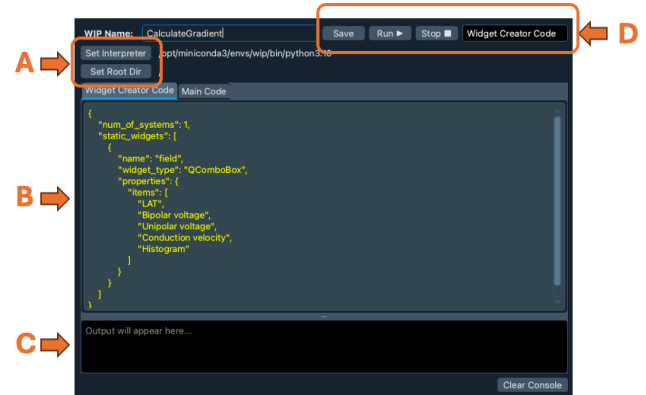


Figure 4: The Work-in-Progress (WIP) Editor. The editor allows full customization of user-defined algorithms for processing electroanatomic mapping data. Key components include buttons to set the interpreter and root directory (A), editor panels for both the user-interface defined through a JSON object and the Python code for the custom analysis algorithm (B), a console which shows system outputs during algorithm execution (C) and run controls (D).

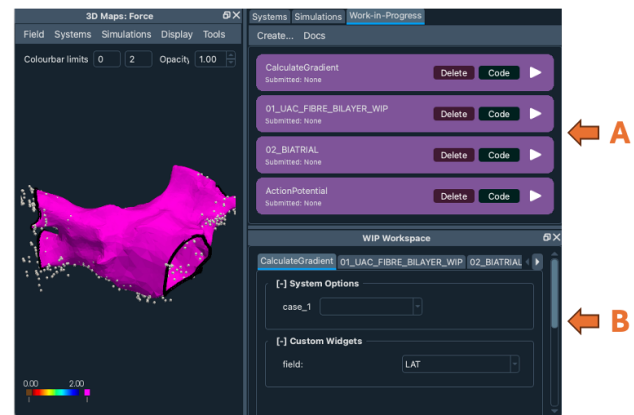


Figure 5: The Work-in-Progress (WIP) manager (A) and workspace (B).

## 4. Discussion

The development and expansion of OpenEP and its graphical interface, EP Workbench, address longstanding challenges in the quantitative analysis of electroanatomic mapping (EAM) data. Historically, the scale and complexity of EAM datasets have limited widespread analyses to specialized centers with dedicated engineering support. OpenEP overcomes these barriers by providing a lightweight data structure, reducing per-patient file sizes by orders of magnitude, and offering validated, reproducible tools for data manipulation. EP Workbench builds on this foundation by offering a user-friendly, cross-platform interface that allows both technical and non-technical users to perform advanced analyses.

The incorporation of data cleaning and feature extraction tools enhances the utility of EP Workbench, ensuring high-quality input for downstream analyses. The modular design allows users to tailor the interface to specific workflows, increasing usability across diverse



clinical and research settings. Importantly, the Work-in-Progress (WIP) engine represents a significant innovation. By enabling user-defined algorithms within the graphical interface, the WIP tool bridges the gap between rapid prototyping and production-level software integration. This advance will foster translational research and expedite the deployment of novel analyses across the electrophysiology community.

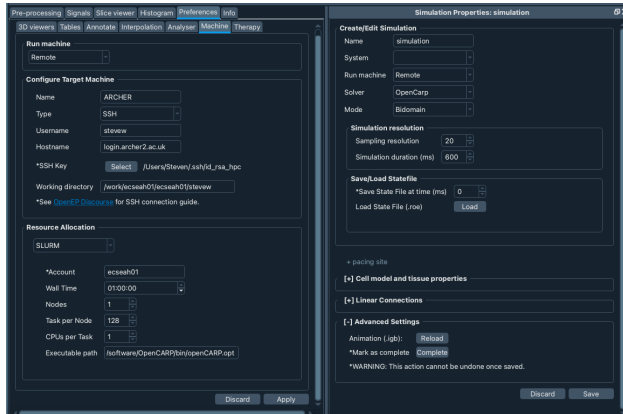


Figure 6: The remote connection preferences pane (left panel) and the simulation properties pain (right panel).

Furthermore, the integration with high performance computing (HPC) environments is a critical step toward truly at-scale analysis. It enables the processing of large, multi-patient datasets and the execution of computationally intensive tasks such as simulations and large-batch feature extraction.

Together, these developments lay the groundwork for transformative projects like the OpenEP|NET study, which will generate a large, open-access EAM dataset linked to clinical outcomes. As such, EP Workbench and OpenEP will not only democratize access to electrophysiological data analysis but also support the growing movement toward data-driven diagnostics and personalized arrhythmia therapy.

## 5. Conclusions

OpenEP and EP Workbench are reshaping the landscape of electrophysiology research by making scalable, quantitative analysis of electroanatomic data accessible to all. With powerful tools, a flexible interface, and support for custom algorithms and HPC integration, they are set to drive a new era of data-driven discovery and clinical innovation in arrhythmia care.

## Acknowledgments

The authors acknowledge the support of the British Heart Foundation (RE/24/130012) and the Medical Research Council (APP53906). SEW is supported by the British Heart Foundation (FS/20/26/34952). COS is supported by the Sir Henry Wellcome Fellowship, Wellcome Trust, 221650/Z/20/Z.

## References

- [1] Bindini, L., et al., *All-in-one electrical atrial substrate indicators with deep anomaly detection*. Biomedical Signal Processing and Control, 2024. **98**.
- [2] Hartwig, V., et al., *A Novel Workflow for Electrophysiology Studies in Patients with Brugada Syndrome*. Sensors (Basel), 2024. **24**(13).
- [3] Latrofa, S., et al., *Endocardial repolarization dispersion in BrS: A novel automatic algorithm for mapping activation recovery interval*. J Cardiovasc Electrophysiol, 2024. **35**(5): p. 965-974.
- [4] Zahid, S., et al., *Conduction velocity is reduced in the posterior wall of hypertrophic cardiomyopathy patients with normal bipolar voltage undergoing ablation for paroxysmal atrial fibrillation*. J Interv Card Electrophysiol, 2023.
- [5] Baldazzi, G., et al., *Computer-aided detection of arrhythmogenic sites in post-ischemic ventricular tachycardia*. Sci Rep, 2023. **13**(1): p. 6906.
- [6] Whitaker, J., et al., *Frequency Domain Analysis of Endocardial Electrograms for Detection of Nontransmural Myocardial Fibrosis in Nonischemic Cardiomyopathy*. JACC Clin Electrophysiol, 2023. **9**(7 Pt 1): p. 923-935.
- [7] Elliott, M.K., et al., *Effect of scar and pacing location on repolarization in a porcine myocardial infarction model*. Heart Rhythm O2, 2022. **3**(2): p. 186-195.
- [8] Mendonca Costa, C., et al., *Determining anatomical and electrophysiological detail requirements for computational ventricular models of porcine myocardial infarction*. Comput Biol Med, 2022. **141**: p. 105061.
- [9] Gharaviri, A., et al., *In Vivo Analysis of Conduction Pattern Dynamics: System Development and Application Using OpenEP*, in *2022 Computing in Cardiology Conference (CinC)*. 2022.
- [10] Williams, S.E., et al., *Updates on OpenEP: The Open-Source Platform for Elec-trophysiological Data Analysis*, in *2022 Computing in Cardiology Conference (CinC)*. 2022.
- [11] Rossi, A., et al., *Transmural voltage gradient dispersion and heterogeneity in brugada syndrome phenotype - a novel workflow for advanced mapping using endocardial unipolar electrograms J-point elevation*. EP Europace, 2022. **24**(Supplement\_1).
- [12] Latrofa, S., et al., *Assessment of repolarization dispersion by using endocardial unipolar electrograms in patients with brugada syndrome and spontaneous coved-type electrocardiographic pattern*. European Heart Journal, 2023. **44**(Supplement\_2).
- [13] Latrofa, S., et al., *A novel workflow for advanced mapping of repolarization dispersion from endocardial unipolar electrograms in patients with coved-type brugada syndrome*. EP Europace, 2023. **25**(Supplement\_1).

Address for correspondence:

Dr Steven E Williams. Institute for Neuroscience and Cardiovascular Research, Chancellor's Building, 49 Little France Crescent, Edinburgh, EH16 4SB. [steven.williams@ed.ac.uk](mailto:steven.williams@ed.ac.uk)