Updates on OpenEP: The Open-Source Platform for Electrophysiological Data Analysis

Steven E Williams[1,2,3], Paul Smith[2], Ali Gharaviri[1], Christopher O'Shea[4], Adam Connolly[2], Louisa O'Neill[2], Irum Kotadia[2], Iain Sim[2], Neil Bodagh[2], Neil Grubb[3], John Whitaker[2], Matthew Wright[5], Steven Niederer[2], Mark O'Neill[2,5], Martin Bishop[2] and Nick Linton[6]

[1]Centre for Cardiovascular Science, The University of Edinburgh. [2]School of Biomedical Engineering and Imaging Sciences, King's College London. [3]NHS Lothian, Lothian, UK. [4]Institute of Cardiovascular Sciences, University of Birmingham, [5]Department of Cardiology, Guy's and St Thomas' NHS Foundation Trust. [6]Imperial College

Abstract

OpenEP is an open-source library for electrophysiological data analysis, first released in 2020. This paper provides an update on the features and tools added since initial release.

Software development has been performed in Matlab (R2020a/R2021b) and Python3. Development is ongoing in the areas of data parsing and data analysis. Data parsing: The system continues to support parsers for Carto3, Velocity and Precision. In addition, a parser for the Kodex electroanatomic mapping system has been added. Data analysis: An extensible architecture for data interpolation has been added. This new architecture has exposed hitherto unrecognized variation in interpolation schemes in clinical mapping systems and will permit the optimization of interpolation methods against 'gold standard' simulation or histological data. Similarly, an architecture for conduction velocity vector measurement has been added. We have refined the ablation lesion quantification tools permitting time-domain analysis of ablation lesion formation.

A Python-based graphical interface is now under development, with the beta-testing program for the interface planned. The interface provides the ability to visualize, manipulate and analyze electrophysiology data. To facilitate this development, we have implemented an open-source Python3 version of OpenEP: openep-py.

1. Introduction

Electroanatomic mapping data, collected during routine electrophysiology procedures encapsulates a vast amount of data describing atrial and ventricular health. Owing to its complexity, clinical studies of electroanatomic data (and to a lesser extent electrophysiology data) frequently apply qualitative or semi-quantitative techniques. Studies undertaking fully quantitative analysis of electroanatomic mapping data are frequently only deliverable in large or expert centres with available technical and engineering resources to handle this data. Moreover, owing to the size of data exported from electroanatomic mapping systems, which frequently runs to several gigabytes of data per patient, analysis of electrophysiology data at scale has not previously been possible.

To address these challenges, 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.

OpenEP was first released in December 2020 [1]. Since this time OpenEP has been under active development. This paper provides an update on the features and tools added since initial release.

2. Methods

2.1. Software development

Software development has been performed in Matlab (R2020a/R2021b) and Python3. The open-source Python version of OpenEP, `openep-py`, currently supports Python versions 3.8-3.10 and is based on `PyVista`, `Numpy`, `Scipy` and `Numba`. The graphical interface is built using `openep-py`, `PyVistaQt`, `Vedo` and `Pyside`.

For the development of `openep-py` we have followed best practices in modern software engineering. We have unit testing to ensure the accuracy of the code and use GitHub actions for continuous integration. We will soon be adding user-tutorials and releasing a package on PyPI for ease of installation.

We are also in the process of adding unit tests for functions in the Matlab version of OpenEP, `openep-`

core. Wherever possible we will rely on publicly-available datasets for testing but will perform additional local tests using confidential datasets.

OpenEP remains hosted on Github (https://github.com/openep). Within the OpenEP Github, the repository openep-core contains the Matlab version and openep-py contains the Python version. 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. Licensing

The Matlab implementation (openep-core) remains licensed under the permissive Apache-2.0 license. The Python implementation of OpenEP (openep-py) is licensed under the GNU GPL-3.0 license.

2.3. Communication

During development the OpenEP team have made use of several avenues of communication. Firstly, the OpenEP website, hosted at https://openep.io, provides general information on the project and source code documentation for openep-core. Through the website, links are provided to the git repositories and discussion forum. Secondly, a discussion forum is available at https://openep.discourse.group. The discussion forum is a publicly accessible central location for question and answers about OpenEP. Thirdly, reference code documentation for openep-py is provided via ReadTheDocs and is available at https://openep-py.readthedocs.io/en/latest/. Finally, there are actively maintained issue trackers for openep-core, openep-py and openep-gui, available through Github.

3. Results

3.1. Data parsing

Data parsers for Carto3 (Biosense Webster) and Precision (Abbott) continue to be supported and updated. Support has been added for the newer EnsiteX mapping system from Abbott. Data parsers are only currently available within openep-core and are called from within Matlab as follows, for Carto3, Precision and EnsiteX, respectively:

```
% Matlab
userdata = importcarto_mem()
userdata = import_precision()
userdata = import_ensitex()
```

Each of these functions returns a data structure containing the electroanatomic mapping data, fully described on the data structure webpage (https://openep.io/data/). Conventionally, the OpenEP data structure is stored in a variable named userdata with the following fields described in **Table 1**.

A data parser for the Kodex electroanatomic mapping system (EPD/Phillips, Best, Netherlands) is included in this development of OpenEP. This data parser is called as follows:

```
% Matlab
userdata = import_kodex()
```

An example local activation time map created using Kodex and displayed using OpenEP is shown in **Figure 1**.

Table 1. Top level fields of the OpenEP data structure.

Field	Description
`userdata.<system>Folder`	Absolute path to the import location
`userdata.electric`	Electrograms and surface ECG data exported from the electroanatomic mapping system. Includes locations, signals and annotations
`userdata.notes`	System-added notes regarding data parsing and processing. Added via the `addNote()` OpenEP function.
`userdata.surface`	Data describing the geometric cardiac chamber model together with any interpolated electro-anatomic maps created from electrical data.
`userdata.rf`	Ablation data including ablation locations, radiofrequency generator settings and quantification for example Ablation Index (Carto) or Lesion Size Index (Abbott).

Regression testing has been established for the data import parsers. To support testing as part of the Pull Request & Merge process, a repository of publicly available datasets is being established, which is hosted via Zenodo. Frequently, datasets are not suitable for public sharing due to data security issues. In such cases, local testing can be performed on data parsing modules prior to accepting any Pull Requests.

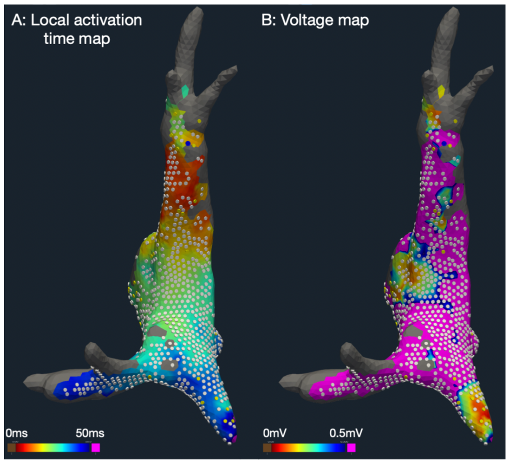


Figure 1*: Example local activation time (**A**) and voltage maps (**B**), drawn using OpenEP, following parsing of data from the Kodex electro-anatomic mapping system. Clinical data was collected using the Kodex electroanatomic mapping system during high right atrial pacing, cycle length 550ms, in a porcine right atrium.*

The ability to import all 12-lead ECG channels along with each mapping point has been added to OpenEP. The new data is stored within a three-dimensional array within the OpenEP data structure (located within `userdata.electric.ecg`). This data is only accessible where it has been exported by the electroanatomic mapping system. For example, the Carto3 mapping system currently exports all 12-leads, whereas the Kodex mapping system does not export any of the 12 leads. The required leads are selected during import allowing the user to prioritize data availability or data size depending on the specific use case.

The Python version of OpenEP, `openep-py`, does not yet support parsing data directly from electroanatomic mapping systems. However, it does support reading and writing OpenEP datasets (.mat files) as well as files from the simulation engine OpenCARP.

3.2. Data analysis

Data interpolation. An extensible architecture has been added for data interpolation with `openep-core` (**Figure 2**). The interpolation system is structured around a Matlab Class definition which provides details of the interpolation scheme employed, the relevant parameters for that scheme and the interpolation functions themselves.

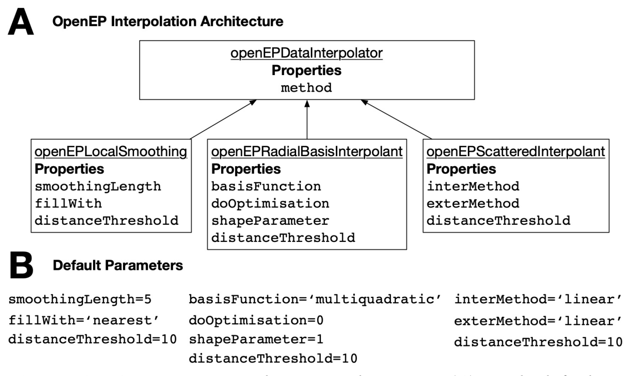


***Figure** 2. OpenEP interpolation architecture (A) and default parameters (**B**).*

Performing interpolation using this new architecture involves organizing the data for interpolation, instantiating an object of the interpolation class, and performing interpolation. Default values for the interpolation parameters are provided so that interpolation can be performed without modifying any parameters of the interpolation class. For example, interpolation of a voltage map can be performed as follows:

```
%Matlab

% organize data
vertices = getVertices(userdata);
X = getElectrogramX(userdata);
voltages = getVoltages(userdata);

% create a data interpolator
int = openEpDataInterpolator();

% perform interpolation
voltageMap = int.interpolate(X, votlages, vertices)

% draw a voltage map using the interpolated data
drawMap(userdata ...
        , 'type', 'bip' ...
        , 'data', voltageMap ...
        , 'coloraxis', [0.499 0.500] );
```

An example voltage map created using the above approach and the OpenEP example dataset 2 (available from https://github.com/openep/openep-examples) is shown in **Figure 3**.

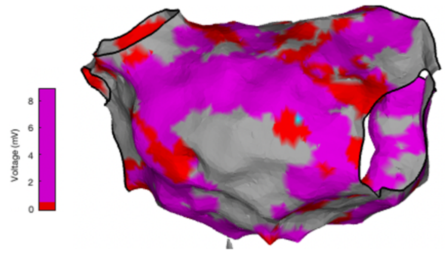


***Figure** 3. An example of voltage map interpolation using OpenEP example 2. See text for code.*

An investigation of the correlation between interpolated local activation time maps and interpolated voltage maps with the same maps exported from clinical systems was performed, showing a stronger correlation for local activation time maps than voltage maps (**Figure 4**).

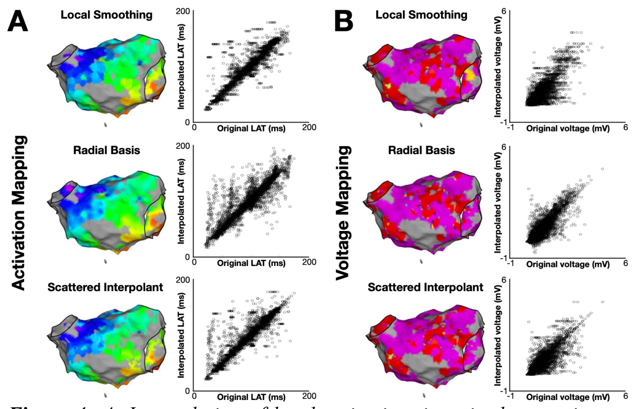


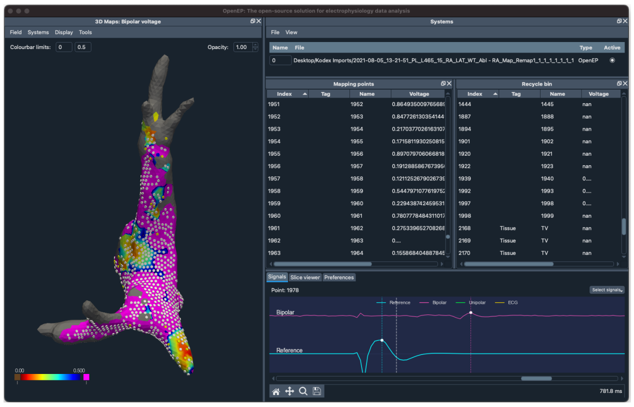
***Figure 4: A.** Interpolation of local activation times is shown using each of the three available interpolation schemes. Correlation with the system-interpolated data for each method is shown. **B.** Interpolation of voltage mapping data using the same interpolation schemes together with correlation between original and interpolated data.*

3.3. Graphical interface

To support interaction with electroanatomic mapping data, a graphical interface for OpenEP is under development (see **Figure 5**). The interface is a cross-platform desktop application for the visualization and analysis of clinical and simulated electrophysiology data. It uses `openep-py` for loading, visualizing, analyzing and exporting electroanatomic mapping data.

Multiple cases can be loaded via the system manager, facilitating qualitative and quantitative comparisons between cases. The graphical interface allows visualization of both the mapping data and the electrical data stored within OpenEP datasets. Electrogram annotations can be updated via the graphical interface and new interpolated activation or voltage maps are then created that are based on the new annotations. Tools are also provided to manipulate the geometrical cardiac chamber model, which can: remove unnecessary geometry; add anatomical structures, divide the domain into regions for segmental analysis and perform registration between datasets.

OpenEP datasets can be exported from these modifications for further analysis within `openep-core`.



Name	File	Type	Active
0	Desktop/Kodex Imports/2021-08-05_13-21-51_PL_L465_15_RA_LAT_WT_Abl - RA_Map_Remap1_1_1_1_1_1_1_1	OpenEP	●

Mapping points

Index	Tag	Name	Voltage
1951		1952	0.86493500976568
1952		1953	0.847726130354144
1953		1954	0.2170377026163107
1954		1955	0.1715811930250815
1955		1956	0.897079706066818
1956		1957	0.191288586767395
1957		1958	0.121125267902673
1958		1959	0.544797107761975
1959		1960	0.229438742459531
1960		1961	0.7807778484311017
1961		1962	0.27533965270826
1962		1963	0....
1963		1964	0.15586840488784

Recycle bin

Index	Tag	Name	Voltage
1444		1445	nan
1887		1888	nan
1894		1895	nan
1901		1902	nan
1920		1921	nan
1922		1923	nan
1939		1940	0....
1992		1993	0....
1997		1998	0....
1998		1999	nan
2168	Tissue	TV	nan
2169	Tissue	TV	nan
2170	Tissue	TV	nan

Figure 5: OpenEP graphical interface. The interface is under development with a beta testing program shortly to be launched.

4. Discussion

In this paper we present an update on the development of OpenEP since initial release. With the exception of the graphical interface, all code is publicly available under Apache-2.0 or GPL-3.0 licenses.

In our initial work we highlighted the ease of use of the OpenEP toolkit for data analysis. We have continued to adhere to this design principle in this work, such that new OpenEP functions can be called with few (or no) arguments, permitting analysis with default values to be performed easily.

Within this work we highlight two new resources to support the use of OpenEP. Firstly, a new Discussion Forum has been launched (https://openep.io/forum/). Rather than support discussion of OpenEP via generic discussion platforms such as Mathworks FileExchange or StackOverflow, we have taken the route of providing a project specific forum. Project specific forums have been proposed as a tool to optimize communication between project members, and to cultivate community and ecosystems [2]. For these reasons we were motivated to provide a project-specific forum for OpenEP. The forum has proved to be a valuable resource for supporting OpenEP users.

Secondly, we have provided a Zenodo community for sharing raw data files for input to OpenEP. This is an essential step for supporting regression testing for the data parsing modules of OpenEP. The availability of publicly accessible data for regression testing of the parsing modules is challenging owing to restrictions placed upon clinical electroanatomic mapping data. Data sets created using clinical electroanatomic mapping systems in experimental settings provides a suitable source of data not limited by these restrictions. We are establishing a library of such datasets within Zenodo which can be used for regression testing [3].

In our original OpenEP paper, we highlighted six areas where OpenEP did not currently provide an equivalent functionality to that used within contemporary literature. These included availability of the 12 lead ECG for analysis and regional analysis of atrial and ventricular electroanatomic mapping data. Each of these features is now available within this further development of OpenEP.

5. Conclusions

OpenEP is an open-source platform for electroanatomic mapping data analysis. This paper provides an update on features added to the platform since initial release.

Acknowledgments

The authors acknowledge the support of the British Heart Foundation Centre for Research Excellence Award III (RE/18/5/34216). 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. SEW acknowledges support from the British Heart Foundation (PG/19/44/34368)

References

[1] SE Williams, CH Roney, A Connolly, I Sim, J Whitaker, D O'Hare, I Kotadia, L O'Neill, C Corrado, M Bishop, SA Niederer, M Wright, M O'Neill and NWF Linton, "OpenEP: A Cross-Platform Electroanatomic Mapping Data Format and Analysis Platform for Electrophysiology Research," *Front. Physiol.*, vol. 12, pp. 646023, Feb. 2021.

[2] YS Nugroho, S Islam, K Nakasi, I Rehman, H Hata, RG Kula, M Nagappan and K Matsumoto, "How are project-specific forums utilized? A study of participation, content and sentiment in the Eclipse ecosystem," *Empir. Softw. Eng.*, vol 26, pp. 132, Sept. 2021.

[3] SE Williams. "openep-testingdata", available from: https://doi.org/10.5281/zenodo.6651600

Address for correspondence:
Dr Steven E Williams.
Centre for Cardiovascular Science, Chancellor's Building, 49 Little France Crescent, Edinburgh, EH16 4SB
steven.williams@ed.ac.uk