A Python Toolkit for Automated Electrophysiological Analysis of Microelectrode Array Recordings

Nikesh Bajaj[1,2], Kiran H K Patel[2], Danya Agha-Jaffar[2], Jesús Requena Carrión[1], Fu Siong Ng[2]

[1] Queen Mary University of London, London, United Kingdom
[2] Imperial College London, London, United Kingdom

Abstract

Microelectrode Array (MEA) systems are widely used in cardiac electrophysiology to record extracellular field potentials from cardiomyocytes with high temporal and spatial resolution. These recordings enable the analysis of critical cardiac parameters such as conduction velocity and arrhythmic events by evaluating the electrical and morphological characteristics of electrograms (EGMs) to study cardiac function and drug effects. Typically, researchers develop their own custom programs to analyse MEA recordings, often requiring frequent manual interventions that are time-consuming and error-prone, leading to a lack consistency across studies. In this paper, we present an open-source, Python-based toolkit for automated electrophysiological analysis of MEA recordings, designed to streamline data preprocessing, feature extraction, visualisation, and batch analysis. With a focus on reproducibility, transparency, and ease of use, this toolkit offers a robust alternative to manual processing. It enables cardiac researchers to efficiently analyse MEA data at scale, while allowing extension or adaptation of analysis modules to suit specific experimental needs.

1. Introduction

The contact electrograms (EGMs) and their morphology are used to investigate the electrophysiology of the myocardium for guiding the targeted treatment strategies. The Microelectrode Array (MEA, MultiChannel Systems, Reutlingen, Germany) system is used to record extracellular field potentials from cardiomyocytes with high temporal and spatial resolution [1]. An MEA plate (See Figure 1 left panel) contains 60 gold electrodes arranged in an 8×8 grid (inter-electrode distance: 700 μm), excluding the corner electrodes. Cells (or tissues) are electrically stimulated with a biphasic impulse ($\pm$1000 mV for 2 ms) delivered as a train at a rate of 1 Hz from a selected pacing site (e.g. North). Typically, a 10-second recording is obtained from each electrode at a sampling frequency of 25 kHz [2]. The voltage, duration, and frequency of the stimulus can be adjusted according to the experimental design. The electrophysiological properties of cells and tissues under various treatments and conditions are then analysed from multiple MEA recordings.

To analyse MEA recordings, researchers often develop their own in-house software and programmes, which typically require manual intervention for labelling and signal processing. These approaches are time-consuming and prone to human error, resulting in a lack of consistency and reproducibility across studies. In this paper, we present an open-source Python toolkit that streamlines the processing and analysis steps for MEA recordings. The toolkit automatically localises stimuli, extracts cycles and aggregated EGMs, computes electrophysiological features from the recordings, and generates effective visualisations. For electrophysiological analysis, the tool computes activation time (AT), conduction velocity, activation flow, and morphological features of EGMs. The processing pipeline also identifies poor-quality channels and EGMs using three configurable criteria. Furthermore, the pipeline supports the computation of repolarisation time (RT) and produces summarised tables in spreadsheet format for subsequent statistical analysis.

The developed toolkit provides a single-line command to perform a complete analysis of an MEA recording. At the same time, each processing step is fully customisable, allowing researchers to follow individual steps and modify implemented methods. This enables researchers to efficiently analyse MEA recordings at scale, with an emphasis on reproducibility, transparency, and ease of use.

2. Processing Steps

MEA systems typically produce recordings in HDF format, though some devices use BDF, which can be converted via Multi Channel DataManager software [3]. Using Python, the full multi-channel recording (i.e. 60 channels) is imported as an array (Figure 1, middle). The tool then executes a series of processing steps to compute electrophysiological features, as described below.

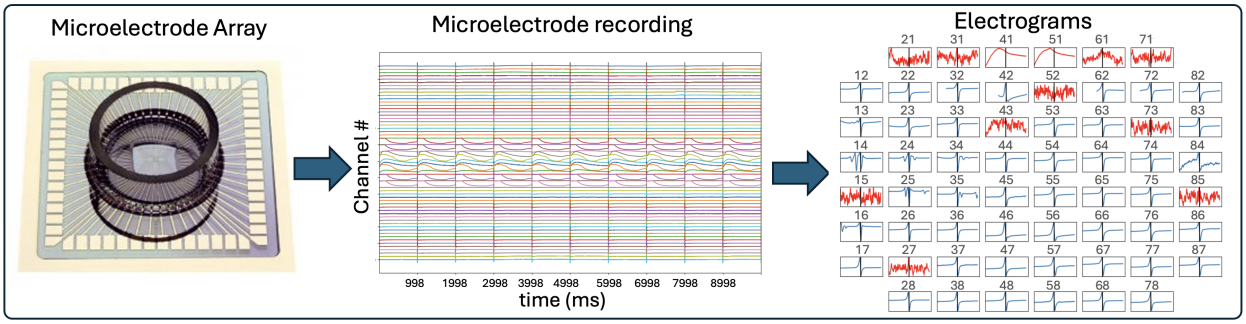

Figure 1: Overview: An MEA plate (left), a 10-second MEA recording (middle), extracted EGMs (right).

2.1. Stimulation Localisation

A biphasic stimulus ($\pm$1000 mV for 2 ms) has a sharp deflection in middle, while transition from -1000 mV to +1000 mV. This sharp deflection is identified by $max\{dv(t)/dt\}$ (maximum positive slop) as depicted in panel A of Figure 2. Given the frequency and voltage of stimuli, the train of stimulus locations is detected using one channel (default electrode 47, avoid pacing or ground electrode). For adapting to different stimulus, such as unipolar, alternative options (i.e. $min\{dv(t)/dt\}$, $max\{|dv(t)/dt|\}$) are provided in Python function *get_stim_loc*.

2.2. Aligning and Aggregating Cycles

After the locations of each stimulus across all channels are identified, all cycles are aligned with the corresponding stimulus for each channel. While aliging the cycles, the first 2 ms of the recording immediately following $max\{dv(t)/dt\}$ are excluded. Depending on the stimulus frequency, the duration of the recording (e.g. 500 ms for 1 Hz) is selected for alignment, as shown in panel A of Figure 2. For an incomplete final cycle, NaN values are padded. The durations for exclusion and inclusion of the signal can be adjusted as required. Once all cycles are aligned across channels, an averaged cycle for each channel is produced by computing the arithmetic mean. Averaging cycles reduces noise and yields clearer electrograms. By default, the final cycle is excluded (this can be modified), as an incomplete cycle may introduce artifacts and distort EGMs. Although cycle aggregation is generally considered good practice, the tool also allows researchers to select individual cycles to facilitate the analysis of electrophysiological changes across cycles.

2.3. Activation Time

For the contact electrograms (EGMs), AT is defined as the time of the maximum negative slope ($\max\{-dv(t)/dt\}$). Following electrical stimulation of cells or tissues, there exists a specific duration (depending on the study and treatment) within which an electrogram is expected to occur. Given this expected range (e.g. within 50 ms), an electrogram is identified by computing $\max\{-dv(t)/dt\}$, and this time point is labelled as the local activation time (LAT) of the electrode.

2.4. Repolarisation Time

Similar to AT, RT is defined by the time of the maximum deflection of the T-wave electrogram; however, either a positive or a negative deflection may be considered for RT (the alternative and Wyatt methods). As RT follows AT, the expected range for RT excludes the already computed AT. Once RT is determined, the Action Potential Duration (APD) for each channel can be calculated using $APD = RT - AT$. The T-wave in MEA recordings, depending on the cells (or tissues), is often very weak and noisy, which can lead to inaccuracies in its computation.

2.5. Extracting EGMs

With identified ATs, EGMs for each channel are extracted over a defined duration (e.g. 5 ms). Each EGM is then processed to remove drift using the Savitzky–Golay filter, as shown in Figure 2 panel C. The EGM duration and parameters for the processing step can be adjusted.

2.6. EGM Feature Extraction

From the collection of preprocessed EGMs, a predefined set of morphological features is extracted for each channel. For the purpose of computation, the first left peak is defined as p_{l_1} and the first right peak as p_{r_1}, occurring before and after LAT, respectively (indicated in Figure 2, panel B,). The following morphological features are computed from each EGM:

- **Peak-to-peak voltage**: The total voltage range is computed as the difference between peaks p_{r_1} and p_{l_1} (i.e.

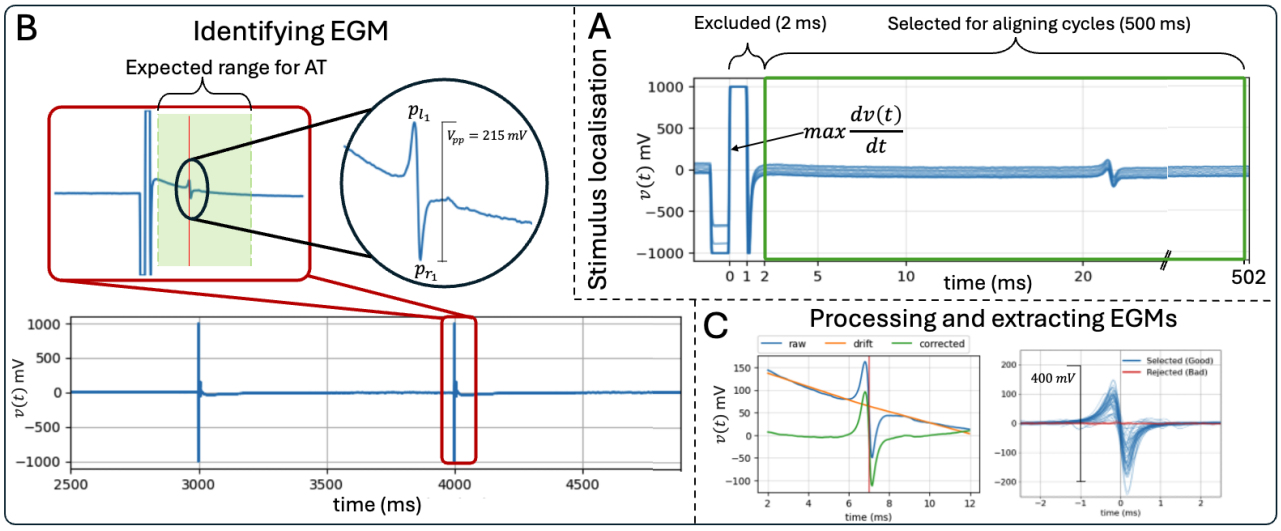

Figure 2: Stimulus and EGM Processing: (A) Stimulus localisation with $max\{dv(t)/dt\}$ and aligning cycles. (B) Identifying EGM with expected range. (C) Extracting and processing EGMs

$V_{pp} = |p_{l_1} - p_{r_1}|$). Compared to $max-min$, this approach avoids unusually high peaks within the selected duration. The voltage range of EGM depends on the cells and tissue.

- **Fractionation Index**: The Fractionation Index is computed from the number of peaks crossing a pre-defined voltage threshold (V_{T_f}) after LAT (e.g. p_{r_2}, p_{r_3}, labeled as fractionating peaks). A Fractionation Index of 1 indicates that no fractionation is detected, whereas a value of 2 indicates one peak crossing the defined threshold after LAT i.e. $p_{r_2} > V_{T_f}$ [1].
- **EGM duration**: Two approaches are used to compute EGM duration: (i) based the width of p_{l_1} and p_{r_1} peaks (ii) incorporating fractionating peaks (i.e. p_{r_2}, p_{r_3}), in which case EGM duration is longer.
- **Energy and dispersion**: The average energy of EGM is computed as $1/T \int_T |v(t)|^2 dt$. The standard deviation of EGM voltage is calculated as voltage dispersion.

Although the thresholds and parameters for the above computations are adjustable, the tool allows users to extract the EGMs and perform custom computations for new features. The feature values for each electrodes can be visualise as MEA-grid (See Figure 3, panel A).

2.7. Identifying Bad Channels and EGMs

The tool automatically identifies poor-quality channels and EGMs, based on three criteria with configurable thresholds.

1. **Based on Stimuli**: After a stimulation pulse, the voltage recorded at the electrodes typically returns to resting state (baseline) and continue to record the electrical activity of cells (or tissue) (See Figure 2 panel A). However, for some electrodes, specifically from pacing sites, the voltage does not return to baseline, resulting in corrupted recordings. A channel with average peak duration above the threshold (i.e. 2 ms) is labeled as a bad channel.
2. **Based on Activation Time**: In case of noisy channel or absence of a valid EGM, an activation time point is identified at random locations. A channel with unusual activation time (e.g. out of expected range) is labeled as a bad channel.
3. **Based on EGM voltage**: Typically, in noisy channels, the estimated EGMs exhibit very low voltages resembling background noise. An EGM with an unusually low peak-to-peak voltage (below the threshold, e.g. 5 mV) is labelled as a bad channel (See Figure 2 panel C, right).

Although the three criteria above identify the majority of poor-quality channels and EGMs, a list of *Good* and *Bad* channels can also be explicitly provided to prevent misclassification. In Figure 1 right panel, EGMs corresponding to the identified bad channels are shown in red.

2.8. Activation and Conduction Maps

The conduction velocity $CV = (\delta_x^2 + \delta_y^2)^{-1/2}$ and the direction of activation flow $\theta = \tan^{-1}(\delta_y/\delta_x)$ are computed using the gradient of local activation times; δ_x and δ_y, in x and y directions, respectively. The gradient is computed using finite differential normalised by dividing with inter-electrode distance. The activation and conduction maps are produced after dropping the values corresponding to bad channels and applying interpolation. In Figure 3, right-most of panel A, shows the matrix of LATs. The conduction and activation maps (Figure 3, panel B) are computed after excluding the values above 50 ms (bad

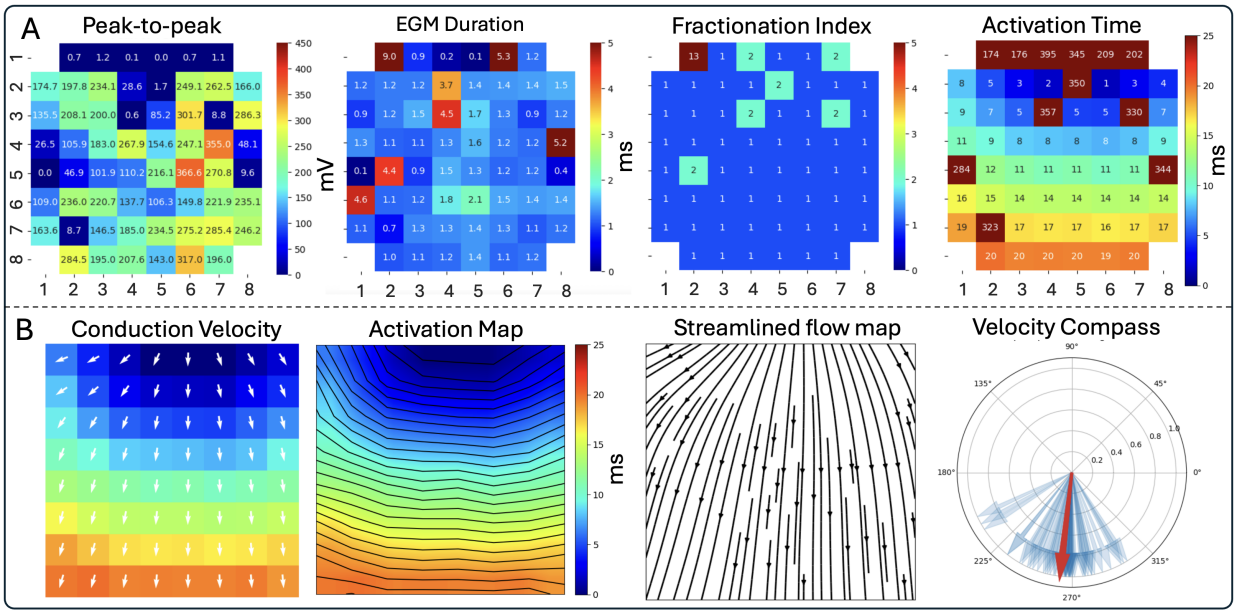

Figure 3: Visualisations: Panel A shows visualisation of features and panel B shows visualisation of maps.

channels) and re-estimating using interpolation. Additionally, a streamlined flow map and a velocity compass is produced to indicate the direction of the activation flow.

2.9. Summary Reports and Visulisations

A report for an MEA recording is generated by summarising the values of electrophysiological features, excluding bad channels. A detailed report of each feature for every channel is also produced for subsequent analysis. The tool provides functionalities to customise the visualisations illustrated in Figure 3 for reporting purposes.

3. Resources: Documentation and Demo

The proposed toolkit is shared as a module of an open-source Python library; *SpKit* [4], with full documentation - https://spkit.github.io/modules/bsp_mea and detailed APIs with doc-strings. For demonstration, two example scripts, including a sample MEA recording, are shared under *examples* section, which can be executed on the interactive cloud-based environment (*binder*). The demo scripts include one-line command and step-by-step processing.

4. Conclusion

In this paper, we introduce an open-source Python toolkit for the automated electrophysiological analysis of MEA recordings. The toolkit streamlines all processing and analysis steps, enabling researchers to efficiently analyse multiple MEA recordings with an emphasis on consistency, reproducibility, and transparency across the studies. The tool also offers features to customise processing steps and adapt to new experimental settings. Full documentation, detailed APIs, and example demonstrations are provided to ensure the ease of use.

References

[1] Chowdhury RA, Tzortzis KN, Dupont E, Selvadurai S, Perbellini F, Cantwell CD, Ng FS, Simon AR, Terracciano CM, Peters NS. Concurrent micro-to macro-cardiac electrophysiology in myocyte cultures and human heart slices. Scientific Reports 2018;8(1):6947.

[2] Reppel M, Pillekamp F, Lu ZJ, Halbach M, Brockmeier K, Fleischmann BK, Hescheler J. Microelectrode arrays: a new tool to measure embryonic heart activity. Journal of Electrocardiology 2004;37:104–109.

[3] Multi Channel Systems MCS GmbH. Multi Channel DataManager, Version: V1.14.10, 2023-02-21. URL https://www.multichannelsystems.com/software/multi-channel-datamanager.

[4] Bajaj N, Requena Carrión J. SpKit: Signal Processing Toolkit, 2024. URL https://spkit.github.io. Version 0.0.9.7, https://spkit.github.io.

Address for correspondence:

Nikesh Bajaj
G.O. Jones Building, Queen Mary University of London, Mile End, E14NS, London, UK
nikesh.bajaj@qmul.ac.uk