



# MEAnalyzer – a Spike Train Analysis Tool for Multi Electrode Arrays

Raha M. Dastgheyb<sup>1</sup> · Seung-Wan Yoo<sup>1</sup> · Norman J. Haughey<sup>1,2</sup>

Published online: 4 July 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

Despite a multitude of commercially available multi-electrode array (MEA) systems that are each capable of rapid data acquisition from cultured neurons or slice cultures, there is a general lack of available analysis tools. These analysis gaps restrict the efficient extraction of meaningful physiological features from data sets, and limit interpretation of how experimental manipulations modify neural network activity. Here, we present the development of a user-friendly, publicly-available software called MEAnalyzer. This software contains several spike train analysis methods including relevant statistical calculations, periodicity analysis, functional connectivity analysis, and advanced data visualizations in a user-friendly graphical user interface that requires no coding from the user. Widespread availability of this user friendly and mathematically advanced program will stimulate and enhance the use of MEA technologies.

**Keywords** Neuron · Electrophysiology · MEA · Multi-electrode array · Functional connectivity · Spike trains · Spike analysis · Burst analysis · Data visualization

## Introduction

Multi-electrode arrays (MEA) are powerful electrophysiological assay systems typically used to measure extracellular electrical activity from groups of cells in tissue culture and slice preparations (Gross and Schwalm 1994; Canepari et al. 1997; Pine 1980; Novak and Wheeler 1988). While the ability to make measurements of intracellular action potentials from single cells has been around since the 1940s (Hodgkin and Huxley 1945), MEA experiments are much more recent, with the first published paper in 1972 (Thomas Jr. et al. 1972). There are now several different manufacturers that offer MEA systems that are each capable of microsecond recording

intervals and ultimately result in an immense amount of data. For example, a 60-electrode MEA making voltage measurements at 25 kHz for 5 min will result in 7,500,000 data points. There is vast potential in the ability to collect so much data, but the ability to correctly and reproducibly analyze this is a limiting factor in moving the field forward. As the field grows, so does the importance of accessible analytical tools and the ability to make observations from the data to guide future studies.

Many labs design their own code to handle this complex data, but not all are released to the scientific community. This reduces experimental reproducibility and transparency of analysis (Schofield et al. 2009). A comparison of available analysis software is available in supplemental Fig. 1 (Pastore et al. 2016; Huang et al. 2008; Bongard et al. 2014; Wagenaar et al. 2005; Hazan et al. 2006; Meier et al. 2008; Goldberg et al. 2009; Kwon et al. 2012; Lidierth 2009; Somerville et al. 2010; Bokil et al. 2010; Vato et al. 2004; Cajigas et al. 2012; Mahmud et al. 2012; Mahmud et al. 2014; Bologna et al. 2010; Novellino et al. 2009; Egert et al. 2002; Georgiadis et al. 2015). NeuroExplorer®(Nex Technologies, Littleton, MA) is currently the most advanced software package available for spike train analysis. It is compatible with most commonly used data acquisition systems but is not available free of charge and lacks advanced data visualizations. Of the tools

---

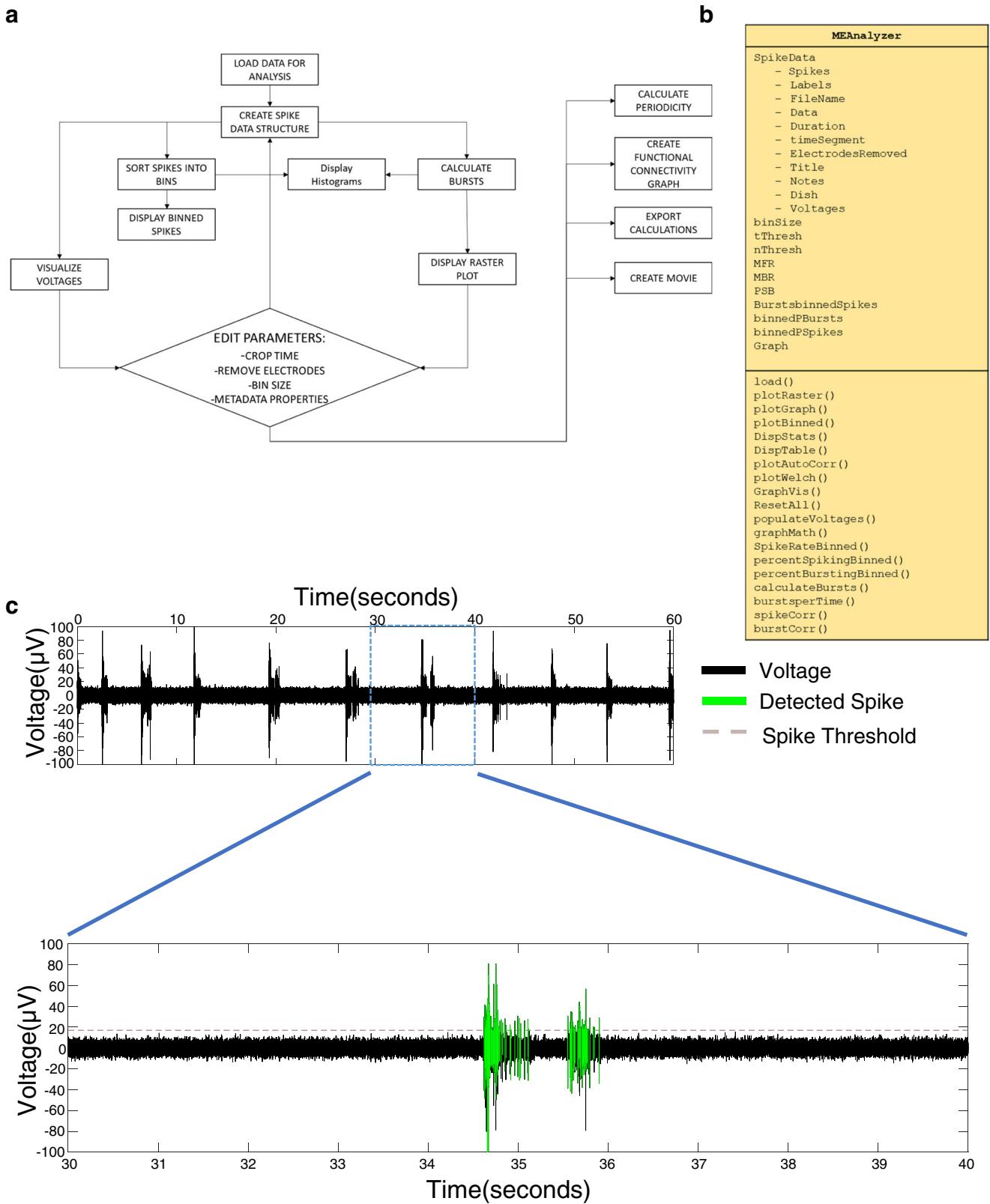
**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s12021-019-09431-0>) contains supplementary material, which is available to authorized users.

---

✉ Raha M. Dastgheyb  
rdastgh1@jhmi.edu

<sup>1</sup> Department of Neurology, The Johns Hopkins University School of Medicine, 600 N. Wolfe Street, Baltimore, MD 21287, USA

<sup>2</sup> Department of Psychiatry, The Johns Hopkins University School of Medicine, 600 N. Wolfe Street, Baltimore, MD 21287, USA



that have been released to the public, we are not aware of any that combine all relevant statistical calculations, periodicity analysis, functional connectivity analysis, and advanced data visualizations in a user-friendly graphic user interface (Supp

Fig. 1). To address this need, we developed MEAnalyzer specifically for the analysis of spike trains. Widespread availability of this user-friendly and mathematically advanced program will stimulate and support the use of MEA technology.

**Fig. 1** MEAnalyzer Workflow and Class Diagram. **a** Simplified MEAnalyzer flow diagram. The user begins by loading spike timestamp trains from either an exported HDF5 file, or a saved file from MEAnalyzer. The spike trains and any user-inputted variables are used to create a *SpikeData* structure. Plate-wide calculations are made based on default settings and displayed to the user to edit calculation parameters. Final calculations are then made and can be exported. **b** MEAnalyzer class diagram of major objects and classes. **c** Example 60 s voltage trace (with 10 s zoomed view) from one electrode displaying a spike detection threshold of 5 SD above baseline voltage and the resulting detected spikes

We designed MEAnalyzer to allow users without coding expertise to reproducibly analyze spike train data from a variety of MEA layouts, and to quickly produce publication-quality visualizations to convey meaningful information from the data. We give the user multiple options to change analysis and visualization parameters so that they can explore their data from various angles. Visualization is an important scientific tool for research and investigation (Earnshaw and Wiseman 2012). Visualizations allow the user and their audience to prepare, validate, communicate, interpret, and verify data (Brodlie et al. 2012). This is especially important when the data is as immense and complicated as it is with MEA data. At the same time, we acknowledge that there are certain pitfalls that come when analyzing large data sets. In order to put these into perspective and to help guide the user in making informed decisions about analysis parameters and sources of error, we have included simulations that provide insight into how analytical results change with different data and analysis parameters.

## Materials and Methods

### Software Implementation

MEAnalyzer was implemented using MATLAB 2019a, a high-level programming language with widely used analysis packages and a command interface. MEAnalyzer was coded and compiled for dissemination on Windows PCs running Windows 7, 8, 8.1, or 10 with any Intel or AMD ×86–64 processor and OpenGL support. The MATLAB compiler allows for distribution of the application to users without the requirement of a MATLAB license along with free installation of the MATLAB Runtime environment, which is a standalone set of shared libraries that allows for the execution of compiled MATLAB applications. Upon installation of MEAnalyzer (25 megabytes), the MATLAB runtime will be automatically downloaded and installed (1.4 Gigabytes). MEAnalyzer uses an Object Oriented Programming (OOP) design, which allows for future expansion (Micallef 1987). The class diagram and workflow for this programming is shown in Fig. 1.

## Source Data and Preparation for Analysis

Within MEAnalyzer the user has several options to load spike train data, to control analysis parameters, and to adjust data visualization modes. The user may either load spike train data from an HDF5 file, a previously saved MEAnalyzer file, or a formatted text file with spike train time stamps. The MEAnalyzer program is currently optimized for analyzing spike data from a variety of MEA plates, including MultiChannel Systems(MCS) 8 × 8, 6 × 10, 60-4Q, and 356 electrode arrangements. An external MEA software interfacing with the hardware, such as MC\_Rack, performs voltage measurements and identifies spikes as time stamps where the voltage exceeds a threshold (Fig. 1c). If using MC\_Rack, MCS' DataManager can be used to export spike time stamps and voltages to an HDF5 file, a generic scientific format useable on all platforms and designed for the management of large data (Teeters et al. 2015). MEAnalyzer then automatically extracts relevant information and metadata to create a data structure called *SpikeData* which contains the following fields: *Spikes*, *Triggers*, *Labels*, *Date* (of the original recording), *FileName* (of loaded file), *Duration*, *TimeSegment*, *StimulatedElectrodes*, and optional *Voltages*. It also creates the following empty fields which the user may modify: *ElectrodesRemoved*, *Title*, *Notes*, and *Dish*. Alternatively, the user may also open a previously saved file from MEAnalyzer that contains the *SpikeData* structure. Spikes are loaded as time stamps for better speed and efficiency, and to reduce disk and memory usage. The *Spikes* field consists of a {1xn} cell array where n is the number of electrodes, and each cell is a [1xm] array containing the time stamps of the m spikes detected for that electrode.

When a file is loaded, the binned spike rates are displayed along with a raster plot with optionally indicated bursts. The user has the option to crop the time segment to be included for analysis, adjust the *binSize* (the length of time during which spikes are summed), and can remove “bad” electrodes from analysis. The user may deem an electrode inadequate for analysis if it is damaged, inactive, or has inappropriately detected spikes (i.e. spike detection is noisy and inconsistent). If voltage data was included in the uploaded file, users can view voltages and spikes on the *Voltage* tab as a visual aid in determining which electrodes to exclude from analysis. The *Analyze* tab also displays preliminary calculations which can aid the user in deciding which electrodes to include in analysis.

### Raster Plot Visualization Tools

The raster plot is a visual display of spike activity, where the x-axis is time and each point on the y-axis represents an individual electrode. A vertical line is plotted at each time point where a spike has been detected for that electrode. The

creation of raster plots from MEA experiments that exhibit exceptionally rapid and constant spike activity will be slower, as the performance of plotting raster data suffers due to the amount of input data being graphically analyzed (even with the most up to date drivers and graphics engines). MATLAB will continue to plot individual vertical lines for each spike, even when vertical lines are visually indistinguishable as individual lines. To overcome this problem, we used a single loop (sequence of steps that are iterated repeatedly) to create a primitive line object with separators that can be called with a single plot function, instead of using nested loops to plot each vertical line in the raster. This approach speeds the graphic plotting of complex raster data. Raster plot visualization allows for a quick overview of all the spike data during a user-defined experimental time-frame.

### Binned Spike and Burst Frequencies

Overall plate activity is calculated and displayed by looping through all the spike time stamps and binning time stamps into user-defined time windows that are defined by *binSize*. *binnedSpikes* are converted to instantaneous firing rates by dividing the summed total of each bin by the time window. Alternatively, network bursts (*binnedPSpikes*) can be displayed as the percent of electrodes firing in each time bin. The binned spike plot allows for a visualization of entire plate activity and is a summary of all electrodes.

### Defining and Calculating Network Burst Activity

Bursts are calculated according to thresholds set by the user and are defined as a series of at least *nThresh* spikes occurring in *tThresh* time (Cotterill et al. 2016). The bursts are stored as a cell array with each burst in the format of [*bStart\_time*, *bEnd\_time*, *numberofSpikes*], which allows for calculations of burst length and identification of overlapping bursts. The cell array of bursts identified is saved as a global variable that can be accessed by other functions to save processing time and to eliminate redundant calculations. Unless otherwise specified, all bursts shown in this manuscript are defined as at least 4 spikes occurring within a 0.1 s time window.

### Spike Train Calculations

The imported spike trains can be mathematically described as a regular stochastic point process which consists of a series of events (Snyder 1991). The *MFR* (Mean Firing Rate), or spike rate, for each electrode is defined as the temporal average of spikes for that electrode:

$$MFR_i = \frac{n_i^{spikes}}{\text{TimeSegment}(2)-\text{TimeSegment}(1)}$$

where  $i$  is the index (position in the  $\{1xn\}$  array) of the electrode and  $n_i^{spikes}$  is the number of spikes detected on that electrode. The function *calcMFR* (*SpikeData*) returns an array in which each index contains the MFR of the corresponding electrodes. Similarly, the function *calcMBR* (*SpikeData*, *tThresh*, *nThresh*) returns an array where each index contains the MBR, or Mean Burst Rate, of that electrode.

### Interval Histograms

The interval of time between events (spikes or bursts) has traditionally been used to characterize spike trains (Johnson 1978). MEAnalyzer includes options for creating histograms based on inter-spike and inter-burst intervals, in addition to histograms of burst lengths, and the number of spikes in each burst. Axis limits and bin sizes are all fully adjustable according to the user's specifications.

### Evaluation Tools for Periodicity and Oscillatory Behavior

Spontaneous neuronal firing in mature neurons is known to occur in a periodic fashion, meaning that they occur in patterns of repeated intervals (Kamioka et al. 1996; Llinas 1988). Periodicity can be evaluated using autocorrelation or Welch's periodogram. These measurements are calculated based on binned measurements of user chosen events as network bursts (% of electrodes Spiking), spike rate, or the percent of electrodes bursting.

Autocorrelation measures the similarity between a signal and a copy of itself that has been shifted by a time lag. If the signal demonstrates periodic behavior there will be a peak at the lag that represents the cycle length. The normalized autocorrelation function  $r(\tau)$  measures the probability that the next event will occur at time  $t + \tau$ .

$$r(\tau) = \frac{\sum_{t=1}^{T-1} (y_t - \bar{y})(y_{t+\tau} - \bar{y})}{(T-1)Var(y)}; \tau = 0, \pm 1*fs, \pm 2*fs, \dots$$

where  $fs$  is the sampling frequency ( $binSize^{-1}$ ) and  $\tau$  is the time lag.

As an alternative to autocorrelation, the power spectrum of burst events is comparable to the Fourier transform of the autocorrelation function (Wiener 1949). We used Welch's power spectral density estimate correction instead of a standard power spectrum, as it reduced variance of the periodogram by breaking the time series into overlapping segments, computes a modified periodogram for each segment, and then averages the segments for an

estimate of the power spectral density (PSD) (Welch 1967). The PSD reports how much of expected signal power is at each frequency and is represented as a function of the frequency by:  $P_{xx}(f) = \frac{1}{f_s} \sum_{m=-\infty}^{\infty} R_{xx}(m) e^{-j2\pi mf/f_s}$ . Peaks in the power spectrum correspond to the repeating periodic intervals. Calculating periodicity in this manner allows for identification of multiple periodic frequencies.

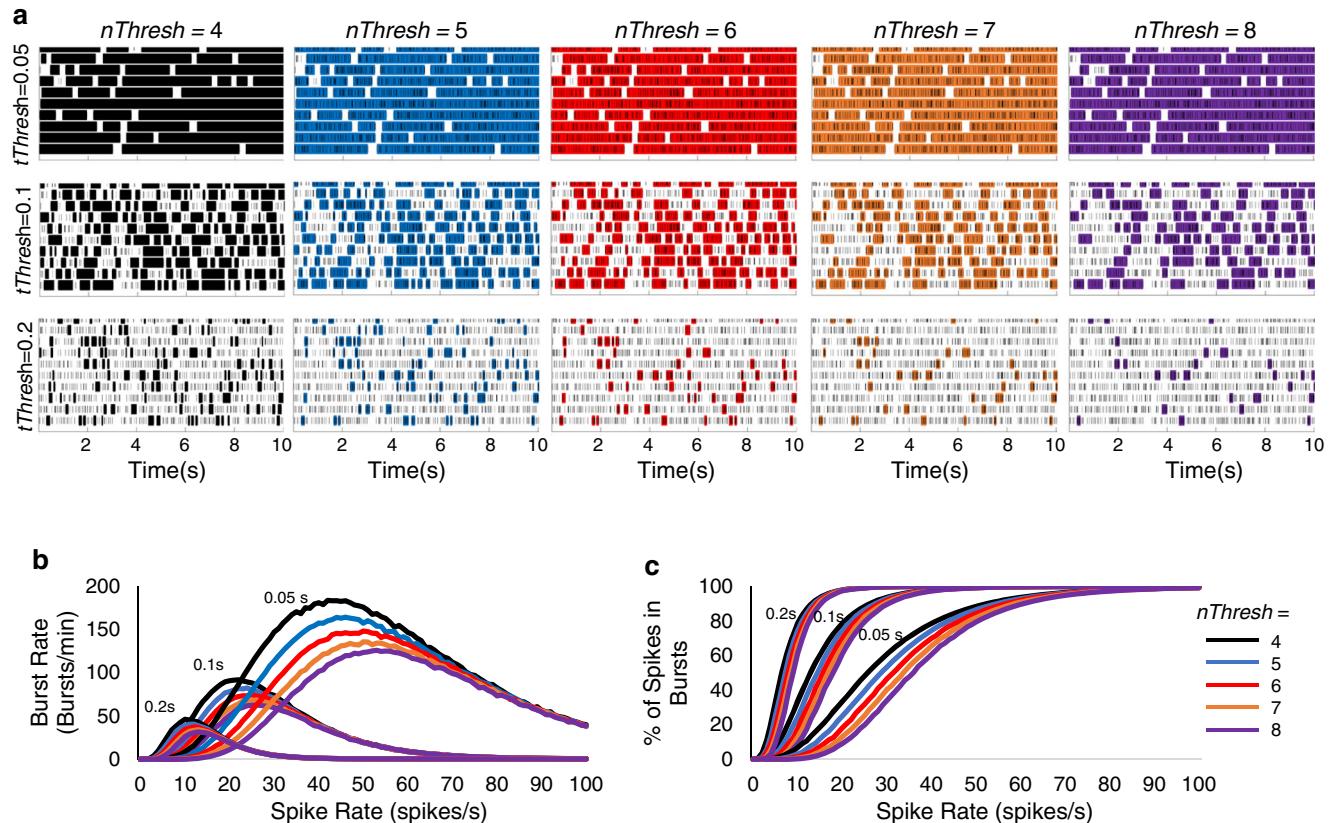
## Analysis and Visualization of Functional Connectivity

Spontaneous neuronal activity largely occurs in a synchronized and connected fashion in mature neurons (Kamioka et al. 1996; Chiappalone et al. 2006; Maeda et al. 1995). To map and visualize the spatial organization of neural network connectivity we created functional connectivity graphs using conventional graph theory where each electrode is considered as a node. Network graphs for functional connectivity were generated using adjacency matrices formed by creating undirected edges between electrodes that showed co-activation (Poli et al. 2015). MEAnalyzer allows the user to define co-activation as either (a) overlapping bursts, (b) cross-correlation of bursts, or (c) cross-correlation of spikes. With

overlapping bursts (method a), an edge is created between two electrodes if they have at least one overlapping burst. Cross-correlations (Wiegner and Wierzbicka 1987) of bursts or spikes (methods c and d) indicate synchronicity by creating an edge between two nodes if at a lag time( $\tau$ ) of 0, the average of the normalized cross-correlations for b) bursts or c) spikes has a peak that is greater than the user-defined threshold (Knox 1981; Gerstein 2000). For example, a cross-correlation of 0.5 indicates a greater than 50% chance of two nodes exhibiting simultaneous bursts or spikes. Cross-correlation of the time series  $y_1$  and  $y_2$  at time lag  $\tau$  for a recording of length T is defined as (Salinas and Sejnowski 2001):

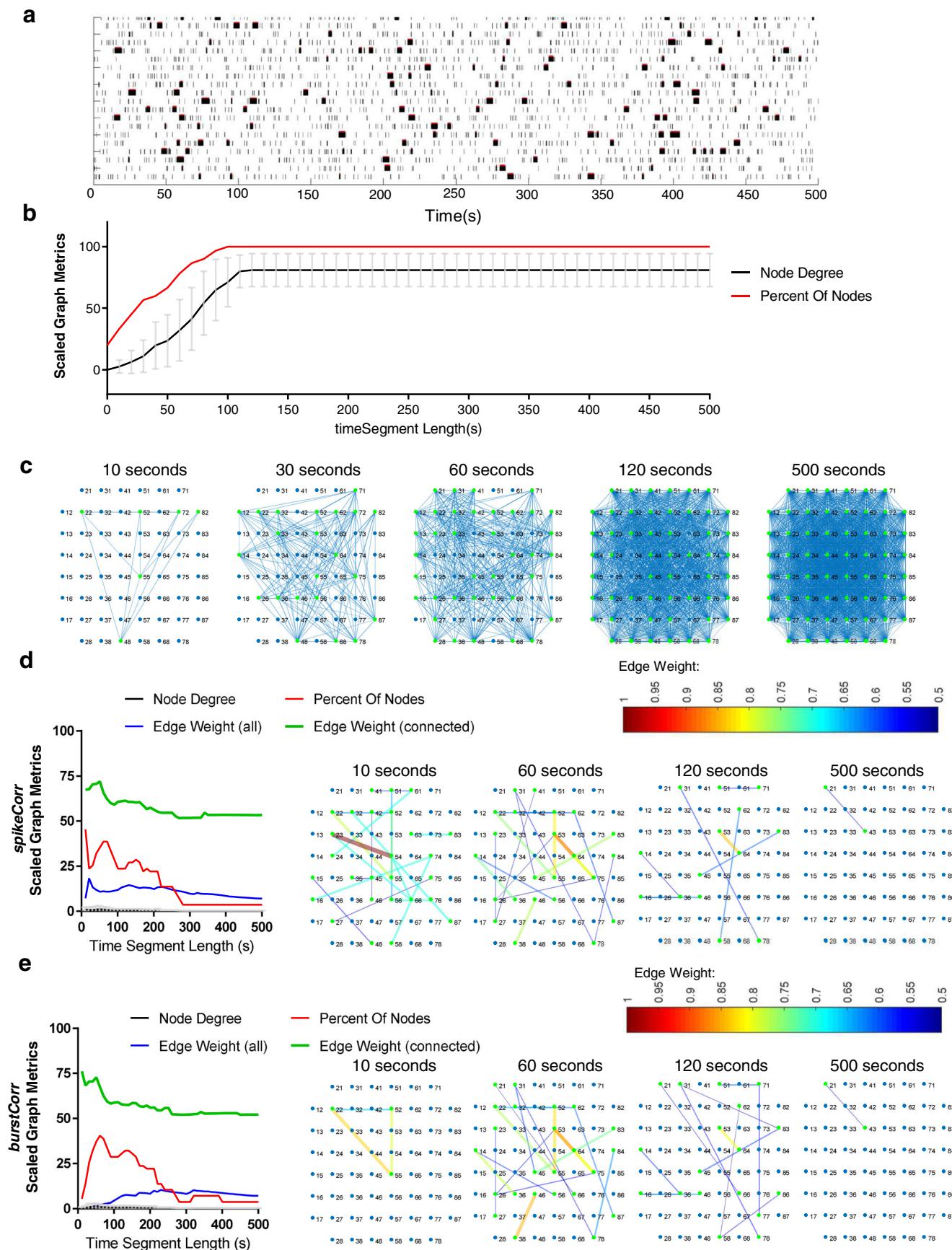
$$C_{y_1 y_2}(\tau) = \frac{1}{T} \sum_{t=1}^{T-\tau} (y_{1,t} - \bar{y}_1)(y_{2,t+\tau} - \bar{y}_2); \tau = 0, \pm 1, \pm 2, \dots$$

Once network graphs are created, we use several graph metrics to evaluate network properties: Node degree, global efficiency, percent of electrodes, and percent of edges. Node degree is defined as the total number of connections for each node, expressed as a percentage of total nodes (Poli et al. 2015). Global efficiency is defined as  $E(G) = \frac{2}{n(n-1)} \sum_{i \neq j} \frac{1}{d(i,j)}$



**Fig. 2** Burst Rate Changes with Varying Burst Parameters. **a** Representative raster plots of spikes (black lines) and bursts (box overlay) of bursts calculated with changing  $n_{Thresh}$  (columns) and  $t_{Thresh}$  (rows) using 10 s segments from 10 channels of the same 120 s

random spike train of 5 spikes/s. **b, c** Simulation results for average burst rate and the percent of spikes in bursts detected using random spike trains with changing spike rates and varying  $n_{Thresh}$  and  $t_{Thresh}$



**Fig. 3 Effect of TimeSegment length on Connectivity Graphs** **(a)** Representative raster plot of 10 electrodes over a 500 s time segment. The random spike train was simulated with a given burst rate of 1 burst/min and spike noise added at a rate of 0.5 spikes/s. **b** Simulation results for the percent of nodes included in the graph and average node degree of a connectivity graph based on overlapping bursts for a random spike train with increasing time segment lengths. **c** Representative connectivity graphs for different time segment lengths where the graph edges are based on overlapping bursts. **d** Connectivity graph metrics and representative connectivity graphs for different time segment lengths ranging from 10 to 500 s where the graph edges are based on  $spikeCorr \geq 0.5$  (**e**) Graph metrics and representative connectivity graphs for different time segment lengths ranging from 10 to 500 s where the graph edges are based on  $burstCorr \geq 0.5$

where  $d(i,j)$  is the shortest path length between  $i$  and  $j$  (Latora and Marchiori 2001). Other metrics include the percent of electrodes included as nodes ( $\frac{n_{nodes}}{n_{electrodes}}$ ) and the percent of possible edges ( $\frac{n_{edges}}{\frac{n_{electrodes}(n_{electrodes}-1)}{2}}$ ).

Once the connectivity graph is calculated it is visually displayed to the user with nodes placed at the topographical location of the electrode array. MEAnalyzer allows user-control of visualization methods and color schemes to incorporate one or more of the metrics into the visual display by changing size or color to represent the desired measurement. Node size and color can be adjusted to represent either the node degree, the spike rate, or the burst rate of a given electrode. If a cross-correlation method was used to calculate the connectivity graph, the edge width and color can be adjusted to represent the cross-correlation between functionally connected electrodes. This allows for visual analysis and interpretation of how neuronal populations on each electrode are functionally related to each other.

### Spatio-Temporal Visualizations

We added a tool to visualize temporal spike activity for each electrode in conjunction with spatial organization mapping. This allows the user to export binned spike rates for each electrode in a movie format to visualize spike intensities and patterns of spike activity over a user defined period of time. It does so by binning the spikes of each electrode into time intervals defined by the user and then creating a movie in which each frame consists of a 3-dimensional bar graph which a bar for each electrode where the x- and y- coordinates correspond to the topographical location of the electrode and the height of the bar corresponds to the number of spikes in that time bin.

A movie tool was also created for visualizing changes in functional connectivity for graphs that were based on overlapping bursts. The user designates a time interval for the functional connectivity graphs as well as a time shift for the segments. Each frame of the movie displays a raster plot where the current time segment is highlighted. Above it is displayed

the functional connectivity graph based on overlapping bursts of the highlighted segment. In each following frame, the time segment is shifted over by the designated shift time.

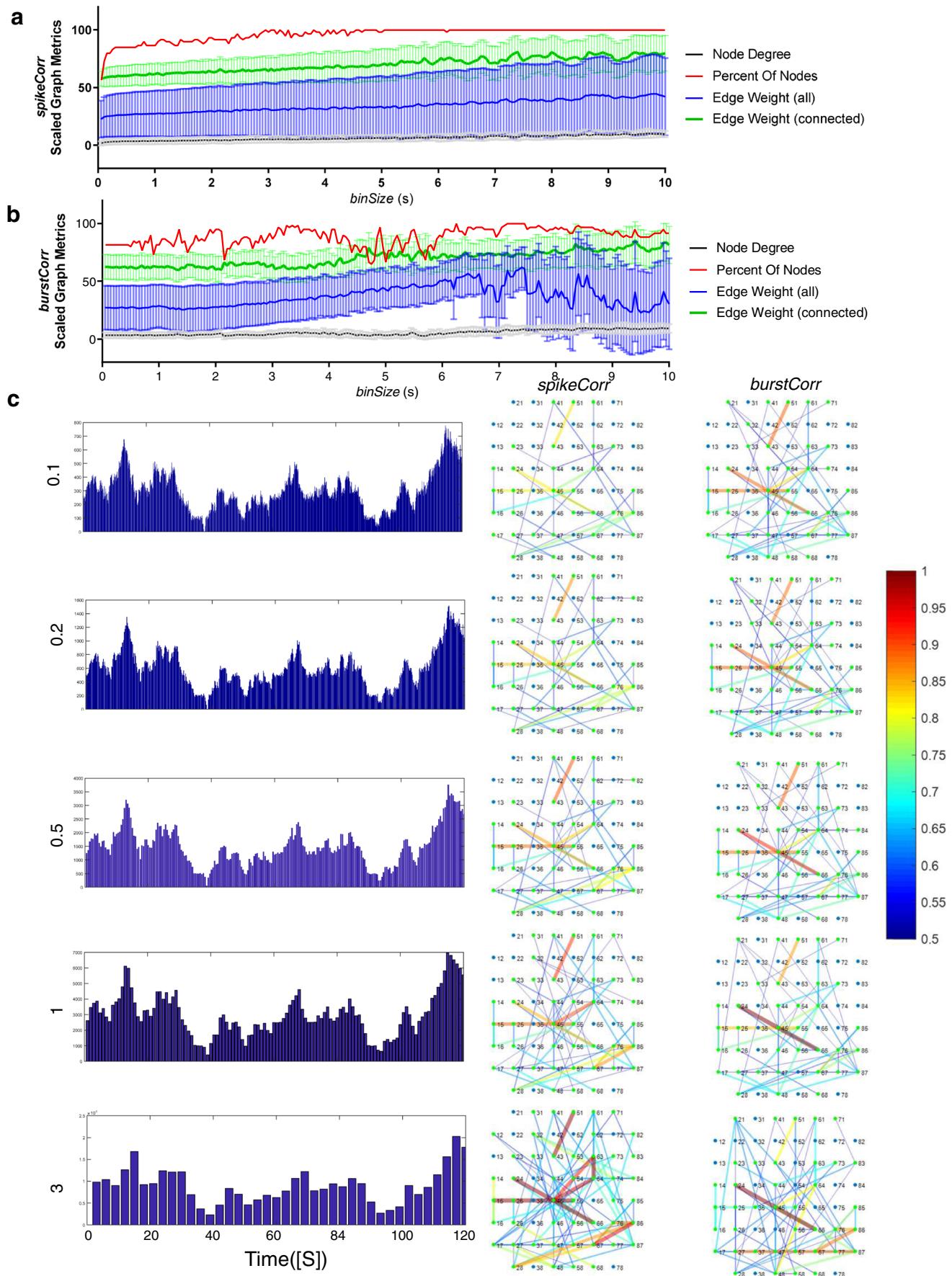
### Testing on Simulated Data

To assess MEAnalyzer code and to investigate the effect of analysis parameters on MEAnalyzer output we created a series of functions to generate simulated *SpikeData* sets. The function *simSpike()* creates a *SpikeData* structure of 60 electrodes with random spike patterns over a defined time range. The inputs to *simSpike()* are *MFR*, *sSTD*, and *timeSegment*, all of which respectively represent numbers defining the mean firing rate of all the electrodes, the desired standard deviation of the mean firing rate, and the range of time over which the simulated spikes are to be arranged. The spike time stamps for each electrode are created independently of the other electrodes.

The spiking behavior of neurons is generally not random, but occurs in patterns of burst activity (Jefferys and Haas 1982). To simulate the effect of random burst activity we designed the function *simBurst()* which requires all of the inputs of *simSpike()* with the addition of *MBR*, *bSTD*, *PSB*, and *pSTD* which are variables that represent the desired mean burst rate, the standard deviation of the mean burst rate, the average percent of spikes that are in bursts, and the standard deviation of spikes that are in bursts. These functions allowed us to simulate electrode activity.

### Application of MEAnalyzer to Experimental Data

Primary rat cortical neurons were co-cultured with astrocytes at a cell density of 800,000 cells/well on multichannel electrode array plates (MCS 60MEA200/30iR-Ti, Multi Channel Systems MCS GmbH) containing 60 electrodes (30  $\mu$ m diameter, 200  $\mu$ m spacing). MEA plates were coated with polyethyleneimine (0.05%), followed by laminin (20  $\mu$ g/ml). Recordings were conducted using an MEA2100 system (Multi Channel System MCS GmbH, Reutlingen Germany) that incorporated a heated stage (set to 37 °C), and a permeable membrane cover to prevent media evaporation and ion concentration (Potter and DeMarse 2001). Voltage measurements were made using MC\_RACK software at a 25 kHz sampling rate, and filtered using a 2nd order Butterworth filter with a 200 Hz cutoff frequency. Spike detection was also conducted in MC\_RACK, and spikes were identified as instantaneous time points of voltages that exceeded a threshold of 5 standard deviations from the baseline voltage measurements. Recordings were conducted to measure spontaneous neuronal activity 22, and 27 days in vitro and were converted to HDF5 files using DataManager before being imported into MEAnalyzer for proof-of-concept analysis.



**Fig. 4 Effect of changing *binSize* on Connectivity Graphs Based on *burstCorr()* and *spikeCorr()*.** Simulations were conducted on a random 120 s spike train with a given burst rate of 1 burst/min. **a** Simulation results for percent of included nodes, average node degree, and edge weights for graphs based on  $spikeCorr \geq 0.5$  for increasing *binSize* values. **b** Simulation results for percent of included nodes, average node degree, and edge weights for connectivity graphs based on  $burstCorr \geq 0.5$  for increasing *binSize* values. **c** Representative raster plot and representative binned spikes for different *binSize* values. **d** Corresponding connectivity graphs based on  $spikeCorr \geq 0.5$  and  $burstCorr \geq 0.5$ . Line width and color correspond to edge weight, which is the cross-correlation value

## Results

### Effects of Burst Detection Parameters on Identified Bursts

We determined how changing thresholds for burst identification ( $nThresh$  = the minimum number of spikes required to define a burst, and  $tThresh$  = the time frame during which the set number of spikes must occur) modifies the relationship between spike rates, and burst identification. Random spike trains were generated from 1 to 100 Spikes/s and  $nThresh$  was set to either 4, 5, 6, 7 or 8, and  $tThresh$  was 0.05, 0.1, or 0.2 s (Fig. 2a). We found that as  $nThresh$  increased, the number of bursts and percent of spikes detected as bursts decreased. As  $tThresh$  increased, there was a shift in the maximum burst rate that could be detected before the burst rate would begin to decrease. In the toe region of the graph, a smaller  $tThresh$  resulted in fewer bursts being identified (Fig. 2b). This data demonstrates that a larger  $nThresh$  and smaller  $tThresh$  reduces the risk of erroneously detecting bursts.

### Effect of TimeSegment Length on Functional Connectivity Graph

Functional connectivity graphs are created utilizing an adjacency matrix that is calculated using one of several possible user-selected methods. The overlapping bursts method creates an edge between two nodes (electrodes) if they have at least one overlapping burst. To investigate how the length of the time segment affects the functional connectivity graph created using this method, we simulated random spike trains with a burst rate of 1 burst/min, and spike noise at a rate of 0.5 spikes/s randomly dispersed over 1 h using *simBurst()* (Fig. 3a). Functional connectivity graphs were created for all time ranges in increments of 10 s (Fig. 3b–c). Graph metrics were calculated for each functional connectivity graph. The results demonstrate that if network connectivity is calculated based on overlapping bursts, all graph metrics increase as the length of time increases until the graph is completely full, in this case this occurred in less than 120 s. This means that for long time segments it will not be possible to show differences in the functional connectivity graphs of MEA experiments using the overlapping bursts method to create the adjacency matrix

because there will be no room for change in the connectivity graph metrics.

The effects of *timeSegment* length on the connectivity graph metrics calculated using correlation matrices was investigated using a random spike train of 1 burst/min and random noise of 0.5 Spikes/s, using *spikeCorr()* and *burstCorr()* with a correlation weight threshold of 0.5 to create the graphs (Fig. 3d–e). All calculated connectivity graph metrics were higher for shorter time periods and leveled out to near zero as time segment length increased (by 100 s for cluster coefficient and 300 s for the percent of nodes included in the connectivity graph).

The effect of *binSize* on graphs based on cross-correlations was investigated by altering *binSize* for a random spike train of 120 s bursting at a rate of 1 burst/min. Graph metrics for graphs created using both *spikeCorr()* and *burstCorr()* were calculated for each bin size. All metrics were relatively stable and both correlation methods yielded similar results (Fig. 4). This indicates that changes in *binSize* result in relatively minor changes to the connectivity graphs.

### Application to Experimental Data

As a proof-of-concept for the application of MEAnalyzer, we used it to analyze spike train time stamps from primary rat hippocampal neurons that were recorded at 22 and 27 DIV. We first selected our electrode layout (MCS  $8 \times 8$  60 MEA) and then imported the HDF5 file that had been exported from DataManager. (Fig. 5a). Based on an inspection of the voltage data from when the recordings were made, we only included electrodes in our analysis that provided functional and informative data. Defective electrodes (defined as electrodes that have noisy baseline voltages that exceed 60  $\mu$ V) were removed from analysis. We cropped the spike train data to a 120 s time segment. A quick overview of summary statistics is given in the ‘Stats at a Glance’ panel. MEAnalyzer displayed raster plot and binned spikes for the edited plate which gave a visualization of patterns and timing for both electrode-specific data and plate-wide data and can be changed to visualize either normalized binned spikes, total percent spiking, or total percent bursting in each time bin.

Electrode-specific calculations were displayed on the *Analyze* tab (MFR, MBR, percent of spikes in bursts). These calculations can aid the user in filtering out inactive electrodes that are below a user-defined spike rate. This tab also displayed periodicity analysis using autocorrelation or Welch’s periodogram. We chose to calculate this based on network bursts (percent of included electrodes that were spiking in each time bin), but also could have chosen to calculate them based on binned spike rate or percent bursting. We copied the data to the clipboard so we could paste the data in excel and easily visualize the comparisons between the periodicity at 22DIV and 27 DIV.



**Fig. 5 Screenshots of MEAnalyzer application to experimental data.** a The Load tab in MEAnalyzer was used to import spike train data. Electrodes with high voltage noise or low spike activity were excluded from analysis using the *Electrode* panel. *TimeSegment* was chosen using the *Crop Time* panel, and *binSize* and burst calculation parameters in the *Burst Definition* panel were left as default. MEAnalyzer displays the raster plot with overlaid bursts as well as binned spikes and plate-wide spike quantifications. b The *Numerical Analysis* panel under the *Analyze* tab displays the spike calculations for each individual electrode. The *AutoCorrelation* and *Periodicity* panels allow the user to investigate oscillatory behaviors in neuronal activity. c The *Graph* tab of MEAnalyzer displays calculations for network connectivity and allows the user to specify the parameters and criteria for the calculations, as well as specifying options for the visualization

The functional connectivity graph is visualized based on user set criteria in the *Graph* tab (Fig. 5c). We chose to calculate our based both on Spike Correlations and Burst Correlations. We initially customized the visualizations so that node size would represent burst rate, node color would represent MFR, and edge color would represent correlation (edge weight). We also selected a different colormap for node color and edge color. Due to the many edges in the graphs for this data, we chose to keep the visualization simple by not adding a variable for edge width. We also used the visualization options in the *Histograms* tab to examine inter-spike intervals, inter-burst intervals, burst length, and the number of spikes in bursts.

Binned Spike and Raster plots at 22 DIV showed evidence of plate-wide neuronal activity that was synchronous and periodic with most spikes appearing to occur in burst patterns. After 27 DIV, neuronal activity was still synchronous and periodic, but bursts were more frequent with more spikes in each bin (Fig. 6a). Spike metrics calculated in MEAnalyzer were exported to excel files for external statistical comparisons. Quantitative analysis of spike rates, burst rates, and percent of spikes contained in bursts were significantly higher at 27 DIV compared with 22 DIV (Fig. 6b). Since we were analyzing measurements made from the same MEA plate, we had the ability to use paired tests. Mean Firing Rate increased from  $6.3 \pm 3.2$  Spikes/s to  $13.7 \pm 4.7$  Spikes/s ( $p < 0.0001$ , Wilcoxon matched-pairs signed rank test), mean burst rate increased from  $10.8 \pm 2.8$  to  $18.5 \pm 1.45$  Bursts/min ( $p < 0.0001$ , Wilcoxon matched-pairs signed rank test), and the percent of spikes in bursts rose from  $88.9 \pm 14.9$  to  $95.1 \pm 3.6$  ( $p < 0.0001$ , Wilcoxon matched-pairs signed rank test).

The *Histogram* tab displayed results from algorithms that have been used historically (ISI, IBI), as well as new visualizations for burst length and the number of spikes in bursts (Fig. 6c). We switched axis limits from “auto” to “manual” so that we could adjust the axis limits to match for both 22 and 27 DIV. The results showed a decreasing shift in ISI and IBI, indicative of a higher MFR and a higher MBR. There was also a shift to the right in the histograms for both burst length and the number of spikes in bursts.

Two methods were used quantify periodic burst activity: Welch’s Periodogram and Autocorrelation (Fig. 6c). Both methods showed distinct peaks corresponding to the periodic intervals. The periodic interval decreased from 7.3 s at 22 DIV to 6.4 s at 27 DIV. This shorter periodic interval indicates more rapid firing at 27 DIV, as evidenced by the higher MFR and MBR. All of these results were summarily visualized with the 4-dimensional movies that showed patterns of electrode activity over time (Supplementary movies 1 and 2).

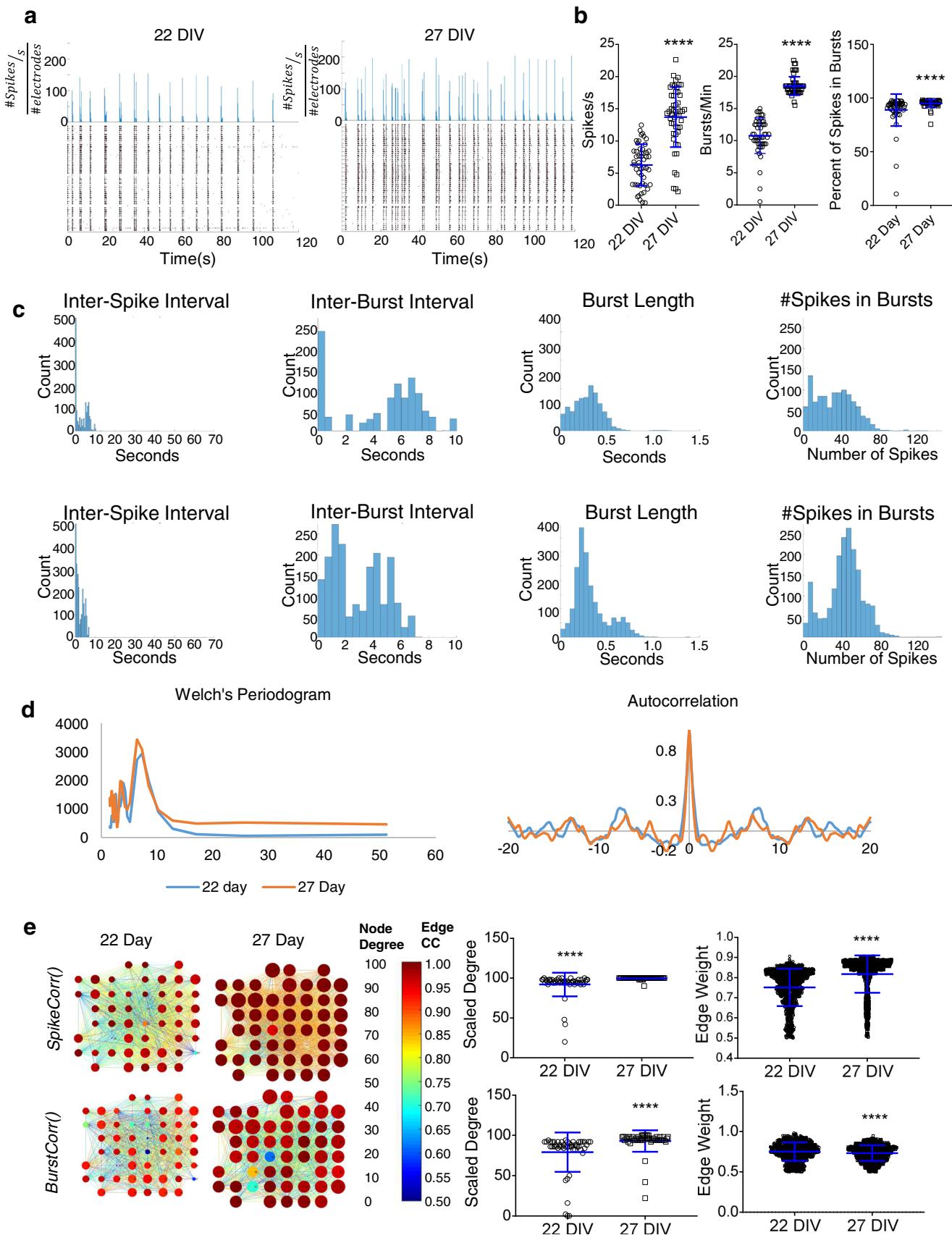
Functional connectivity was investigated using both *spikeCorr()* and *burstCorr()* with a correlation weight threshold of 0.5 (Fig. 6d). MEAnalyzer’s visualization options were used to visualize connectivity maps. Node size was set to represent burst rate, node color was set to represent degree, and edge color was set to represent the correlation between two electrodes. Using *spikeCorr()* resulted in a functional connectivity graph that included 98.1% of electrodes at 22DIV and 100% of electrodes at 27DIV. Graph fullness (the percent of possible edges) increased from 92.0% to 99.6%. Average node degree mirrored graph fullness and increased from  $92.0 \pm 2.1$  to  $99.6 \pm 0.2$ ,  $p = 0.0004$ . This can be seen on the visualization of the connectivity graph by the increase in node size from 22DIV to 27DIV. The average correlation, or edge weight, of the included connections, increased from  $0.752 \pm 0.003$  to  $0.818 \pm 0.003$ . The increase in correlation weight is visualized on the connectivity graph by changes in edge color that shift towards the red end of the color spectrum (Fig. 6d).

Functional connectivity graphs created using *burstCorr()* included 96.1% of electrodes at 22DIV and 100% of electrodes at 27DIV. Graph fullness increased from 79.2% to 93%. Similarly, average Node Degree increased from  $79.2 \pm 3.4$  to  $93.0 \pm 1.9$ ,  $p = 0.0006$ . The average weight of the included edges decreased from  $0.752 \pm 0.004$  to  $0.733 \pm 0.003$ .

Graph connectivity based on overlapping bursts was visualized by creating a movie of 3 s time intervals shifted 1 s for each frame. Node size was set to represent the average burst rate for each electrode over the entire 120 s duration, and node color set to represent Node Degree for each time interval (Supplementary movies 3 and 4).

## Effect of Spike Detection on Spike Train Analysis

The accuracy of MEAnalyzer’s solutions is dependent on the accuracy of the spike train that is loaded for analysis (spike detection is determined outside of MEAnalyzer; by MC\_Rack in our experiments). Spike detection is most commonly conducted using a threshold method, where any time stamp with a voltage greater than a set threshold is considered a spike. The threshold is often determined for each electrode as a user-set number of standard deviations away from baseline noise. The region chosen to identify baseline standard deviation may have a great impact on spike detection, as it will affect the threshold used to detect spikes. To investigate this, spike



**Fig. 6 MEAnalyzer Analysis Results of Experimental Data.** MEAnalyzer was used to analyze the same plate of primary rat cortical neurons 22 and 27 DIV. **a** Binned spike and raster plot comparison between the two conditions. **b** Quantitation of bursting metrics and statistical analysis between the two conditions. Spike Rate, Burst Rate, and the Percent of Spikes in Burst were all significantly higher at the 27 Day time point (Wilcoxon's signed rank test,  $p < 0.0001$ ). **c** Histograms for Inter-Spike Interval (ISI), Inter-Burst Interval (IBI), burst length, and the number of spikes in bursts for each condition (**d**) Periodicity analysis and comparison between the two time points. The Welch's Periodogram and Autocorrelation plots both show peaks occurring earlier for cultures 27 DIV, indicating more frequent oscillations. **e** Connectivity analysis and comparison between the two conditions. Connectivity graphs were created for each condition using both *spikeCorr()* and *burstCorr()*, with *binSize* = 0.2 and a weight threshold of 0.5. Node color represents node degree and node size represents normalized Burst Rate. Line color corresponds to the correlation between the two electrodes

detection was conducted in MC\_Rack on the filtered voltage data for the 22-day plate, with detection thresholds set at standard deviations (SDs) that ranged from 3.5 to 19 in 0.5 SD intervals. MEAnalyzer was then used to examine the resulting spike trains (Fig. 7). As the SD threshold increased, all parameter measurements decreased. By 7.5 SD Spike Rate, Burst Length, and the Number of Spikes in each burst had fallen to less than 50% of the rate measured with the 5 SD threshold. Burst Rate and the Percent of Spikes in Bursts did not reach the same level of change until 11.4 and 14.5 SD, respectively (Fig. 7a). Functional connectivity graph metrics were more resistant to change as the SD threshold was altered. Node Degree calculated based on *burstCorr()* fell to 80% of its 5SD value by 10SDs, as did Node Degree based on *spikeCorr()*. Cluster coefficient calculations based on *burstCorr()* fell to 80% of its 5SD value by 10.5 SDs while cluster coefficient based on *spikeCorr()* did not reach the same level until 15 SDs.

## Discussion

We developed MEAnalyzer in order to make the analysis of MEA spike trains freely available and reproducibly for the average user. Hypothesis-driven science is dependent on the ability to make observations, and MEAnalyzer allows users access to customizable visualizations that can aid in making new observations and asking new questions. Each MEA system has its own software that perform a variety of analyses functions, but each lack some functionality that is important for a comprehensive analysis of data. An increase in the consistency of how MEA data is analyzed will ultimately increase the reproducibility of MEA experiments across laboratories. We hope that by making this analysis software freely available, laboratories conducting MEA experiments will have the option to use universally available software to analyze data. The software will be periodically updated to include additional

functionality, and we encourage input from users for recommendations of additional analyses/visualizations that would enhance the output of MEA experiments.

MEAnalyzer specializes in analyzing imported spike train data from spike time stamps. Excluding spike detection and voltage data from MEAnalyzer reduces computational time and file size. Voltage amplitude is not taken into consideration as it is representative of a population of cells and is not comparable between arrays. Therefore, spike time stamps are the element that is used for analysis. Spike detection is done externally and care should be taken during spike detection as inaccurate spike detection is the primary source for error in analyzing MEA recordings from either falsely detecting spikes or missing spikes (Harris et al. 2016). Although voltage data is not used for analysis, MEAnalyzer allows for the uploading and viewing of voltage traces with spike overlays if the user wishes to inspect voltage detection accuracy.

Our simulations of random spike data provide insights into the effect of analysis parameters on reducing input error; especially considering that noise or false spike detection would most likely be randomly dispersed. Unlike more advanced spike simulations methods like SpikeNet (Delorme and Thorpe 2003), GENESIS (Bower and Beeman 2007), or NEURON (Hines and Carnevale 1997) which aim to understand neuronal function, our simulations were intended to aid in understanding how changing analysis parameters in MEAnalyzer changes the resulting metrics. We found that changing the analysis parameters for what would be considered a burst (through *tThresh* and *nThresh*) showed that requiring a larger number of spikes in a shorter period of time produced more stringent burst detection with less error (Fig. 2c-d). It is possible that the effects of randomly detected bursts, due to noise or erroneous spike detection, are compounded over time. The analysis method most likely to be affected by errors due to compounded spike detection is network graphs based on overlapping bursts. In simulations using random burst activity we found as the length of time for the analysis increased, the fullness of the connectivity graph increased until it appeared that all nodes were connected. Thus, the “overlapping burst” method for creating connectivity graphs is only suitable for very short periods of experimental time. For visual discrimination of network connectivity over longer recording times we created the “graph movie” option which shows spike and burst activity in real time so that compounding effects are minimized.

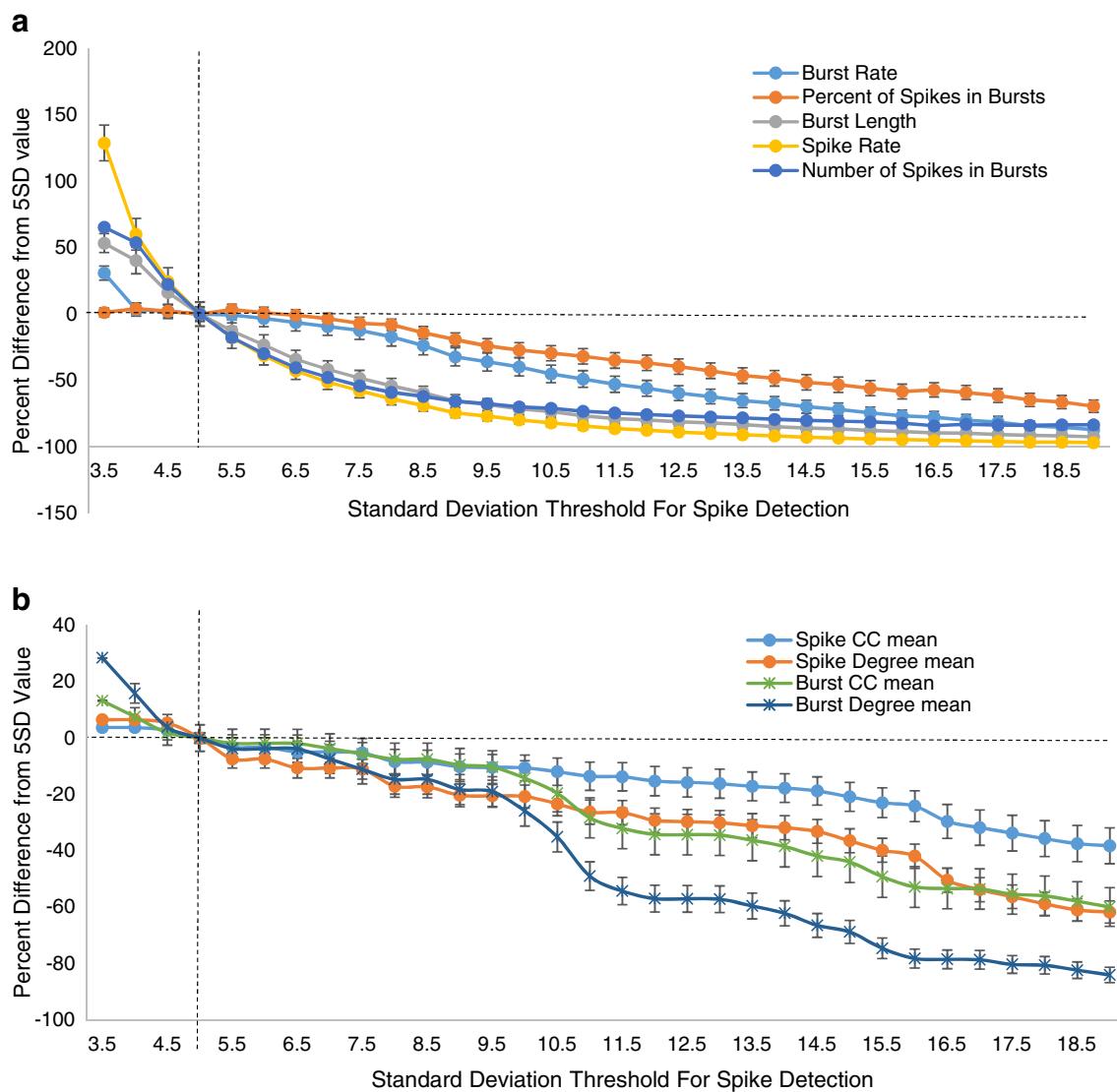
Our simulations suggest that calculating network connectivity based on the correlation of activity (using binned spikes or binned bursts) is a more suitable approach for longer experimental time frames. However, our simulations using *simBurst()* suggest that *binSize* may affect the calculations for network connectivity. Burst timestamps created randomly without reference to other electrodes produced minimal

correlations between electrodes. These random correlations between electrodes increased as bin size increased so that longer bin sizes increased the apparent strength correlation between two nodes.

In addition to the advanced spike train calculations included in MEAnalyzer, traditional analysis methods such as mean firing rates and interval histograms have been included. The inclusion of these methods makes it possible to compare experiments analyzed with MEAnalyzer to previously analyzed or published experiments. While important, these measures contain limited information about spatio-temporal patterns. Functional connectivity and periodicity measures provide more information about this. Two versions of periodicity measures are included, as autocorrelation is the easiest to interpret

but Welch's periodogram allows for the identification of multiple periodic intervals.

Applications to experimental data demonstrate that MEAnalyzer can quantify differences between experimental conditions. In the case of primary rodent cortical neurons measured at different developmental time points (5 DIV apart), MEAnalyzer quantified increased spike rate, and burst rate. Analysis of functional connectivity was displayed topographically, and network graphs / visualization options were easily modified to allow facilitate data interpretation and parameters for analyses. Data visualizations were also useful to guide threshold settings to determine what is considered an edge in the adjacency matrix. For example, functional connectivity graphs created from overlapping bursts would not be



**Fig. 7** Spike Train Analysis is dependent on proper spike detection.

MEAnalyzer was used to analyze voltage data with Spike detection conducted at varying thresholds. Spike detection was done in MC\_Rack based on thresholding set at designated standard deviations above baseline. The standard deviations used ranged from 3.5 to 19, in 0.5

intervals. Results are presented as the percentage difference from the values for spike detection conducted at a 5 SD above background threshold. **a** Statistical calculations for Burst Rate, Percent of Spikes in Bursts, Burst Length, Spike Rate, and the Number of Spikes in Bursts. **b** Functional connectivity graph metrics of Node Degree

appropriate for an experiment where the user requires continuous analyses over long time durations, as random bursts would be compounded over time and all nodes in the connectivity graph would appear to be connected. Alternatively, a cross-correlation analysis would not be appropriate for experimental analysis of short time frames or for experiments with minimal spike activity, as this would result in an artificially high cross-correlation between nodes. The variety of analysis options provided in MEAnalyzer ensures the user will be able to choose settings appropriate for their particular experimental set-up.

Although MEAnalyzer is a powerful analysis tool for MEA data analysis, it has some practical limitations. MEAnalyzer is implemented in MATLAB, which can be less time efficient than other programming languages. We have made every effort to make the code as efficient as possible to prevent redundant calculations. Although these same algorithms can be implemented in a faster processing language, maintaining the code in MATLAB provides ease of expansion to update with new functionality.

MEAnalyzer is currently designed to analyze one multi-electrode array at a time, and does not have an internal comparison between multiple arrays. To compare between plates the user must export the analyzed data from each experiment into a separate statistical software program for statistical analysis. We have made every effort to make this easier by allowing for easy export of all analyzed data and allowing for manual adjustment of figure options (axis limits, bin sizes, etc) so that figures can be comparable. Currently, the functional connectivity graphs in MEAnalyzer are undirected, but the creation of directed graphs may also be useful for users interested in using electrical stimulation methods. Additional metrics that can be added are weighted global efficiency and weighted clustering coefficients (among others). MEAnalyzer is fully expandable, and upgrades will be periodically released to ensure that MEA data from all acquisition systems can be analyzed.

Multi-electrode array experiments have many purposes (Spira and Hai 2013), from neuropharmacological testing for toxicity (Bradley et al. 2018) to cardiac experiments (de Korte et al. 2017) to neuronal plasticity. MEAnalyzer can help to consistently and reproducibly analyze these experiments with a wide array of features. We are unaware of any other free stand-alone software that includes all of these features in an easy-to-use GUI interface at no cost (Supp. Figure 1). MEAnalyzer has already been used to investigate functional changes in neuronal activity (Chaudhuri et al. 2018; Miller et al. 2019). While there is no perfect one-size fits all framework for analyzing MEA experiments, MEAnalyzer provides a multitude of analysis options that can be customized for the needs of any experiment.

**Information Sharing Statement** MEAnalyzer is freely available at Mathworks Fileshare (<https://www.mathworks.com/matlabcentral/fileexchange/68260-meanalyzer>; which is linked to the Github Repository at <https://github.com/RDastg1/MEAnalyzer>). This also contains the HDF5 files for the spike trains biological measurements used as examples in this manuscript and a detailed user manual.

## References

- Bokil, H., Andrews, P., Kulkarni, J. E., Mehta, S., & Mitra, P. P. (2010). Chronux: A platform for analyzing neural signals. *Journal of Neuroscience Methods*, 192(1), 146–151. <https://doi.org/10.1016/j.jneumeth.2010.06.020>.
- Bologna, L. L., Pasquale, V., Garofalo, M., Gandolfo, M., Baljon, P. L., Maccione, A., Martinoia, S., & Chiappalone, M. (2010). Investigating neuronal activity by SPYCODE multi-channel data analyzer. *Neural Networks*, 23(6), 685–697. <https://doi.org/10.1016/j.neunet.2010.05.002>.
- Bongard, M., Micol, D., & Fernandez, E. (2014). NEV2lkit: A new open source tool for handling neuronal event files from multi-electrode recordings. *International Journal of Neural Systems*, 24(4), 1450009. <https://doi.org/10.1142/S0129065714500099>.
- Bower, J. M., & Beeman, D. (2007). Constructing realistic neural simulations with GENESIS. *Methods in Molecular Biology*, 401, 103–125. [https://doi.org/10.1007/978-1-59745-520-6\\_7](https://doi.org/10.1007/978-1-59745-520-6_7).
- Bradley, J. A., Luithardt, H. H., Metea, M. R., & Strock, C. J. (2018). In vitro screening for seizure liability using microelectrode Array technology. *Toxicological Sciences*, 163, 240–253. <https://doi.org/10.1093/toxsci/kfy029>.
- Brodie, K. W., Carpenter, L. A., Earnshaw, R. A., Gallop, J. R., Hubbald, R. J., Mumford, A. M., et al. (2012). *Scientific visualization: Techniques and applications*. Springer Science & Business Media.
- Cajigas, I., Malik, W. Q., & Brown, E. N. (2012). nSTAT: Open-source neural spike train analysis toolbox for Matlab. *Journal of Neuroscience Methods*, 211(2), 245–264. <https://doi.org/10.1016/j.jneumeth.2012.08.009>.
- Canepari, M., Bove, M., Maeda, E., Cappello, M., & Kawana, A. (1997). Experimental analysis of neuronal dynamics in cultured cortical networks and transitions between different patterns of activity. *Biological Cybernetics*, 77(2), 153–162. <https://doi.org/10.1007/s004220050376>.
- Chaudhuri, A. D., Dastgheib, R. M., Yoo, S. W., Trout, A., Talbot, C. C., Jr., Hao, H., et al. (2018). TNFalpha and IL-1beta modify the miRNA cargo of astrocyte shed extracellular vesicles to regulate neurotrophic signaling in neurons. *Cell Death & Disease*, 9(3), 363. <https://doi.org/10.1038/s41419-018-0369-4>.
- Chiappalone, M., Bove, M., Vato, A., Tedesco, M., & Martinoia, S. (2006). Dissociated cortical networks show spontaneously correlated activity patterns during in vitro development. *Brain Research*, 1093, 41–53. <https://doi.org/10.1016/j.brainres.2006.03.049>.
- Cotterill, E., Charlesworth, P., Thomas, C. W., Paulsen, O., & Eglen, S. J. (2016). A comparison of computational methods for detecting bursts in neuronal spike trains and their application to human stem cell-derived neuronal networks. *Journal of Neurophysiology*, 116(2), 306–321. <https://doi.org/10.1152/jn.00093.2016>.
- de Korte, T., Guilbot, S., Printemps, R., Hautefeuille, L., Le Grand, M., Wilbers, R., et al. (2017). Assessment of compound effects on electrophysiology of hiPSC-derived ventricular cardiomyocytes using

- patch clamp and multielectrode Array (MEA) analysis. *Journal of Pharmacological and Toxicological Methods*, 88, 196–196. <https://doi.org/10.1016/j.vascn.2017.09.092>.
- Delorme, A., & Thorpe, S. J. (2003). SpikeNET: An event-driven simulation package for modelling large networks of spiking neurons. *Network*, 14(4), 613–627.
- Earnshaw, R., & Wiseman, N. (2012). *An introductory guide to scientific visualization*. Springer Science & Business Media.
- Egert, U., Knott, T., Schwarz, C., Nawrot, M., Brandt, A., Rotter, S., & Diesmann, M. (2002). MEA-tools: An open source toolbox for the analysis of multi-electrode data with MATLAB. *Journal of Neuroscience Methods*, 117(1), 33–42. [https://doi.org/10.1016/S0165-0270\(02\)00045-6](https://doi.org/10.1016/S0165-0270(02)00045-6).
- Georgiadis, V., Stephanou, A., Townsend, P. A., & Jackson, T. R. (2015). MultiElec: A MATLAB based application for MEA data analysis. *PLoS One*, 10(6), e0129389. <https://doi.org/10.1371/journal.pone.0129389>.
- Gerstein, G. L. (2000). Cross-correlation measures of unresolved multi-neuron recordings. *Journal of Neuroscience Methods*, 100(1–2), 41–51.
- Goldberg, D. H., Victor, J. D., Gardner, E. P., & Gardner, D. (2009). Spike train analysis toolkit: Enabling wider application of information-theoretic techniques to neurophysiology. *Neuroinformatics*, 7(3), 165–178. <https://doi.org/10.1007/s12021-009-9049-y>.
- Gross, G. W., & Schwalm, F. U. (1994). A closed flow chamber for long-term multichannel recording and optical monitoring. *Journal of Neuroscience Methods*, 52(1), 73–85. [https://doi.org/10.1016/0165-0270\(94\)90059-0](https://doi.org/10.1016/0165-0270(94)90059-0).
- Harris, K. D., Quiroga, R. Q., Freeman, J., & Smith, S. L. (2016). Improving data quality in neuronal population recordings. *Nature Neuroscience*, 19(9), 1165–1174. <https://doi.org/10.1038/nn.4365>.
- Hazan, L., Zugaro, M., & Buzsaki, G. (2006). Klusters, NeuroScope, NDManager: A free software suite for neurophysiological data processing and visualization. *Journal of Neuroscience Methods*, 155(2), 207–216. <https://doi.org/10.1016/j.jneumeth.2006.01.017>.
- Hines, M. L., & Carnevale, N. T. (1997). The NEURON simulation environment. *Neural Computation*, 9(6), 1179–1209.
- Hodgkin, A. L., & Huxley, A. F. (1945). Resting and action potentials in single nerve fibres. *The Journal of Physiology*, 104(2), 176–195.
- Huang, Y., Li, X., Li, Y., Xu, Q., Lu, Q., & Liu, Q. (2008). An integrative analysis platform for multiple neural spike train data. *Journal of Neuroscience Methods*, 172(2), 303–311. <https://doi.org/10.1016/j.jneumeth.2008.04.026>.
- Jefferys, J. G. R., & Haas, H. L. (1982). Synchronized bursting of CA1 hippocampal pyramidal cells in the absence of synaptic transmission. *Nature*, 300(5891), 448–450. <https://doi.org/10.1038/300448a0>.
- Johnson, D. H. (1978). The relationship of post-stimulus time and interval histograms to the timing characteristics of spike trains. *Biophysical Journal*, 22(3), 413–430. [https://doi.org/10.1016/S0006-3495\(78\)85496-4](https://doi.org/10.1016/S0006-3495(78)85496-4).
- Kamioka, H., Maeda, E., Jimbo, Y., Robinson, H. P., & Kawana, A. (1996). Spontaneous periodic synchronized bursting during formation of mature patterns of connections in cortical cultures. *Neuroscience Letters*, 206(2–3), 109–112.
- Knox, C. K. (1981). Detection of neuronal interactions using correlation-analysis. *Trends in Neurosciences*, 4(9), 222–225. [https://doi.org/10.1016/0166-2236\(81\)90070-9](https://doi.org/10.1016/0166-2236(81)90070-9).
- Kwon, K. Y., Eldawlatly, S., & Oweiss, K. (2012). NeuroQuest: A comprehensive analysis tool for extracellular neural ensemble recordings. *Journal of Neuroscience Methods*, 204(1), 189–201. <https://doi.org/10.1016/j.jneumeth.2011.10.027>.
- Latora, V., & Marchiori, M. (2001). Efficient behavior of small-world networks. *Physical Review Letters*, 87(19). <https://doi.org/10.1103/PhysRevLett.87.198701>.
- Lidierth, M. (2009). sigTOOL: A MATLAB-based environment for sharing laboratory-developed software to analyze biological signals. *Journal of Neuroscience Methods*, 178(1), 188–196. <https://doi.org/10.1016/j.jneumeth.2008.11.004>.
- Llinás, R. R. (1988). The intrinsic electrophysiological properties of mammalian neurons - insights into central nervous-system function. *Science*, 242(4886), 1654–1664. <https://doi.org/10.1126/science.3059497>.
- Maeda, E., Robinson, H. P. C., & Kawana, A. (1995). The mechanisms of generation and propagation of synchronized bursting in developing networks of cortical-neurons. *The Journal of Neuroscience*, 15(10), 6834–6845.
- Mahmud, M., Bertoldo, A., Girardi, S., Maschietto, M., & Vassanelli, S. (2012). SigMate: A Matlab-based automated tool for extracellular neuronal signal processing and analysis. *Journal of Neuroscience Methods*, 207(1), 97–112. <https://doi.org/10.1016/j.jneumeth.2012.03.009>.
- Mahmud, M., Pulizzi, R., Vasilaki, E., & Giugliano, M. (2014). QSpikes tools: A generic framework for parallel batch preprocessing of extracellular neuronal signals recorded by substrate microelectrode arrays. *Frontiers in Neuroinformatics*, 8, 26. <https://doi.org/10.3389/fninf.2014.00026>.
- Meier, R., Egert, U., Aertsen, A., & Nawrot, M. P. (2008). FIND—a unified framework for neural data analysis. *Neural Networks*, 21(8), 1085–1093. <https://doi.org/10.1016/j.neunet.2008.06.019>.
- Micallef, J. (1987). Encapsulation, reusability and extensibility in object-oriented programming languages.
- Miller, S. J., Philips, T., Kim, N., Dastgheyb, R., Chen, Z., Hsieh, Y. C., Daigle, J. G., Datta, M., Chew, J., Vidensky, S., Pham, J. T., Hughes, E. G., Robinson, M. B., Sattler, R., Tomer, R., Suk, J. S., Bergles, D. E., Haughey, N., Pletnikov, M., Hanes, J., & Rothstein, J. D. (2019). Molecularly defined cortical astroglia subpopulation modulates neurons via secretion of Norrin. *Nature Neuroscience*, 22, 741–752. <https://doi.org/10.1038/s41593-019-0366-7>.
- Novak, J. L., & Wheeler, B. C. (1988). Multisite hippocampal slice recording and stimulation using a 32 element microelectrode Array. *Journal of Neuroscience Methods*, 23(2), 149–159. [https://doi.org/10.1016/0165-0270\(88\)90187-2](https://doi.org/10.1016/0165-0270(88)90187-2).
- Novellino, A., Chiappalone, M., Maccione, A., & Martinola, S. (2009). Neural signal manager: A collection of classical and innovative tools for multi-channel spike train analysis. *International Journal of Adaptive Control and Signal Processing*, 23(11), 999–1013. <https://doi.org/10.1002/acs.1076>.
- Pastore, V. P., Poli, D., Godjoski, A., Martinola, S., & Massobrio, P. (2016). ToolConnect: A functional connectivity toolbox for in vitro networks. *Frontiers in Neuroinformatics*, 10, 13. <https://doi.org/10.3389/fninf.2016.00013>.
- Pine, J. (1980). Recording action-potentials from cultured neurons with extracellular micro-circuit electrodes. *Journal of Neuroscience Methods*, 2(1), 19–31. [https://doi.org/10.1016/0165-0270\(80\)90042-4](https://doi.org/10.1016/0165-0270(80)90042-4).
- Poli, D., Pastore, V. P., & Massobrio, P. (2015). Functional connectivity in in vitro neuronal assemblies. *Front Neural Circuits*, 9, 57. <https://doi.org/10.3389/fncir.2015.00057>.
- Potter, S. M., & DeMarse, T. B. (2001). A new approach to neural cell culture for long-term studies. *Journal of Neuroscience Methods*, 110(1–2), 17–24.
- Salinas, E., & Sejnowski, T. J. (2001). Correlated neuronal activity and the flow of neural information. *Nature Reviews Neuroscience*, 2(8), 539–550. <https://doi.org/10.1038/35086012>.
- Schofield, P. N., Bubela, T., Weaver, T., Brown, S. D., Hancock, J. M., Einhorn, D., et al. (2009). Post-publication sharing of data and tools. *Nature*, 461(7261), 171–173. <https://doi.org/10.1038/461171a>.
- Snyder, D. L. (1991). Random point processes in time and space. In M. I. Miller, & D. L. Snyder (Eds.), (2nd ed. / ed.). In *New York :: Springer-Verlag*.

- Somerville, J., Stuart, L., Sernagor, E., & Borisuk, R. (2010). iRaster: A novel information visualization tool to explore spatiotemporal patterns in multiple spike trains. *Journal of Neuroscience Methods*, 194(1), 158–171. <https://doi.org/10.1016/j.jneumeth.2010.09.009>.
- Spira, M. E., & Hai, A. (2013). Multi-electrode array technologies for neuroscience and cardiology. *Nature Nanotechnology*, 8(2), 83–94. <https://doi.org/10.1038/nnano.2012.265>.
- Teeters, J. L., Godfrey, K., Young, R., Dang, C., Friedsam, C., Wark, B., Asari, H., Peron, S., Li, N., Peyrache, A., Denisov, G., Siegle, J. H., Olsen, S. R., Martin, C., Chun, M., Tripathy, S., Blanche, T. J., Harris, K., Buzsáki, G., Koch, C., Meister, M., Svoboda, K., & Sommer, F. T. (2015). Neurodata without Borders: Creating a common data format for neurophysiology. *Neuron*, 88(4), 629–634. <https://doi.org/10.1016/j.neuron.2015.10.025>.
- Thomas, C. A., Jr., Springer, P. A., Loeb, G. E., Berwald-Netter, Y., & Okun, L. M. (1972). A miniature microelectrode array to monitor the bioelectric activity of cultured cells. *Experimental Cell Research*, 74(1), 61–66.
- Vato, A., Bonzano, L., Chiappalone, M., Cicero, S., Morabito, F., Novellino, A., & Stillo, G. (2004). Spike manager: A new tool for spontaneous and evoked neuronal networks activity characterization. *Neurocomputing*, 58, 1153–1161. <https://doi.org/10.1016/j.neucom.2004.01.180>.
- Wagenaar, D., DeMarse, T. B., & Potter, S. M. (2005). MeaBench: A toolset for multi-electrode data acquisition and on-line analysis. *2005 2nd Internatinoal Ieee/Embs Conference on Neural Engineering*, 518–521.
- Welch, P. D. (1967). Use of Fast Fourier Transform for Estimation of Power Spectra - a Method Based on Time Averaging over Short Modified Periodograms. *Ieee Transactions on Audio and Electroacoustics*, Au15(2), 70-&, <https://doi.org/10.1109/Tau.1967.1161901>.
- Wiegner, A. W., & Wierzbicka, M. M. (1987). A method for assessing significance of peaks in cross-correlation histograms. *Journal of Neuroscience Methods*, 22(2), 125–131.
- Wiener, N. (1949). *Extrapolation, interpolation, and smoothing of stationary time series, with engineering applications*. Cambridge: Technology Press of the Massachusetts Institute of Technology.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.