Journal of Neuroscience Methods 117 (2002) 33–42

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

MEA-Tools: an open source toolbox for the analysis of multi-electrode data with MATLAB

U. Egert[a,*], Th. Knott[b], C. Schwarz[c], M. Nawrot[a], A. Brandt[a], S. Rotter[a], M. Diesmann[a,1]

[a] *Neurobiology and Biophysics, Institute for Biology III, Albert-Ludwigs-University Freiburg, Schänzlestr. 1, 79104 Freiburg, Germany*
[b] *NMI, Naturwissenschaftliches und Medizinisches Institut an der Universität Tübingen in Reutlingen, Markwiesenstr. 55, 72770 Reutlingen, Germany*
[c] *Abteilung für Kognitive Neurologie, Neurologische Universitätsklinik Auf der Morgenstelle 15, 72076 Tübingen, Germany*

Received 14 January 2002; received in revised form 8 March 2002; accepted 11 March 2002

Abstract

Recent advances in electrophysiological techniques have created new tools for the acquisition and storage of neuronal activity recorded simultaneously with numerous electrodes. These techniques support the analysis of the function as well as the structure of individual electrogenic cells in the context of surrounding neuronal or cardiac network. Commercially available tools for the analysis of such data, however, cannot be easily adapted to newly emerging requirements for data analysis and visualization, and cross compatibility between them is limited. In this report we introduce a free open source toolbox called microelectrode array tools (MEA-Tools) for the analysis of multi-electrode data based on the common data analysis environment MATLAB (version 5.3–6.1, The Mathworks, Natick, MA). The toolbox itself is platform independent. The file interface currently supports files recorded with MCRack (Multi Channel Systems, Reutlingen, Germany) under Microsoft Windows 95, 98, NT, and 2000, but can be adapted to other data acquisition systems. Functions are controlled via command line input and graphical user interfaces, and support common requirements for the analysis of local field potentials, extracellular spike activity, and continuous recordings, in addition to supplementary data acquired by additional instruments, e.g. intracellular amplifiers. Data may be processed as continuous recordings or time windows triggered to some event.

Keywords: Microelectrode arrays; Multi-electrode recording; Electrophysiology; Field potential analysis; Spike analysis; Data analysis

1. Introduction

Recent developments of theories, models, and subsequently, experimental designs in the neurosciences have fostered a range of new techniques that require the acquisition, storage, visualization and analysis of simultaneously recorded neuronal activity (Deribaupierre et al., 1986; Abeles and Gerstein, 1988; Nicolelis, 1999). These experiments take into consideration that the biophysical properties, the function and the structure of individual neurons are influenced by the activity in the neuronal network surrounding them and the interaction between its members. Many of these experiments focus on understanding the functional neuronal correlate of perceptual or behavioral processes. Standard techniques for the analysis of electrophysiological data, however, are often insufficient to detect and represent the associated changes, which led to the development of new tools for multi-electrode data such as gravitational clustering (Abeles and Gerstein, 1988; Lindsey et al., 1989), joint-PSTH (Gerstein et al., 1989; Aertsen and Gerstein, 1991; Vaadia et al., 1995), and other representations of population activity (e.g. Georgopoulos et al., 1993).

For various reasons commercial software tends to have only limited possibilities to incorporate new tools or modify existing ones and share them with colleagues. Furthermore, most companies consider the file structure used in their acquisition systems to be intellectual property, which complicates the use of common tools

[*] Corresponding author. Tel.: +49-761-203-2862; fax: +49-761-203-2860.
E-mail address: egert@biologie.uni-freiburg.de (U. Egert).
[1] Current address: Max Planck Institut für Strömungsforschung, Bunsenstrasse 10, 37073 Göttingen, Germany.

PII: S0165-0270(02)00045-6

for further processing. Recently, a small number of companies emerged as key players in the technological development. None of them, however, includes extensive data analysis tools, nor exists a common standard for data exchange.

It would therefore be desirable to have at hand a tool that allows users to retrieve and analyze data from different hardware systems, to modify and to share them across different operating systems. With this goal in mind, we developed the MATLAB based toolbox MEA-Tools (http://www.brainworks.uni-freiburg.de/projects/mea/meatools/overview.htm), an open source development distributed under GNU public license[2], which regulates its free use and redistribution.

The general idea of this toolbox is to provide a basic set of routines and graphical user interfaces (GUIs) that facilitate access to the original data files and to feed these to an expandable set of tools for their visualization and analysis. It also serves as framework to which additional functions can be added. File access is supported by a minimal set of routines that can be adapted or exchanged for different data acquisition systems. To facilitate the adaptation and extension of the MEA-Tools functionality as well as the addition of new tools for data analysis, the source code and extensive documentation are freely available.

2. Methods

The toolbox was written for MATLAB version 5.3 and tested up to version 6.1, and was originally developed to handle spike, triggered field potential and continuous recordings from multi-electrode arrays in files recorded with the MCRack software (Multi Channel Systems, Reutlingen, Germany, MCRack comes with their MEA 1060 and ME system data acquisition systems for multi-electrode recordings in vitro and in vivo). MCS supplies an object linking and embedding (OLE) interface to its files and several basic tools for use with MATLAB. While access to the files via the OLE interfaces requires Microsoft Windows (Windows 95, 98, NT, 2000 tested), the toolbox itself is in principle compatible with other operating systems. We use this system for extracellular recording from acute brain slices, cell cultures and organotypic tissue cultures with planar, thin-film micro-electrode arrays (MEAs) with 60 recording electrodes. An average experiment in our lab generates 400–600MB of data. Considering that some investigators mentioned the need for up to 1000 electrodes, it is likely that future projects will produce considerably larger amounts of data, which emphasizes the need to minimize the memory space required for a particular procedure.

Except for the file access interface, only standard MATLAB data types are used, so all other functions should work within the framework of MATLAB on other operating systems.

2.1. File interface

File access is supported by two file-type specific dynamic link libraries (DLL, *mcstream.dll* and *mcstreammex.dll*), which are the only non-MATLAB external functions used. To port the toolbox to other operating systems, adapt it to other data acquisition systems, or retrieve data from other file formats, would require suitable adaptation or replacement of these DLLs or other suitable methods for file access. The authors guide for the MEA-Tools gives further details on the requirements for these routines.[3]

Based on these interfaces, the central routine (*nextdata.m*) accessing a DLL (*mcstreammex.dll*) creates a structure-like *datastream object* collating information about the recording settings (e.g. sampling period, date, number of electrodes, version information, etc.) and the data types (*streams*) stored in the file. Fields in this global variable need to be filled by appropriate routines for each data acquisition system or file format.

The second function of *nextdata.m* is to retrieve a specified dataset from the file into the MATLAB workspace. Again, this is implemented through a DLL. All other implemented functions interact with the datastream object or call *nextdata.m*. The Author's Guide section in the MEA-Tools documentation gives further details on the requirements for these routines. In the case of MCRack, the C++ source code for *mcstream.dll* and *mcstreammex.dll* is supplied by Multi Channel Systems.

For each section of the toolbox dealing with a particular data type we designed the tools to segregate three levels of processing. Because of the size of the files and the total amount of data for an experiment we defined a first step that extracts important properties of the data and collects them in the workspace. An intermediate level processes these derived data, requiring no further access to the files. Tools that are useful to display a variety of results were collected in a separate menu. At this step a common data input format is used, which was chosen to represent the data from one electrode in a row of the data matrix.

[2] Version 2 of June 1991. For further details on GNU public license concept and conditions please refer to http://www.gnu.org/copyleft/gpl.html#TOC1.

[3] See http://www.brainworks.uni-freiburg.de/projects/mea/meatools/authors_guide.html.

3. Results

3.1. General requirements of multi-electrode recordings

While many tasks for the analysis of electrophysiological single-electrode recordings can be implemented quite easily, multi-electrode recording generates several additional demands and restrictions that have to be considered when designing tools for their analysis, such as larger amounts of data, multiple data types in one data set, (e.g. neuronal single cell spikes, multi-unit activity, local field potentials (LFPs)), partial overlap between signal sources, and need for automated data processing. In addition, the spatial patterns of the activity need to be related to the anatomical structure of the tissue. The MEA-Tools concept incorporates this demand for the representation of data collected with planar electrode arrays with fixed electrode positions for several layout styles. In the following section we will introduce the general structure of the toolbox and the central data types used.

Beyond handling raw data files, many of the tools will accept simple matrices (corresponding to the data class 'double' in MATLAB), e.g. variables holding the results from intermediate processing steps. The individual routines are accessible through GUIs and via command line calls to facilitate their incorporation into other functions. GUIs can be stored with their parameter settings and retain their functionality when reopened, provided the original data set is available in the base workspace. This facilitates the reproduction of analysis sequences.

A hyperlink matrix documenting links between individual files is available in HTML format. The concept and structure of the toolbox is sufficiently transparent so that the user may add additional functions, examples for which are given in the Author's Guide documents.

3.2. Data types

Functionally, the MEA-Tools file access routine creates a data object, called datastream object, in the MATLAB workspace that holds all file header and data record information. To minimize the amount of main memory occupied, the actual voltage values or other time-varying data are only retrieved as needed. A datastream object is needed for all routines requiring file access. Pre-processed data are handled as matrices, where rows correspond to data sets from one electrode.

3.3. Menu structure

Although several of the individual toolbox functions are used with different analyses, the menu structure was split into signal type oriented sections (Fig. 1). Spike processing and analysis have little overlap with the tools needed to evaluate LFPs. Similarly, continuous recordings (gap-free) are generally processed in a different way. Separate submenus therefore guide through common procedures used for these signal types. A fourth menu supports the processing of various supplementary data types.

Display functions for a variety of data types as well as examples for dynamic displays were combined in a separate menu. The final menu section gives access to documentation, a list of frequently asked questions, and credits contributors.

3.4. Spikes analysis section

Spike Tools were designed to load, view, sort and analyze aspects of electrophysiological spike data recorded extracellularly. As explained above, the central file access routines are currently available for one acquisitions system only, but could with few changes be expanded for others. In addition, several functions can be used with any MATLAB-array of appropriate dimensions.

Spike analysis requires extensive pre-processing to extract spike events from raw data for classification according to their shapes (spike sorting). The analysis of the resulting time series then requires very different techniques, e.g. to assess interactions between neurons based on correlation analyses. Because various tools for time series analysis are readily available (e.g. NeuroExplorer), we focused on incorporating new approaches for spike processing, instead of merely reproducing the set of classical functions.

Parsing the file includes retrieval of spike times, associated channels, and the calculation of principle components (PCs) for each channel. This can be done individually for each set of spikes or based on predefined PCs. Stimulus artifacts can be excluded from this calculation if trigger events were recorded with the stimuli. The procedure can be run in batch mode across several files. Other descriptors, such as spike amplitudes, are easily added to the resulting dataset and can be used in the next processing step.

Spike sorting is a widespread pre-processing step in spike data analysis. The goal is to identify spikes of similar shapes under the assumption that these are likely generated by the same neuron. Even though several spike sorting tools have been presented (Sarna et al., 1988; Salganicoff et al., 1988; Sahani, 1999; Hulata et al., 2000; Kim and Kim, 2000), for everyday use the task itself is currently not solved optimally by any of the existing approaches, in particular not for large sets of data. In the MEA-Tools, we implemented a version using PC analysis of the spike shape. Clusters of the component scores can be identified and delimited manually in pair-wise two-dimensional projections (Fig. 2). The procedure is extended by utilities to review

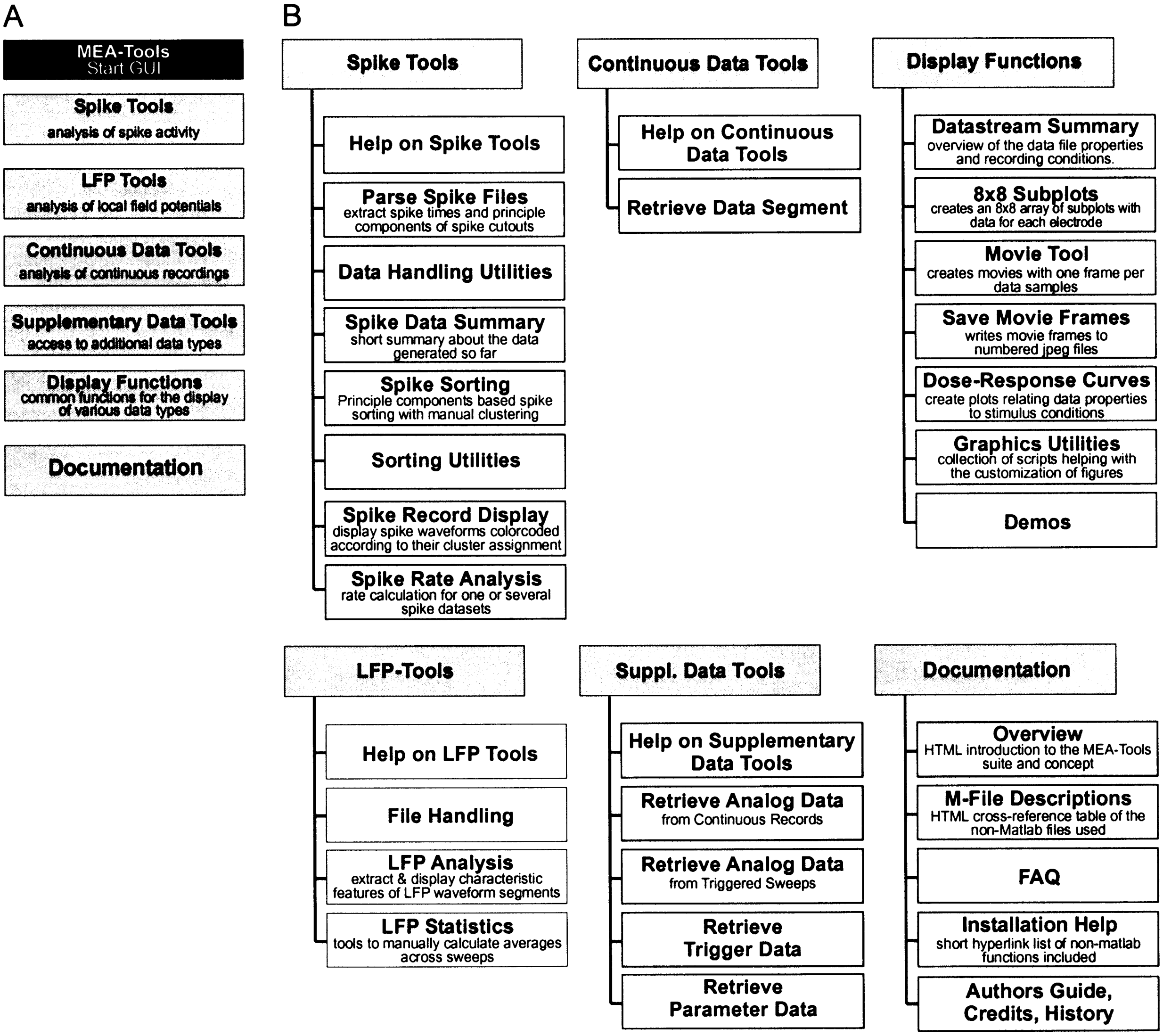

A
MEA-Tools Start GUI
Spike Tools analysis of spike activity
LFP Tools analysis of local field potentials
Continuous Data Tools analysis of continuous recordings
Supplementary Data Tools access to additional data types
Display Functions common functions for the display of various data types
Documentation
B
Spike Tools
Help on Spike Tools
Parse Spike Files extract spike times and principle components of spike cutouts
Data Handling Utilities
Spike Data Summary short summary about the data generated so far
Spike Sorting Principle components based spike sorting with manual clustering
Sorting Utilities
Spike Record Display display spike waveforms colorcoded according to their cluster assignment
Spike Rate Analysis rate calculation for one or several spike datasets
Continuous Data Tools
Help on Continuous Data Tools
Retrieve Data Segment
Display Functions
Datastream Summary overview of the data file properties and recording conditions.
8x8 Subplots creates an 8x8 array of subplots with data for each electrode
Movie Tool creates movies with one frame per data samples
Save Movie Frames writes movie frames to numbered jpeg files
Dose-Response Curves create plots relating data properties to stimulus conditions
Graphics Utilities collection of scripts helping with the customization of figures
Demos
LFP-Tools
Help on LFP Tools
File Handling
LFP Analysis extract & display characteristic features of LFP waveform segments
LFP Statistics tools to manually calculate averages across sweeps
Suppl. Data Tools
Help on Supplementary Data Tools
Retrieve Analog Data from Continuous Records
Retrieve Analog Data from Triggered Sweeps
Retrieve Trigger Data
Retrieve Parameter Data
Documentation
Overview HTML introduction to the MEA-Tools suite and concept
M-File Descriptions HTML cross-reference table of the non-Matlab files used
FAQ
Installation Help short hyperlink list of non-matlab functions included
Authors Guide, Credits, History

Fig. 1. Top (A) and second level (B) menu structure of the MEA-Tools. The six main items were designed to match tasks used with particular data types and for common display functions.

individual or groups of spikes, identify groups, and mark them for assignment to a cluster. The results can be reviewed with respect to the original spike series. PCs and cluster boundaries can be reused on other spike data records from the same experiment.

We did not include automatic clustering because the available techniques that produce reliable results require unacceptable processing time for experiments resulting in 10^4 and often more than 10^5 spikes detected on one electrode.

At this point we have not included facilities to use on stereo- or tetrode-based data that allow the comparison of waveforms recorded with different, closely positioned, electrodes from the same neuron and use these for spike sorting. These in some cases provide more information than is available from single-electrode recordings and thus improve the possibility to separate different sources. Since sorting is based on a matrix of parameters (the component scores) describing the waveform of each spike, it can be easily extended by calculating the desired descriptive parameters for other types of base transformations and feeding these into the display routines. Several additional utilities facilitate calculating PCs from matrices of waveforms, merging data from different files, editing of descriptive variables as well as import and export of data from and to ASCII files. The time series thus created are available for further processing.

In MEA experiments, it is often desirable to review the results of an analysis with respect to the geometry of the MEA, that is, the position of its electrodes in the tissue. The calculations therefore provide the possibility to store the results in the common format used by the display functions described below.

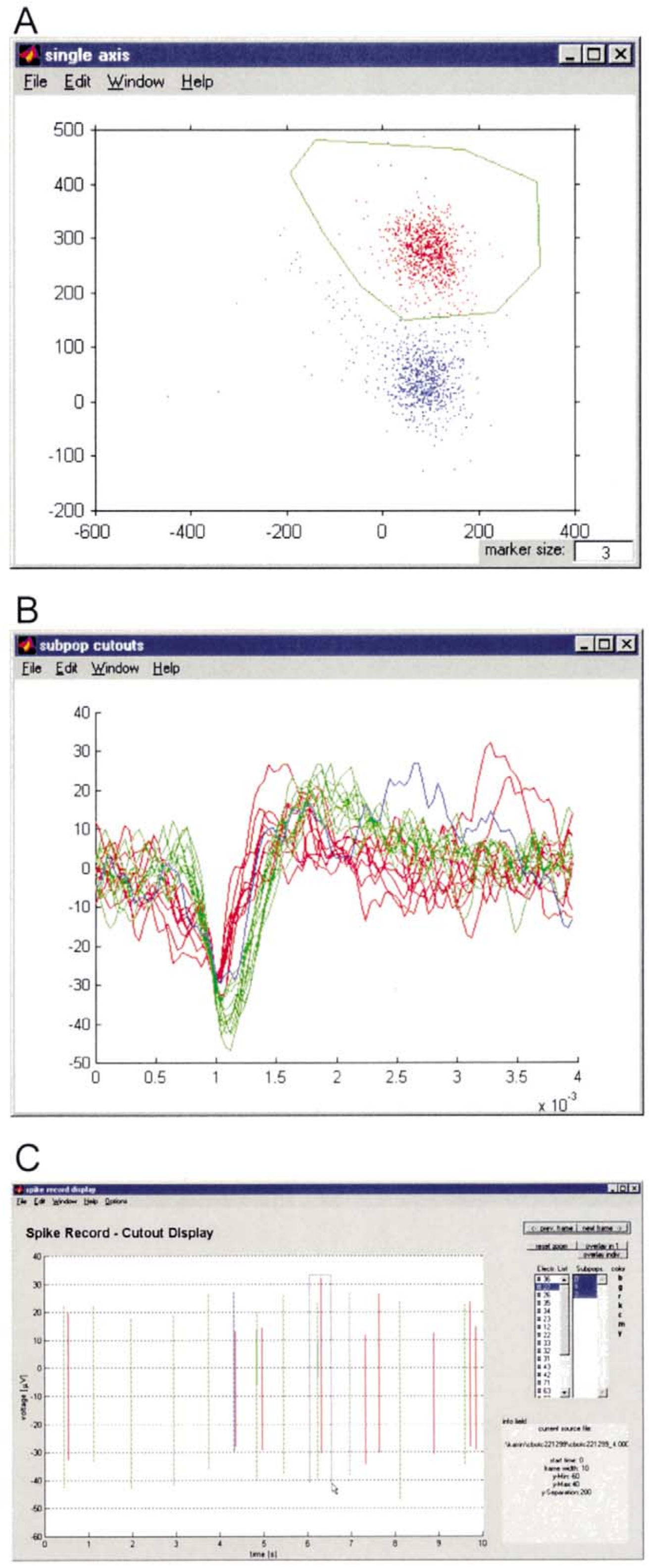
3.5. Local field potential analysis section

These tools are used not only to analyze conventional LFP waveforms, but are suitable for other trial-like data as well, in particular the display functions. The only prerequisite is that a common trigger time can be defined for each trial. For example, the tool was used to analyze LFP responses to repeated stimuli in hippocampal and cortical slices, episodes of cardiac myocyte action potentials, long-term potentiation protocols, etc. The analysis of LFPs generally evaluates the waveform of the response of a population of cells in which the individual responses of numerous cells overlap, e.g. the size and timing of certain components relative to an external event and relative to each other. Again, it is necessary to analyze and summarize these data with respect to the underlying tissue architecture. Frequently, pre-processing such as filtering, subtraction of other waveforms, averaging, and spatial interpolation is required before the values of interest (minima, maxima, delays, slopes, etc.) can be determined. The GUI for the extraction of LFP features incorporates the most common of these procedures. Preset tasks can be applied to an arbitrarily large set of trials in a batch process.

Since the display suitable for a particular parameter and experiment may vary, the graphical evaluation of the results has been separated from the LFP extraction procedure. These functions include line, 2D and 3D plots, pseudocolor and animated displays. The values displayed are extracted to a separate variable for command line access. The dimensions of this variable depend on the type of electrode array used and are again compatible with other display tools. The functionality of this tool set is illustrated by an example in Appendix A.

3.6. Gap-free recordings

Continuous, gap-free data records, including external data streams, e.g. from intracellular electrodes, which can be co-recorded on our system, are handled with a separate tool. Since the structure and the goals of analysis of such data vary widely, the parameters of interest cannot be defined in general. The MEA-Tools provide a GUI-based utility to retrieve arbitrary large stretches from the file into a matrix, either formatted to fit the display utilities or for a few electrodes only. This tool was used, for example, for continuous records from cardiac myocyte cultures or of spontaneous LFPs in organotypic cultures. The data can be viewed via the common display functions described in the next paragraph.



current source file:
\\kain\cbolc221299\cbolc221299_4.00C
start time: 0
frame width: 10
y-Min: -60
y-Max: 40
y-Separation: 200

Fig. 2

Fig. 2. Illustration of some main elements of the spike sorting routine. Two-dimensional projections of selected PC weights for each spike allow the identification of clusters (A). These can be manually delimited, resulting in the segregation of spike cut-outs with similar waveforms, identified by their respective colors (B). These assignments are used to separate the spikes within the time series of spikes recorded at a given electrode (C) and to create separate time series for each presumptive neuron.

3.7. Supplementary data streams

Like other data acquisition systems, MCRack supports the recording of several other data sources, such as intracellular voltages recorded with an external amplifier, digital signals and trigger events. Furthermore, some online procedures are available for simple analysis tasks. These data streams are accessible through the MEA-Tools routines in a separate section to facilitate the addition of utilities accessing other file-types.

3.8. Display functions

Display functions for multi-electrode recordings extend those used for single-electrode studies to represent the results (a) with respect to the geometry of the electrode array, and (b) in comparison between the electrodes recorded simultaneously. This requires the reduction of results to suitable descriptive parameters and their display in multiple dimensions. An example is given in Appendix A.

The display functions in the MEA-Tools that represent geometrical relations are designed to work with any type of data as long as it is presented in a numeric $M \times N$ matrix, e.g. $64 \times N$ to be compatible with 8×8 arrays. They can thus be used to visualize raw data recorded directly or the results of preceding steps of data analysis. The simplest display of that kind is the representation of the data recorded, e.g. voltages, in subplots arranged according to the layout of the electrode array (Fig. 3). Other display styles are available.

While being accurate in a technical sense, spatio-temporal patterns are difficult to perceive in this representation. To visualize such structures in 2D or 3D pseudocolor plots proved to be very useful. In the primary rendering, for example, the voltage distribution at an instant in time, the picture is dominated by the coarse checkerboard pattern. Interpolating across the patches reveals patterns correlated with anatomical structures (Fig. 4). Obviously, changes across time can then be interpreted as frames in a movie and viewing such a movie is frequently very instructive. The Movie Tool creates a MEA-geometry oriented pseudocolor movie of either the data in an MCRack record or any $64 \times N$ array, which can be viewed in 2D or 3D display with different degrees of interpolation and from various perspectives.[4]

In many cases, changes of a particular parameter, for example LFP amplitudes, latencies or other changes of firing rates, pseudocolor rendering of ratios or differences give compact displays of the results, relating such changes to the position of the electrodes in the tissue (Fig. 5).

When changes are induced by variation of experimental conditions, they are often summarized in a stimulus/response or dose/response curve. This mostly requires averaging of the response across a temporal window, e.g. while a particular concentration of the drug under test is applied, or across the results from several triggered sweeps, e.g. trials with a certain stimulus condition. The dose–response curve function is designed to assist such calculations.

3.9. Documentation

All m-files and GUIs are documented in MATLAB style. The interdependencies between these files are summarized in a hyperlinked HTML document[5]. The documentation, besides the general descriptions on concept and use, therefore includes a hyperlink library summarizing the files, their help text and calls on each other. The individual files were further classified according to their field of application in a MATLAB style hyperlinked *contentclasses.m* file. Furthermore, links were included to guides for authors, with help on adding tools to the menus and on adjusting the tools for new MEA types. We hope that this documentation, together with extensive documentation within the code, will help new contributors to add functions they would like to contribute to the neuroscience community.

While providing the framework for an open source toolbox, we are aware that numerous functions currently in use are not included. We hope that primitives and advanced functions used in various labs working with multi-electrode data will be contributed and thus allow a rapid distribution in the scientific community. In addition, since Linux does not provide OLE functionality and the UNIX version of the GUI functions in MATLAB differ somewhat from the Microsoft Windows version the toolbox is currently not fully functional under Linux. This concerns access to the MCRack data files in particular.

Acknowledgements

Parts of this work were funded by the German BMBF and the Land Baden-Württemberg. We would also like

[4] Examples recorded from various preparations may be downloaded as MPEG-movie illustrating the extracellular signal spreading through the preparation from our website: www.brainworks.uni-freiburg.de/projects/mea/meatools/moviedemo/movietool_example.html. In these, all movie frames are interpolations based on the signal on 60 electrodes.

[5] Created with a modified version of the HTML toolbox written by B.K. Alsberg.

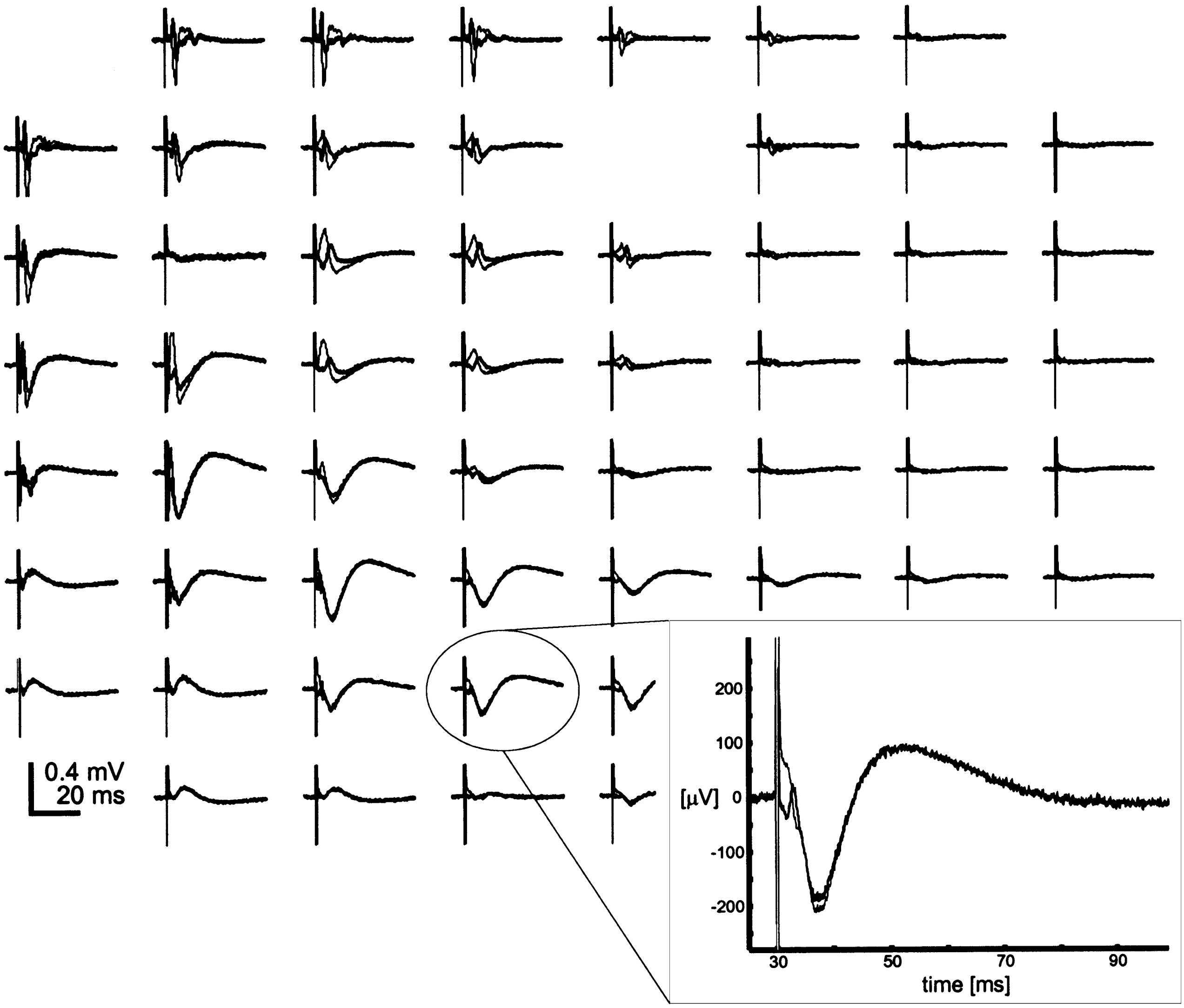


Fig. 3. MEA record of LFP responses in an acute slice of the rat hippocampus. The subplots are positioned according to the geometry of the electrode array used. Each subplot represents the voltage at that electrode from 5 ms prior to 35 ms after the electrical stimulus (marked by the vertical line indicating the stimulus artifact generated by the 320 ms pulse). Thin traces represent single trial raw data; thick traces show postsynaptic components revealed by subtraction of the response in O–Ca^{2+} buffer.

to thank the users of the MEA-Tools for beta-testing and pointing out opportunities for improvement and C.-L. von Schlabrendorff for overhauling some of the files. We thank Th. Wachtler-Kulla and H. Thurm for helpful comments on the manuscript.

Appendix A: Microelectrode arrays

In our laboratory we use planar, substrate-integrated MEAs to analyze LFPs and spike activity from in vitro preparations ranging from acute slices of the neocortex, cerebellum and hippocampus to organotypic cultures of various brain tissues, retina explants and cardiac myocyte cultures (Egert et al., 1998, 2000; Banach et al., 2000; Jost et al., 2001). MEAs are planar arrays of 60 electrodes produced with thin-film technology on glass substrates (Nisch et al., 1994). The electrodes are positioned on an 8 × 8 grid with the corner positions empty. Extracellular recordings are performed with a MEA-1060 system.

The questions asked in these preparations and the analyses necessary to answer them differ widely. The following example illustrates the functionality of the toolbox for a data set of triggered sweeps with LFPs recorded with the MCRack software in an acute brain slice of the rat hippocampus. Common tasks in LFP analysis include filtering, peak identification, detection of minima and maxima, and averaging across trials. In this example we generate a plot giving the degree of potentiation at each recording site in a long-term potentiation experiment. Italic print indicates menu

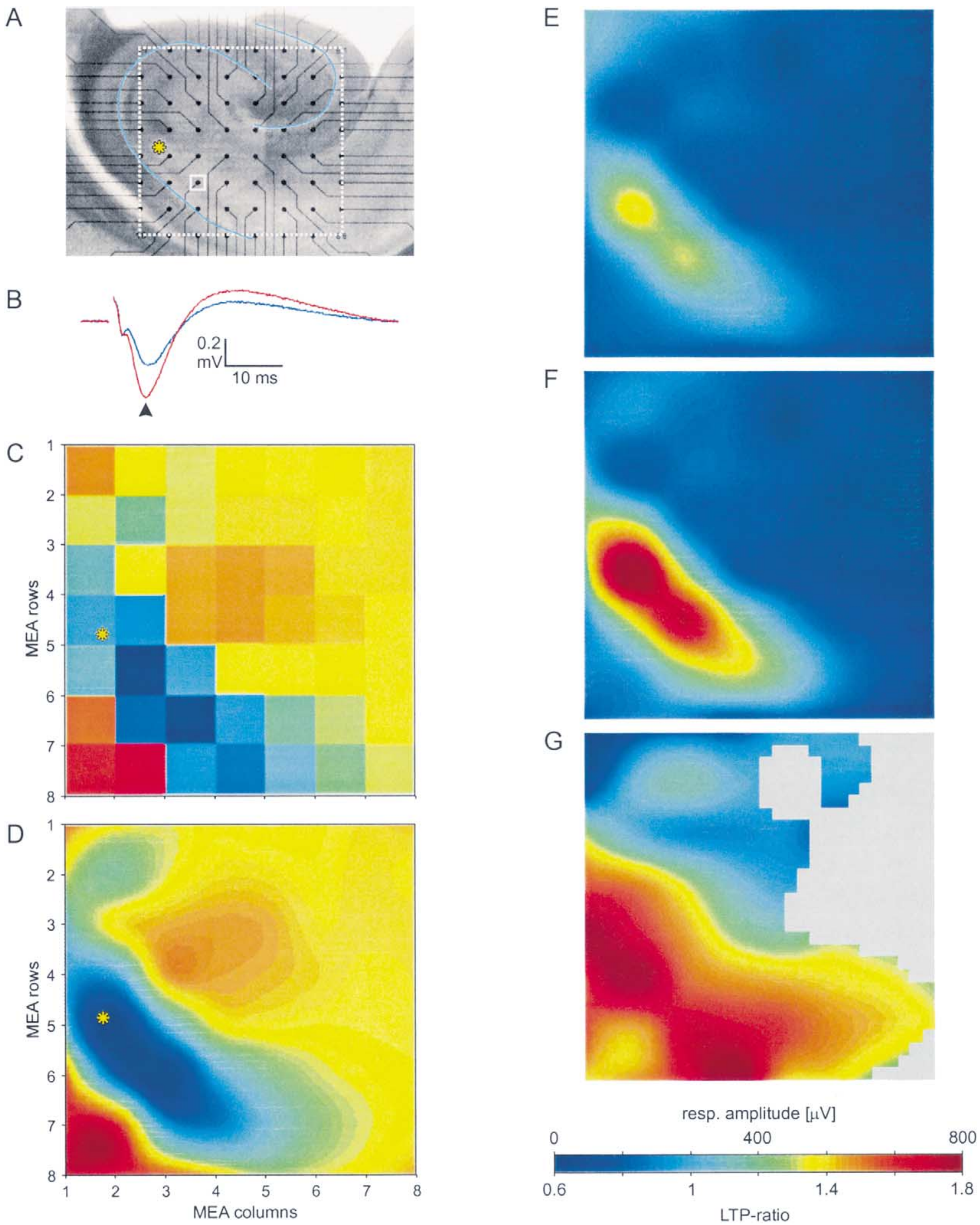
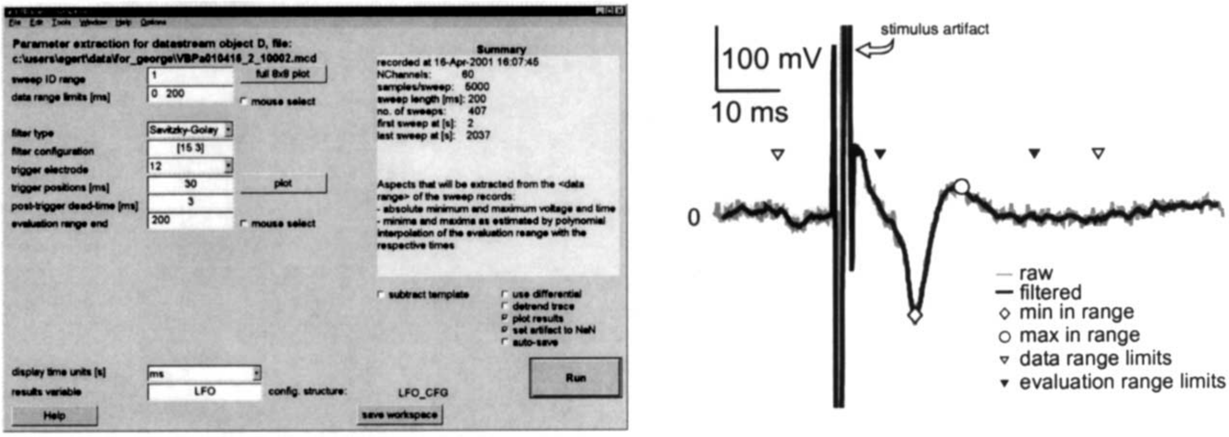


Fig. 4

items or GUI field labels. Per default, all times are given in ms (selectable *display time units*). The results are stored in the *results variable* entered in the corresponding field.

Triggered sweeps often encompass a stretch of time recorded before the trigger event and a time window after that event. Multiple trigger events may occur in some studies.



Parameter extraction for datastream object D, file:
c:\users\egert\data\for_george\VBPa010416_2_10002.mcd

sweep ID range: 1
data range limits [ms]: 0 200 □ mouse select
filter type: Savitzky-Golay
filter configuration: [15 3]
trigger electrode: 12
trigger positions [ms]: 30
post-trigger dead-time [ms]: 3
evaluation range end: 200 □ mouse select

Summary
recorded at 16-Apr-2001 16:07:45
NChannels: 60
samples/sweep: 5000
sweep length [ms]: 200
no. of sweeps: 407
first sweep at [s]: 2
last sweep at [s]: 2037

Aspects that will be extracted from the <data range> of the sweep records:
- absolute minimum and maximum voltage and time
- minima and maxima as estimated by polynomial interpolation of the evaluation reange with the respective times

□ subtract template
□ use differential
□ detrend trace
☑ plot results
☑ set artifact to NaN
□ auto-save

display time units [s]: ms
results variable: LFO
config. structure: LFO_CFG

Fig. 5. The interface guiding the user through the extraction of important properties of LFPs. When the GUI (left panel) is created the fields are filled with information from the datastream object and a default configuration file, which can be adjusted by the user. Various options are available via the *Options* submenu in the main menu or through popup-menu within the GUI. The meaning of some of the fields and the values found are illustrated for a single-electrode sample (right panel). This type of analysis is performed for all sweeps on all electrodes in the data file.

File access is established by defining a 'datastream object' in the workspace, either with a GUI menu command *LFP Tools/File Handling Utilities/Load Sweep Data* or on the MATLAB command line.

The analysis of the LFPs is handled via the *LFP Tools/LFP extraction* menu command. The first entry allows the user to indicate the *sweep ID range* to analyze by entering a list of integers in MATLAB notation identifying a sequence of sweeps in any order, or a corresponding variable.

Minimizing the amount of data transfer and analysis, and thus the memory load, will optimize processing speed. Defining *data range limits* for the relevant time window, which needs to include all trigger events, crops the data retrieved from the file. *Data range limits* may be assigned by mouse clicks in a preview window of a single set of data from one electrode.

Depending on the data type used, it is often necessary to apply low pass filtering to the data. While a sliding window average may sometimes suffice, *filter type* will allow you to select a Savitzky–Golay filter (Savitzky and Golay, 1964), which preserves the temporal structure and amplitude of low frequency components to a larger extent, while providing the same noise reduction. The *filter configuration* can be defined by giving the half width of the filter as number of samples and the order of the polynomial used to fit the data locally (i.e. two integer values). Large numbers for the first parameter will lower the cut-off frequency, large values for the second will increase it. With sampling frequencies of 25 kHz, a half width of 15 samples and 3rd order polynomial are suitable settings for LFPs. The results are shifted for phase correction to match the peak latencies to the unfiltered waveform.

Trigger positions (in this case a single stimulation time) are read from the datastream object per default, but can be defined manually or with a mouse click in a sample trace window from the selected *trigger electrode*. The analysis window evaluated for each trigger event is delimited by a '*post-trigger dead time*' period and the '*end of the evaluation range*' window. Again, manual and mouse selection is supported. If multiple trigger times are given, windows of the same length will be evaluated after each trigger.

In experiments using electrical stimuli, a stimulus artifact may overlap in part with the neuronal response. If a sample of that stimulus artifact can be obtained without the neuronal response, e.g. after adding the Na^+-channel blocker tetrodotoxin, this template can be subtracted from the data before processing. For the LFP processing procedures, individual templates can be

Fig. 4. Pseudocolor representation of the properties of LFPs in the hippocampus after electrical stimulation. LFPs were induced in the hippocampal slice at the position marked by the asterisk (A). The prominent morphological features of the hippocampus, the dentate gyrus and the CA region, were clearly visible (marked by lines). The blue line in B illustrates the response observed at one electrode (white square in A) under standard conditions. The red line gives the LFP response after LTP induction. The voltages of the baseline response found on all electrodes at one instant in time (9 ms after the stimulus in this example) were used to calculate a surface representing the estimated distribution of activity in the tissue (C). The swatches are defined by the electrode positions at their corners and give a coarse estimate for that area. The dominating checkerboard pattern, however, hampers the comparison of the distribution with the tissue anatomy. Interpolation (bicubic spline) helps to reveal the structure and indicates that the voltage distribution follows prominent anatomical features of the slice (D). The panels E–G illustrate how the processing of the LFPs can be used to estimate synaptic plasticity in this preparation. After a baseline period, the slice was stimulated with three bursts of pulses (tetanic stimulus) which result in a long lasting increase of the postsynaptic response. The response magnitudes to the test stimulus obtained before and after these bursts are shown in E and F respectively. The ratio of these respones (G, calculated for suprathreshold values of 3 × standard deviation of the baseline only) reveals the location of maximal potentiation along the CA1 region. Movies and series of JPEG images can be generated to create MATLAB or standalone MPEG movies.

defined for each sweep. A separate function assists in the generation of one or several templates with the help of a MATLAB script.

All results are stored in a variable rapidly accessible during further processing steps without referring to the raw data files again. Various display options are available to review the validity of the results for individual sweeps, the overall outcome, and for different aspects of the data extracted, e.g. amplitudes, minima, times etc. The latter is best accessible via the display routine *display LFP features* in the LFP Tools submenu.

References

Abeles M, Gerstein GL. Detecting spatiotemporal firing patterns among simultaneously recorded single neurons. J Neurophysiol 1988;60:909–25.

Aertsen AMHJ, Gerstein GL. Dynamic aspects of neuronal cooperativity: fast stimulus-locked modulations of effective connectivity. In: Krüger J, editor. Springer series in synergetics: neuronal cooperativity. Berlin: Springer, 1991:52–67.

Banach K, Egert U, Hescheler J. Excitation spread between heart cells derived form embryonic stem (ES) cells. Pflug Arch Eur J 2000;439(Suppl. 6):13–9.

Deribaupierre F, Abeles M, Deribaupierre Y. Functional interaction between neurons at different tangential distances in the cat auditory-cortex. Experientia 1986;42:709.

Egert U, Schlosshauer B, Fennrich S, Nisch W, Fejtl M, Knott Th, et al. A novel organotypic long-term culture of the rat hippocampus on substrate-integrated multielectrode arrays. Brain Res Prot 1998;2:229–42.

Egert U, Schrenk K, Kapfhammer J, Metzger F. Electrophysiological characterization of long and short-term drug effects in acute slices and organotypic cultures of the cerebellum. N-S Arch Pharmacol 2000;361:359.

Georgopoulos AP, Taira M, Lukashin A. Cognitive neurophysiology of the motor cortex. Science 1993;260:47–52.

Gerstein GL, Bedenbaugh P, Aertsen AMHJ. Neuronal assemblies. IEEE Trans Biomed Eng 1989;36:4–14.

Hulata E, Segev R, Shapira Y, Benveniste M, Ben-Jacob E. Detection and sorting of neural spikes using wavelet packets. Phys Rev Lett 2000;85:4637–40.

Jost B, Aertsen A, Egert U. Dopamine D_2/D_3 receptor mediated modulation of cerebellar spike activity. In: Kreutzberg GW, Elsner N, editors. Proceedings of the 28th Göttingen Neurobiology Conference. Stuttgart: Thieme, 2001, p. 865.

Kim KH, Kim SJ. Neural spike sorting under nearly 0-dB signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier. IEEE Trans Biomed Eng 2000;47:1406–11.

Lindsey BG, Shannon R, Gerstein GL. Gravitational representation of simultaneously recorded brainstem respiratory neuron spike trains. Brain Res 1989;483:373–9.

Nicolelis MAL. Methods for neural ensemble recordings. Boca Raton: CRC Press, 1999.

Nisch W, Böck J, Egert U, Hämmerle H, Mohr A. A thin film microelectrode array for monitoring extracellular neuronal activity in vitro. Biosens Bioelectron 1994;9:737–41.

Sahani M. Latent variable models for neural data analysis. Pasadena, CA; PhD Thesis, Caltech, 1999.

Salganicoff M, Sarna M, Sax L, Gerstein GL. Unsupervised waveform classification for multi-neuron recordings: a real-time, software-based system. I. Algorithms and implementation. J Neurosci Meth 1988;25:181–7.

Sarna MF, Gochin P, Kaltenbach J, Salganicoff M, Gerstein GL. Unsupervised waveform classification for multi-neuron recordings: a real-time, software-based system. II. Performance comparison to other sorters. J Neurosci Meth 1988;25:189–96.

Savitzky A, Golay MJE. Smoothing and differentiation of data by simplified least squares procedures. Anal Chem 1964;36:1627–39.

Vaadia E, Haalman I, Abeles M, Bergman H, Prut Y, Slovin H, et al. Dynamics of neuronal interactions in monkey cortex in relation to behavioural events. Nature 1995;373:515–8.