Biomed Tech 2013; 58 (Suppl. 1) © 2013 by Walter de Gruyter · Berlin · Boston. DOI 10.1515/bmt-2013-4181

STIMFIT: A FAST VISUALIZATION AND ANALYSIS ENVIRONMENT FOR CELLULAR NEUROPHYSIOLOGY

Schlögl A[1], Jonas P[1], Schmidt-Hieber C[2,3] and Guzman SJ[1],

[1]Institute of Science and Technology (IST), Austria

[2]Wolfson Institute for Biomedical Research, University College London, London, UK

[3]Department of Neuroscience, Physiology and Pharmacology, University College London, London, UK.

alois.schloegl@ist.ac.at

Abstract: *Stimfit is a free cross-platform software package for viewing and analyzing electrophysiological data. It supports most standard file types for cellular neurophysiology and other biomedical formats. Its analysis algorithms have been used and validated in several experimental laboratories. Its embedded Python scripting interface makes Stimfit highly extensible and customizable.*

Keywords: *Electrophysiology, patch-clamp, data analysis, biomedical data formats, free software.*

Introduction

Electrical activity that arises from cellular and subcellular processes is commonly studied with intracellular techniques (e.g patch-clamp or sharp electrodes) in repetitive epoch-like events at high precision and temporal resolution. A detailed study of the acquired electrical signals generally requires both fast data visualization and ready access to complex analysis routines. We developed Stimfit[1], an open-source software package available for several operating systems, for fast and easy visualization of such recordings. Using the BioSig[2] library as an optional backend for file I/O [1], Stimfit supports more than 20 biomedical formats, including those most commonly used in cellular electrophysiology (see Table 1). It features robust algorithms for analysis in neurophysiology (e.g. calculations of synaptic and action potential latencies, rise and slope values, thresholds, etc.). These measurements are indicated to the user by a pair of cursors on the screen (see Figure 1) and are updated upon navigation along the recording. In addition, detection routines of spontaneous events are also provided [2, 3]. An implementation of the Levenberg-Marquardt algorithm [4] is used for fitting data to standard mathematical functions (single and multiexponentials) and common models in cellular neuroscience. Moreover, Stimfit is highly customizable with an embedded Python shell giving access to common scientific Python libraries such as NumPy or SciPy [5]. In summary, we present a cross-platform analysis environment that is easy to use and can be adapted to individual experimental requirements with Python [6].

[1]Stimfit is available on line from `http://www.stimfit.org`

[2]BioSig is available on line from `http://biosig.sourceforge.net/`

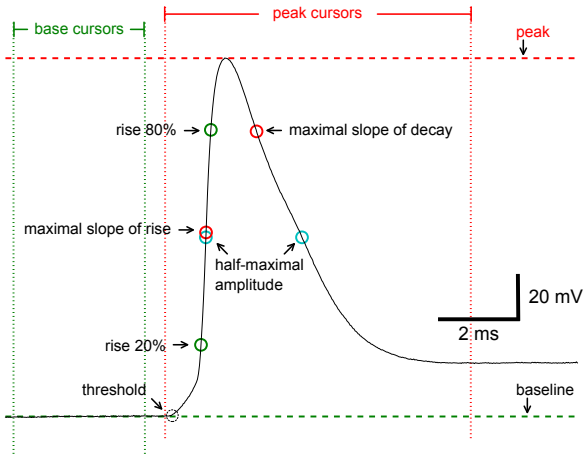

Figure 1: Principal measurements obtained when base and peak cursors are positioned on an action potential in a Stimfit session.

Table 1: File formats supported by Stimfit - only formats relevant for cellular electrophysiology are listed. (*) indicates support through biosig, (+) indicates improved support through biosig.

file type	read	write
Axon text	yes	yes
Axon binary v1	yes	no
Axon binary v2	yes	no
CFS binary	yes(+)	yes
HDF5 files	yes	yes
Axograph	yes	no
HEKA	yes(+)	no
Igor binary	yes(*)	yes
GDF	yes(*)	yes(*)
NEURON	yes(*)	no

Methods

Stimfit is written in C++ with some external libraries (see Table 2) and extensions in Python. To create the GUI, the wxWidgets library was chosen because it provides cross-platform support [7]. The BioSig library has recently been added to support additional biomedical data formats and the GDF file format [8]. The source code can be compiled with an ANSI/ISO C++ compliant compiler and has been tested with the GNU compiler for GNU/Linux, Mac OS

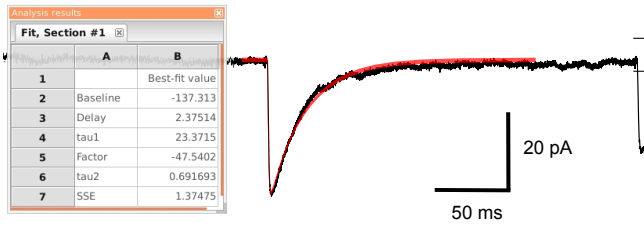

Figure 2: Fitting of an averaged excitatory postsynaptic current to a biexponential function by least-square optimization implementation in Stimfit.

Table 2: List of C++ external libraries.

library	purpose
BioSig	Biomedical file formats, input/output
boost	Shared pointers and arrays
FFTW	Fast Fourier transform for filtering
HDF5	Support to HDF5 file format
LAPACK	Linear algebra, non-linear regression
levmar	Non-linear regression
wxWidgets	Graphical user interface
Python	Python interpreter
wxPython	Embedded Python shell

X and with Microsoft Visual C++ 2008® for Microsoft® Windows. Alternatively, the the MinGW-cross-compiler environment (MXE) [9] tool-chain can be used for building the Windows version to avoid proprietary compilers. Some regressions when using MXE, such as missing Python and ABF support, will eventually be fixed in future releases.

Fast visualization by avoiding line plot redundancies
To efficiently plot data obtained at high sampling rates, two algorithms are employed. A down-sampling algorithm is used to reduce the number of lines plotted between two pixel columns by choosing the maximum and minimum sampling value within a column and plotting one single line between them. An anti-aliasing algorithm is used to connect pixel columns.

Event detection and fitting algorithms
Spontaneous or miniature synaptic events can be detected by minimizing the sum of squared errors between a template waveform and recording periods. To decide whether an event has occurred, the user can choose between two criteria: either the correlation coefficient between optimal template and recording [3], or the ratio of template scaling factor and noise standard deviation [2].

Python integration
To provide a customizable analysis platform, Stimfit includes a command-line Python interface to control the GUI and access the analysis algorithms. In addition, the Python shell gives access to the extensive ecosystem of scientific libraries that are available for Python, such as NumPy or SciPy. In addition, a stand-alone Python module (stfio) allows users to read and write standard electrophysiology file formats from Python without the Stimfit GUI.

Discussion

Robust algorithms for visualization and analysis are necessary for our understanding of the electrical signals in neuroscience. A free software model facilitates reproducibility, one of the cornerstones of scientific progress. For that reason, Stimfit is released under the GNU general public license (GPL), and all its libraries and dependencies are released under free software licenses. Among all programming languages under the GPL license, Python was chosen as a scripting language for Stimfit because of its widespread use in neuroscience [10]. Thus, as an alternative to proprietary software solutions, Stimfit combines an intuitive user interface with a general-purpose programable environment.

Bibliography

[1] C. Vidaurre, T. H. Sander, and A. Schlögl, "BioSig: The free and open source software library for biomedical signal processing," *Computational intelligence and neuroscience*, vol. 2011, p. 12 pages, 2011.

[2] J. D. Clements and J. M. Bekkers, "Detection of spontaneous synaptic events with an optimally scaled template.," *Biophys J*, vol. 73, pp. 220–229, Jul 1997.

[3] P. Jonas, G. Major, and B. Sakmann, "Quantal components of unitary EPSCs at the mossy fibre synapse on CA3 pyramidal cells of rat hippocampus.," *J. Physiol.*, vol. 472, pp. 615–663, Dec 1993.

[4] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++." [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004. [Accessed on 13 Mar. 2013.].

[5] T. E. Oliphant, "Python for scientific computing," *Computing in Science Engineering*, vol. 9, no. 3, pp. 10–20, 2007.

[6] G. van Rossum, "The python programming language," *Python Software Foundation*, 2007.

[7] J. Smart, K. Hock, and S. Csomor, *Cross-Platform GUI Programming with wxWidgets (Bruce Perens Open Source)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2005.

[8] A. Schlögl, "GDF - a general dataformat for biosignals." [web page] `http://arxiv.org/abs/cs/0608052`, 2006. [Accessed on 13 Mar. 2013.].

[9] V. Grabsch, "Mingw-cross-compiler environment." [web page] `http://mxe.cc`, 2007-2013. [Accessed on 13 Mar. 2013.].

[10] A. P. Davison, M. L. Hines, and E. Muller, "Trends in programming languages for neuroscience simulations.," *Front Neurosci*, vol. 3, no. 3, pp. 374–380, 2009.