# STIMFIT; A FAST VISUALIZATION AND ANALYSIS ENVIRONMENT FOR CELLULAR NEUROPHYSIOLOGY

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

[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 software for viewing and analyzing electrophysiological data that is released for all major operative system. It supports the standard file types for cellular neurophysiology and other biomedical formats. Its analysis algorithms have been used and validated in several experimental laboratories. Stimfit is extensible and highly customizable due to a Python scripting interface.*

***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. Due to the high precision and temporal resolution of the intracellular techniques, a detailed analysis on acquired electrical signals is possible but it generally requires both user visualization and a complex analysis. We developed Stimfit[1], an open source environment that allows fast and easy visualization of recordings with a common graphical user interface (GUI) that is available for several operative systems [1]. Using the BioSig[2] library as a backend for file I/O [2] 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 visualized to the user by locating 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 [3, 4]. An implementation of the Levenberg-Marquardt algorithm [5] is used to provide a highly customizable user experience 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 [6] shell giving access to common scientific Python libraries such as NumPy or SciPy [7]. We present a cross-platform analysis environment that is easy to use and can be adapted to individual experimental requirements with Python.

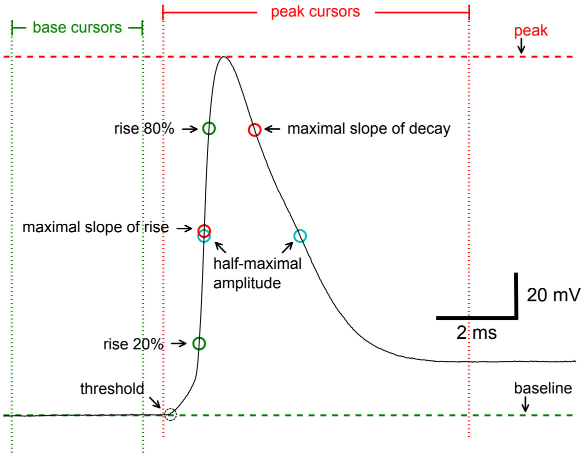


Figure 1: Principal measurements obtained when base and peak cursors are positioned on an action potential in a Stimfit session.

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

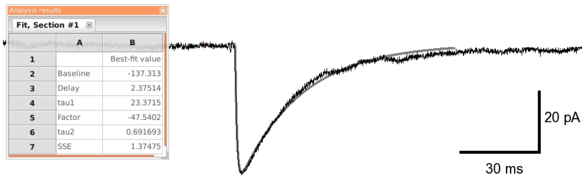
| **file type** | **read** | **write** |
|---|---|---|
| Axon text | yes | yes |
| Axon binary v1 | yes | no |
| Axon binary v2 | yes | no |
| CFS binary | yes(+) | yes |
| HDF5 files | yes | yes |
| Axograph | yes | no |
| HEKA | yes(+) | no |
| Igor binary | yes(*) | yes |
| GDF | yes(*) | yes(*) |
| Neuron | yes(*) | no |
| many more (*) | | |

## 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 a cross-platform support and a native environment for the Stimfit user. The libbiosig library has been recently added to support additional biomedical data formats and the GDF file format [8] as default. In general, the source code can

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

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



|   | A | B |
|---|---|---|
| 1 |  | Best-fit value |
| 2 | Baseline | -137.313 |
| 3 | Delay | 2.37514 |
| 4 | tau1 | 23.3715 |
| 5 | Factor | -47.5402 |
| 6 | tau2 | 0.691693 |
| 7 | SSE | 1.37475 |

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

be compiled with any ANSI/ISO C++ compiler and has been tested with the GNU compiler for GNU/Linux and Mac OS X and with Microsoft Visual C++ 2008® for Microsoft® windows. However, the tool-chain for building the windows version has been extended by using the MinGW-cross-compiler environment (MXE) [9], to avoid proprietary compilers and to maintain all dependencies to the external libraries. This resulted in some regressions (Python and ABF format are currently not usable), but eventually this should be fixed.

**Fast visualization by avoiding line plot redundancies**

To present data obtained at high sampling rates, two algorithms are employed. A down-sampling algorithm that reduces the number of lines plotted between two consecutive sampling points by choosing the maximum and minimum sampling value within a column pixel and plotting one single line between them. An anti-aliasing algorithm is used to connect pixel columns.

**Event detection and fitting algorithms**

Detection of spontaneous or miniature synaptic events is performed by minimization of the sum of squared errors between an template waveform and the time course of a recording. To decide if an event has occurred the user can choose a threshold value above the correlation coefficient between template and recording [4] or above the noise standard deviation [3]

**Python integration**

To provide a customizable analysis platform, Stimfit is expanded with a command-line Python interface that controls the GUI and access the algorithms for analysis. This allows the user to evaluate the results obtained with Stimfit in a more specific scientific context (e.g with NumPy). In addition, a standalone Python module (stfio) allows reading and writing standard electrophysiology file formats without the GUI.

## Discussion

Robust algorithms for visualization and data analysis are necessary for our understanding of the electrical signals in neuroscience. However, to ensure scientific progress, reproducibility should be guaranteed by a *free software* model of development. For that reason, Stimfit is released under the GNU general public license (GPL), and all its libraries and dependencies rely on GPL licenses. Among all programming languages under the GPL license, Python was chosen as extension for Stimfit because of its clear syntax and widely use in neuroscience [10]. As alternative to proprietary software solutions, Stimfit combines both an intuitive user-interface with a customizable environment with a general-purpose programming language of in neuroscience.

Table 2: List of C++ external libraries.

| library | purpose |
|---|---|
| BioSig | 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 |

## Bibliography

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.].

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

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

[8] A. Schlögl, “GDF - a general dataformat for biosignals,” vol. 19, pp. 1–15, 2006.

[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.