

ParamAP: Standardized Parameterization of Sinoatrial Node Myocyte Action Potentials

Christian Rickert¹ and Catherine Proenza^{1,*}

¹Department of Physiology and Biophysics, University of Colorado–Anschutz Medical Campus, Aurora, Colorado

ABSTRACT Sinoatrial node myocytes act as cardiac pacemaker cells by generating spontaneous action potentials (APs). Much information is encoded in sinoatrial AP waveforms, but both the analysis and the comparison of AP parameters between studies is hindered by the lack of standardized parameter definitions and the absence of automated analysis tools. Here we introduce ParamAP, a standalone cross-platform computational tool that uses a template-free detection algorithm to automatically identify and parameterize APs from text input files. ParamAP employs a graphic user interface with automatic and user-customizable input modes, and it outputs data files in text and PDF formats. ParamAP returns a total of 16 AP waveform parameters including time intervals such as the AP duration, membrane potentials such as the maximum diastolic potential, and rates of change of the membrane potential such as the diastolic depolarization rate. ParamAP provides a robust AP detection algorithm in combination with a standardized AP parameter analysis over a wide range of AP waveforms and firing rates, owing in part to the use of an iterative algorithm for the determination of the threshold potential and the diastolic depolarization rate that is independent of the maximum upstroke velocity, a parameter that can vary significantly among sinoatrial APs. Because ParamAP is implemented in Python 3, it is also highly customizable and extensible. In conclusion, ParamAP is a powerful computational tool that facilitates quantitative analysis and enables comparison of sinoatrial APs by standardizing parameter definitions and providing an automated work flow.

INTRODUCTION

Sinoatrial node myocytes (SAMs) perform their function as cardiac pacemaker cells by generating spontaneous action potentials (APs). Sinoatrial APs are characterized by a spontaneous depolarization of the membrane potential during diastole and by a slow upstroke velocity. Despite widespread recognition of these characteristics, limited quantitative information is available about the averages and variability of sinoatrial AP waveform parameters, or about the relationship between specific attributes of the sinoatrial AP and the underlying ionic currents. The collection and standardized analysis of SAM AP waveform parameters is therefore crucial for the development of improved mathematical models of the sinoatrial AP and is an important step toward ultimately gaining insight into the molecular processes that drive pacemaking. However, the analysis and comparison of AP parameters between studies is currently hindered by the lack of standardized parameter definitions and the absence of automated analysis tools.

To address these issues, we have developed a standalone computational tool, ParamAP, which automates sinoatrial AP analysis, and standardizes parameter definitions. ParamAP can be run on Windows, Mac OS X, or Linux operating systems. It accepts data input in the form of standard text files that can be generated by any data acquisition platform and it computes a total of 16 AP waveform parameters, which are output as text and graphical files.

ParamAP is available under the GNU General Public License 2.0 and archived on Zenodo to make it uniquely citable (1). File archives containing bundled installations of ParamAP, the executable source code, a user manual, and an example file are available at <https://github.com/crickert1234/ParamAp/>.

MATERIALS AND METHODS

ParamAP implementation

The implementation of ParamAP is based on recent installations of Python 3 (2), NumPy (3), SciPy (4), and Matplotlib (5). ParamAP runs its code from the command line (Windows) or the terminal (Linux, Mac OS X). However, a graphical user-interface has been implemented for the visualization of the data and for documentation of the results.

Submitted January 5, 2017, and accepted for publication July 5, 2017.

*Correspondence: catherine.proenza@ucdenver.edu

Editor: Eric Sobie.

<http://dx.doi.org/10.1016/j.bpj.2017.07.001>

© 2017 Biophysical Society.

The parameterization is performed in 10 consecutive steps:

- 1) Experimental data is imported from standard text files using NumPy's `loadtxt` (`np.loadtxt`) function and displayed with Matplotlib's `plot` function.
- 2) The user is given the option to provide numerical input to customize the analysis of a given dataset by providing parameters like the time interval, limiting values for the data, the filter strength, and peak detection scopes.
- 3) The imported data is filtered using the Savitzky-Golay (SG) filter (6) from SciPy's signal processing modules (`sp_sig.savgol_filter`).
- 4) All maxima (peaks) in the filtered data are detected using `sp_sig.argrelemax`, and all minima (troughs) in the filtered data are detected using `sp_sig.argrelemin`, respectively.
- 5) All detected extrema are examined for validity and excluded from analysis, if they do not comply with user-defined criteria.
- 6) The time intervals between maxima and minima are calculated using a k -dimensional tree (7) for nearest-neighbor lookup (`sp_spat.KDTree`) from SciPy's spatial algorithms and data structures modules to assign minima to their corresponding maxima. The result of this analysis is displayed with Matplotlib's `plot` function.
- 7) Raw unfiltered data segments consisting of time intervals with a central maximum and two adjacent minima are averaged.
- 8) The resulting average AP and its first derivative are then SG-filtered to improve the robustness of the peak/trough detection.
- 9) The AP waveform parameters are determined from the filtered averaged data and the filtered first derivative of the averaged data. The result of this analysis is displayed with Matplotlib's `plot` function.
- 10) After the analysis of the input data, three files are saved (the input parameters, the previously displayed output data, and a summary of output parameters) using the functions `np.savetxt` and `matplotlib.backends.backend_pdf.PdfPages`.

Details of the implementation can be found in the in-line comments of the source code. Instructions for the installation and use of ParamAP are described in the accompanying User Manual in the [Supporting Material](#).

Savitzky-Golay filtering and AP detection

SG filtering in ParamAP is used to improve the robustness of the maxima/minima detection by reducing high-frequency noise in the raw traces. It secondarily improves the alignment of data segments containing individual APs for averaging, thus improving the signal-to-noise ratio (SNR) of the averaged trace and the parameterization accuracy. The filter settings can be customized by adjusting the running average window size with the manual user input options for a given run (see the User Manual in the [Supporting Material](#) for more details).

For the identification of APs, which are comprised of a central maximum and two adjacent minima, individual distances from each of the minima to each of the maxima could be calculated. However, such a calculation would become computationally demanding as the number of APs increased. Instead, ParamAP employs the nearest-neighbor lookup using a k -dimensional tree search algorithm for one dimension, where the dataset is iteratively split into an arbitrary number of half-datasets using the median maxima as delimiters. This approach significantly reduces the number of calculated distances required to assign two neighboring minima to their corresponding maxima.

RESULTS AND DISCUSSION

SAM AP parameter definitions

The sinoatrial AP waveform parameter definitions implemented in ParamAP are based on previously published definitions (8–18) with some refinements made to allow

automated analysis and to accommodate a wide range of AP waveforms and firing rates. ParamAP computes a total of 16 parameters in three categories: membrane potentials, rates of change of the membrane potential, and time intervals (Fig. 1; Table 1).

Membrane potentials

The peak potential (PP) is defined as the maximum membrane potential during an AP. PP is used to localize individual APs in continuous recordings. Detection of PP relies on the use of the SG filter in the initial data preparation to reduce the number of potential maxima caused by noise artifacts. Further elimination of potential maxima is achieved by limiting candidate maxima to a user-defined range of acceptable PP values. The AP firing rate (FR) is determined from the number of identified APs divided by the time interval from the first to the last AP maximum.

The maximum diastolic potentials (MDP_1 , MDP_2) are the minimum membrane potentials that precede and follow

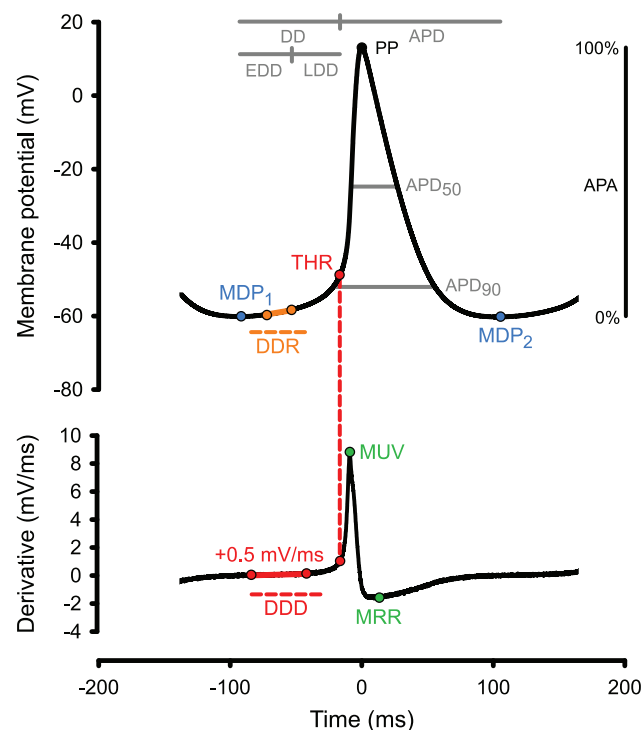


FIGURE 1 Schematic illustration of AP parameters reported by ParamAP. Shown here are the representative AP waveform (*top*) and first derivative of AP (*bottom*). PP (*top*, black dot), and MDP₁ and MDP₂ (*top*, blue dots), mark the extrema of the AP waveform. DDR (*top*, orange dots, and line) is determined from the best linear fit segment from a search within a predefined time interval. EDD, LDD, DD, APD, APD₅₀, and APD₉₀ are indicated with horizontal gray lines (*top*). APA is indicated by a vertical black line (*top*). MUV and MRR (*bottom*, green dots) mark the extrema of the first derivative of the AP waveform. THR (*top and bottom*, red dots, and dashed red line) is determined from the best linear fit (*bottom*, red dots, and red line) within a predefined time interval in the first derivative trace (DDD; *bottom*, dashed red line) plus 0.5 mV/ms. To see this figure in color, go online.

TABLE 1 List of AP Parameters Reported by ParamAP

Abbreviation	Parameter	Unit	Definition
APs	action potentials	—	number of averaged APs in a given run
FR	AP firing rate	AP/min	AP firing rate
CL	cycle length	ms	time interval MDP ₁ -MDP ₂
DD	diastolic duration	ms	EDD plus LDD
EDD	early DD	ms	time from MDP ₁ to end of linear fit for DDR
LDD	late DD	ms	time from end of linear fit for DDR to THR
APD	AP duration	ms	time from THR to MDP ₂
APD ₅₀	AP duration 50	ms	time interval at 50% of maximum repolarization
APD ₉₀	AP duration 90	ms	time interval at 90% of maximum repolarization
MDP ₁	maximum diastolic potential 1	mV	minimum potential preceding PP
MDP ₂	maximum diastolic potential 2	mV	minimum potential after PP
THR	threshold potential	mV	potential separating DD and APD
PP	peak potential	mV	maximum potential of AP
APA	AP amplitude	mV	potential difference of PP minus MDP ₂
DDR	diastolic depolarization rate	mV/ms	potential change rate at end of EDD
MUV	maximum upstroke velocity	mV/ms	maximum of derivative between MDP ₁ and PP
MRR	maximum repolarization rate	mV/ms	minimum of derivative between PP and MDP ₂
TRR	transient repolarization rate	mV/ms	maximum negative potential rate change after PP, if distinct from MRR

Please refer to the text for detailed explanations of the parameter definitions.

each PP, respectively. Detection of MDP is challenging because the comparatively slow change in membrane potential during diastole means that noise artifacts could introduce multiple local minima. SG filtering and averaging are used to reduce the number of local membrane potential minima considered in the determination of MDP. The minima ultimately accepted as MDP₁ and MDP₂ meet the criteria of being the closest minima to a given maximum while falling within a window of expected maximum half-cycle length (user-definable, with a default value of 250 ms). A half-cycle length criterion is used to exclude subthreshold membrane potential oscillations from the AP averages.

The action potential amplitude (APA) is the difference in the membrane potential between PP and MDP₂.

Threshold potential (THR) is defined as the potential at a time point occurring after MDP₁ but before PP at which the first derivative of the AP waveform deviates from linearity by the empirically determined value of 0.5 mV/ms. The linearity is determined from an iterative linear regression fit of a segment consisting of 40–80% of the first derivative trace between MDP₁ and the beginning of APD₉₀. The determination of THR combines and extends previous approaches (8–14) to provide robust analysis of a wide range of AP waveforms.

Rates of change of membrane potential

The maximum upstroke velocity (MUV) is defined as the maximum value of the first derivative in the epoch between MDP₁ and PP. The maximum and transient repolarization rates (MRR and TRR) are determined as minimum values of the derivative between PP and the time point at 90% repolarization. MRR is defined as the minimum closest to the time of 90% repolarization. If a second minimum is detected

between PP and the time of 90% repolarization, the minimum closer to PP is defined as TRR. This transient repolarization phase is present in APs from transitional and atrial-like myocytes, but is absent from classical SAM APs, such as the one depicted in Fig. 1.

The diastolic depolarization rate (DDR) is determined iteratively from a linear regression fit of a segment consisting of at least half of the interval corresponding to the initial 10–50% of MDP₁-THR. Interestingly, the best linear fit usually occurs at the end of this interval, matching the definition of DDR used by Bucchi et al. (12) for rabbit sinoatrial myocyte APs. However, the iterative fit approach allows determination of DDR across a wide range of AP firing rates and waveforms.

AP epoch durations

The cycle length (CL) is defined as the time interval between MDP₁ and MDP₂. The diastolic duration (DD) is defined as the time interval between MDP₁ and THR (9,12,14). The early diastolic duration (EDD) consists of the time interval from MDP₁ to the end of the segment used for the best linear fit from the determination of DDR. The late diastolic duration (LDD) is defined as the time interval from the end of EDD to THR. The action potential duration (APD) is the time interval between THR and MDP₂. The action potential durations at 50 and 90% repolarization (APD₅₀ and APD₉₀) are defined as the time intervals spanning across the AP at potentials corresponding to 50 and 90% repolarization.

Validation

ParamAP was validated by comparing manually determined AP parameters from a previous study (see Table

S2 in (15)) with parameters automatically obtained by ParamAP for the same dataset. As shown in Table 2, ParamAP returned results that are very similar to the previous analysis for most parameters. Significant differences were observed only for LDD and APD₉₀ owing to differences in parameter definitions. LDD was previously defined as the duration between 1 and 10% of the maximum dV/dt (15), whereas in ParamAP, LDD is defined as the interval from EDD to THR, so that it is independent of the upstroke velocity. Similarly, APD₉₀ was previously defined as the interval between THR and 90% repolarization, whereas ParamAP defines APD₉₀ as the interval spanning across the AP at 90% repolarization, so that it is independent of THR. The parameterization results for parameters with identical definitions are therefore validated by the comparison with the previous study, whereas the results for most parameters with modified definitions are insignificantly different. ParamAP's robust analysis of a wide range of AP waveforms is illustrated in Fig. 2.

Capabilities

As an open-source software tool, ParamAP is highly extensible and customizable. For example, the SG filter could be extended with an optimized filter algorithm to reject specific artifacts, such as stimulation artifacts in atrial or ventricular AP recordings. In a similar manner, the performance-optimized averaging algorithm could be applied for time intervals that are defined differently from our current implementation using extrema, for example, with an external Schmitt trigger signal as reference. A possible application is the analysis of neuronal APs synchronized with an external signal.

TABLE 2 Validation of ParamAP

Parameter	Parameterization Method			
	Larson et al., 2013 (15)		ParamAP	
MDP ^a (mV)	-59.2	± 2.0 (12)	-59.3	± 2.0 (12)
THR (mV)	-45.9	± 1.3 (13)	-50.1	± 2.1 (13)
DD (ms)	51.1	± 5.6 (14)	66.7	± 13.0 (14)
EDD (ms)	22.7	± 2.9 (14)	29.2	± 6.3 (14)
LDD (ms)	17.7	± 2.6 (14)	37.5	± 7.1 (14)*
DDR (mV/s)	133.7	± 15.4 (13)	115.9	± 17.9 (13)
MRR ^a (mV/ms)	-2.3	± 0.2 (14)	-2.3	± 0.2 (14)
MUV ^a (mV/ms)	24.3	± 4.4 (13)	23.2	± 4.8 (13)
PP ^a (mV)	11.1	± 1.6 (13)	11.0	± 1.7 (13)
APA ^a (mV)	70.1	± 2.9 (11)	70.1	± 2.9 (11)
APD ^a (ms)	81.9	± 7.3 (14)	87.5	± 8.0 (14)
APD ₅₀ (ms)	35.6	± 3.3 (14)	30.9	± 1.9 (14)
APD ₉₀ (ms)	50.8	± 2.8 (14)	69.4	± 6.8 (14)*

Parameterization results are from manual analysis in a previous study (see Table S2 in (15)) compared to the results of automated analysis of the same dataset by ParamAP. Definitions, except as indicated in the footnote, are slightly modified in ParamAP. Asterisks mark significant differences ($p > 0.05$; t -test/ U -test) between the two analyses. Numbers of cells analyzed for each parameter are shown in parentheses.

^aParameters with identical definitions.

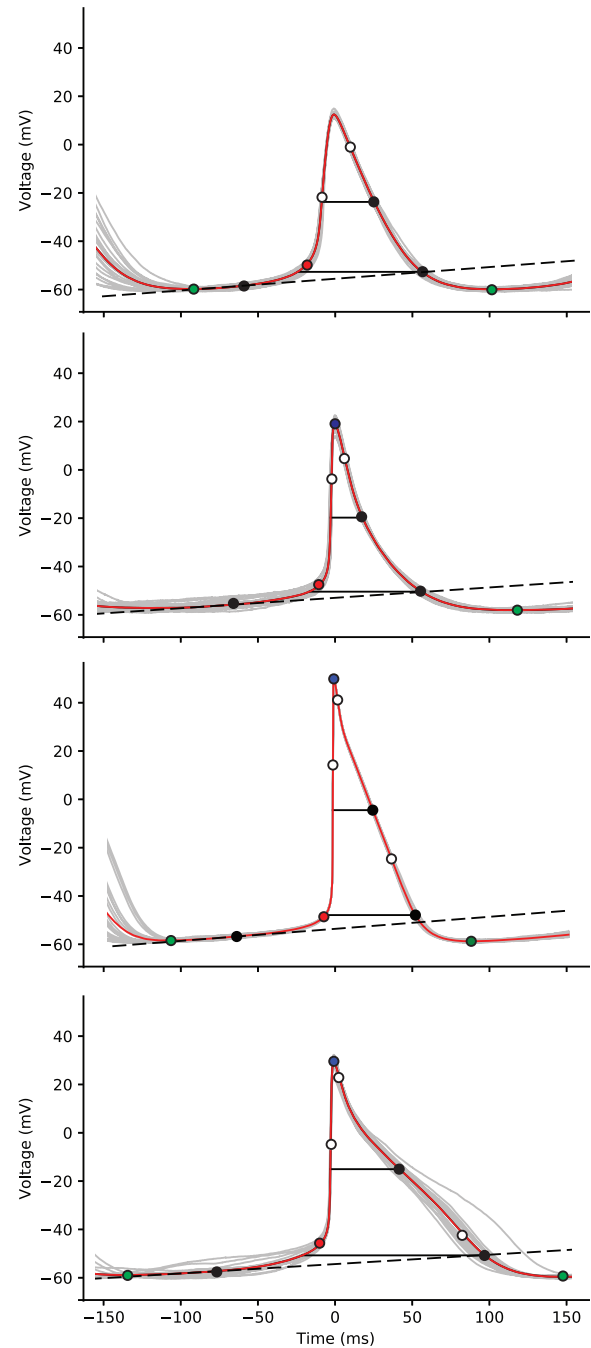


FIGURE 2 Robust analysis of a wide range of AP waveforms. Given here are representative results returned by ParamAP for four distinct AP waveforms. Shown are membrane potentials (green and blue circles), rates of change of membrane potential (open circles), THR (red circle), DDR (dashed line), and APDs (horizontal lines and black circles). See User Manual in the Supporting Material for complete details of ParamAP output. To see this figure in color, go online.

Limitations

ParamAP's overall run-time is mainly determined by the time required for the file input from the hard drive to the

main memory. An immediate reduction in run-time can therefore be achieved by processing the input files from a solid-state drive. In addition to file input/output, the memory allocation of the NumPy arrays can be a performance-limiting step. In our experiments, we commonly observed an ~ 10 -fold increase in peak memory requirement compared to the file size on the hard drive. However, because main memory modules are available at relatively low cost, we have refrained from implementing our own memory-optimized data structure competing with NumPy's readily available array objects and functions.

The detection of a significant number of false-positive extrema can also be computationally demanding. In the case of a low SNR, the SG filter can be applied taking into account a larger number of data points for filtering, effectively increasing the SNR. Similarly, the reliability of the extrema detection will be less prone to noise artifacts by increasing the number of data points considered for the detection.

The performance of ParamAP could also benefit from a parallelized implementation that would avoid the global interpreter lock active in CPython. However, both NumPy and SciPy are currently incompatible with global interpreter lock-free implementations of Python, i.e., Jython or IronPython. Instead of parallelizing the analysis of a single file, large datasets with multiple files should therefore be analyzed in parallel using separate copies of ParamAP.

CONCLUSIONS

ParamAP is a powerful computation tool that automates and standardizes the analysis and of SAM APs. In particular, it streamlines the analysis of large data sets in a reproducible manner. The iterative algorithm for the determination of DDR and THR is robust, allowing the analysis of a wide range of different AP waveforms with the same algorithms. Furthermore, the open-source implementation allows users to extend and customize ParamAP to their individual applications.

SUPPORTING MATERIAL

ParamAP User Manual and four figures are available at [http://www.biophysj.org/biophysj/supplemental/S0006-3495\(17\)30750-6](http://www.biophysj.org/biophysj/supplemental/S0006-3495(17)30750-6).

AUTHOR CONTRIBUTIONS

C.R. and C.P. designed the study. C.R. implemented and validated ParamAP. C.R. and C.P. wrote the manuscript.

ACKNOWLEDGMENTS

We thank Drs. Stefano Morotti, Emily J. Sharpe, Hicham Bichraoui, and Daniel C. Bartos for constructive criticism during the development of ParamAP and for feedback on the manuscript and the User Manual.

This work was supported by National Institutes of Health (NIH) R01-HL088427 to C.P.

REFERENCES

1. Rickert, C., and C. Proenza. ParamAP: Standardized parameterization of sinoatrial node myocyte action potentials. *Zenodo*, <http://dx.doi.org/10.5281/zenodo.823742>.
2. Dubois, P. F. 2007. Guest editor's introduction. Python: batteries included. *Comput. Sci. Eng.* 9:7–9.
3. van der Walt, S., S. C. Colbert, and G. Varoquaux. 2011. The NumPy array: a structure for efficient numerical computation. *Comput. Sci. Eng.* 13:22–30.
4. Jones, E., T. Oliphant, ..., P. Peterson. 2001. SciPy: open source scientific tools for Python. <https://www.python.org/>.
5. Hunter, J. D. 2007. Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* 9:90–95.
6. Savitzky, A., and M. J. E. Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* 36:1627–1639.
7. Bentley, J. L. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM.* 18:509–517.
8. Opthof, T., B. de Jonge, ..., L. N. Bouman. 1985. Functional and morphological organization of the guinea-pig sinoatrial node compared with the rabbit sinoatrial node. *J. Mol. Cell. Cardiol.* 17:549–564.
9. Kodama, I., M. R. Nikmaram, ..., J. M. Owen. 1997. Regional differences in the role of the Ca^{2+} and Na^{+} currents in pacemaker activity in the sinoatrial node. *Am. J. Physiol.* 272:H2793–H2806.
10. Verheijck, E. E., A. Wessels, ..., L. N. Bouman. 1998. Distribution of atrial and nodal cells within the rabbit sinoatrial node: models of sinoatrial transition. *Circulation.* 97:1623–1631.
11. Verheijck, E. E., A. C. van Ginneken, ..., L. N. Bouman. 1999. Contribution of L-type Ca^{2+} current to electrical activity in sinoatrial nodal myocytes of rabbits. *Am. J. Physiol.* 276:H1064–H1077.
12. Bucchi, A., M. Baruscotti, ..., D. DiFrancesco. 2007. Modulation of rate by autonomic agonists in SAN cells involves changes in diastolic depolarization and the pacemaker current. *J. Mol. Cell. Cardiol.* 43:39–48.
13. Bogdanov, K. Y., V. A. Maltsev, ..., E. G. Lakatta. 2006. Membrane potential fluctuations resulting from submembrane Ca^{2+} releases in rabbit sinoatrial nodal cells impart an exponential phase to the late diastolic depolarization that controls their chronotropic state. *Circ. Res.* 99:979–987.
14. Zaza, A., R. B. Robinson, and D. DiFrancesco. 1996. Basal responses of the L-type Ca^{2+} and hyperpolarization-activated currents to autonomic agonists in the rabbit sino-atrial node. *J. Physiol.* 491:347–355.
15. Larson, E. D., J. R. St Clair, ..., C. Proenza. 2013. Depressed pacemaker activity of sinoatrial node myocytes contributes to the age-dependent decline in maximum heart rate. *Proc. Natl. Acad. Sci. USA.* 110:18011–18016.
16. Severi, S., M. Fantini, ..., D. DiFrancesco. 2012. An updated computational model of rabbit sinoatrial action potential to investigate the mechanisms of heart rate modulation. *J. Physiol.* 590:4483–4499.
17. Verkerk, A. O., G. S. C. Geuzebroek, ..., R. Wilders. 2012. Effects of acetylcholine and noradrenalin on action potentials of isolated rabbit sinoatrial and atrial myocytes. *Front. Physiol.* 3:174.
18. St Clair, J. R., E. J. Sharpe, and C. Proenza. 2015. Culture and adenoviral infection of sinoatrial node myocytes from adult mice. *Am. J. Physiol. Heart Circ. Physiol.* 309:H490–H498.

Biophysical Journal, Volume 113

Supplemental Information

**ParamAP: Standardized Parameterization of Sinoatrial Node Myocyte
Action Potentials**

Christian Rickert and Catherine Proenza

ParamAP User Manual

C. Rickert and C. Proenza

Table of Contents

1. Introduction.....	1
2. Hardware and Installation	2
2.1 Recommended minimum hardware	2
2.2 Bundled installation.....	2
2.3 Manual installation.....	2
2.3.1 Install Python 3.....	3
2.3.2 Install NumPy, SciPy, Matplotlib	3
2.3.3 Test the manual installation	4
3. Using ParamAP.....	4
3.1 Data format.....	4
3.2 Data analysis	5
3.2.1 Launch ParamAP from the bundled installation.....	5
3.2.2 Launch ParamAP from the manual installation.....	6
3.2.3 Data file organization	6
3.2.4 Sample data analysis.....	7
3.3 User input	12
3.4 Minimum sampling rate and Savitzky-Golay filtering	14
3.5 Data documentation.....	15
4. Troubleshooting	16
4.1 Error messages.....	16
5. Contact us.....	17

1. Introduction

ParamAP is a standalone computational tool that uses a template-free detection algorithm to automatically identify and parameterize sinoatrial myocyte APs. ParamAP is written in Python 3 and it can be run on Windows, Mac OS X, or Linux/Ubuntu operating systems (OS). It employs a graphic user interface with automatic and user-editable input modes. ParamAP accepts text input files and returns a total of 16 AP waveform parameters as text and graphical outputs. The software is available under the GNU General Public License (GPL) 2.0.

2. Hardware and Installation

2.1 Recommended minimum hardware

ParamAP can be run on any computer for which Python 3 and the libraries NumPy, SciPy and Matplotlib are available, which includes most common processor architectures and operating systems (Mac OS X, Linux/Ubuntu, Windows). The minimum system requirements are very modest (~ 2 GB system memory, 1 GHz processor frequency, and at least 1 GB of free disk space). As a general rule, the minimum memory required is approximately ten times the largest file size to be analyzed, which depends on the data sampling rate and the length of the recording to be read.

2.2 Bundled installation

Bundled 64-bit installations of ParamAP with Python 3, NumPy, SciPy, and Matplotlib are available for 64-bit versions of Windows, Mac OS X and Linux/Ubuntu. File archives containing bundled installations, the executable source code, and an example file are available at <https://github.com/crickert1234/ParamAP/>.

Examples:

(all OS) Click on the appropriate link for your OS. Copy and extract the file archive to the desired location on your computer:
[ParamAP-1.0-Windows-x64.zip](#)
[ParamAP-1.0-MacOSX-x64.zip](#)
[ParamAP-1.0-Ubuntu-x64.zip](#)

2.3 Manual installation

Users with 32-bit Windows or Linux operating systems can download and manually install ParamAP:

Examples:

(all OS) Download the executable source code for all OS. Copy and extract the file archive to the desired location on your computer:
[ParamAP-1.0.zip](#)

ParamAP is implemented in CPython 3 (“Python 3”), using modules from NumPy, SciPy, and Matplotlib. Therefore, manual installation of ParamAP starts with the download and the installation of Python 3 with the additional modules. Although we provide basic instructions for the installation of Python 3 and the additional modules, this user manual does not cover all the necessary steps in great detail. Furthermore, the installation procedure might change slightly for future versions of Python 3 and the additional modules. For more assistance during installation, we suggest an internet search using the terms “install python3 numpy scipy matplotlib” plus the name of your current OS to find step-by-step guides or video tutorials.

2.3.1 Install Python 3

This website contains the installer files and additional installation instructions for the *Python 3.5.2 release*: <https://www.python.org/downloads/release/python-352/>.

Examples:

- (Windows) Download and install “Windows x86-64 executable installer”
 `python-3.5.2-amd64.exe`

- (Mac OS X) Download and install “Mac OS X 64-bit/32-bit installer”
 `python-3.5.2-macosx10.6.pkg`
 `ActiveTcl8.5.18.0.29889-macosx10.5-i386-x86_64-threaded.dmg`

- (Ubuntu) Run in Terminal
 `sudo apt-get install python3`
 `sudo apt-get install python3-tk`
 `sudo apt-get install idle3`

The Python 3 binaries need to be included in your system's path variables (%PATH% for Windows or \$PATH for Linux/Mac OS X) during or after the installation. The path variables ensure the availability of the newly installed binaries to the OS independent from the user's current working directory. In addition to Python 3, we recommend installing Python's integrated development environment (IDLE), which can be used to run and edit Python software independent from the OS used or from third-party text editors.

2.3.2 Install NumPy, SciPy, Matplotlib

NumPy, SciPy, and Matplotlib are required modules and are sometimes referred to as being part of the “SciPy Stack”. They can be downloaded from <http://scipy.org/install.html>. The downloaded modules need to be compatible with the installed version of Python 3 and the computer architecture (32- or 64-bit) from the previous step. Module installer file names containing the string “cp35” or “35” are compatible with all versions of Python 3.5.x., while file names containing the string “amd64” indicate 64-bit versions of the modules.

Examples:

- (Windows) Download files and run pip in Windows command line
 `pip install numpy-1.11.2+mkl-cp35-cp35m-win_amd64.whl`
 `pip install scipy-0.18.1-cp35-cp35m-win_amd64.whl`
 `pip install matplotlib-1.5.3-cp35-cp35m-win_amd64.whl`

- (Mac OS X) Run in Command Prompt
 `sudo pip3 install numpy scipy matplotlib`

- (Ubuntu) Run in Terminal
 `sudo apt-get install python3-numpy`
 `sudo apt-get install python3-scipy`

```
sudo apt-get install python3-matplotlib
```

These commands can be used verbatim, if they are executed from within the download directory of the module installer files. However, the module installer files can be used remotely (from another directory), if the fully qualified file path is provided in addition to the module installer file names.

2.3.3 Test the manual installation

It is important to ensure that the modules are correctly installed so that they are readily available for the Python 3 interpreter. More information on Python 3 setup and usage can be found at <https://docs.python.org/3/using/index.html>.

Examples:

- | | |
|------------|---|
| (Windows) | Start Python 3 from the Start menu
“Python 3.5.2 Python 3.5.2 (64-bit)” |
| (Mac OS X) | Start Python 3 in Command Prompt
python3 |
| (Ubuntu) | Start Python 3 in Terminal
python3 |
| (all OS) | Enter these commands successively into the Python 3 interpreter window
import numpy
import scipy
import matplotlib |

The installation of the modules was completed successfully, if the import commands execute silently (i.e. if they do not cause any error messages to appear). An error message like “[...] Import Error: No module named numpy” notifies the user that the module NumPy has not been installed successfully. Please check that the missing modules have been installed for the correct version of Python (version number and computer architecture). Furthermore, the modules need to be available in your system's path variable. Start Python 3 in a new Windows command line, Command Prompt or Terminal so that the path variable is up-to-date after the installations.

3. Using ParamAP

3.1 Data format

The units for the data input files are automatically imported from file headers in pClamp .atf files, where they are read from the final line of the header. However, in case of invalid or missing units – such as text files from other acquisition platforms that lack the pClamp header -- the user is prompted to provide the units manually. The pClamp .atf header lines are characterized by a quotation mark (“) as their first

character – line number two being an exception. Valid values for the X-axis are “ms” and “s”, valid units for the Y-axis are “mV” and “V”. ParamAP scans up to 25 lines for a valid final header line.

Data in input files should be tab delineated, as is common for .txt files, with a single pair of numerical values in each data line – representing time and voltage – each separated by a tab character (“\t”). All lines should end with a carriage return and a line-feed signal (“\r\n”). The last line of the input file should be empty, i.e. only contain the “\r\n” signal.

Examples:

(all OS)

File header for input files

```
"ATF 1.0\r\n                                # ignored
7      2      \r\n                          # "      "
"AcquisitionMode=Gap Free"\r\n              # "      "
"Comment="\r\n                              # "      "
"YTop=100"\r\n                              # "      "
"YBottom=-100"\r\n                          # "      "
"SweepStartTimesMS=0.000"\r\n               # "      "
"SignalsExported=whol Vm"\r\n               # "      "
"Signals=" "whol Vm"\r\n                   # "      "
"Time (s)" "Trace #1 (mV)"\r\n              # used to extract the units
```

Data lines for input files

```
100.000\t1.000\r\n
100.100\t1.383\r\n
100.200\t3.000\r\n
```

Last line for input files

```
\r\n
```

3.2 Data analysis

3.2.1 Launch ParamAP from the bundled installation

ParamAP can be run directly from the bundled installations without prior installation.

Examples:

- (Windows) Execute the compiled version of ParamAP from the Windows Explorer
ParamAP.exe
- (Mac OS) Execute the compiled version of ParamAP in Command Prompt
./ParamAP.app
- (Ubuntu) Execute the compiled version of ParamAP in Terminal
./ParamAP.bin

3.2.2 Launch ParamAP from the manual installation

ParamAP can be run from IDLE, Python's integrated development environment (<https://docs.python.org/3.5/library/idle.html>). IDLE is bundled with the Python 3 installers for Microsoft and Mac OS X. However, Linux users need to install IDLE separately (see above). IDLE provides a simple user interface and basic functions for editing and executing Python files.

Examples:

- (Windows) Start IDLE from the Start menu
"Python 3.5.2 | IDLE (Python GUI)"
- (Mac OS X) Start IDLE from the Applications folder
"Python 3.5.2 | IDLE (Python GUI)"
- (Ubuntu) Start IDLE from the Dash in the Unity Launcher
"Python 3.5.2 | IDLE (Python GUI)"
- (all OS) Open and run ParamAP with IDLE using the menu
"File | Open..."
"Run | Run Module"

ParamAP can also be run independently from IDLE, using the Windows command line or the Mac OS X Command Prompt or Ubuntu's Terminal. Make sure to set the executable flag for `ParamAP.py` and to pass the file to the Python interpreter as the first argument:

```
python3 ParamAP.py
```

3.2.3 Data file organization

After the first run of `ParamAP.py`, a data folder with the name "ParamAP" will be created in the same folder. All files in this folder with the extension ".atf" will be considered for data analysis. Files in subdirectories are not analyzed.

Examples:

- (all OS) File hierarchy used by ParamAP

<code>./ParamAP.py</code>	<code>#executable</code>
<code>./ParamAP/</code>	<code>#data folder</code>
<code>./ParamAP/file.atf</code>	<code>#data file</code>
<code>./ParamAP/file2.atf</code>	<code>" "</code>
<code>./ParamAP/file3.atf</code>	<code>" "</code>
<code>./ParamAP/subdirectory/</code>	<code>#archive folder</code>
<code>./ParamAP/subdirectory/file4.atf</code>	<code>#archived data file</code>
<code>./ParamAP/subdirectory/file5.atf</code>	<code>" "</code>
<code>./ParamAP/subdirectory/file6.atf</code>	<code>" "</code>

3.2.4 Sample data analysis

An example run of ParamAP is shown below for the sample data file that is distributed with the software. Parameters can be entered via the keyboard and confirmed with the Enter key. Default values or previously entered values can be accepted by simply confirming with the Enter key. Default values are displayed after each question for user input as values (numerical input) or as capitalized letters (Boolean input). After entering the user input to the Python interpreter, the accepted values are printed in the next line. All parameter inputs are defined in section “3.3 User Input” below.

(all OS) Example run of ParamAP

```
=====
ParamAP version 1.0, Copyright (C) 2017 Christian Rickert

Laboratory of Cathy Proenza
Department of Physiology & Biophysics
University of Colorado, Anschutz Medical Campus

ParamAP is distributed in the hope that it will be useful, but it comes
without any guarantee or warranty. This program is free software; you can
redistribute it and/or modify it under the terms of the GNU General Public
License:

https://www.gnu.org/licenses/gpl-2.0.en.html

=====

Use automatic mode? [y/N]:
False

Run time series analysis? [y/N]:
False

Analyze action potentials? [Y/n]:
True

NOTE: Enter “y” or “n”, or a numerical value at the prompt – depending on the type of prompt.
Alternatively, confirm the default values (capital letters or numbers) by pressing the “Return” or the
“Enter” key without further input.

-----
FOLDER: /home/user/Desktop/ParamAP
-----
FILE: ParamAP_example

>> READING... [OK]
```

>> PLOTTING...

[OK]

NOTE: The “Raw data” window must be closed to continue.

>> SETTING...

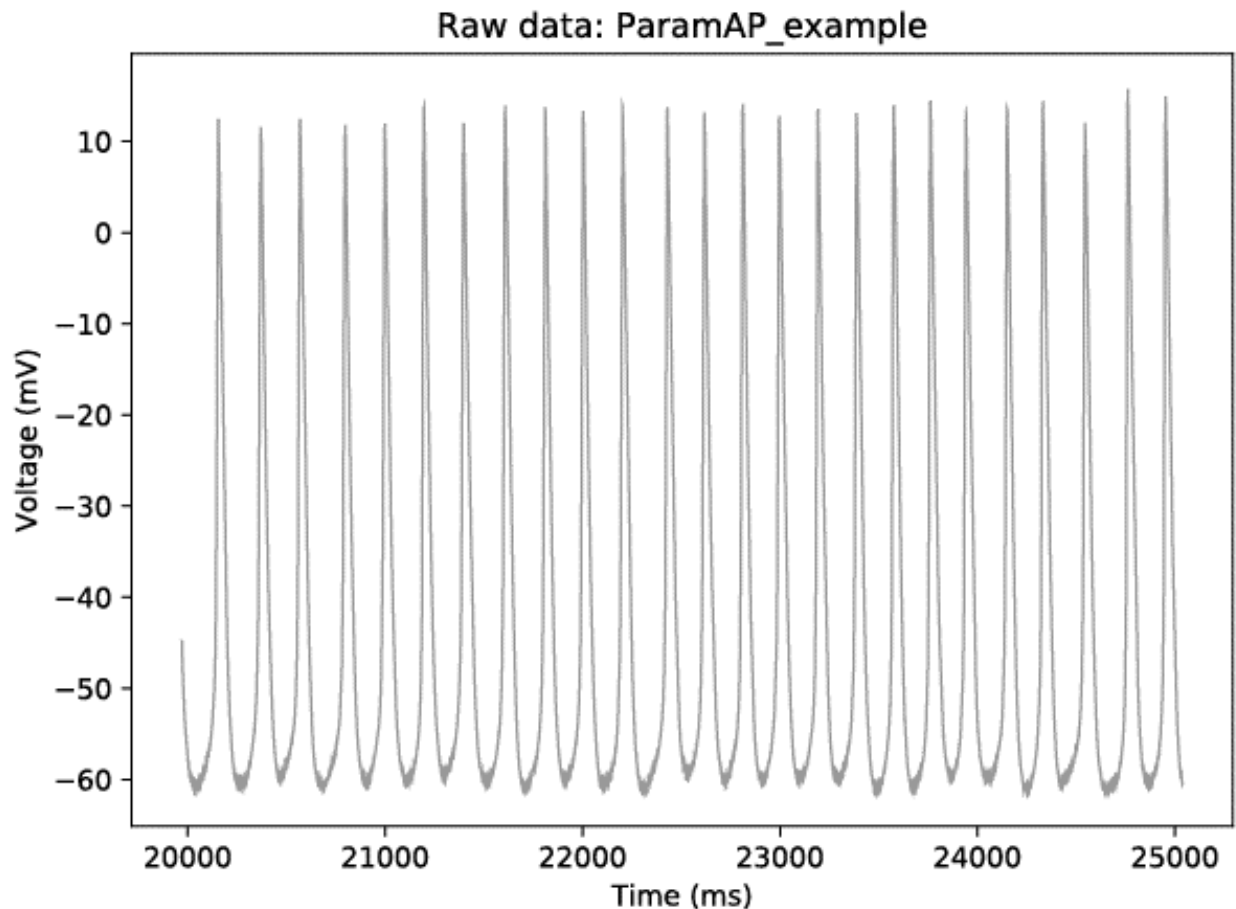


Figure 1: Raw data plot. After completion of reading the input file, ParamAP displays the raw data (*grey line*). Control elements at the bottom of the window (not shown) can be used to reset the view (house symbol), to go back and forth to the recent views (arrows pointing left and right), to pan the graph (cross symbol), to zoom into and out of the graph (magnification glass symbol), to configure the subplots (graph symbol), and to save the plot manually (floppy disk symbol).

Enter analysis start time [19969.76] ms:
19969.760000000002

Enter analysis stop time [25038.88] ms:
25038.879999999997

Enter upper limit for maxima [50.0] mV:
50.0

Enter lower limit for maxima [-10.0] mV:
-10.0

Enter upper limit for minima [-50.0] mV:
-50.0

Enter lower limit for minima [-90.0] mV:
-90.0

Enter maximum peak half width [250.0] ms:
0.25

Enter minimum peak amplitude [50.0] mV:
50.0

Enter running average window size [25]:
25

Enter window multiplier for derivative [1.0]:
1.0

Enter window multiplier for maxima [4.0]:
4.0

Enter window multiplier for minima [16.0]:
16.0

>> FILTERING...	[OK]
>> SEARCHING...	[OK]
>> REDUCING...	[OK]
>> PLOTTING...	[OK]

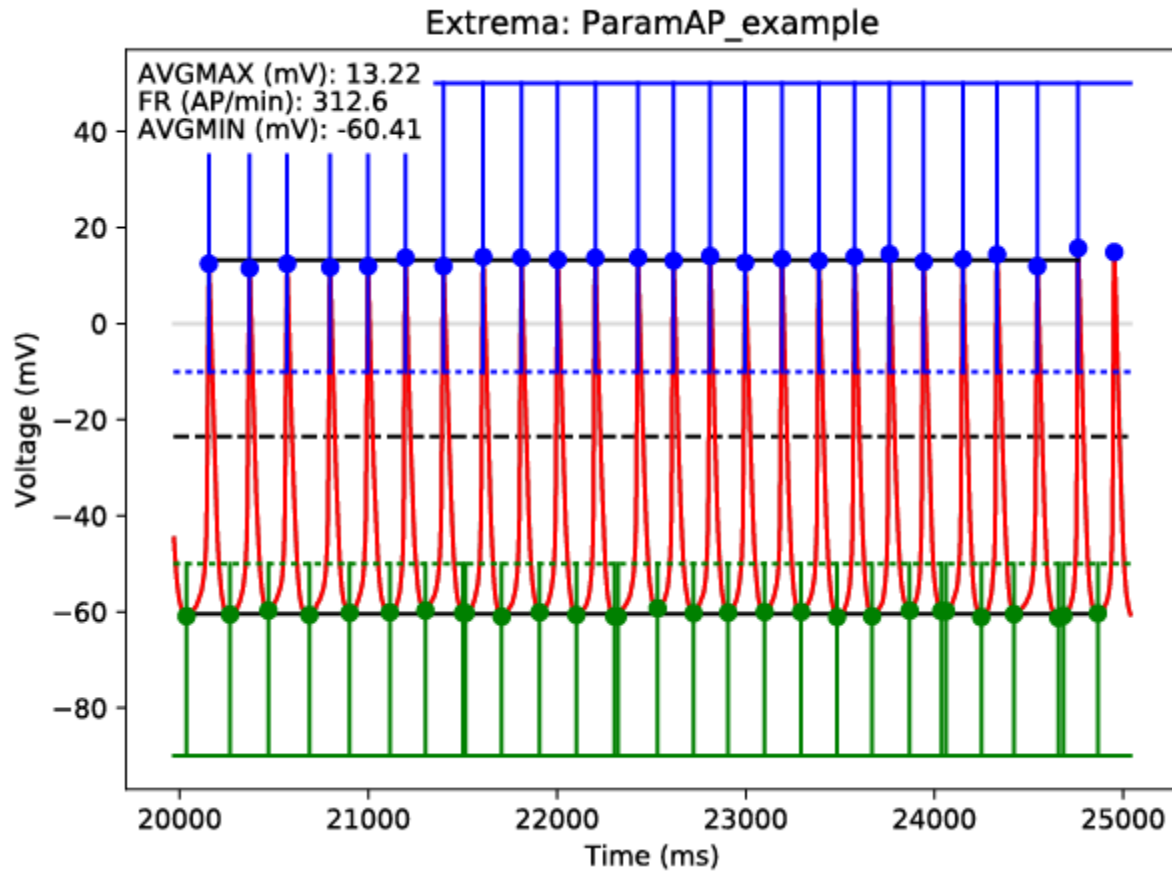


Figure 2: Extrema detection plot. The filtered raw data plot is shown as a *red line*. The maxima (*blue dots*) and minima (*green dots*) are detected. Maxima and minima that have been assigned to their corresponding APs (“accepted”) are marked with *blue and green vertical lines*, respectively. The average of all detected maxima and all detected minima is displayed as a *black dashed line*, while the averages of the accepted maxima (AVGMAX) and the accepted minima (AVGMIN) are displayed as *solid black lines*. Firing rate (FR) is determined from the number of accepted maxima divided by the time interval from the first accepted maximum to last accepted maximum. Time intervals with triplets consisting of one accepted maximum (*blue lines*) and two nearest accepted minima (*green lines*) are used for averaging. Note that the final peak is detected (*blue dot*) but is not accepted (no vertical blue line) because it is not associated with a subsequent accepted minimum.

NOTE: The “Extrema” window must be closed to continue.

```
>> SAVING... [OK]
>> AVERAGING... [OK]
>> ANALYZING... [OK]
>> PLOTTING... [OK]
```

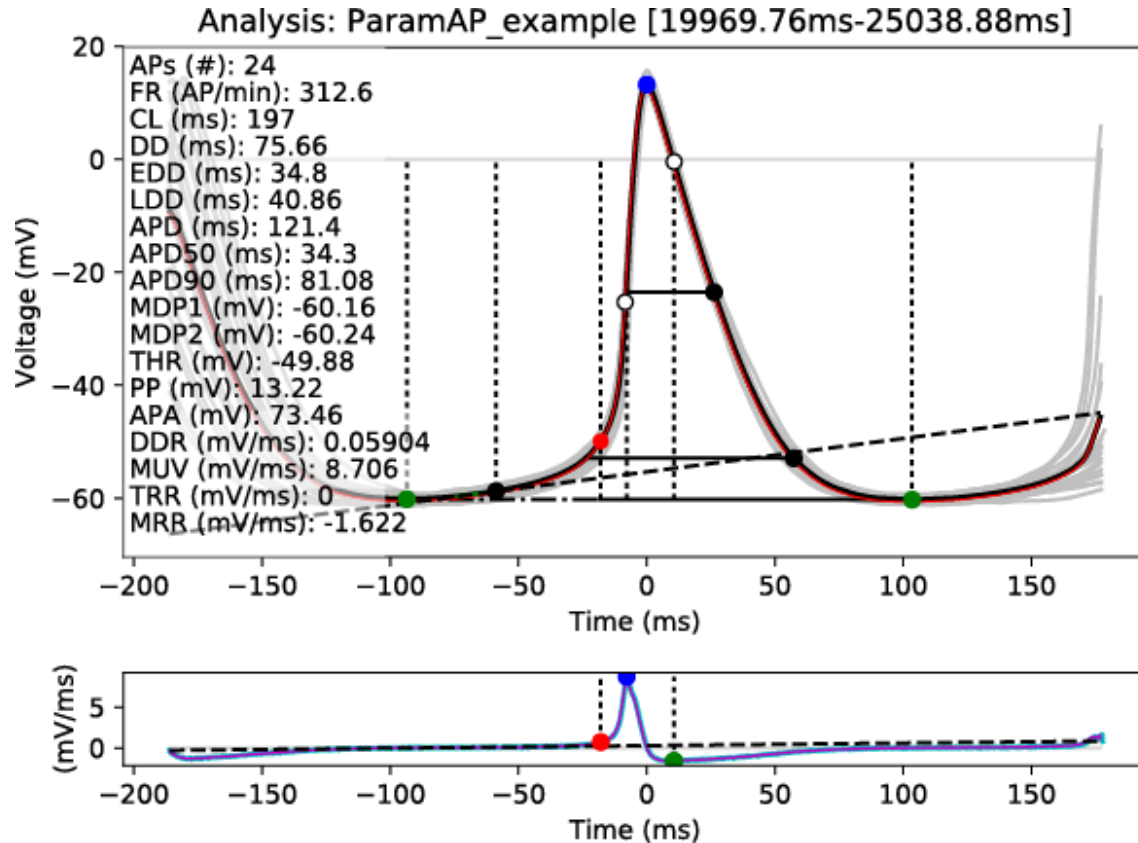


Figure 3: Analysis plot. *Top:* The APs (grey lines) from the previous step are averaged (black line) and filtered (red line). The maximum (blue dot, PP) and the minima (green dots, MDP₁ and MDP₂) are identified. Furthermore, the diastolic depolarization rate (dashed black line, DDR) and the corresponding end of the early diastolic duration (black dot, left) are determined. The action potential durations APD₅₀ and APD₉₀ are plotted as horizontal black lines with their respective membrane potential values (black dots). The threshold potential (red dot), the maximum upstroke velocity (white dot left of maximum, MUV), and the maximum repolarization rate (white dot right of maximum, MRR) are identified with the derivative of the filtered data. Notice the vertical dotted black lines indicating (from left to right) the beginning of the cycle length (CL), the beginning of the late diastolic duration (LDD), the beginning of the action potential duration (APD), MUV, MRR, and the end of CL. Furthermore, the diastolic duration (DD) and the APD are plotted with dashed and solid horizontal black lines, respectively. **Bottom:** The first derivative of the filtered data (cyan line) and the filtered first derivative (magenta line). The maximum (blue dot) corresponds to MUV, the minimum (green dot) to MRR. The best linear fit (dashed black line) within the DD is used to determine THR.

NOTE: The “Analysis” window must be closed to continue.

>> SAVING...

[OK]

Continue with next file? [Y/n]:

True

>> CLEANING...

[OK]

=====
End of run: 1 file processed.
=====

3.3 User input

In this section, the user input parameters are explained. Default values are shown in brackets. If a user enters an invalid input to any of the parameter prompts, ParamAP will notify the user of the error and ask again for a valid parameter input.

Use automatic mode? [y/N]:

The automatic mode can be used to run the data analysis without user interaction. Based on the start parameters supplied in the ParamAP.py file, the software will try to detect the extrema and perform the data analysis in batch. In order to automatically adjust the acceptable ranges for the maxima and the minima for each individual file, the ranges are dynamically defined as the averages of the extrema plus/minus four times the standard deviations, respectively. However, the automatic mode works best with a homogeneous dataset and a good signal-to-noise ratio (SNR). In particular, user interaction is required to avoid artifacts or specify parameters on a case-by-case basis.

Run time series analysis? [y/N]:

The run time series can be used to analyze consecutive time intervals in individual AP recordings. This can be used to document time-dependent or drug-induced changes of parameters in the course of a recording. For example, averages for specified time intervals from a continuous action potential (AP) recording can be analyzed sequentially. The first time interval starts at the specified <start time> and ends at <start time> + <frame time>. The second interval begins with <start time> + <frame time> and ends at <start time> + 2*<frame time>. This continues, until the end of the last interval given by <start time> + n*<frame time> exceeds the <stop time>. The frame time is only queried when performing a run time series analysis.

Analyze action potentials? [Y/n]:

If a user only wants to make use of ParamAP's extrema detection feature, the averaging and parametrization can be skipped. Using this “beat detection mode” will increase the start values for the running average by a factor of ten and the window multiplier for maxima by 50%. The window multiplier for minima will be set to the same value as the window multiplier for maxima value. These changes are intended to allow for the analysis of recordings with a low SNR or small signal amplitude (e.g., cell-attached or extracellular recordings of firing rates). It could also be used to analyze rates in Ca^{2+} imaging or other types of data with an oscillating signal if the Y-axis units are entered as mV or V.

Enter analysis start time [first time stamp] ms:

The start time of the time interval is the first time-stamp listed in the X-axis of the input file. It is used for the data averaging and analysis. Change this parameter to avoid artifacts at the beginning of your data file. This parameter is also the start time for a time series analysis.

Enter analysis stop time [last time stamp] ms:

The end time of the time interval used for the data averaging and analysis. Change this parameter to avoid artifacts at the end of your data file. This parameter is also the end time for a time series analysis, as explained for the query of the time series analysis. The default value is the last time-stamp listed in the X-axis of the input file.

Enter analysis frame time [5000.0] ms:

The time increment used for the time series analysis (see above). Increase this parameter to add more data for averaging and to increase SNR. Decrease this parameter in order to increase the temporal resolution of your AP parametrization, while decreasing SNR.

Enter upper limit for maxima [50.0] mV:

The upper limit for the detection of maxima. In the non-automatic mode, data above this limit are not analyzed in search of maxima. In the automatic mode, all data are analyzed, but those maxima that are above this limit are automatically rejected. Decrease this parameter to avoid the detection of high-intensity artifacts in your data.

Enter lower limit for maxima [-10.0] mV:

The lower limit for the detection of maxima. In the non-automatic mode, data below this limit are not analyzed in search of maxima. In the automatic mode, all data are analyzed, but those maxima that are below this limit are automatically rejected. Increase this parameter to avoid the detection of low-intensity AP in your data.

Enter upper limit for minima [-50.0] mV:

The upper limit for the detection of minima. In the non-automatic mode, data above this limit are not analyzed in search of minima. In the automatic mode, all data are analyzed, but those minima that are above this limit are automatically rejected. Decrease this parameter to avoid the detection of baseline-peaks in your data.

Enter lower limit for minima [-90.0] mV:

The lower limit for the detection of minima. In the non-automatic mode, data below this limit are not analyzed in search of minima. In the automatic mode, all data are analyzed, but those minima that are below this limit are automatically rejected. Increase this parameter to avoid the detection of baseline-troughs in your data.

Enter maximum peak half width [250.0] ms:

The maximum peak half width for the assignment of minima to their corresponding maxima. Limit this parameter to exclude subthreshold membrane potential oscillations from the AP averages.

Enter minimum peak amplitude [50.0] mV:

The minimum peak amplitude for the assignment of minima to their corresponding maxima. Limit this parameter to reject the assignment of maxima, caused by baseline-peaks, to nearby minima.

Enter running average window size [25]:

The running average window size is estimated from the extrapolated number of data points in a time interval of 0.5 ms. The running average has no units and is the base value for the Savitzky-Golay (SG) filter and the detection of the extrema. The larger the value of the running average is, the more data points are used for filtering or extrema detection. However, for both the SG filtering of the derivative and the detection of the extrema, the running average is multiplied with the respective window multiplier values

first. The running average value must be odd and larger than the order of the polynomial function used for filtering – the standard value for the order of the polynomial function is 2.

Enter window multiplier for derivative [1.0]:

The window multiplier for the derivative trace is multiplied with the running average window size for the Savitzky-Golay filter applied to the first derivative of the AP.

Enter window multiplier for maxima [4.0]:

The window multiplier for maxima is multiplied with the running average window size to determine the number of data points on each side of a reference data point that are considered for the localization of a maximum. Increase this value to make the search for maxima more global (more robust against noise) and decrease this value to make the search more local (less robust against noise).

Enter window multiplier for minima [16.0]:

The window multiplier for minima is multiplied with the running average window size to determine the number of data points on each side of a reference data point that are considered for the localization of a minimum. Increase this value to make the search for minima more global (more robust against noise) and decrease this value to make the search more local (less robust against noise).

3.4 Minimum sampling rate and Savitzky-Golay filtering

Data should be acquired with a minimum sampling rate that is determined by the maximum upstroke velocity (MUV), the fastest component of the AP. While the MUV for “classical” sinoatrial node myocytes is ~20 mV/ms, APs from transitional and atrial-like myocytes can exhibit MUV values as high as ~250 mV/ms, corresponding to a maximum signal frequency of ~ 1 kHz. According to the Nyquist theorem, a frequency of 1 kHz requires an acquisition rate of 2 kHz (assuming a low-pass filter cutoff of ≥ 400 Hz). However, since the Nyquist theorem technically applies to the determination of frequencies, the actual minimum sampling rate should be significantly higher than the calculated minimum value. Users should thus employ an empirical check of sufficient sampling rate by inspecting the waveform of the first derivative of the averaged AP to ensure that undersampling artifacts are not present (such as a gradual or distorted representation of the waveform maximum). Using this approach, we have determined that a sampling rate of 50 kHz is sufficient for a MUV of up to 250 mV/ms (with a low-pass Bessel filter cutoff of 10 kHz).

The default value for the Savitzky-Golay (SG) filter in ParamAP is dynamically adjusted for the acquisition rate. The algorithm evaluates the number of data points within a 0.5 ms time interval to determine the sampling rate and then sets the frame size of the SG filter accordingly. For the recommended sampling rate of 50 kHz, the default frame size is 25. The user should confirm that the SG filter does not distort the signal by examining the peak potential (PP). As shown in **Figure 4**, a filter frame size that is too large will result in oversmoothing of the data, as indicated by a reduced and delayed peak in the average trace compared to the raw data traces.

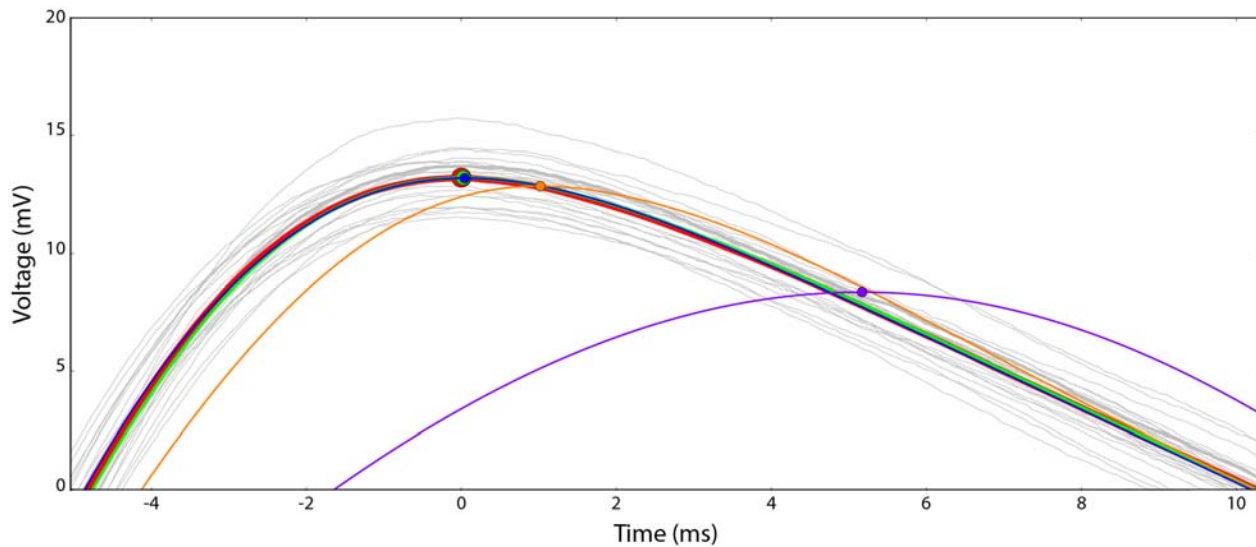


Figure 4: Effects of SG filter settings on the determination of PP. Raw data traces (grey lines) acquired at a sampling rate of 50 kHz are averaged and SG filtered with the default settings (red line, frame size = 25). An increase of the frame size to 251 (green line), 501 (blue line), 1001 (orange line), and 2001 (purple line) further smooths the averaged data. However, the corresponding values for PP (circles) decrease in value and shift to later times.

3.5 Data documentation

ParamAP creates several files for documentation during the parametrization process.

Examples:

(all OS)

Result files from ParamAP

`./ParamAP/ap_2016_12_01_0000.pdf`

`./ParamAP/ap_2016_12_01_0000.txt`

`./ParamAP/ParamAP.txt`

#for each file:

#graphical summary

#textual summary

#once per session:

#parameter summary

Two output files for each input file are created for documentation during the parametrization, a portable document format (PDF) file that contains all the graphical output from the analysis run (for examples see above) and a text file that contains the user input as well as the average plus the filtered average of the input data. Furthermore, a third text file named “ParamAP.txt” is created for each session, in which the parameters are saved line-by-line for each of the files analyzed. The tab-separated values in the “ParamAP.txt” file can be used to import the results of the analysis into other applications for visualization or statistical analysis.

4. Troubleshooting

Users are likely to encounter errors during their data analysis. The most common errors are caused by invalid user input, oscillations from a low SNR, or artifacts from insufficient sampling rates. ParamAP tries to catch the most common errors and to resume the data analysis.

4.1 Error messages

Run failed. Detection of extrema or threshold failed.

This error is commonly seen, if the detection of the maxima and/or the minima failed or if the MUV is below 0.5 mV/ms. In the first case, check that the value for the running average is reasonable and that the window multiplier for the maxima and/or the minima allow for the detection of extrema with the given running average. Try to use small values (1-8) for maxima and higher values (8-64) for minima to begin with. Once all the extrema are detected – i.e. each AP is marked by a maximum and two minima – then increase the values incrementally to avoid noise artifacts and false-positive detections.

In the second case, either the type of AP that the user tries to analyze does not meet the criteria for the definition of THR and should be determined using a different criterion or the AP average exhibits a very low SNR that introduces artifacts into the first derivative trace. The former case requires the user to customize the line of code in ParamAP that defines THR. The latter case requires the user to increase the window multiplier for the derivative.

Run failed. File access denied by system.

This access error occurs if a file that ParamAP tries to access is exclusively accessed by another program (e.g., text editor, PDF viewer, virus scanner, etc.). In this case, close all other software and retry.

Run failed. Identification of action potentials failed.

The identification of action potentials can fail if there is not at least one AP that can be identified (by a single maximum with two trailing minima) in the analyzed recording. If you are certain about the presence of APs in your dataset, check the time interval settings, adjust the limits for the extrema, reduce the window multiplier for the maxima and/or the minima, increase the maximum peak half width, or decrease the minimum peak amplitude parameter.

Run failed. Error was: _____ .

This error will occur, if none of the former errors has been identified. The message will specify the exact cause of the error. However, you will need to read and understand ParamAP's source code in order to fix the causes of these kind of errors.

Run skipped. Canceled by user.

The user can interrupt ParamAP during file processing by sending the interrupt signal SIGINT to the Python 3 interpreter. Use the key combination CTRL+C on Windows, Mac OS X, and Linux. If a user prompt is currently awaiting user input, the interrupt will occur after pressing the Return/Enter key.

5. Contact us

Your feedback is welcome! Please contact us if you run into problems while using ParamAP or if you need information from this manual to be revised. We also welcome feature requests for future versions of ParamAP. Please contact us using this email address: christian.rickert@ucdenver.edu.