

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	2
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.....	5
3.2.3 Data file organization	6
3.2.4 Sample data analysis.....	6
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.....	16

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 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 Ubuntu.

Examples:

(all OS)

Click on the appropriate link for your OS. Use the installer or copy and extract the file archive to the desired location on your computer:

[ParamAP-1.0-Windows-x64.exe](#)

[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)

[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 computer architecture (32- or 64-bit) and the installed version of Python 3 from the previous step. Module installer file names containing the string “cp35” or “35” are compatible with all versions of Python 3.5.x.

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
<code>python3</code> |
| (Ubuntu) | Start Python 3 in Terminal
<code>python3</code> |
| (all OS) | Enter these commands successively into the Python 3 interpreter window
<code>import numpy</code>
<code>import scipy</code>
<code>import matplotlib</code> |

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

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

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...

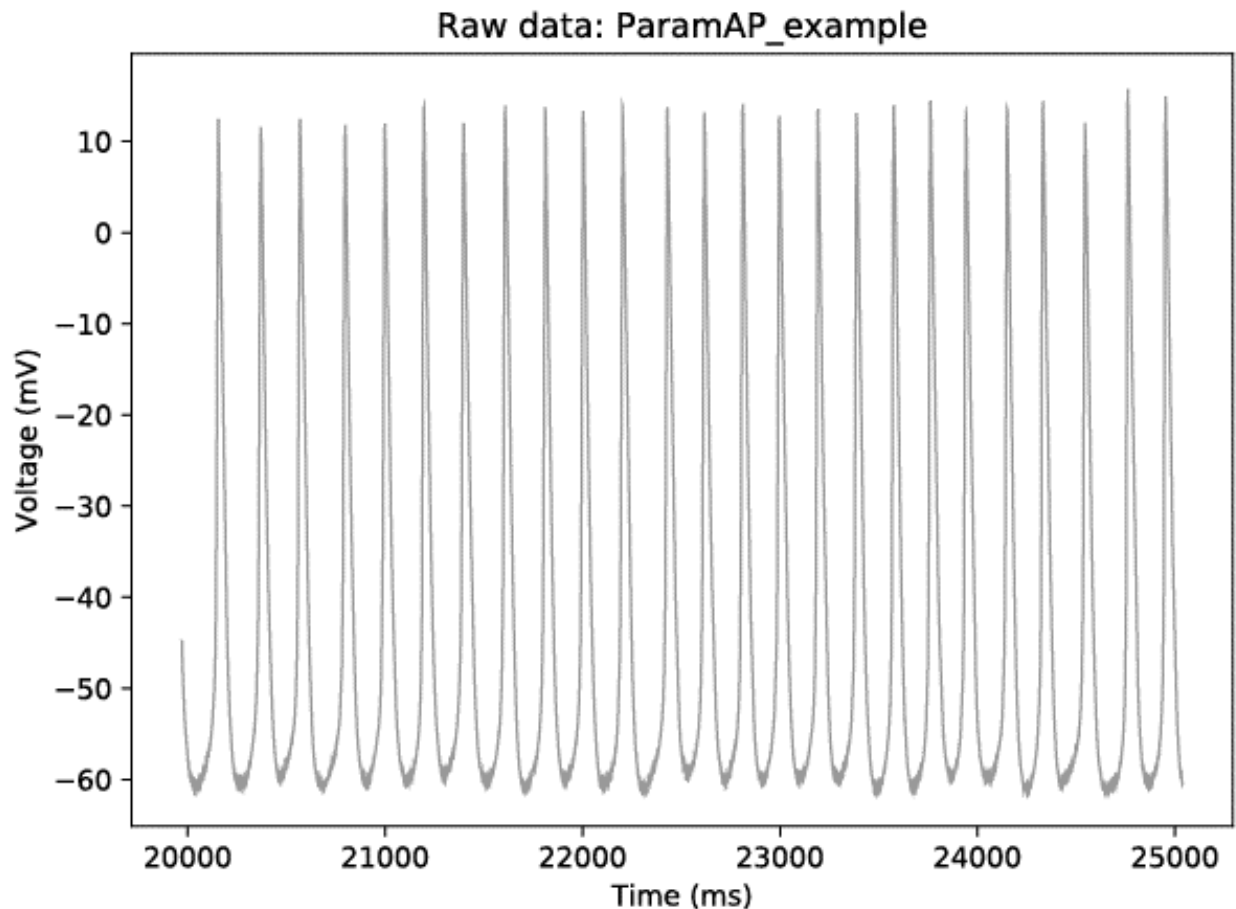


Figure 1: Raw data plot. After completion of reading the input file, ParamAP displays the raw data (*grey line*). Control elements at the bottom of the window (not shown) can be used to reset the view (house symbol), to go back and forth to the recent views (arrows pointing left and right), to pan the 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]

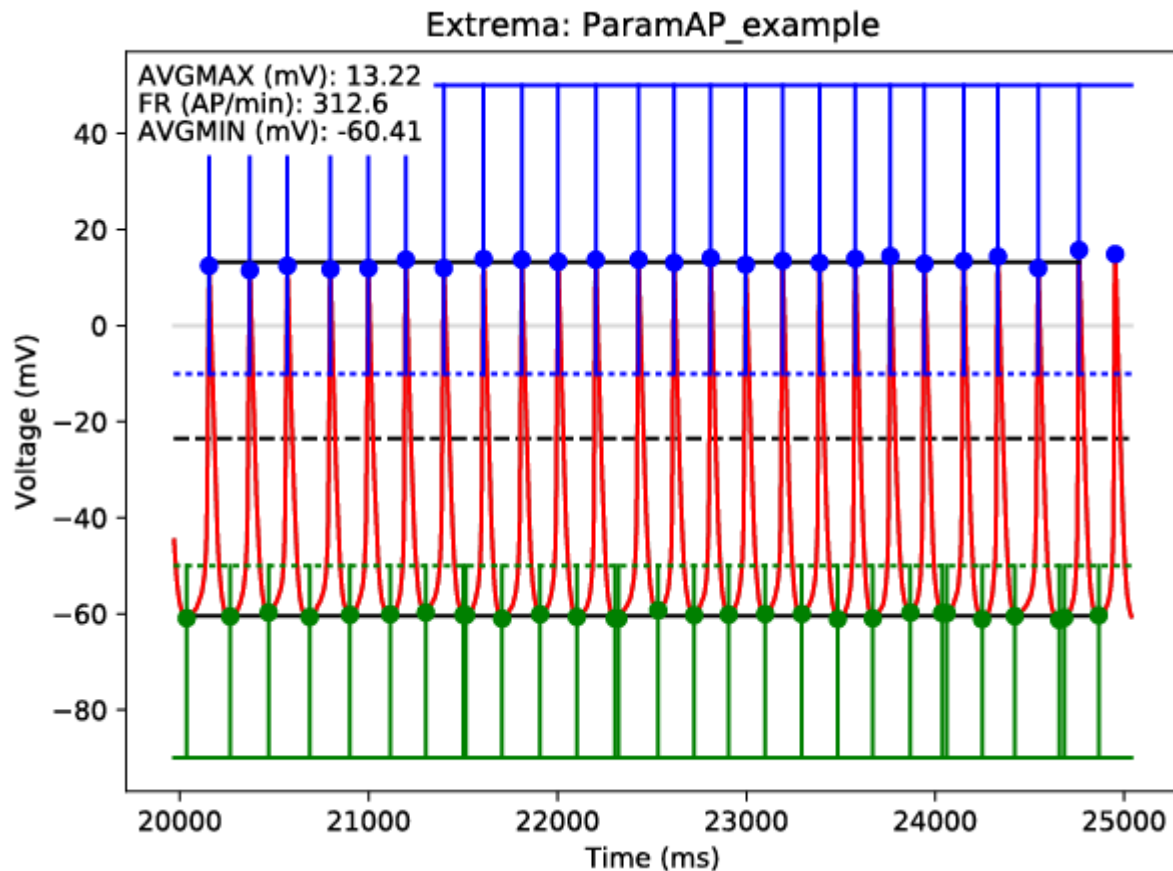


Figure 2: Extrema detection plot. The filtered raw data plot is shown as a *red line*. The maxima (*blue dots*) and minima (*green dots*) are detected. Maxima and minima that have been assigned to their corresponding APs (“accepted”) are marked with *blue and green vertical lines*, respectively. The 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]
```

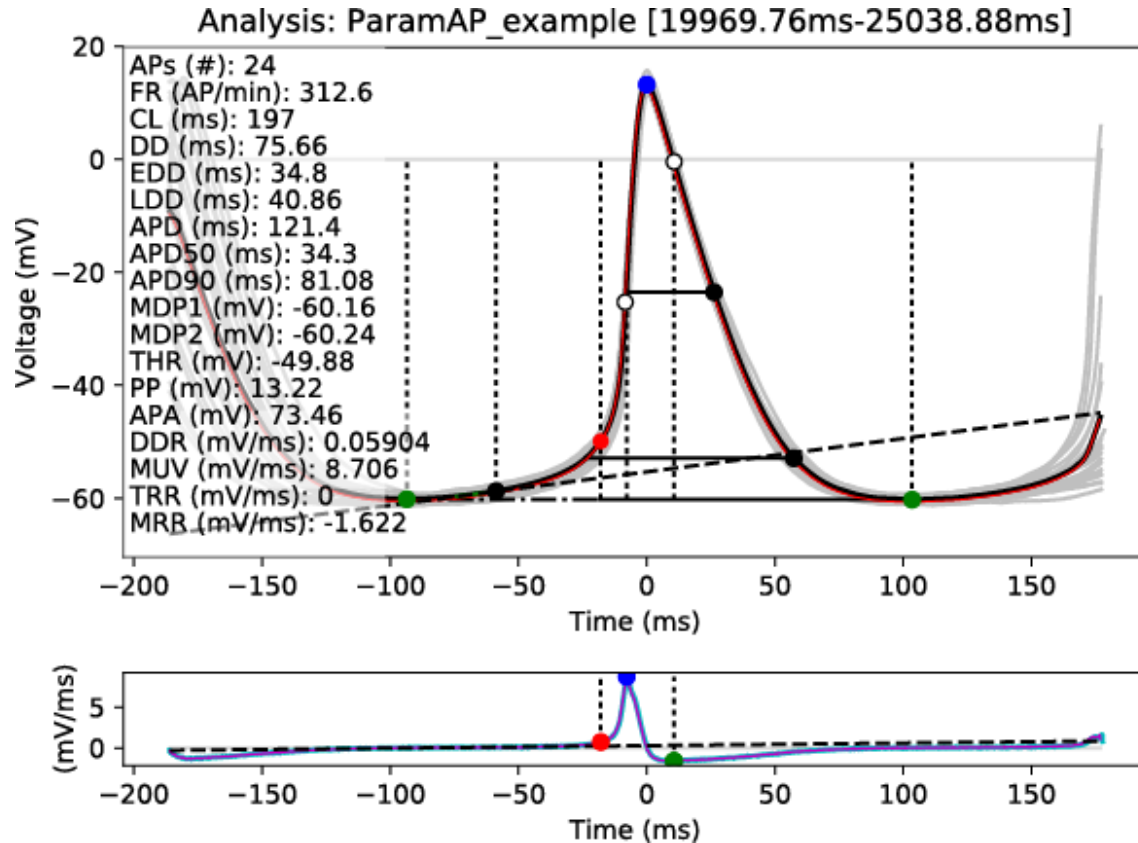


Figure 3: Analysis plot. *Top:* The APs (grey lines) from the previous step are averaged (black line) and filtered (red line). The maximum (blue dot, PP) and the minima (green dots, MDP₁ and MDP₂) are identified. Furthermore, the diastolic depolarization rate (dashed black line, DDR) and the corresponding 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 derivative of the filtered data (cyan line) and the filtered 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 bad 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 reject the assignment of minima, caused by baseline-troughs, to nearby maxima.

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, how many points on each side of a data point are used 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, how many points on each side of a data point are used 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 have 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 counts the number of data points within a 0.5 ms time interval to determine the sampling rate and then sets the frame size of 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.

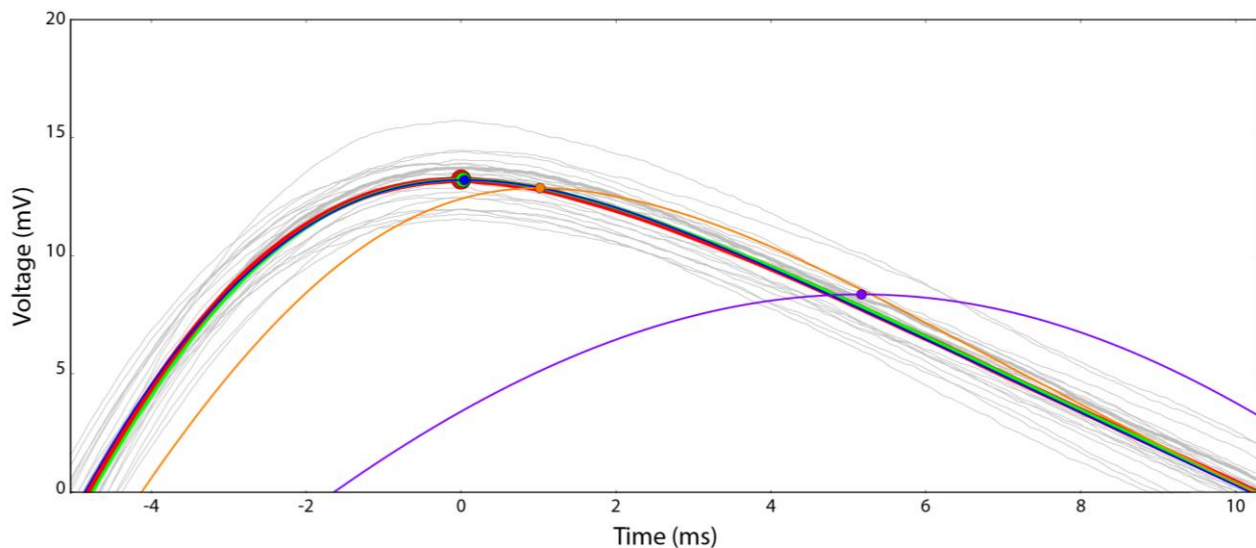


Figure 4: Effects of SG filter settings on the determination of PP. Raw data traces (grey lines) acquired at a sampling rate of 50 kHz are averaged and SG filtered with the default settings (red line, frame size = 25). An increase of the frame size to 251 (green line), 501 (blue line), 1001 (orange line), 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, another 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 comma-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, artifacts from a low SNR, or insufficient sampling rates. ParamAP tries to catch the most common errors and resume 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.

In the second case, 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. This case requires that the user customizes the line of code in ParamAP that determines THR.

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's not at least one AP – 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, try to reduce the window multiplier for the maxima and/or the minima, reduce the maximum, 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.

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.