Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2011, Article ID 251215, 7 pages
doi:10.1155/2011/251215

*Research Article*

# Desktop Software for Patch-Clamp Raw Binary Data Conversion and Preprocessing

**Ning Zhang,[1] Yutao Tian,[2] Lei Zhang,[2] Zhuo Yang,[2] Tao Zhang,[2] and Jishou Ruan[1]**

*[1] Chern Institute of Mathematics and College of Mathematical Science, Nankai University, Tianjin 300071, China*
*[2] Key Laboratory of Bioactive Materials, Ministry of Education, College of Life Sciences, Nankai University, Tianjin 300071, China*

Correspondence should be addressed to Ning Zhang, zhni@mail.nankai.edu.cn

Received 8 March 2011; Accepted 2 May 2011

Academic Editor: Paul Cristea

Since raw data recorded by patch-clamp systems are always stored in binary format, electrophysiologists may experience difficulties with patch clamp data preprocessing especially when they want to analyze by custom-designed algorithms. In this study, we present desktop software, called PCDReader, which could be an effective and convenient solution for patch clamp data preprocessing for daily laboratory use. We designed a novel class module, called clsPulseData, to directly read the raw data along with the parameters recorded from HEKA instruments without any other program support. By a graphical user interface, raw binary data files can be converted into several kinds of ASCII text files for further analysis, with several preprocessing options. And the parameters can also be viewed, modified and exported into ASCII files by a user-friendly Explorer style window. The real-time data loading technique and optimized memory management programming makes PCDReader a fast and efficient tool. The compiled software along with the source code of the clsPulseData class module is freely available to academic and nonprofit users.

## 1. Introduction

Since the advent of patch-clamp electrophysiology at the end of the 1970s [1], there has been an ever-increasing interest in the application of its power to investigate ion-channel-related bioscientific questions in unprecedented detail. By the use of the technique, investigators are able to measure both electrical currents and transmembrane potential responses from individual cells. Over the past decade, patch-clamp technology has led to major advances in the knowledge of the structure and function of ion channels and in drug development, becoming an essential tool in electrophysiology and neuroscience [2–5].

Currently, the world of electrophysiology data acquisition and analysis is practically divided between two major software packages—PULSE from HEKA and pClamp from Axon Instruments. The software produced by HEKA and Axon employ different data acquisition and analysis systems. Pulse+PulseFit package consists of two parts: PULSE, the main acquisition program, and PulseFit to fit raw traces and to derive parameters. Clampex and Clampfit are the acquisition component and the analysis component of the pCLAMP 9 suite, respectively. Although both commercial packages provide rich set of features, electrophysiologists may still experience some difficulties with data analysis especially when it comes to implementing a novel analysis algorithm (e.g., detrended fluctuation analysis (DFA)), which is not implemented in the commercial software packages. In addition, data recorded by patch-clamp systems are always stored in binary format, because of its advantages—it occupies less space than text format both on the hard disk and in the memory and also takes less time to read/write for the same reason. However, it is usually easier to implement custom analysis of data stored in text file format. Furthermore, text format has wider support; the text format, such as the comma-separated-values (CSV) format, is supported by virtually all existing data analysis packages. Therefore, it is really the problem of the absence of generally accepted organization of multiple binary values into a single data file stream, because it is more difficult for electrophysiologists to program a binary file loading module than to get the data from an ASCI file.

There are other programs available to analyze patch-clamp data, most of which are very highly featured and include methods for analysis. For instance, data obtained from patch-clamp instruments can be further analyzed and plotted by means of Igor Pro. Some 3rd-party analysis packages are able to import pClamp files directly for analysis of data recorded in pClamp 9 suite, such as Origin, SigmaPlot [6], and DataView [7]. Marrannes and Prins [8] also designed IVOC and IROC programs to facilitate the analysis of drug effects on currents through ion channels in the cell membrane measured by means of voltage clamp. The programs are written in Igor Pro (WaveMetrics) and read exported files or raw data files generated by the PULSE (HEKA Electronic) or pClamp (Axon Instruments) systems. However, most of these programs are commercial packages, some are free. In the latter group only a few can read the binary raw data recorded from HEKA instruments. In addition, modules of some existing tools (e.g., Igor Pro, Origin) could not be easily used for electrophysiological researchers who are not familiar with programming using the specific language (e.g., the script language in Igor Pro or in Origin), which makes it still hard to read data from binary files.

There are some software tools developed for fa-cilitating solving these problems, such as DataAccess (http://www.bruxton.com/DataAccess/index.html), which is capable of reading both the pClamp and HEKA PULSE file formats, WinWCP (http://spider.science.strath.ac.uk/sipbs/page.php?show=software_winWCP), which supports some of HEKA acquisition hardware, without any conversion of HEKA PULSE format.

In this study, we developed a software called PCDReader (Patch-Clamp Data Reader) for patch-clamp raw binary data conversion and preprocessing. Differing from some earlier studies such as IVOC and IROC [8], which programmed in Igor Pro, we designed a novel class module to directly read the raw data files recorded from HEKA instruments without any other program support (such as Igor Pro and Origin). We have preprocessed the raw data by using PCDReader in our previous patch clamp experiments, such as [9–11].

## 2. Computational Methods and Theory

*2.1. File Format Overview.* The idea of PULSE is to order the data of an experiment in Trees. A single PULSE "Experiment," that can hold a variable number of single electrophysiological experiments, consists of at least three files: *.pgf, *.pul, and *.dat (Table 1). Each of the three files has a tree structure, separately. PCDReader is designed to recognize and read all the information stored in these files.

*2.2. Outline of PCDReader.* PCDReader is a Windows-based application that is implemented using Visual Basic programming language and Win32 API (Application Programming Interface). It consists of four main components: (i) a Data Quick Viewer to quickly obtain (with <1 s file loading time) parts of the raw data, (ii) a Spike-wave Viewer to visualize the spike waves of traces, (iii) a Data Conversion dialog (with data preprocessing options) to export raw data into ASCII text files, and (iv) a Parameter Viewer to view *.pgf and *.pul files in an Explorer style window. By the Parameter Viewer, users can export all information in *.pgf and *.pul files into ASCII text and can also modify parameters and save the modifications back to the raw binary files. Figure 1 depicts an outline of the proposed system.

It has been successfully tested on most 32-bit Windows systems (Windows 98, Windows Me, Windows NT, Windows 2000, Windows XP, and Windows 2003). It can be run on any PC compatible computers with the minimal requirements of a 386DX processor and 64 Mb RAM. The program has also been optimized both in memory management and data loading and writing processing (see below).

PCDReader is programmed in an objected-oriented programming style. Object-oriented technique is more versatile because of its modularity, data abstraction, and inheritance characteristics. PCDReader employs many different classes to accomplish different tasks, for instance, raw data reading class, memory management class, traces drawing class, and Windows interface class. A complete description of every class module is beyond the scope and length of this paper and here we point out only the raw data reading class below.

*2.3. Class Module for Reading Raw Data.* We design a novel class module, called clsPulseData, to directly read the raw data along with the parameters recorded from HEKA instruments without any other program support (such as Igor Pro and Origin). A Class Module encapsulates objects' properties and methods, which is a fundamental concept of object-oriented programming (OOP). Data analysis algorithm programmers may use the Class Module to create their own custom objects for easily retrieving data from raw binary files. It is not necessary for them to redesign the data retrieving program when they analyze data. Technical information about the format of the raw binary files was obtained from the PULSE+PULSEFIT Manual (8.52).

clsPulseData encapsulates as much as possible properties and functions for raw data loading, which can be easily called in the main procedure. Some of the properties and methods of the clsPulseData class are listed in Table 2. The parser in the class will automatically recognize and read the data and return values to the main procedure as required.

clsPulseData does not load all data in memory at the time of file opening but only constructs the data structure and stores the pointers. The event of reading data stream from raw files occurs only if one of the following conditions is met: (a) the main procedure calls a reading method; (b) the main procedure sends a data preprocessing command. In addition, clsPulseData reads only part of the data required at the moment. This kind of real-time data loading technique makes the program optimized a quick and efficient data processing tool.

## 3. Program Features

*3.1. Raw Data Quick Viewer.* A quick viewer window was designed, which provides a rapid access to the raw data (Figure 2). The real-time data loading technique makes it possible to view any part of the raw data quickly. In

Table 1: Brief description of the three main files in a PULSE experiment.

| Extension | Type | Tree structure | Brief description |
|---|---|---|---|
| *.pgf | Stimulus Templates file | Root, Stimulation, Segment | Has the stimulus templates used (Stim Tree), which will be copied into the running experiment. |
| *.pul | Acquisition Parameters file | Root, Group, Series, Sweep | Has the complete data tree that includes all the amplifier settings and also contains the pointer to the data stored in the raw data file. |
| *.dat | Data file | Root, Group, Series, Sweeps, Traces | Has only the actual raw data without any timing or scaling information. |

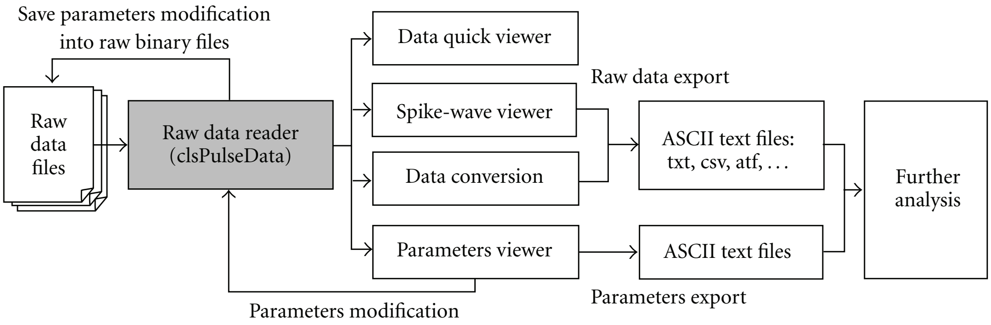
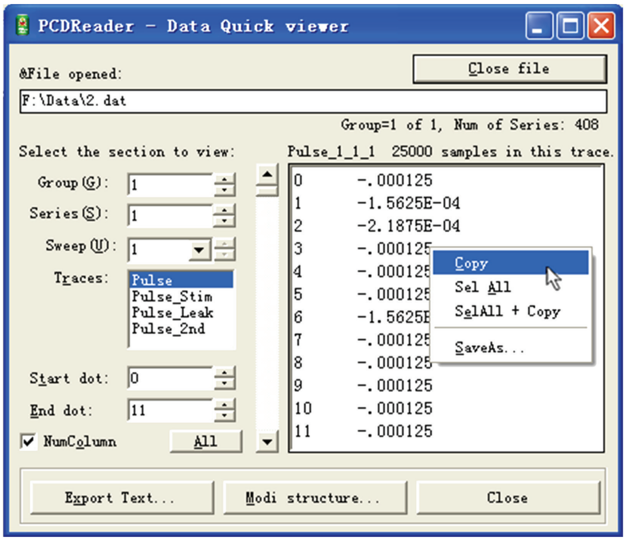


Figure 1: System outline of PCDReader software. Raw data can be viewed in several windows and can be exported into ASCII files. Parameters can be viewed, exported, and modified and modification can be saved back into the binary files.

Table 2: List of some of the properties and methods of the clsPulseData class.

| | Name | Description |
|---|---|---|
| Properties | CacuExpression | Set or return the calculation expression for raw data preprocessing. |
| | DataFormat | Return the data format of a given Sweep (i.e., int16, int32, real32, or real64). |
| | FileNameDat, FileNamePul, FileNamePgf | Set or return the *.dat, *.pul or *.pgf file name (with entire path), respectively. |
| | GroupCount, SeriesCount, SweepCount | Return the number of Groups in the experiment, the number of Series in a given Group, and the number of Sweeps in a given Series, respectively. |
| | HasLeak, HasSecond | Return whether the leak and the second trace exist or not, respectively. |
| | PtrInPgfSeries | Return the pointer of a Series in *.pgf file. |
| | PtrInPulGroup, PtrInPulSeries, PtrInPulSweep | Return the pointer of a Group, a Series, and a Sweep in *.pul file, respectively. |
| | SampleRate | Return the sample rate. |
| Methods | CloseFile | Close the opened files, release all resources. |
| | OpenPulseDataFile | Open the *.dat file along with the *.pul and *.pgf files. Only construct the data structure in memory, without loading all data from files. |
| | PulseReadData | Read and return a continuous data stream from raw data file given a trace and a range of start and end time. |
| | SweepIdxToIdxes | Given an index of Sweep in the data structure, return the corresponding indexes of Group, Series, and Sweep in raw data, respectively. |



PCDReader – Data Quick viewer

File opened: F:\Data\2.dat

Group=1 of 1, Num of Series: 408

Select the section to view:

Group (G): 1

Series (S): 1

Sweep (U): 1

Traces: Pulse, Pulse_Stim, Pulse_Leak, Pulse_2nd

Start dot: 0

End dot: 11

☑ NumColumn

Pulse_1_1_1 25000 samples in this trace.

```
0     -.000125
1     -1.5625E-04
2     -2.1875E-04
3     -.000125
4     -.000125
5     -.000125
6     -1.5625E
7     -.000125
8     -.000125
9     -.000125
10    -.000125
11    -.000125
```

Figure 2: Raw data quick viewer in PCDReader software.

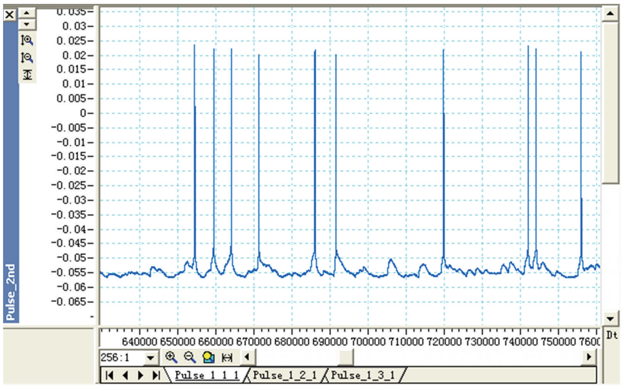
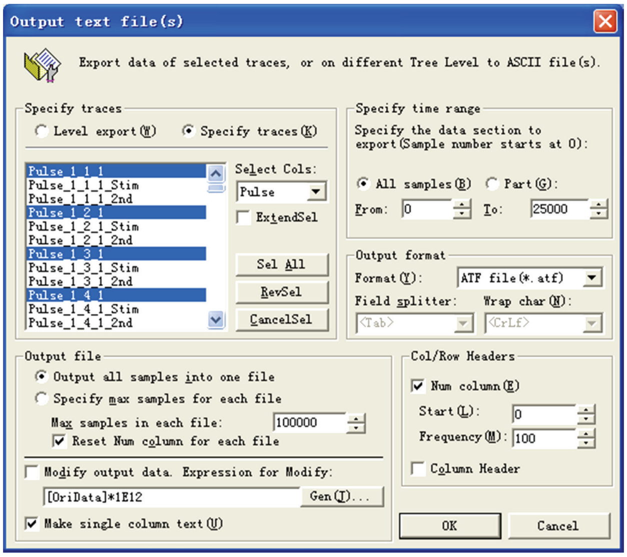


Figure 3: A sample of spike wave (whole-cell recording) visualization in PCDReader. The spike waves are for the spontaneous discharges (recorded in the second trace) in hippocampus slice CA1 region pyramid neuron (Wistar Rat). The data was acquired from HEKA instruments in our laboratory.



Output text file(s)

Export data of selected traces, or on different Tree Level to ASCII file(s).

Specify traces: ○ Level export (W) ● Specify traces (K)

Pulse_1_1_1, Pulse_1_1_1_Stim, Pulse_1_1_1_2nd, Pulse_1_2_1, Pulse_1_2_1_Stim, Pulse_1_2_1_2nd, Pulse_1_3_1, Pulse_1_3_1_Stim, Pulse_1_3_1_2nd, Pulse_1_4_1, Pulse_1_4_1_Stim, Pulse_1_4_1_2nd

Select Cols: Pulse

□ ExtendSel

Specify time range: Specify the data section to export (Sample number starts at 0):

● All samples (B) ○ Part (G):

From: 0 To: 25000

Output format: Format (Y): ATF file (*.atf)

Field splitter: <Tab> Wrap char (N): <CrLf>

Output file:

● Output all samples into one file

○ Specify max samples for each file

Max samples in each file: 100000

☑ Reset Num column for each file

□ Modify output data. Expression for Modify: [OriData]*1E12

☑ Make single column text (U)

Col/Row Headers:

☑ Num column (E)

Start (L): 0

Frequency (M): 100

□ Column Header

Figure 4: The data export dialog.

the screen, users can select the Group, Series, Sweep, and the Trace, as well as the start and end point to display. The selected section of raw data will be rapidly shown in the window on the right hand, which can then be copied to the clipboard or saved to an ASCII text file by a right-click mouse pop-up menu. As for selection of start and end point, there is also a scrollbar in the middle left of the window to drag as well as to input values into the text boxes.

This is a useful tool for researchers especially when they want to quickly view only a specific part of the raw data. There is no necessity to wait for a long time for the whole data file to be loaded.

*3.2. Spike Wave Viewer.* PCDReader has a graphical interface that allows for the spike waves visualization for traces. There is a selector with some labels at the bottom, each for one sweep. Users can select a sweep by clicking the corresponding label to view both its traces in the graph window and the data value in the table right. Data are shown by rows and columns in the table just as that shown in Igor Pro software. Real-time movable cursors, scroll, zoom, resize, and selection functions capabilities are also supplied in the graphic window. A sample of spike waves visualized in this window is presented in Figure 3. Users can select a part of the data by dragging to highlight it. The selection can then be exported to a file.

*3.3. Binary Data Conversion.* PCDReader provides a versatile dialog to export binary data, with several data preprocessing options. The dialog is shown in Figure 4.

Users are allowed to specify which part of raw data to export, by selecting fields in the top left window and typing start and end point in the top right window. There is also a drop-down menu for quick selection of the same type traces in the top left window. For instance, one can simply select all the “stim” traces (such as Pulse_1_1_1_Stim, Pulse_1_1_2_Stim, Pulse_1_2_1_Stim, and Pulse_1_3_1_Stim) by clicking the “Pulse_Stim” item of the drop-down menu. PCDReader also supports “Level Export” to generate files automatically at different levels, that is, Group level, Series level, or Sweep level. For instance, if a user exports raw data in a Series level, each Series will be exported into one file in spite of the number of Sweeps it contains. This provides a convenient way to preprocess for batch data analysis one unit by one unit (Series by Series in this example). Otherwise, without this feature, one must do the complicated procedure manually as follows: select all Sweeps in Series 1, export them into one file; select all Sweeps in Series 2, export them into another file; and so forth; repeat until all Series are exported. There will be no necessity to export the whole file and then manually separate it into several parts, either. Note that the “Level Export” functions are not shown in Figure 4. They will be displayed in the place of the top left window in the dialog if the user selects the “Level Export” radio button.

Then the exported data can be loaded by other software to make further analysis. PCDReader provides an option to set the maximum samples in each exported file. Large data files will be separated into several small parts automatically, with each one not exceeding the maximum number of samples. This is useful and provides a possible way for memory-consuming algorithms to analyze part by part the samples from a large-size raw data file.

Another special feature of the program is that data modification by a user-defined expression is permitted during the export processing. Any valid expression (+, −, ∗, /, ^) and many functions such as Sin(), Cos(), Exp(), Log(), and Sqr() are supported. This preprocessing feature provides a batch calculation based on raw values to generate more proper data for further analysis. It can be used in, for example, data value magnification such that each sample is multiplied by $10^{12}$, which can be simply done by typing the expression "[OriData]*1E12" in the expression text box in the dialog. Users have another option to add row headers (num column) and column headers in the exported files. The num column can also be customized by specifying the start value and the sampling frequency in the text box.

Exported data can be saved in several kinds of ASCII text format, such as CSV, ATF, and user-defined TXT files. An ATF format file can then be loaded directly into Clampfit software for further analysis. User-defined TXT files can be generated by specifying the field separator (space, comma, semicolon, tab, etc.) and the wrap character (line feeds, carriage returns, etc.). Furthermore, a progress window will be shown to indicate the export progress, and a "Cancel" button is also provided for user to cancel at any time during the processing.

*3.4. The Parameters Viewer.* The Parameters Viewer is used for *.pul files and *.pgf files. Information in *.pul files and *.pgf files can be presented in an Explorer style window, with left part displaying the tree structure and right part displaying all the parameters corresponding to the selected node (Figure 5). The right window includes various kinds of controls, such as text boxes, drop-down menus, and check boxes. There are also some command buttons in the right window to click to open another pop-up window, for example, the Trig parameters in a stimulation record. Every parameter entry can be customized by users in the configuration file "PlsFdCfg.ini". Properties of parameter entries that can be customized in the "ini" file are listed in Table 3. Users are able to modify "PlsFdCfg.ini" to customize their own window screen, and the configuration file has a default version for novice users.

A special screen was designed to enable selected parameter entries to be exported to ASCII files. Recursion operations are also supported, which enables the export of the sublevels of a selected item, for example, a series entry is exported together with all its subsweeps in the series.

All parameter entries shown in the right window can be edited and the modification can be saved back to the original binary files. Modified *.pul files and *.pgf files by PCDReader can be loaded again in Pulse+Pulsefit system or any other software which support the format.

*3.5. Software Performance.* The program has been optimized both in memory management and data loading and writing processing. A 2.8 GHz Pentium 4 computer was used to benchmark the performance of PCDReader. The results are shown in Table 4.

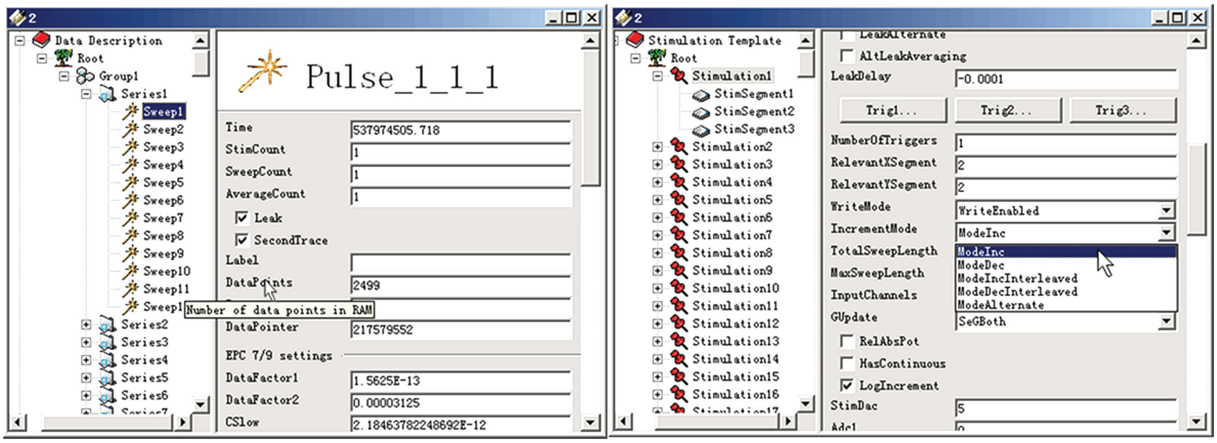
As shown in Table 4, the real-time data loading technique makes the time spent for file opening less than 1 second, even for files more than 100 MB. The time spent for loading and preprocessing data is usually less than 20 min and often less than 5 min when data conversion is without modification, with much less memory occupied. Therefore, PCDReader is a fast and efficient tool for patch-clamp data processing for daily laboratory use.

## 4. Conclusions and Future Plans

In the present study, a desktop software, PCDReader, has been presented for the patch-clamp raw binary data conversion and preprocessing. The program is different from most of the previous programs in the field. It incorporates a class module designed by ourselves that includes many useful properties and functions for raw binary data reading. The class module is a stand-alone program and can run without any support of other programs (e.g., Igor Pro or Origin). The real-time data loading technique and our efforts to optimize the program both in memory management and programming make PCDReader a fast and efficient tool. It provides features for data conversion and preprocessing in several versatile dialogs. In addition, it provides a vivid and easy way (an Explorer style window) to view information in *.pul and *.pgf files and supports the modification and changes being saved back to the original binary files.

Future enhancement of the software could include (i) integrating more programs of newly developed algorithms into the system for data analysis and providing an easy way to integrate custom designed algorithms; (ii) designing new class modules to support more patch-clamp data formats; (iii) extending programming to support additional platforms, such as Linux, and MacOS.

PCDReader, as a desktop application, is made friendly for electrophysiological researchers for ease of use with a graphical user interface. Although some commercial software implement more sophisticated analysis algorithms, the free PCDReader application is potentially useful especially for small laboratories that do not have funding for the expensive commercial packages. Its primal goal is to help a user with some patch-clamp data on his/her hand and the need to preprocess it, with the choice of what to use, and what is out there that can be used. PCDReader could be an effective and convenient solution for daily laboratory use, for the growing number of researchers who are moving beyond



Time 537974505.718
StimCount 1
SweepCount 1
AverageCount 1
☑ Leak
☑ SecondTrace
Label
DataPoints 2499
Number of data points in RAM
DataPointer 217579552
EPC 7/9 settings
DataFactor1 1.5625E-13
DataFactor2 0.00003125
CSlow 2.18463782248692E-12

□ LeakAlternate
□ AltLeakAveraging
LeakDelay -0.0001
NumberOfTriggers 1
RelevantXSegment 2
RelevantYSegment 2
WriteMode WriteEnabled
IncrementMode ModeInc
TotalSweepLength
MaxSweepLength
InputChannels
GUpdate SeGBoth
□ RelAbsPot
□ HasContinuous
☑ LogIncrement
StimDac 5
Adc1 0

Figure 5: The parameters viewer and the parameters export dialog. PCDReader can load *.pul files and *.pgf files directly and display all parameter entries in an Explorer style window. The windows are shown for *.pul files (left) and *.pgf files (right), respectively.

Table 3: Properties of parameter entries that can be customized in "PlsFdCfg.ini" file.

| Property | Values | Description |
|---|---|---|
| Caption | String | Displayed in the right window to indicate a parameter name. |
| ToolTip | String | Brief text displayed in tooltip text when the mouse moves upon a parameter entry. |
| StatusPrompt | String | Brief text displayed in status bar when the corresponding parameter entry has the focus. |
| VariableType | 1 = Byte, 2 = Integer,<br>3 = Long, 4 = Single,<br>5 = Double, 6 = String,<br>7 = custom | Specify the variable type of a parameter entry. |
| Length | Bytes | Only used for VariableType = 7 parameter entries. |
| Control | 1 = Text box,<br>2 = Drop down menu,<br>3 = Check box,<br>4 = command button,<br>5 = separate line | Specify the control used for a parameter entry. |
| Color | RGB color value | Specify the control color for a parameter entry. |
| Items | List of string | The item list. Only used for Control = 7 (drop-down menu) parameter entries. |
| Reserved | | |

Table 4: Processing benchmarks for PCDReader using a 2.8 GHz Pentium 4 computer.

| Data size | Raw data File size | Memory occupancy | File open time | Data conversion processing time[1] | Data conversion processing time[2] |
|---|---|---|---|---|---|
| 1 group, 256 series, 25000 samples per traces | 24.6 MB | 9.9 MB | <1 sec | 23 sec | 3 min 29 sec |
| 1 group, 408 series, 25000 samples per traces | 38.4 MB | 8.7 MB | <1 sec | 40 sec | 4 min 18 sec |
| 1 group, 2 series, 1220000 samples per traces | 50.5 MB | 10.8 MB | <1 sec | 56 sec | 6 min 21 sec |
| 1 group, 713 series, 25000 samples per traces | 69.7 MB | 28.7 MB | <1 sec | 1 min 22 sec | 9 min 48 sec |
| 1 group, 1408 series, 25000 samples per traces | 131 MB | 30.1 MB | < 1 sec | 3 min 16 sec | 17 min 24 sec |

[1] All current trace data was converted into ASCII text format, without data modification.

[2] All current trace data was converted into ASCII text format, with each sample value multiplied by $10^{12}$.

simple use of a software tool for preprocessing raw patch-clamp data and are designing their own analysis algorithms.

## Availability

The compiled software along with the source code of the clsPulseData class module is freely available to academic and nonprofit users and can be downloaded from the website http://www.nkbiox.com/sub/pcdreader/index.htm. There is no password required to enter the system.

## Conflict of Interests

The authors declare that there is no conflict of interest.

## Acknowledgment

This work was partly supported by grants from the National Natural Science Foundation of China (30870827).

## References

[1] E. Neher and B. Sakmann, "Single channel currents recorded from membrane of denervated frog muscle fibers," *Nature*, vol. 260, no. 5554, pp. 799–802, 1976.

[2] P. M. Heyward and M. T. Shipley, "A device for automated control of pipette internal pressure for patch-clamp recording," *Journal of Neuroscience Methods*, vol. 123, no. 1, pp. 109–115, 2003.

[3] S. M. Huber, C. Duranton, and F. Lang, "Patch-clamp analysis of the "new permeability pathways" in malaria-infected erythrocytes," *International Review of Cytology*, vol. 246, pp. 59–134, 2005.

[4] B. G. Kornreich, "The patch clamp technique: principles and technical considerations," *Journal of Veterinary Cardiology*, vol. 9, no. 1, pp. 25–37, 2007.

[5] T. J. Dale, C. Townsend, E. C. Hollands, and D. J. Trezise, "Population patch clamp electrophysiology: a breakthrough technology for ion channel screening," *Molecular BioSystems*, vol. 3, no. 10, pp. 714–722, 2007.

[6] J. M. Hilbe, "Review of SigmaPlot 9.0," *American Statistician*, vol. 59, no. 1, pp. 111–112, 2005.

[7] W. J. Heitler, DataView, 2009, http://www.st-andrews.ac.uk/~wjh/dataview/.

[8] R. Marrannes and E. D. Prins, "Computer programs to facilitate the estimation of time-dependent drug effects on ion channels," *Computer Methods and Programs in Biomedicine*, vol. 74, no. 2, pp. 167–181, 2004.

[9] Z. W. Liu, T. Lei, T. Zhang, and Z. Yang, "Peroxynitrite donor impairs excitability of hippocampal CA1 neurons by inhibiting voltage-gated potassium currents," *Toxicology Letters*, vol. 175, no. 1–3, pp. 8–15, 2007.

[10] T. Yu-Tao, L. Zhao-Wei, Y. Yang, Y. Zhuo, and Z. Tao, "Effect of alpha-cypermethrin and theta-cypermethrin on delayed rectifier potassium currents in rat hippocampal neurons," *NeuroToxicology*, vol. 30, no. 2, pp. 269–273, 2009.

[11] Y. T. Tian, Z. W. Liu, Y. Yao, T. Zhang, and Z. Yang, "Effects of alpha- and theta-cypermethrin insecticide on transient outward potassium current in rat hippocampal CA3 neurons," *Pesticide Biochemistry and Physiology*, vol. 90, no. 1, pp. 1–7, 2008.