« User’s guide for Nanopore Data Analysis package»

Lausanne

2017

**Contents**

[Introduction 3](#_Toc495939688)

[1. User Guide 3](#_Toc495939689)

[1.1 Installation 3](#_Toc495939690)

[1.2 List of required modules 4](#_Toc495939691)

[1.3 Main window 4](#_Toc495939692)

[1.4 Load button 4](#_Toc495939693)

[1.4.1 Loading Axopatch (.dat) 5](#_Toc495939694)

[1.4.2 Loading Chimera (.log and .mat) 5](#_Toc495939695)

[1.5 Low pass filtering problem 6](#_Toc495939696)

[1.6 How to play ping pong? 7](#_Toc495939697)

Introduction

Pyth-ion is a python based program designed for analysis of nanopore ionic current data.

There are several methods for detecting the structure of molecules passing. The first is based on measuring the ionic current through the nanopore. When the molecule enters the nanopore, it causes a sharp drop of the ionic current associated with blockage of the regular flow of ions through the pore.

The second method is more complicated, but allows to explore the molecule in more details. The idea is to measure the ionic current and the current through the membrane, caused by the bias of the voltage at its ends. In this case, when a complex molecule consisting of charged parts passes through the nanopore, it is possible to make conclusions about its internal structure based on knowledge of the impact of elements on the current through the membrane.

Experiments in our laboratory can receive data from two different reading devices. The first, amplifier Axopatch 200B (Molecular Devices, Inc. Sunnyvale, CA) allows us to obtain experimental data only from one channel (ionic) sampled at 100 kHz. The second, the Chimera Instruments VC100 amplifier allows us to receive data from two channels sampled at 4 MHz. Experiments with two channels are now of great interest, since they allow obtaining more information about the analyte. For data analysis, various software is used. The first OpenNanopore program was implemented in Matlab computational environment and allowed to analyze data only from one channel. The speed of data processing was quite slow for large data (several millions of points). Experiments that require reading 3 millions of points from each of two channels can no longer be processed with this software. In this regard, for processing such a large amount of data, a new software was developed in the Python language (freely distributed), which allows us to quickly implement big data analysis from both channels simultaneously. This work presents the main features of the new software for processing experimental data.

The software provides the ability to process data contained in the following types of files: .log (Chimera Amplifiers), .dat (Axopatch).

# User Guide

## Installation

This package can be easily run from a standard Anaconda 3 distribution, the only additional requirement is pyqtgraph.

## List of required modules

Python 3.4 or higher:

* Numpy
* Scipy
* Pandas
* Matplotlib
* Pyqtgraph
* Pyqt5
* H5view
* H5py

## Main window

The screenshot of main window is presented in Figure 1. Detailed explanation of all buttons and figures can be found on the text below.

Изображение выглядит как снимок экрана, компьютер, ноутбук, внутренний

Описание создано с очень высокой степенью достоверности

Figure 1 Pyth-Ion Main window

## Load button

Press Load button on the main window. You can choose a file (see Figure 2) which is suitable for the program (.log, .dat).

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

Figure 2 Load window

### Loading Axopatch (.dat)

Axopatch file can be easily loaded by clicking on the corresponding file on the Load window. Axopatch file can contain information from single source (Ionic current) or from two channels (Ionic and Transverse currents).

### Loading Chimera (.log and .mat)

Chimera data loading has features: the samplerate and the size of such type of files are high (Samplerate approx. 4 MHz, 20 Mb data). In this connection the data should be filtered and downsampled before being shown. In the Figure 3 you can see the window for choosing the filter options. First, the program filters the data with Low Pass Filter at a given frequency (100 kHz at our example) with the help of Scipy’s 3-th order Bessel filter. Then it is downsampled with Scipy Resample function to a given samplerate frequency (201 kHz at our example). Note, that the Low Pass Filter should be less than  . Meanwhile the samplerate after downsampling must be at least two times higher than Low Pass Filter frequency.

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

Figure 3 Window for selecting filter parameters

## Low pass filtering problem

Big data from Chimera requires downsampling before being processed. The initial data has signal components at very high frequencies. When such signal is resampled, high frequency components interact each other which cause a phenomenon called aliasing. To avoid this problem, the low-pass filters can be applied. Artificial filters in the program are not ideal. The result of applying a non-ideal filter is seen in Figure 4. It can be seen from the figure that the filter cuts out both the high-frequency components and some part of the low-frequency ones (see Figure 4). This fact should be taken into account when choosing the cutoff frequency of the filter.

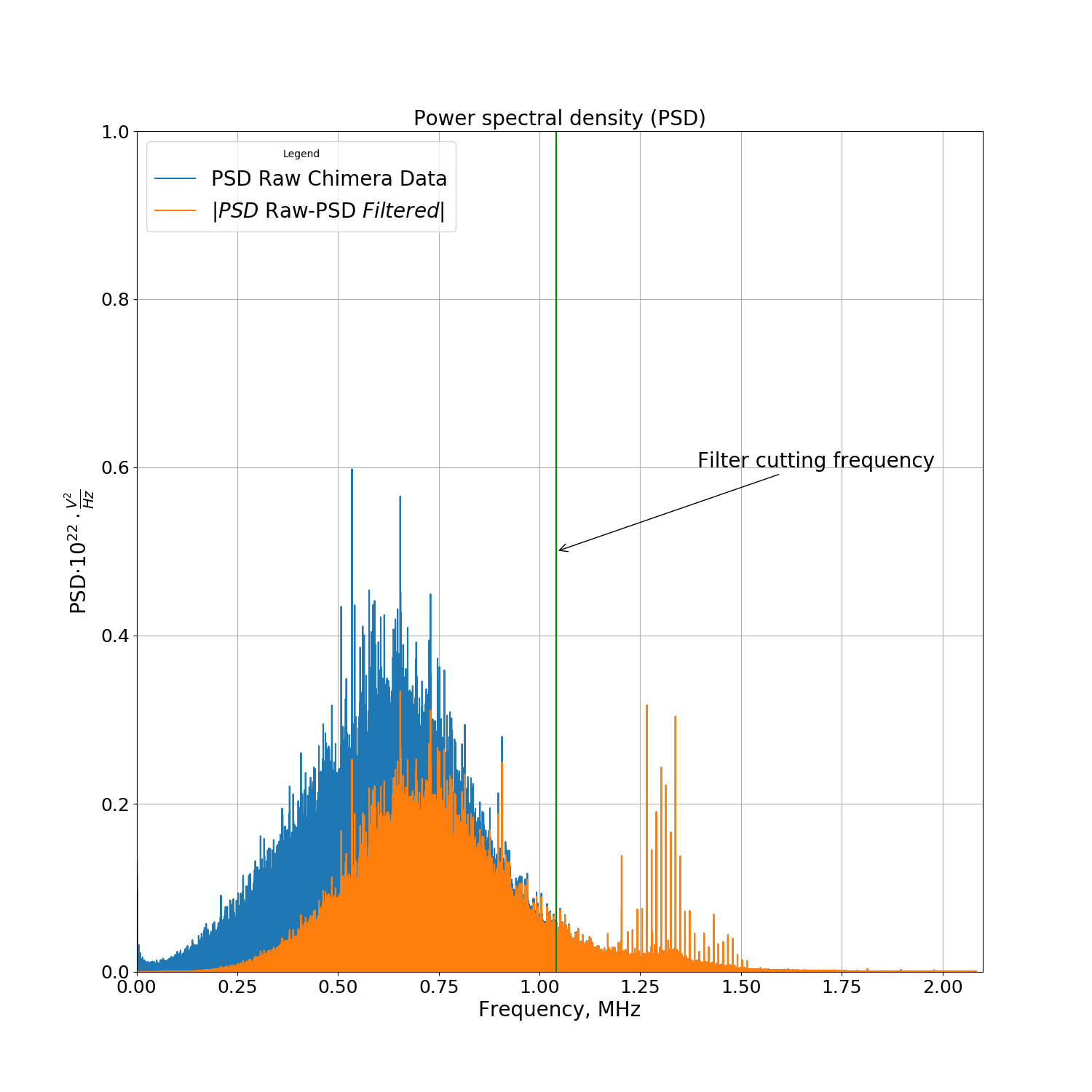


Figure 4 Power spectral density. PSD of raw Chimera Data (blue). |PSD Raw Chimera data – PSD data after filtering| filtered with 3rd order Bessel filter (orange). Green line indicates the cutoff filter frequency.

## How to play ping pong?

Load suitable file. Set the preliminary parameters shown in the Figure 6. Details of the parameters described below. As “prepingpong” button is pressed, the program will try to find points (within the region from “Start search” to “Search end”) where the following condition is satisfied:



(these points are called “zero points” in the program).

As all “zero points” are found press “pingpong” button: the program selects the region for event search. Let us consider it on the example given in the Figure 7. In order to find event, the program starts from “zero points” (event number 1, for instance). Then it evaluates the standard deviation over the region [“zero points”(1)- Delay back trace: “zero points”(1)- Delay back trace+ Safety region]. Then it applies the cumulative sum algorithm CUSUM (within the region [“zero points”(1)- Delay back trace+ Safety region: “zero points”(1)]) to find event parameters such as start, end and depth of event. “Level Threshold” and “Minimal duration” serves as parameters for the flgorithm for minimal event depth and length correspondingly.

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

Figure 6 Searching parameters for ping pong



Figure 7 Example of the ping pong search of events