

Jeremy Bergsman's Processing and Analysis Tool

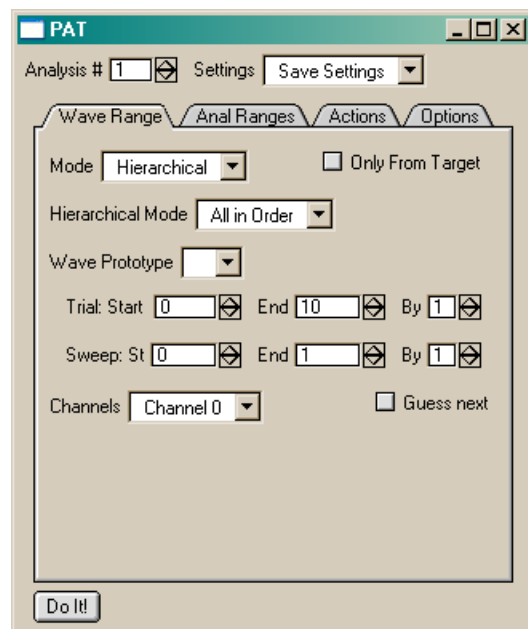
Introduction

The basic idea of this tool is that it performs various actions on multiple waves. You can pick which waves the tool will run through and have one or more actions performed on each wave. The strength of the tool is the ease and flexibility of picking multiple waves, and that multiple actions can be performed simultaneously. There are several built-in actions and it is easy to add more. The actions include things like averaging and duplicating waves (“Processing”) and making numerical measurements from portions of the waves that you specify (“Analysis”). It is designed for analysis of electrophysiological experiments but can do many things useful for any Igor user. Analysis result waves are graphed automatically and a fairly complete record of what was done is saved in the History. Although perverse use will allow it, the tool is designed not to modify processed waves. Instead duplicate waves will be created when needed (see **Actions** below). In this document I have tried to use a **different font** when referring to parts of the PAT panel. Screen shots containing most of these appear with the descriptions.

How it Works

Selecting the PAT tool from the Macro menu or the command window creates a panel called **PAT** which controls the analysis. The analysis is actually run when you press the **Do It!** button which appears at the bottom of the panel. Selecting the tool again will bring the panel to the front. If you close the panel and reopen it you will erase all your current settings (not your saved settings) but your results will be fine. The various parts of the panel are described below, which will, I hope, reveal the means of its operation.

Analysis # simply supplies a number which will be used on the result waves and graphs to keep them separate. It will appear after an underscore at the end of the result wave name. It will also be used to name any graph windows which contain results. You can use this number to look up what you did in the history of the command window. This number is automatically incremented when you run an analysis. Analysis numbers from 1-500 within a given experiment are also recorded so that you don't accidentally overwrite a previous analysis, although you can force an overwrite if you want. Even if you run different actions, most results of a previous analysis with the same number will be erased. (You may defeat overwrite checking—see the code.)



The **Settings** popup allows you to save and set virtually all the remaining settings for reuse later in this or other experiments. Once you have settings that you like, choose the **Save Settings** option. You will be prompted for a name. Once you have saved settings, that name will appear in the popup for later selection when you wish to repeat a function.

Wave Range Tab

The first tab contains the parameters which control which waves the tool works on. There are 4 **Modes** in which it can get waves: (1) according to waves named in a rigid hierarchical naming scheme, (2) according to a pattern in the wave name, (3) waves with names which vary only by a number in the name, and (4) from a function that you can write. This function can be called automatically from PAT without any changes to its code. Any wave names generated by any of these modes can be limited to waves present on the top graph or table by checking **Only From Target**. The parameters below appear based on which value of the Mode popup is selected:

Hierarchical

The hierarchical mode is currently dependent on one of two particular wave naming schemata. It shouldn't be hard to adapt to other similar schemata. The two schemata are:

AXON: X_TTTT_ChY_S where X is anything, with any number of characters, T is the trial number padded on the left with zeros, Y is the channel # (0 or 1), and S is the sweep number (any number of digits). This is the result of Axon binary waves loaded with my modified version of Christian Stricker's procedure. You can get this where you got PAT.

HEKA: X_R_T_S where each part can be any number of characters with X being anything, R the recording number, T the trial number, and S the sweep number. This is a result of loading HEKA waves with the Patcher's Power Tools procedures available from the Wavemetrics web site.

PAT can handle suffixes to the above schemata if they start with an underscore (“_”). To switch between AXON and HEKA naming you must change a string constant in the program. You will see this in the first line of code after the initial comments:

```
Static Strconstant sDataSource="AXON"// Choose AXON, HEKA, or PULSE as appropriate (case sensitive)
```

(PAT comes with a wave browser which can easily navigate waves named according to the above schemata. It uses the same constant to control which type of waves it navigates. The wave browser is also set up to work on data collected with Pulse, from Richard Bookman's lab, and PAT will function with some limitations on this non-hierarchical scheme. In this case the wave names are assumed to be wNNNNN[_Suffix] where N and Suffix are any length, N is a number.)

The **Hierarchical Mode** popup determines how the results of the value-extracting actions (Maximum, Minimum, Biggest, Average, Baseline, Number, PPR, and Fit; see below for more information about actions) are grouped together with respect to the wave hierarchy. **All in Order** and **vs. Time** both keep all the results separate in a single 1D wave. The **vs. Time** option will create a second wave with the time values which correspond to the points in the first wave. (Time values depend on an extractable value in the wave note giving the time. My axon binary file loader puts this time information in the wave note automatically.) **Average by Trial** means that the result wave will have as many points as there are sweeps in a trial, with each point being an average of all the sweeps at the same position of each trial. Use this when each sweep in a trial is different but each trial is the same as another. This is basically "averaging trials together". **Average by Sweep** is the opposite. Use this when each sweep in a trial is the same, but something else is different between trials. You can also **Group Trial and Sweep**, in which case results are placed in a 2D

wave with sweep # indicated by the X index and trial # indicated by the Y index. The 2D wave is also broken into a series of 1D waves and graphed by a function included in the package which can be used to do this for any 2D wave. (In some cases you may need to understand the processing order. All the sweeps for the first trial are processed, followed by all the sweeps for the second trial, etc. This is done for each recording (HEKA) or channel (AXON) in turn.) (The **Hierarchical Mode** popup also controls the behavior of some other features. For example, see wave averaging below.)

Wave Prototype supplies a sample wave to use as a base in coming up with the names of the waves to process. The trial, sweep, and channel/recording of the wave prototype are unimportant as they are controlled by the parameters below. These parameters should be largely self-explanatory except that the **By** boxes control the increment of the range, so for example you can do trials 3, 6, 9, and 12 by setting **Start** to 3, **End** to 12, and **By** to 3. The increment may be a negative number, in which case the **End** value should be less than the **Start** value.

Although you can process more than one channel (Axon) or recording (Heka) with one analysis, the results are kept separate as if they were done separately. The **Guess Next** check box causes the trial range to be incremented to do an adjacent block of trials the same size as the current block. This is useful for rapidly stepping through trials if you want to perform the same type of analysis on each block, but don't want to group all the results together. For example, maybe each block of trials is in a different drug.

Pattern Matching

In this mode you simply use wild cards to specify waves which will be processed. This is very powerful but unlike the other modes there is no control over the order of the processing so this is probably best for processing operations like duplicating, averaging, and smoothing waves. With pattern matching this tool largely supersedes the Wave Averaging tool supplied with Igor, although it currently doesn't calculate an error wave (this would be easy to add). See the Igor manual under WaveList for how to use the * and the ! characters

Incremental

In this mode you supply a sample wave name containing one or more % characters. PAT will substitute the numbers in the range you specify below wherever the % character appears. Again you can increment the value by any integer.

User Func

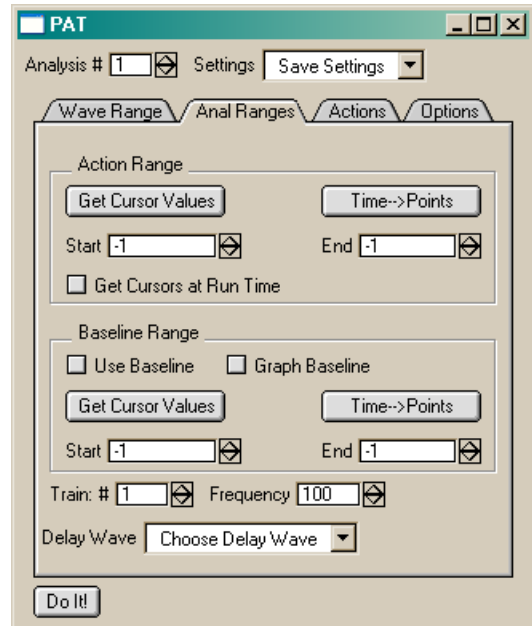
You may write a function which returns a valid wave name (perhaps by monitoring some of the global variables in PAT, or just getting them out of a list you construct). PAT will run until the function returns an empty string. The results are simply numbered in the order they are processed. All user string functions which don't take any parameters will be listed in the popup—just select the one you wrote. Included is a sample user function (SelectedWavesOnTable) which allows processing of all selected waves on the top table.

Anal Ranges Tab

The built-in actions are described below. Many of them analyze only a portion of the wave. This portion is called the **Action Range** and is specified in the **Anal Ranges** tab on the panel. Many actions will also make measurements of and even use (i.e. subtract from other results) a baseline

from the wave if you like. Turn on the baseline feature with the **Use Baseline** check box. If used, the baseline is also extracted from a specified portion of the wave being analyzed called the **Baseline Range**. Some actions require a baseline measurement (see below) and selecting them will turn on the baseline checkbox automatically. PAT will automatically save the measured baseline values in a wave if they were used, but you can have that wave included in the output graphs automatically by checking the **Graph Baseline** check box.

Obviously you must specify the **Action Range** and (if used) the **Baseline Range**. The values must always be in *points* (as opposed to the *X value*) along the wave. There are four ways to specify these values: 1) position the cursors at the relevant points of one of the waves and press the **Get Cursor Values** button in the relevant box on the panel, 2) type the point values into the entry fields, or 3) type the time (or whatever unit is represented by the *X* value of the waves) into the entry fields and press the **Time-->Points** button. This button takes the scaling from the first wave in the top window, so make sure this matches the waves you will be analyzing. In the (4)th approach, the cursor values are taken from the top graph at run time. For this method check the **Get Cursors at Run Time** check box.



The ranges specified above are really only the starting points of how you can define what interval is analyzed. The basic numerical measurements may be repeated at regular intervals within the wave. The number of measurements is given in the **Train #** box. The first interval analyzed is the one specified as above. The following measurements will be repeated according to the **Frequency** specified and according to the *X* scaling of the wave being analyzed (i.e. $N/\text{fundamental unit}$ where *N* is the value you enter in the **Frequency** box). **Frequency** may be a decimal and may even be negative to go backwards from the interval specified explicitly.

Having a **Train #** greater than 1 results in qualitatively different operation, with different windows being created, and different naming of the result waves. In this “train mode”, one result wave is produced for each analyzed wave for each action, rather than one wave that spans all the analyzed waves for each action. However, in every case, the result waves and graph windows should have names which make what was analyzed clear. Trains cannot be used in the group **Hierarchical Mode**.

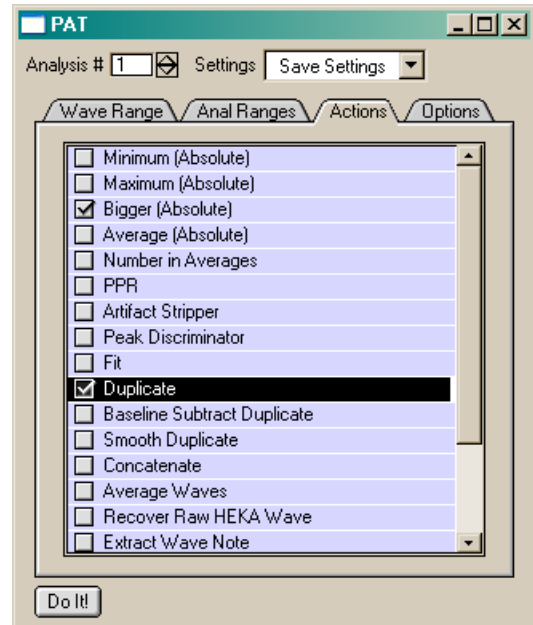
Separate from (but compatible with) the train mode, you may have an arbitrary offset of all the analysis intervals described above for each sweep within a trial. This offset must be specified in a wave, which is then selected in the bottom popup, **Delay Wave**. The sweep number being analyzed is used to look up an offset in the corresponding *X value* of this “delay” wave (although the offsets could be negative and therefore not be delays). The offsets themselves are in *points*. The popup first lists waves with names that start with the letters “Delay”, so it will be convenient to name your delay waves this way. Let’s say you have waves with sweep numbers 1-10 to analyze. The

interesting part of the wave occurs in a logarithmic fashion, starting at point 100, 110, 200, 1100, 10100, ... in consecutive sweeps. You would set the **Action Range Start** to 100 (and the **End** to 100+ the length of the event). Make a wave called, say, DelayTimes10 and fill it with the values 0, 10, 100, 1000, Now, since the X values of DelayTimes10 must match the sweep numbers, you need to change the scaling of this wave to Start=1, Delta=1. Select the wave in the panel and you are ready to analyze.

Actions Tab

This tab contains a list of all the possible actions. Check as many as you would like to be performed. In general the actions should not interfere with each other. If you cannot see with certainty how two actions can be performed simultaneously with unambiguous results you may wish to check the output carefully or perform the analyses separately.

Here is a listing of the actions I've provided. After the description will be a "B" if it can use the baseline feature and/or an "A" if it is only executed over the action range. An asterisk means that use of the baseline or action range is optional. All numerical analysis is performed on a(n optionally) smoothed temporary copy of the wave being analyzed. Enter 0 in the **Smoothing Number** box in the **Options** tab to skip smoothing. Some of the actions have parameters which are set in boxes that appear in the **Options** tab when the action is selected from the list.



- Maximum (A,B*) The maximum value in the **Action Range**.
- Minimum (A,B*) The minimum value.
- Bigger (A,B*) The max or min with the larger absolute value (if used, the baseline is subtracted before the comparison).
- Average (A,B*) The average value of all points in the **Action Range**.
- Number in Averages Number of results averaged together in the value-extracting actions.
- PPR (A,B) If you perform a **Train** analysis, takes the ratio of the second minimum to the first minimum.
- Artifact Stripper (A,B) Tries to remove a stimulus artifact from the analysis range. Looks for largely consecutive stretches of points over 5 SD from the baseline. Dependent on some parameters which are hard coded. Take a look if it's not working for you. If it finds an artifact, it is replaced by NaNs. Kludgy.
- Peak Discriminator (A,B) Finds size and location of the first "real", negative peak. The idea is to skip noise and very short, large peaks (e.g. stimulus artifacts)

and find a peak in an unknown place within a long stretch of data. In the **Options** tab specify the threshold size (in the d units of the wave) and the minimum length of a peak (in points) before it is considered “real”. Works with my data but a kludgy thing.

- Fit (A,B*) Any of the built-in Igor fits is applied. Does not work with **Trains** or **Group** mode. You may force the fit to work as if the left cursor is at X=0, and you may extrapolate the fit to the X values specified in the boxes in the **Options** tab. You may also specify any flags for the fit (see the Igor manual for flag use).
- Duplicate (A*) Duplicate each wave. The duplicate’s name will either substitute what you type in the box in the **Duplicating Options** box (in the **Options** tab) for the first letter, or, if you begin it with an underscore (“_”), append the contents of the box to the end of the wave name. You may also choose to only duplicate within the action range (again in the **Options** tab).
- Baseline Subtract (A*,B) Calls the duplicate action above and subtracts the baseline from the duplicates.
- Smooth (A*,B) Like Baseline Subtract but the duplicate wave is smoothed with the default Igor smoothing with a value determined by the number set in the **Options** tab. Smooth and baseline subtract can be used together.
- Concatenate All the waves are joined end-to-end. See Caveats below.
- Average Waves The processed waves are averaged together. In most modes all the waves are combined in one average. In the hierarchical modes Average by Trial/Sweep, the waves are only averaged together with their partners in the appropriate trial or sweep.
- Recover Raw HEKA Wave The raw Heka wave is calculated from the leak and leak-subtracted waves.
- Extract Wave Note Numerical values stored systematically in the wave notes can be extracted. You must specify the Key for the wave note in the **Options** tab (see the Igor NumberByKey function for insight). Useful for supplying X waves for plotting other results against. The ability to make these plots automatic will be added in a future version. My version of the Axon file loader includes some useful info in the wave notes which can be extracted.
- Call userFunc1 (A*,B*) This just prints out some messages as an example of an easy way to add your own actions to the tool. Pass the variables you need to functions that initialize, process, and clean things up. By writing them as function calls it should be easy to avoid unanticipated problems with interacting with my code.

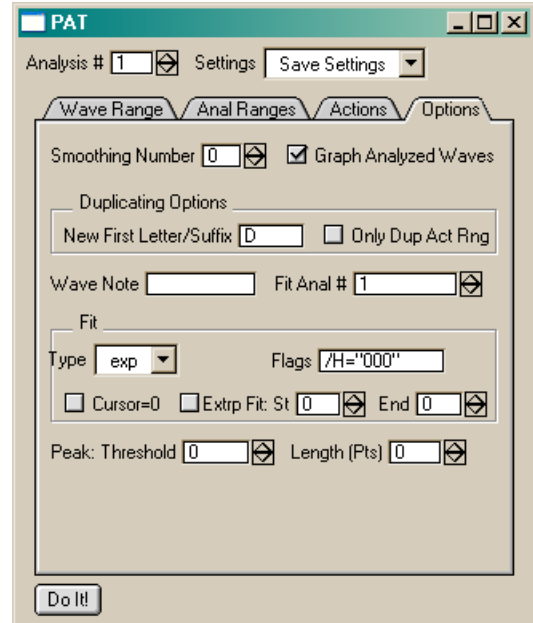
- Minimum vs. Fit

A strange thing I wrote to use the results of a fit to the wave being analyzed as the baseline for a minimum measurement (for overlapping EPSCs). You must specify the analysis number used to generate the fit.

Options

Two items are always present in this tab. (1) The **Smoothing Number** specifies the amount of smoothing applied to a temporary copy of the wave being analyzed before any numerical analysis is done. It also controls the smoothing of the duplicate made by the Smooth Duplicate action. (2) Checking the **Graph Analyzed Waves** checkbox causes a new graph window to be made containing all the waves which are analyzed.

The remaining items in this tab appear when needed by an action checked in the **Actions** tab. Their use should be clear from an understanding of the actions, described above.



Installation

Installation is similar to any Igor user procedure. Just put a shortcut/alias to the folder containing the pat.ipf file in your Igor Procedures folder.

Version information

The PAT panel, once created, is saved with an experiment. If you update to a newer version of PAT and reopen an old experiment with a panel from the older PAT version there may be some funny behavior. This can be avoided by closing the panel and reselecting PAT from the Macro menu. The program will also check the panel version when you press **Do It!** and update the panel if required. When you install a new version with new features requiring new parameters, your old settings waves will have arbitrary values of these new parameters. Usually this should be fine since the new parameters probably are used only by new actions not called by your old settings, but you may wish to pay extra attention to its behavior after updating the version. Opening old experiments after changing the setting waves can also cause problems (I hope to fix this in future versions) which, again, can be fixed by closing and reopening the PAT panel.

Caveats/Notes

- Graphing vs. time depends on being able to determine the time of a wave, which will be dependent on the data acquisition and importing methods.
- There is no interaction between the channels/recordings. They are separate in every way during analysis.
- Does not warn if you are about to duplicate over existing waves
- Note the difference between the averaging hierarchical *modes*, which average numerical results like max or min; the average *action*, which averages the points in the action range to produce a numerical result; and the *average waves action*, which averages waves together point by point.

- If you are in a mode which averages the results and there are no results in a certain position to average together (because there are no waves with the corresponding names), the result will be 0 instead of undefined. You can turn on the N action to check for positions with 0 results.
- Requires Concatenate Waves
- Concatenate works on the whole wave, not the action range (should probably change this).
- The scaling of the concatenated wave is that of the last wave processed
- Converting time to points is based on an arbitrary wave in the top graph when the button is pressed
- Likewise the time output in the history is based on the last wave processed, so it may not apply to all the waves. Always use points to be sure!
- Only tested on waves in the root data folder
- Never tested on a Mac and only limited testing with HEKA
- This is a beta version since it has mainly only been tested in ways I need to use it.

Ideas

- Can be used just to graph a bunch of waves. Turn on the graphing checkbox and turn off all analyses.
- Can create a graph depicting the data structure of the experiment. Turn on N analysis and use the **Average by Sweep** mode. Set the trial **Start** and **End** to the whole experiment and the sweep range to encompass the largest number of sweeps in any trial
- Duplicate, smooth, baseline subtract, etc. all the waves on the top graph by using **Pattern Mode** on the first tab with an * for the pattern and turning on **Only From Target**. Or just do it to those that follow a certain pattern such as all the fits on the top graph (fit*).
- Average whole families of waves at one go with the hierarchical modes **Average by Trial** and **Average by Sweep**

Credits

Written by Jeremy Bergsman in the laboratories of Richard W. Tsien (Stanford), David A. McCormick (Yale), and Pietro De Camilli (Yale). The basic idea is from an analysis tool by Frank Horrigan (Formerly Stanford). Stephen M. Smith (OHSU) helped with the original versions and with HEKA support.

Email me at: jeremy@bergsman.org

This document refers to version 2.73. Updated January 17, 2005.