# Tutorial to bownhill.exe

Bownhill is an interactive DOS command line program, it should run under any Microsoft windows® environment.

## Files

The following files are provided and have to be in the same directory

1. The batch file. e.g. 251115\_NTS\_sym1500\_110-080.bat
2. The set file, e.g. 251115\_NTS\_sym1500\_110-080.set
3. The amplitude histogram, e.g. 251115\_NTS\_sym1500\_110-080.amp
4. bownhill.exe together with
5. rauschb4.dat
6. cygwin.dll
7. emx.exe
8. wgnuplot.exe
9. Bessel\_4.mat (for 4-pole Bessel filters)
10. A sample fit result as screen shot 251115\_NTS\_sym1500\_110-080.jpg is also provided.

## Data format

Since computing speed is the most serious problem for bownhill, the program runs in a DOS-window

(command line).

The amplitude histogram must be given as a 1-column ascii file of integer numbers (also for the issue of speed). The number of rows in this file must be 4096, with 2048 = 0, corresponding to a 12-bit AD converter. If not available, measured current traces can be converted to amplitude histograms by the program Kielpatch (available on request).

Only the baseline noise can be entered as decimal number.

## Set file format

**The set-file** includes all information (except for the filter frequency, which in given in in the batch file, which bownhill needs to start:

The first line just includes notes of the user. It is not used by the program.

The header includes basic information:

|  |  |
| --- | --- |
| delta\_t = 0.0002  | Sampling interval in s, i.e. 0.0002 corresponds to a sampling frequency of 5 kHz. |
| datafile = 251115\_NTS\_sym1500\_110-060.amp | name of the file of the amplitude histogram |
| channel\_no = 1  | number of channels1 |
| no\_samples = 287767 | number of samples, has to be known2 |
| noise\_file = rauschb4.dat  | file for baseline noise3 |
| sigma\_noise = 9.8 | the width of the baseline noise |

**Remarks:**

**1The number of channels n** can be determined from the current trace by conventional methods. n = 2 is used if singe-channel recordings with two channels are used.

n = 3 has been used when the OmpF channel with three pores was investigated (Brauser et al., 2012).

**2The number of samples** should be delivered by the program converting the current trace to an amplitude histogram, e.g. in Kielpatch it is obtained from Files/settings.

**3The noise file rauschb4.dat** must be available, even though it is no longer used in bownhill, because in the mean time is has been replaced by a theoretical Gaussian distribution with the width given by sigma-noise or modified by the command “v” described below. However, bownhill shares the setfile read function with other programs that need this file and will crash if rauschb4.dat is not in the program folder.

Next is the matrix of rate constants for the Markov model that also defines the model topology. The first line (4 4) defines the size of the matrix.

|  |  |
| --- | --- |
| **model parameters kij (from i - to j)** 4 40 373000 0 092000 0 14.8 19650 240 0 00 17400 0 0 | Number of states of the model = number of rows = number of columns Row number = source state, column number = sink state |

The above rate constants k(row,column) are given for a 4-state branched model, namely

0 k12 0 0

k21 0 k23 k24

0 k32 0 0

0 k42 0 0

The next matrix determines, which parameters are fixed and which can be changed by the fitting algorithm:

|  |  |
| --- | --- |
| **Indices of the parameters in K to be fitted**0 2 | This is an example, where no fitting at all is done. The matrix has the size 0x2, thus the number of fitting parameters is 0. |

In this example: k12 and k21 are fitted, all other rate constants are fixed:

|  |  |
| --- | --- |
| 2 21 22 1 | Definition of the matrix size: 2 rows with 2 columnsThe rate constant with indices 1,2, i.e. k12The rate constant with indices 2,1, i.e. k21 |

If you want to fit all 6 rate constants in the branched 4-state model, you put:

|  |  |
| --- | --- |
| 6 21 22 12 33 22 44 2 | 6x2 matrixk12k21k23k32k24k42 |

It is recommended to start with 0 2 i.e., all rate constants fixed, in order to get a quick overview of the situation, but actually this depends on the preferences of the user.

**P of initial state distribution (t=0):**

1 4 one row, 4 columns

.25 .25 0.25 0.25

This is actually of no importance. However, take care that the number of columns (here = 4) is the exact number of states of the gating model and that the sum of the occupation probabilities in the 2nd row is 1.

**assignment states <==> output symbols given below**

1 4 one row, number of columns = number of states

1 2 1 1

First row: The “1” is obligatory (always 1 row) und “4” gives the number of states of the gating model (here it is 4).

Second row: “1” labels a closed state (C), “2” an open state (O). You also can introduce 3 and 4 for substates. The assignment of the numbers to the single-channel of each state is given by “current in DAC units for 1 = C and 2 = O”, below.

The column index of a number in the second row corresponds to the index of the state in the matrix of rate constants, above.

In the example provided by the included .amp, .bat and .set file, the assignment would be

State F on position 1 = closed, state O on place 2 = open, state S on place 3 = closed, state M on place 4 = closed.

The conductivities of these states are given the next matrix “current in DAC units for 1 = C and 2 = O”

**current in DAC units for 1 = C and 2 = O**

1 2 the 1st row gives the names of the current values occurring in the model

2051 2331 the 2nd row gives the values of the current levels of “1” and “2”.

The first number (1 = C) may deviate from 2048 = 0 pA because of electrode potentials. It has to be taken from the current recording. The second number (2 = O) is normally only a first guess of the true single-channel current. It can be taken from the open-peak in the amplitude histogram (or a little bit higher). It is normally readjusted during the fitting routine by the option “c” (c = current value).

If a sublevel is involved the first row would be

1 2 3

**User remarks**

Itrue = 280

Ieff = 209

Chisquare = 6.829

The end of the file is not read by bownhill and can be used by the user to store *Itrue*, *Ieff* and Chisquare as resulting from the final fit.

After a successful fit, the resulting fit parameters should be inserted in the set-file. This enables an immediate start from these values if the user wants a test of the fit in a the next run.

**Testing bownhill**

The required programs and data files are provided in this package to run one example.

**The model used in the example provided in the package is**



The indices used below correspond to F = 1, O = 2, S = 3, M =4.

**The batch-file** starts the fitting routine**.** It contains the command

bownhill -s 071114\_NTSM1\_101-60.set -F 1.5

-s means the next expression is the name of the set-file

-F means the next expression is the filter frequency in kHz multiplied by a factor of 1.5

Caution, the 1.5 in the example is not the factor of 1.5. The frequency of the low-pass filter is 1 kHz. After multiplying it by 1.5 it happens to give 1.5 kHz. Why one of PhD and Diploma students creating and improving bownhill has introduced this strange factor is unknown. Nobody has repaired it so far, because everything runs fine if you enter a filter frequency, which is 1.5 time the real filter frequency.

## Running the example

The bat, amp, and set files given in the package enable a start of the program to see how it works.

Do not expect to get exactly the same error sum (and results) in repetitive runs. The transitions between the states O and S are too rare and thus there is a minor stochastic scatter due to the generation of the amplitude histogram by random generators. However, this also holds for the measured time series, which would show the same scatter in repetitive experiments.

**Start the batch file 251115\_NTS\_sym1500\_110-060.bat (with “return” or double click)**

It calls the set-file 251115\_NTS\_sym1500\_110-060-290.set and starts the fit.

The set-file controls the loading of the amplitude histogram 251115\_NTS\_sym1500\_110-060.amp and all necessary parameters into bownhill.exe.

After the run has finished a DOS window appears and a gnuplot window opens.The gnuplot window shows the measured amplitude histogram in black and the theoretical amplitude histogram in red generated by bownhill for a visual inspection of the quality of the fit.

**The DOS window shows the fit results.**

The rate constants are presented by the same matrix as given in the set file

*Ieff* is calculated as *Ieff* = *Itrue* k12/(k12+k21)

Caution if (k12+k21) = 0! The gating model should be presented in a way that this does nor occur.

Or a version of bownhill has be used which does not do this calculation. Also, if you use a different gating model, the value of *Ieff* might not make sense in your situation.

**The DOS window offers a dialog of how to proceed**

Change (b)aseline

Change baseline noise (v)

Change (c)urrent

Set (r)ate constants by hand

(M)ultiply rate constants by a factor

Change (i)ndices of parameters to be fitted

Change calculation of (e)rror sum

Change (w)eighing factors

Change (n) of simulated data points

(S)tart fitting

(Q)uit program

The letter in brackets have to be entered. Then, a dialog is opened which allows enter starting values for new values of the parameters.

**Some details**

**Change (b)aseline** and **Change baseline noise (v**)

The baseline and baseline noise are determined from the sojourns of the channel in the closed state e.g. in the program Kielpatch.exe. However, sometimes this determination is not exact enough. Then these two commands enable a readjustment to obtain a perfect fit of the closed peak. The baseline has to be given in integers, but the noise may also be entered by decimal numbers.

**Change (c)urrent**. The true single-channel current, *Itrue*, is not a free parameter of the fitting algorithm. It has to be set by hand, and fits with different suggested values of *Itrue* have to be repeated until an optimum fit is reached. Depending on the statistics of the fast gating, *Itrue* can lie outside the peak of the amplitude histogram. From the deviations between measured and theoretical amplitude histograms as shown in the gnuplot window (the png file provided shows an example), a guess of the best *Itrue* can be obtained and verified by a subsequent fit. Often several runs are required.

**Set (r)ate constants by hand**

The dialog asks for the two indices (separated by space) of the rate constant to be changed. Then, the old value is shown and a new one can be typed in.

(If you enter return, the new value is zero, mostly not wanted by the user. Then just repeat the procedure with entering the indices once more. Assignment of values can be repeated several times, if you change your mind. Previous suggestions are overwritten.

The program continues to ask for new suggestions, stop it by entering one of the other commands, but not q, that will end the program. Typing “s” will end the dialog and start fitting.

**(M)ultiply rate constants by a factor.**

The dialog is similar to that of “r”, but now the new suggested value is obtained by multiplying the old rate constant by a factor you suggest. This is useful would when you believe that the ratio of two rate constants is okay, but you want to shift them in parallel.

 In the example of the 4-state model provided here, the ratio of the rate constants *k23* and *k32* determines the relative heights of the closed and open peak. The absolute values of *k23* and *k32* determine the height of the valley between the two peaks. If you want to lift this valley without disturbing the balance of the peaks, use the m-command.

**Change (i)ndices of parameters to be fitted.** The user can decide whether all rate constants should be fitted as free parameters or only part of them or none. In (Schroeder and Hansen, 2009), a strategy is given, of how to approach the final fit stepwise.

**Change calculation of (e)rror sum** The default error is the sum over n2/n. This turned out to be the best compromise between the error sum over *n*2, which ignores the region with small values and the sum over the relative error *n*2/*n*2, which is too sensitive to the high relative scatter of the small values of events, *n*.

**Change (w)eighing factors**

Weighing factors can draw the attention of the fitting routine to regions where the difference between measured and theoretical amplitude histograms is serious. We use this option very rarely.

In the dialog, the three numbers are given in a row to define the range(s) were a different weighing factor than the default value of 1 should be inserted:

Lower value width of the range value of the weighting factor

This can be repeated for more than one range.

To leave this dialogue, type:

0 space 0 space 0 return.

Unfortunately, this last line is obligatory.

**Change (n) of simulated data points**

Since the theoretical amplitude histogram is generated from simulated time series two random generators are employed. If the time series are not long enough, the stochastic nature of this process can cause different results in different runs (which also holds for the measured current traces). At least the theoretical amplitude histogram can be made more reliable if the length *n* is increased. However increasing *n* by a factor of 10 would also increase the computing time by a factor of *n*. Thus, this is done only in extreme cases.

**(S)tart fitting**

After “s” the fit starts, and the dialog is closed. If i = 0 (no free parameters) bownhill delivers the amplitude histogram of the actual parameter set without fit. This can be used to test the influence of rate constants modified by “r” or of *Itrue* (c), baseline location (b) or baseline noise (n).

**(Q)uit program.**

One way to end the program. The other way is to close the DOS window.

**Storage of the results.**

There is a version of bownhill, where the results of all fits are stored in the setfile. This results in a long list of rubbish, since all intermediate wrong fits are listed. Nevertheless, it is available on request.

Instead, it is recommended to replace the parameters in the setfile by the final fit results by hand. (The time required for this is negligible as compared to the time of a fit). This also offers the chance to start a new fit with these parameters if later on the fits are to be reinvestigated.

Unfortunately, there is no option to store the gnuplot graph showing the measured and the theoretical amplitude histogram directly. Thus, it is recommended to make a screenshot and create an image file, e.g. in IrfanView as shown in the jpg-file 251115\_NTS\_1500sym\_110-80.png provided in package together with this instruction.

**Remark**

Do not be disappointed if the test run gives a slightly different error sum. Any difference results from the stochastic elements in the generation of the simulated time series. The S-O transitions are rare. Thus, the random generators may cause a varying number of transitions with the same rate constants. However, this is just what the biological system does. There is also a random generator determining the time of transition and will lead to different results in repeated experiments.

The influence of the stochastic element can be reduced be increasing the length of the time series (enter “n”), if you are willing to wait some hours more.

Literature

Review Article

Schroeder, I. 2015. How to resolve microsecond current fluctuations in single ion channels: The power of beta distributions. Channels. 9:262–280

*Relationship between characteristic regions of the amplitude histograms and the rate constants*

Schroeder, I., Hansen, U.P. 2009b. Using a 5-state model for fitting amplitude histograms from MaxiK: ß-distributions reveal more than expected. *Eur. Biophys. J.* 38:1101–1114

*Simulations how far the beta distribution can look beyond the paw-pass filter frequency and permission to ignore shot noise*

Schroeder, I., Hansen, U.P. 2009. Interference of shot noise with fast gating analysis of ion channels – patchers do not (yet) have to care. *J. Membrane Biol.*  229:153-163

*Application to µs gating in BK (MaxiK*):

Schroeder, I., Hansen, U.P. 2007. Saturation and µs-gating of current indicate depletion-induced instability of the MaxiK selectivity filter. *J. Gen. Physiol*. 130: 83-97.

*Application to the effect of Tl+ in BK*

Schroeder, I., Hansen, U.P. 2008. Tl+-induced µs-gating of current indicate instability of the MaxiK selectivity filter as caused by ion/protein interaction. *J. General Physiol*. 131: 365–378

*Application to the distinction between Ca2+ effect and gating effect in BK*

Schroeder, I., G. Thiel, and U. P. Hansen 2013. Ca2+ block and flickering both contribute to the negative slope of the IV curve in BK channels. J. Gen. Physiol. Vol. 141: 499–505

*Application to fast gating in Kcv channels*

Abenavoli A, DiFrancesco M, Schroeder I, Epimashko S, Gazzarrini S, Hansen, UP, Thiel G, Moroni A (2009) Fast and slow gating are inherent properties of the K+ channel pore module. *J. Gen. Physiol*. 134: 219–229.

*Application to Enrofloxancin block in OmpF*

Brauser A., I. Schroeder, T. Gutsmann, C. Cosentino, A. Moroni, U.P. Hansen, and M. Winterhalter. 2012. Modulation of enrofloxacin binding in OmpF by Mg2+ as revealed by the analysis of fast flickering single-porin current. J. Gen. Physiol. 140:69–82