This demo program is related to the following publication:

Bedard, C., Behuret, S., Deleuze, C., Bal, T. and Destexhe, A. Oversampling method to extract excitatory and inhibitory conductances from single-trial membrane potential recordings. *Journal of Neuroscience Methods*, 2011.

This MATLAB demo program performs the extraction of conductances and preconductances, as explained in the paper. All examples shown here corresponds to testing of the method using models, as shown in Figs. 4, 5 and 7 of the paper. The program first reads the conductances from data files, and uses these conductances to compute the oversampled Vm from the membrane equation. It then extracts the "preconductances" from this oversampled Vm, and next removes the singularities from the preconductances (see paper). Finally, the preconductances are converted to conductances, which are compared to the original conductances.

General remarks:

1. the "data" folder contains line vectors for the excitatory conductance (gge.dat), the inhibitory conductance (ggi.dat) and time (tt.dat). The parameters for the cell and thresholds to remove singularities, as well as the initial voltage, are in a line vector of dimension 8 which name is "Parametres.dat" - everything must be in ascii in those files.

The localisation of data files (in ascii) is

data/exemplex/gge.dat
data/exemplex/ggi.dat
data/exemplex/tt.dat
data/exemplex/Parameters.dat

The demo has several examples, corresponding to some of the figures of the paper. To get "exemple1", write "exemplex=exemple1" (see Extraction.RR at lines 44,47,48 and 49)

2. The organization of the program is the following

ExtractionRR.m : main function

In a first step, the main program generates the oversampled voltage from the conductances located in data/exemplex/... and compares them in a graph. Note that the program actually computes two Vm: the oversampled Vm, and a "normal" Vm at the same sampling frequency as the conductances. The latter is used for comparison and to make sure that the oversampling factor is enough.

In a second step, the program extracts preconductances from the oversampled Vm, remove singularities, convert to conductances, which are then compared to the original conductances.

signalRR.m : function to generate the oversampled Vm from the conductances using the function eqdifferentielleRR.m

The oversampling factor can be set in line 65 in the main function ExtractionRR.m.

algorithmeRR.m : function to extract conductances and preconductances; the singularities are removed in the last part of this function.

graphRR: function to graph the results.

graph 1 : Compares the initial Vm with the oversampled Vm.

graph 2 : Compares the original preconductances with the ones extracted.

graph 3 : Compares the original conductances with the ones extracted.

graph 4 : Compares the extracted preconductances before removing singularities, with the ones with all singularities removed.

3. The "data" folder includes five examples. Each example consists of one file (ANote.txt), who explains the organization of "Parametres.dat".

example1 : conductances generated using a shot-noise stochastic process (similar to Fig. 7 of the paper)

example2 : conductances following a periodic function (Fig. 5 of the paper)

example3 : conductances followsing an oscillatory exponential functions

example4 : conductances generated using the "sinc" function ($\sin(x)/x$)

example5 : constant conductances (Fig. 4 of the paper)

Claude Bédard
Unité de Neurosciences, Information et Complexité (UNIC),
CNRS, 91198 Gif-sur-Yvette, France
bedard@unic.cnrs-gif.fr