Overview

The code performs a Bayesian analysis of decision-making data. There are two metrics per decision—reaction time and choice accuracy (correct/incorrect)—and two experimental variables:

1. the difficulty of the perceptual task, and

2. the presence, strength, and site of experimental interference with neuronal function.

In the odor intensity discrimination task of Wong *et al.*, perceptual difficulty is quantified as the logarithm of the odor concentration ratio to be distinguished; the type of experimental interference is determined by the size and subcellular location of photocurrents evoked in different classes of neurons (no, low-, or high-intensity light; dendritically targeted, untargeted, or no CsChrimson; CsChrimson expressed in αβc KCs or α'β' KCs). The code returns the posterior probability density functions of several model parameters (see below) and computes the Bayesian Information Criterion (BIC) in order to establish:

1. whether the decision strategies formalized by drift-diffusion (DDM) or extrema detection models (EXD) best describe the experimental data, and

2. which model parameter(s) are most likely altered by different types of experimental interference with neuronal function.

## **Installing and running the code**

This code was written and run using:

Windows 10

Intel Xeon processor (4 physical cores) @ 3.6 GHz

64 GB RAM

MATLAB 2019b, using the Statistics and Machine Learning, Optimization, and Parallel Computing (optional) toolboxes.

The code has also been tested successfully in MATLAB 2020b and 2022b.

If you do not have the parallel computing toolbox, the code will take longer to run.

To install the software, copy all .m files to a single folder and either set it as the MATLAB working directory or include it in the MATLAB path. The zip files contain the source data and the output files for verification. Unzip these. It’s convenient to keep them in the working directory.

To allow verification of the code, a test data set and test mode are provided that use a smaller number of samples in the *slicesample* function. The test set contains the first 50 decisions from each genotype–light intensity–difficulty level combination of the entire data set. Verifying the code in this way takes less than five minutes on the machine described above. To configure the code for test mode, open the file “script\_BayesSolve.m”, navigate to section 1, line 21, and change the value of “testingFlag” to true. Then run the script file from the MATLAB command window (type “script\_BayesSolve” and press enter). A prompt window requesting the source data folder opens. Navigate to and select the parent data folder, “test set”. The script should run in a few minutes, printing one line of output for each genotype–stimulation–model combination. A folder in the working directory will be created called “output <time stamp>” that contains the posterior distributions of model parameters for the five different flavors of drift-diffusion model (models 0–4; see below). Two tab-delimited text files summarize the output of both the drift-diffusion and extrema detection models. The output should be identical to that in the folder “test output”.

To analyze the complete data set with the parameters used by Wong *et al.*, set the “testingFlag” back to false and run the script file from the MATLAB command window (type “script\_BayesSolve” and press enter). Select the data folder “behavioural parameters”. The analysis took just under 17 hours to run on the machine described above. The output should be identical to that in the folder “all data output”.

After running the code, relevant fitting parameters and the posterior probability density functions will be saved to the file “fitting <time stamp>.mat”. This can be reloaded by following the instructions at the top of section 1 of script\_BayesSolve.

## Description of data structure

Reaction times and choice accuracies are stored in the comma-separated files “reactiontimes.csv” and “trialaccuracy.csv”. There is one pair of files for each combination of genotype, light intensity ("optogenetic stimulation level"), and task difficulty ("odour concentration ratio"). The directory structure is:

…\<genotype>\<optogenetic stimulation level>\<odour concentration ratio>\<file>

<genotype> is one of:

“NP6024 AMPAR (targeted Chrimson)”

“NP6024 CS (no Chrim)”

“NP6024 NLG1 (untargeted Chrimson)”

“VT030604 AMPAR (prime KC)”

Data from different genotypes are analyzed independently of one another (i.e., they are not grouped in any of the global models). The names of the folders have no value. If you wish to change, or add to, the genotype folders, update the variable “sources” in section 1 of the file “script\_BayesSolve.m” accordingly.

<optogenetic stimulation level> is one of:

"high intensity"

"low intensity"

"no light"

Data from different optogentic stimulation levels are grouped in global models. These folder names have no value (or order) and can be changed quite freely. However, the code is currently limited to accept a maximum of three different stimulation levels. To increase this number, the function “globalHandle.m” will require modification. To analyze data from a single stimulation level, use only one optogenetic stimulation folder, and configure the script\_BayesSolve to run the independent model only (in section 2, change globalFitInds to [] and keep fitIndepFlag set to true).

<odour concentration ratio> is in the format <LHS flow rate>vs<RHS flow rate> where the flow rates take the values 10, 50, 70, 90, or 100.

The difficulty of the task is calculated in the script as logDiff = |log(<LHS>/<RHS>)| in section 1 of script\_BayesSolve. If you wish to change this, modify the parameter logDiff.

## Description of global models

tND and sigmaND refer to the mean and standard deviation of the non-decision time. A is the bound height or evidence threshold. k is the drift rate in drift-diffusion models (DDM) or the standard deviation of successive evidence samples in the extrema detection model (EXD). Independent models are run separately for each optical stimulation level and so have separate A, k, tND, and sigmaND for each stimulation level. The global models assume that some or all parameters are the same across the different optical stimulation levels as summarized in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model number | A | k | tND | sigmaND |
| 1 | Free | Free | Global | Global |
| 2 | Global | Free | Global | Global |
| 3 | Free | Global | Global | Global |
| 4 | Global | Global | Global | Global |

It is possible to modify the decision-making models by changing the functions “pDDM” or “logP\_RT\_acc” and the “globalHandle” function.

## Description of output

After running the script, a folder is created in the working directory called “output <date stamp>” where date stamp is in the format “yy-mm-dd HHMM”. This folder contains the posterior probability density functions of the five different DDM models. The files are named “DDM M<X>. <text description>. <genotype> <optogenetic stimulation level>”. The model number <X> is 0 for independent models, or 1 to 4 for global models, as defined in the table above. The text description lists the global and free (individual) parameters. The genotype is determined as described in the description of data structure. The file names for independent models (number 0) also indicate the stimulation light intensity. For the global models, the stimulation light intensity of the free parameters is included in the column titles.

Separate summary output files for the DDM and EXD models contain the means and confidence intervals of the fitted parameters and the BIC for each model. The summary files are in tab-delimited format, allowing neat copying into your favorite spreadsheet package.