VBA toolbox

GETTING STARTED

VBA toolbox

Official website: https://mbb-team.github.io/VBA-toolbox/

What is the VBA toolbox?

Most models of neurobiological and behavioral data can be broken down into processes that evolve over time and static observation mappings. Given these evolution and observation mappings, the toolbox can be used to simulate data, perform statistical data analysis, optimize the experimental design, etc… In brief, the toolbox provides:

- plug-and-play tools for classical statistical tests
- a library of computational models of behavioral and neurobiological data time series
- quick and efficient probabilistic inference techniques for parameter estimation and model comparison (+ experimental design optimization)
- graphical visualization of results (+ advanced diagnostics of model inversion)

Requirements

This toolbox runs in Matlab. Although it should run in all versions of Matlab, the toolbox has only been extensively tested on Matlab 2013 and higher.

How do I install the toolbox?

Get the toolbox

- Ideally, use Git to clone the repo of the toolbox on your computer:

```
cd ~/path/to/parentDirectory
git clone https://github.com/MBB-team/VBA-toolbox.git
```

You will then be able to stay up to date with the latest versions using the command:

```
cd ~/path/to/parentDirectory/VBA-toolbox
git pull
```

- If you don't want to install Git, you can alternatively download a zip of the latest stable release of the toolbox directly form the download page of the official website: http://mbb-team.github.io/VBA-toolbox/download.

Add the toolbox folder to your Matlab path:

```
cd ~/path/to/parentDirectory/VBA-toolbox
VBA_setup()
```

Note that you might have do run `VBA_setup()` after an update of the toolbox (eg. after a `git pull`).

Enjoy!

You can now try one of the demos or tutorials you can find in the `VBA-toolbox/demos` folder. If you have a recent version of Matlab (>= 2017), you can also run `VBA_test()` to check that everything works as intended on your system.

Structure of the toolbox

- `/` contains all the functions you can use directly in your scripts to call general routines like model simulation and inversion.
- `/core` contains the sub-functions that implement the internal algorithms of the toolbox, like the variational estimation scheme. You should not use those functions directly!
- `/demos` contains a large selection of computational models (ie. evolution and observation functions) you can use directly or adapt to your hypothesis. You will also find in this folder a series of demos that implement those models and tutorials demonstrating the various features of the toolbox.
- `/legacy` contains some old code that will soon disappear but we keep for backward compatibility.
- `/modules` contains a set of tools that complement the toolbox, like DCM generators or advanced models and scripts used in publications.
- `/sandbox` contains code in development that is not yet fully functional and / or tested. Feel free to test them, or wait a bit until they move to the core.
- `/tests` contains unit testing functions. This code help us ensure that the toolbox does what we want it to do…
- `/third-party` contains code we did write ourselves but is needed by the toolbox.
- `/utils` contains plenty of cool tools that you can use directly if you need, like random number generators, mathematical measures (eg. KL divergence), or nifty numerical tricks.

Want more details?

Please visit the wiki pages for tutorials, demos, and advanced features descriptions.

You can also seek help on our dedicated forum. We will always be happy to help you with the toolbox if you need.

How can I participate?

The VBA-toolbox is an open-source, collaborative project. We gladly welcome contributions at all levels:

- flag a bug or request a feature by creating a new issue on Github
- provide your model to other users by integrating it directly in the toolbox. Send us an email, or directly create a pull request.