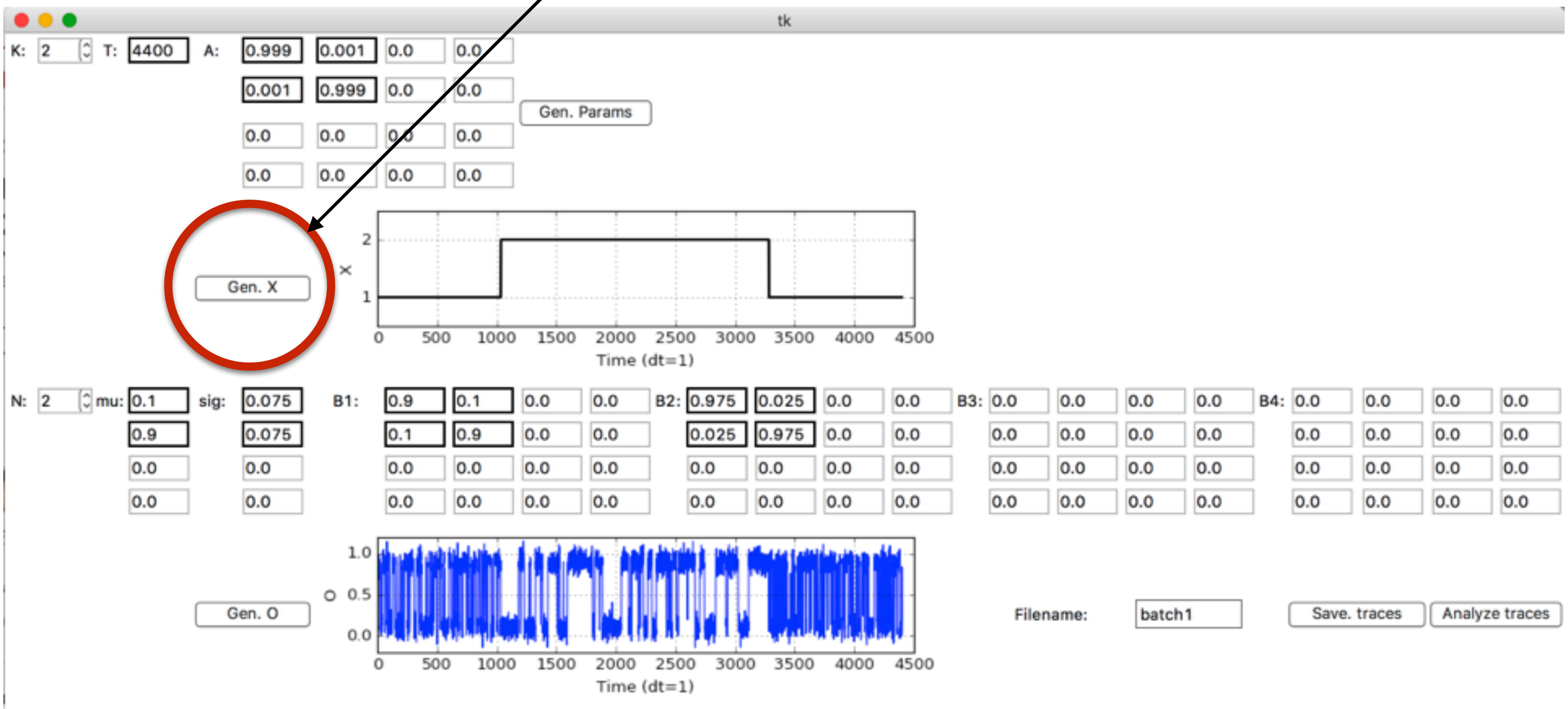
How to use VBDCMM

1. VBDCMM_gui_simul.py (short version)

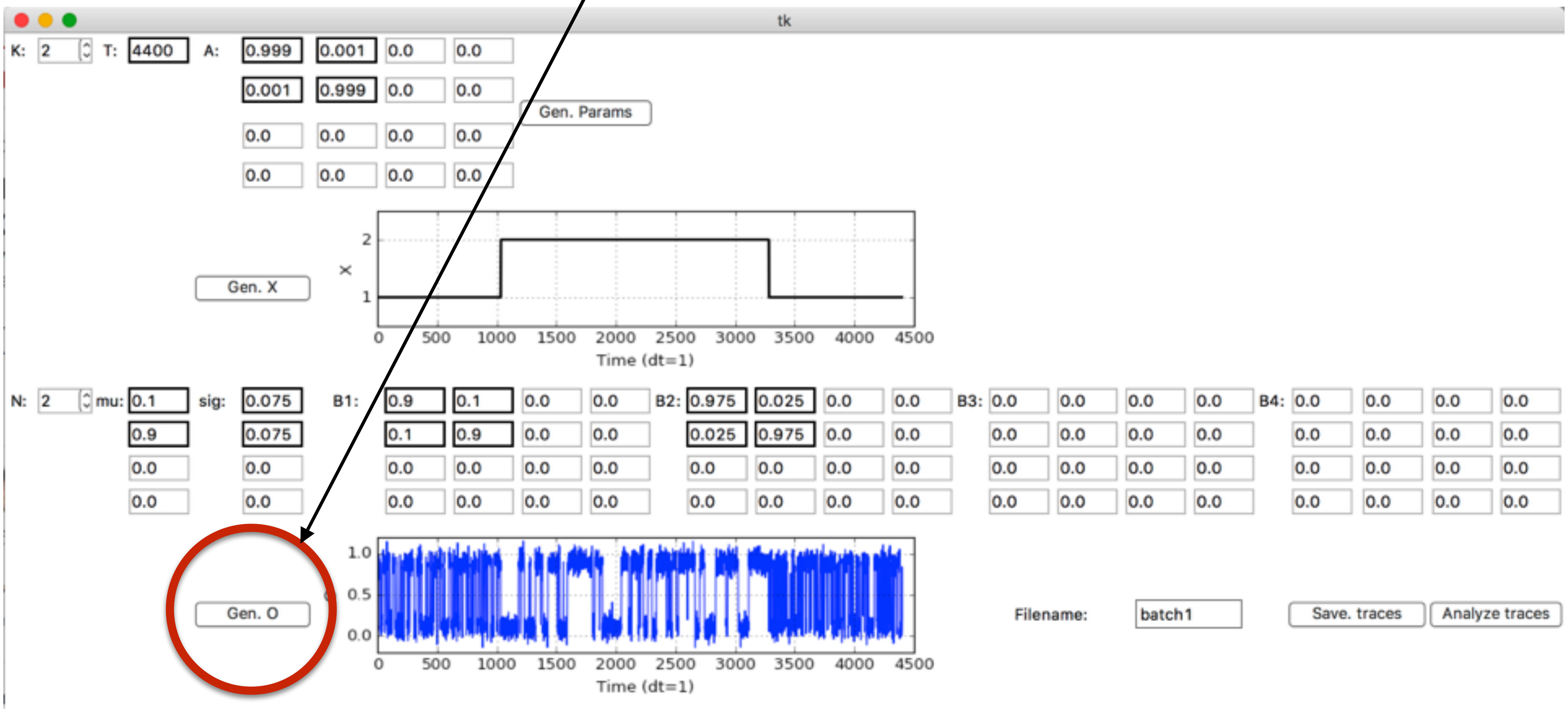
(1) Open terminal.
(2) Move to the folder where VBDCMM source files are downloaded),
(3) Type: `python3 VBDCMM_gui_simul.py`

Click this



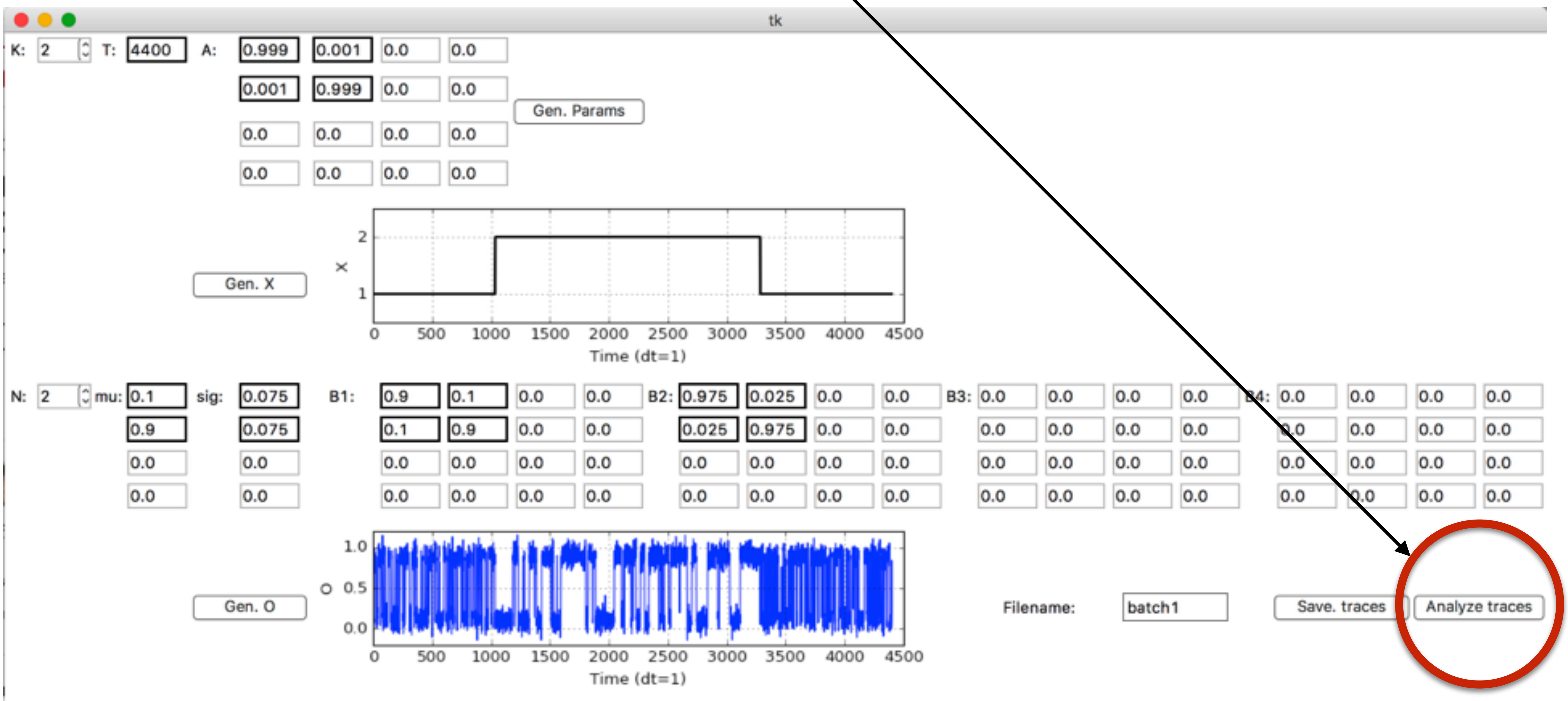
(to generate the sequence of (hidden) internal states. Repeat as much as you like)

Click this



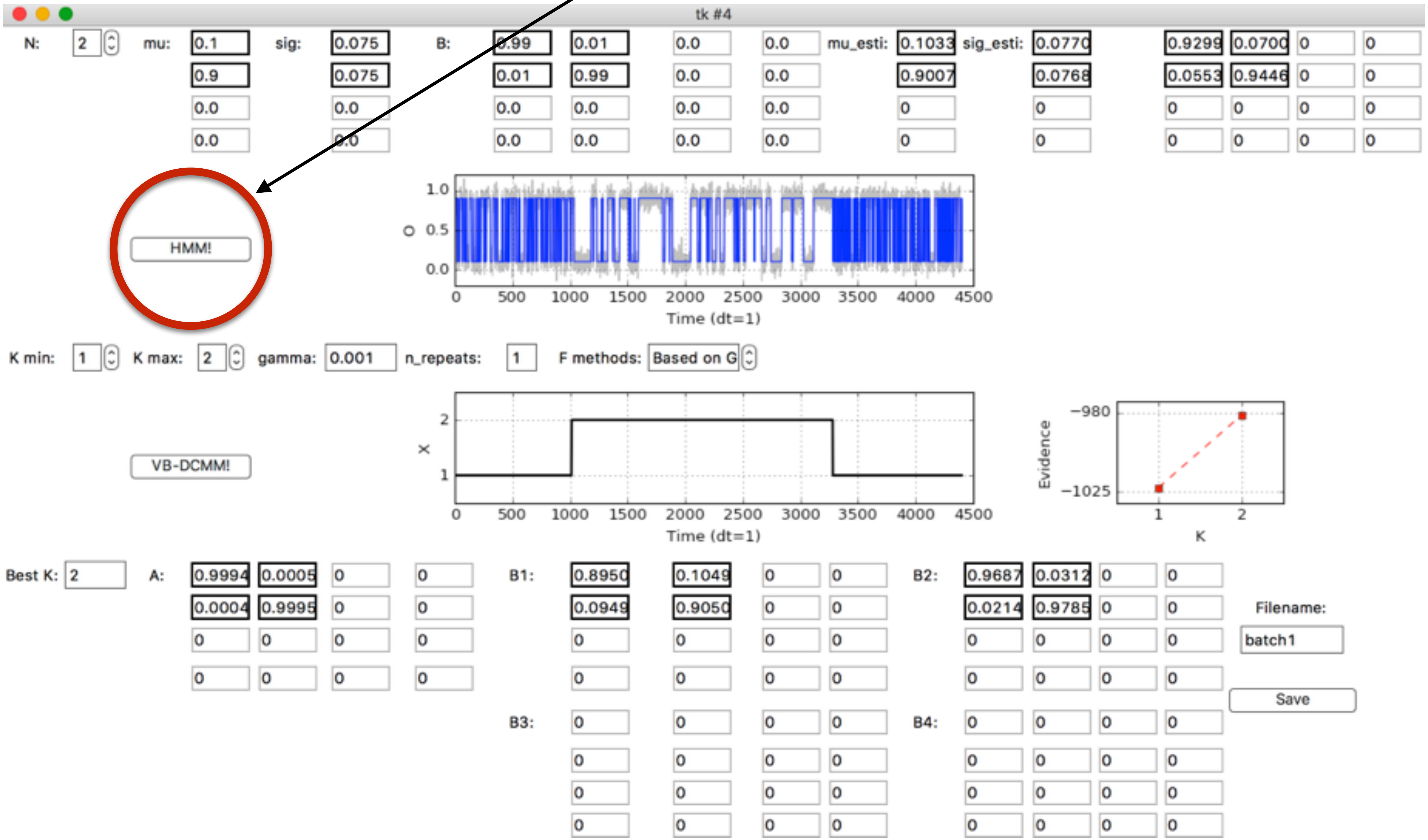
(to generate the sequence of observable states. Repeat as much as you like)

Click this



(to start VBDCMM analysis)

Click this



(to start filter the noise)

Click this

(to find most probable (hidden) sequence of internal state & best model)

2. VBDCMM_gui_simul.py (longer version)

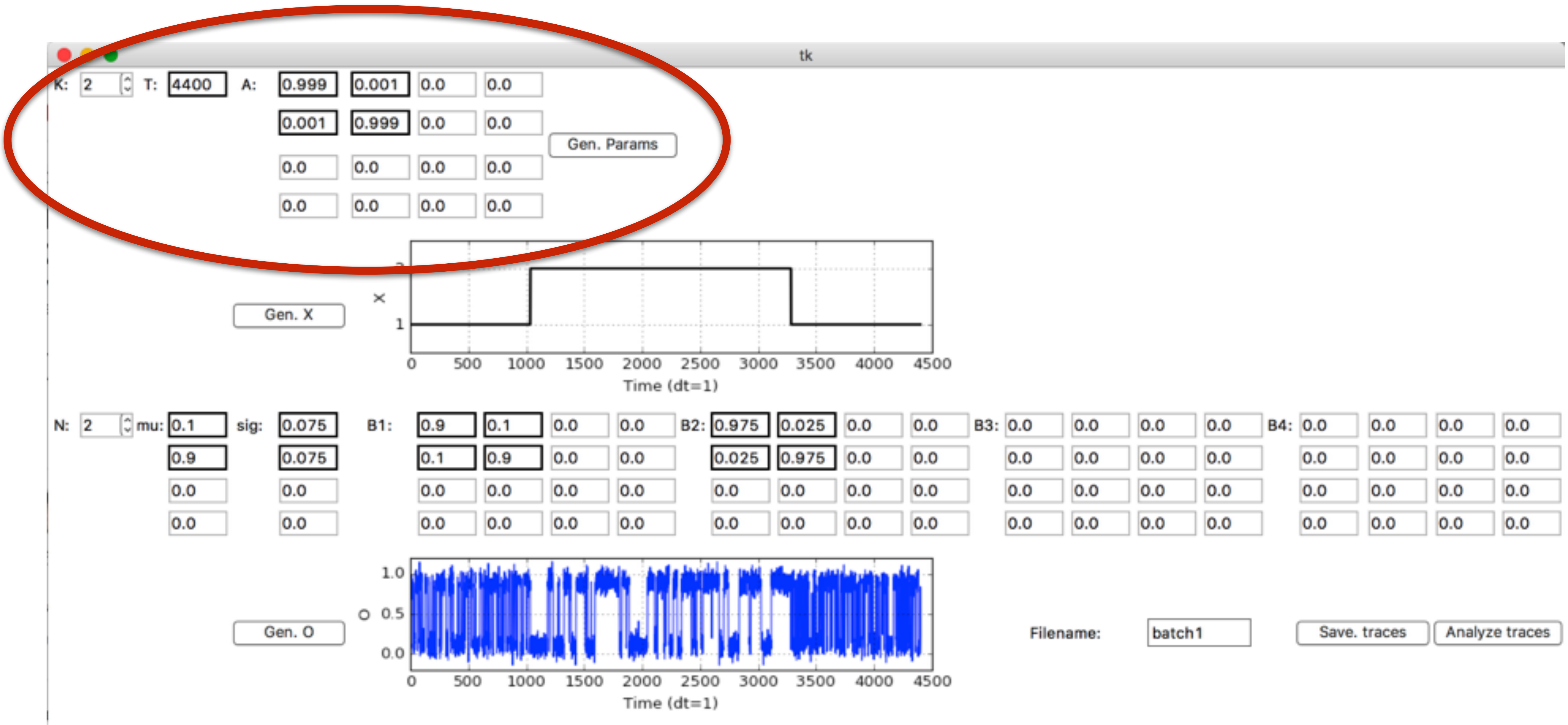
1. Set parameters for the generation of sequence of internal states

K: number of internal states

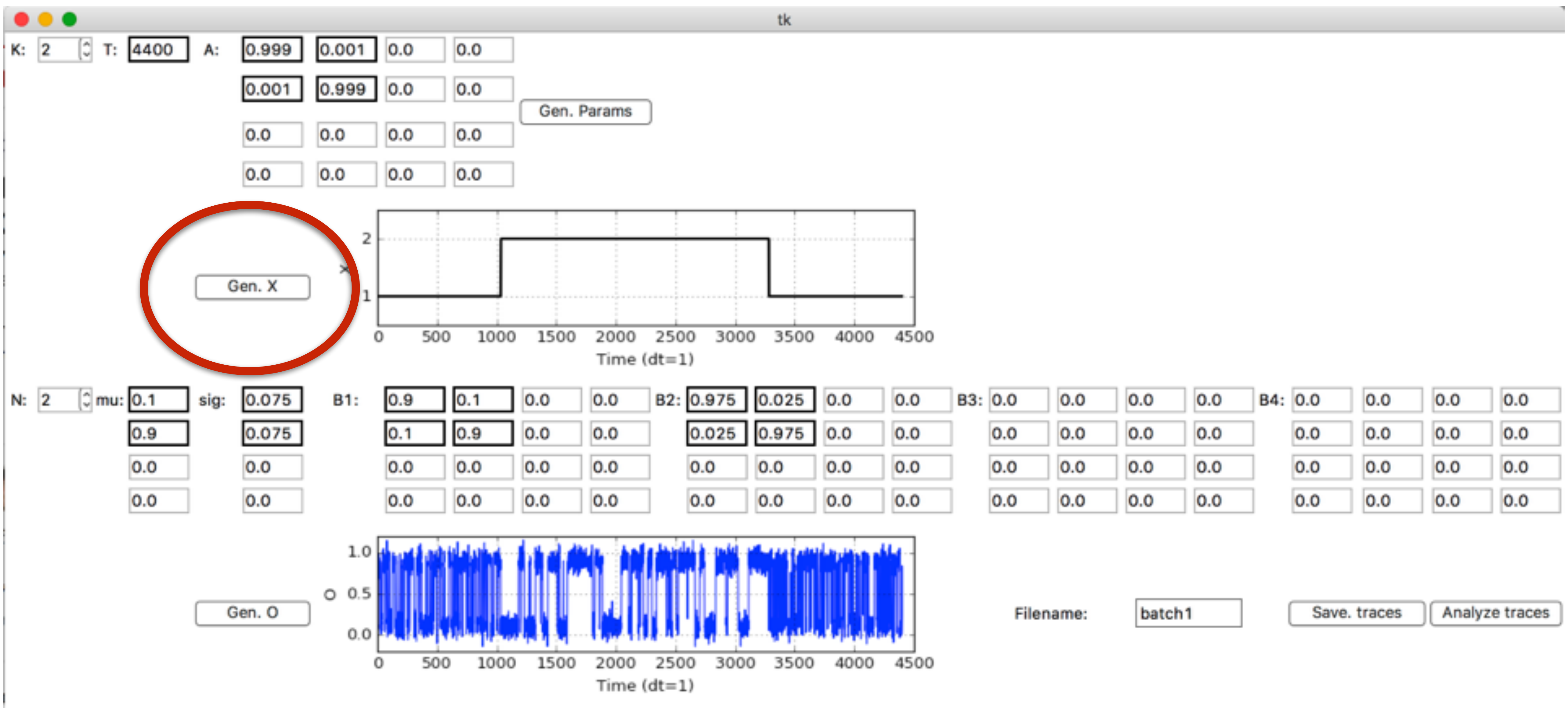
T: Observation time

A: Transition matrix

Gen. Params: Set parameters automatically.



2. Generate a trace of internal state



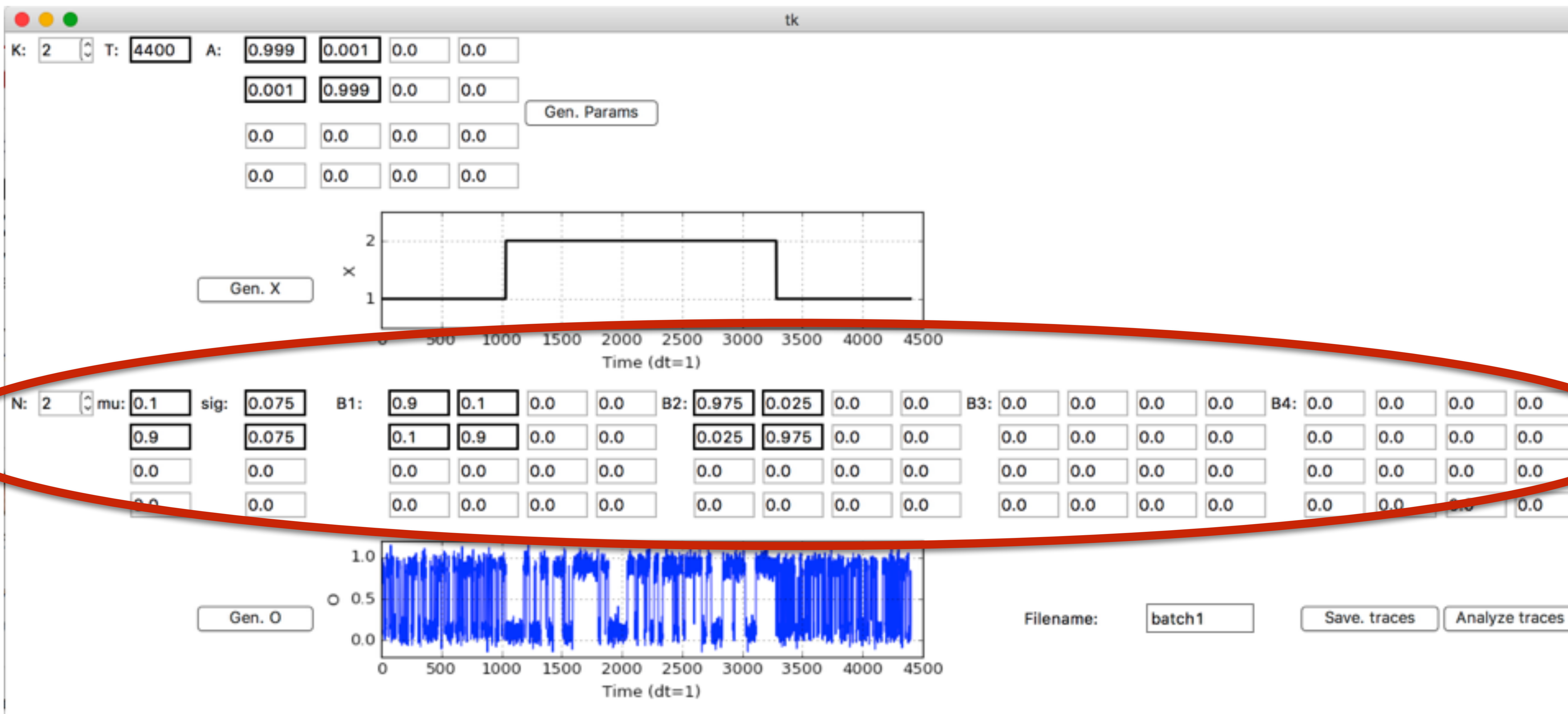
3. Set parameters for generation of sequence of observable states

N: The number of observable states

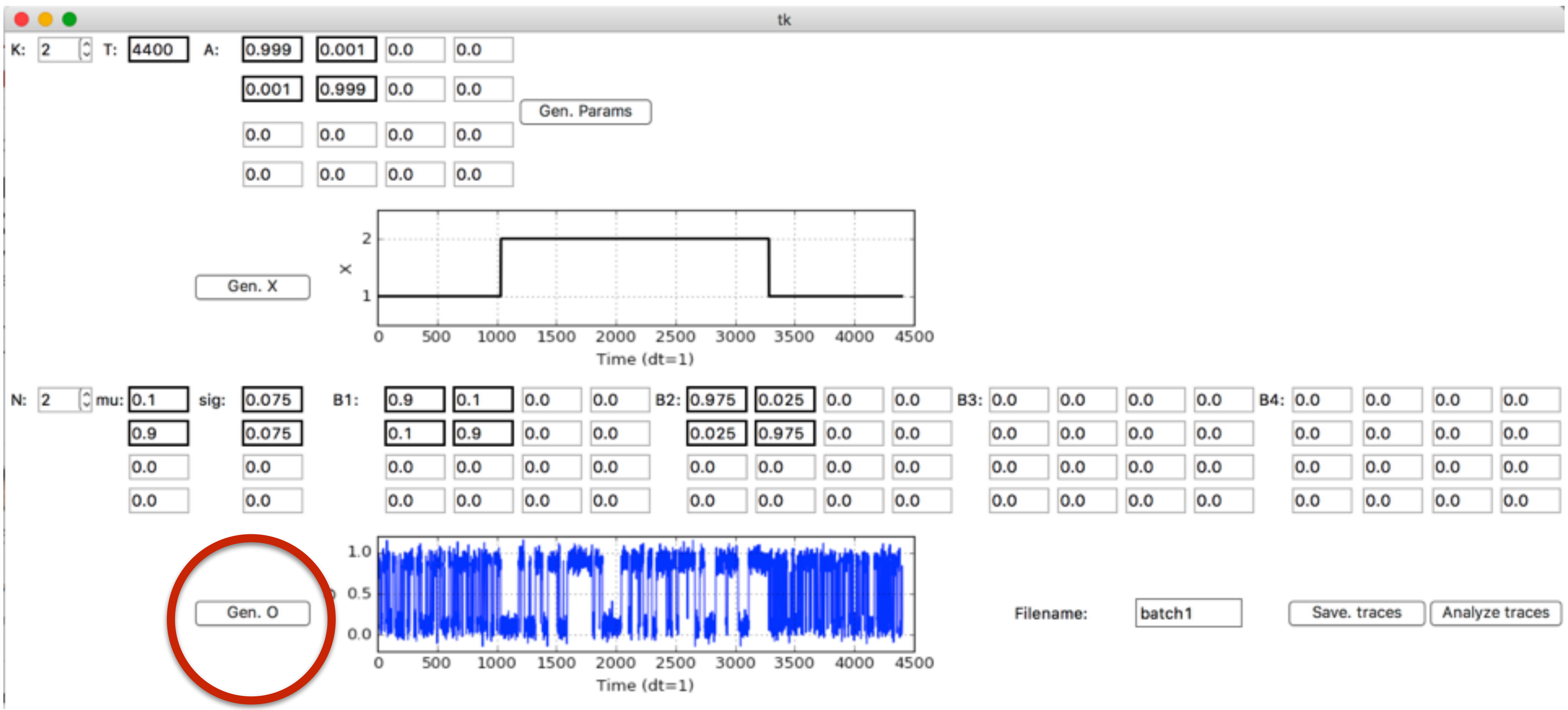
mu: Mean value of each observable state

sig: Std of each observable state

B1—4: Transition matrix of observable state corresponding to internal state 1—4.



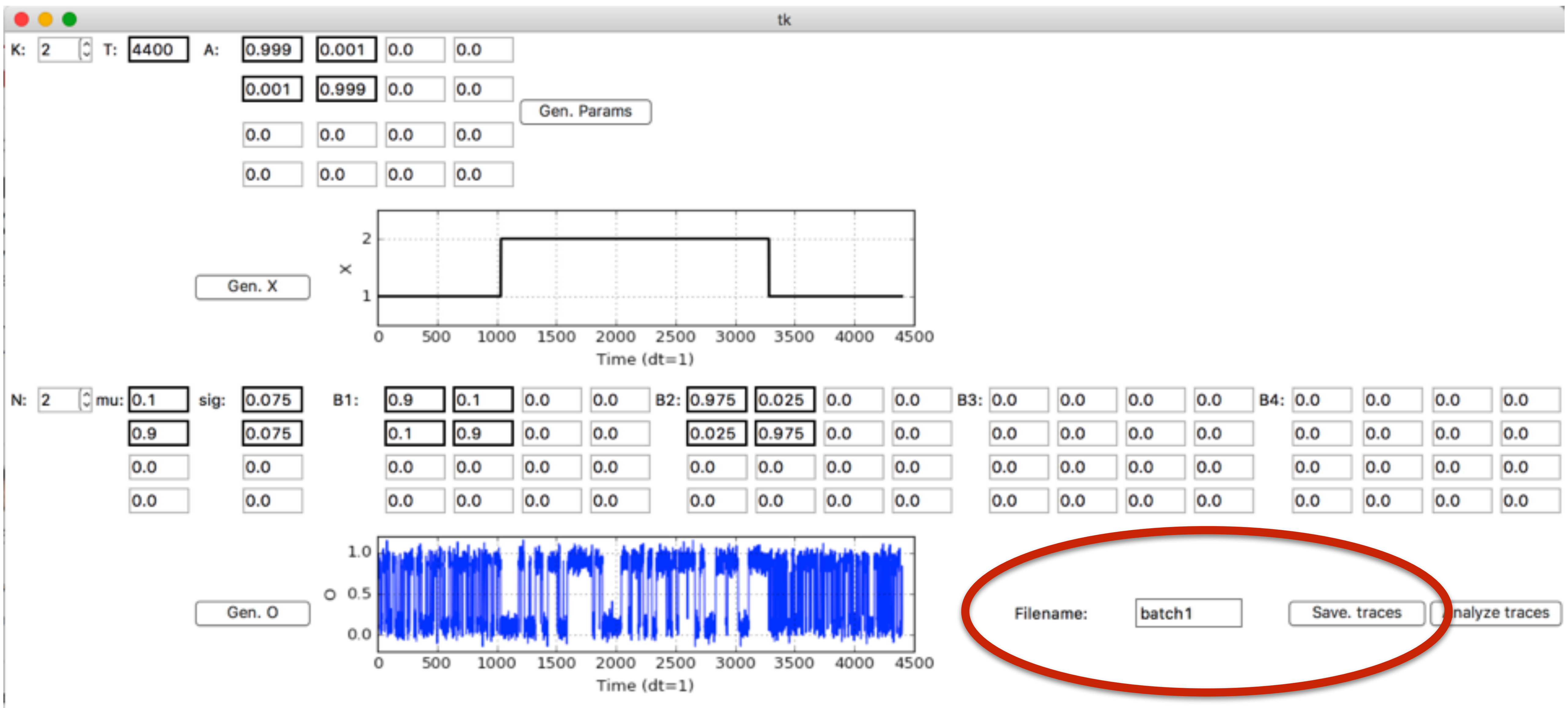
5. Gen. O: Generate the trace of observable state.



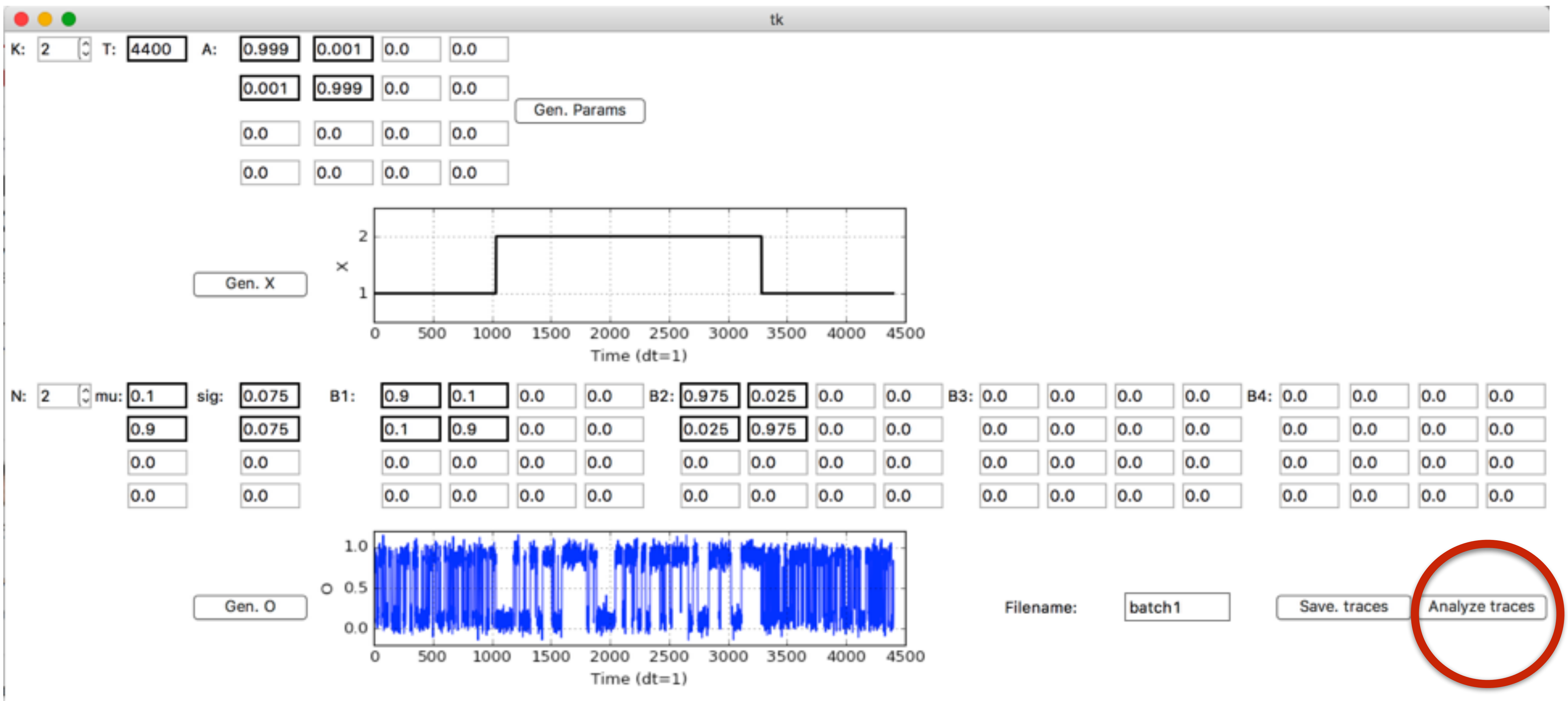
6. Save traces

Filename: The name of save-files.

Save. traces: Save data. Files will be saved in the same folder with “VBDCMM_gui_simul.py” file



7. Start VBDCMM



8. Filter the noise via Hidden Markov Modelling (HMM)

N: The number of (guessed) observable states.

mu: The initial guess of mean value of each observable state.

sig: The initial guess of std value of each observable state.

B: The initial guess of transition matrix.

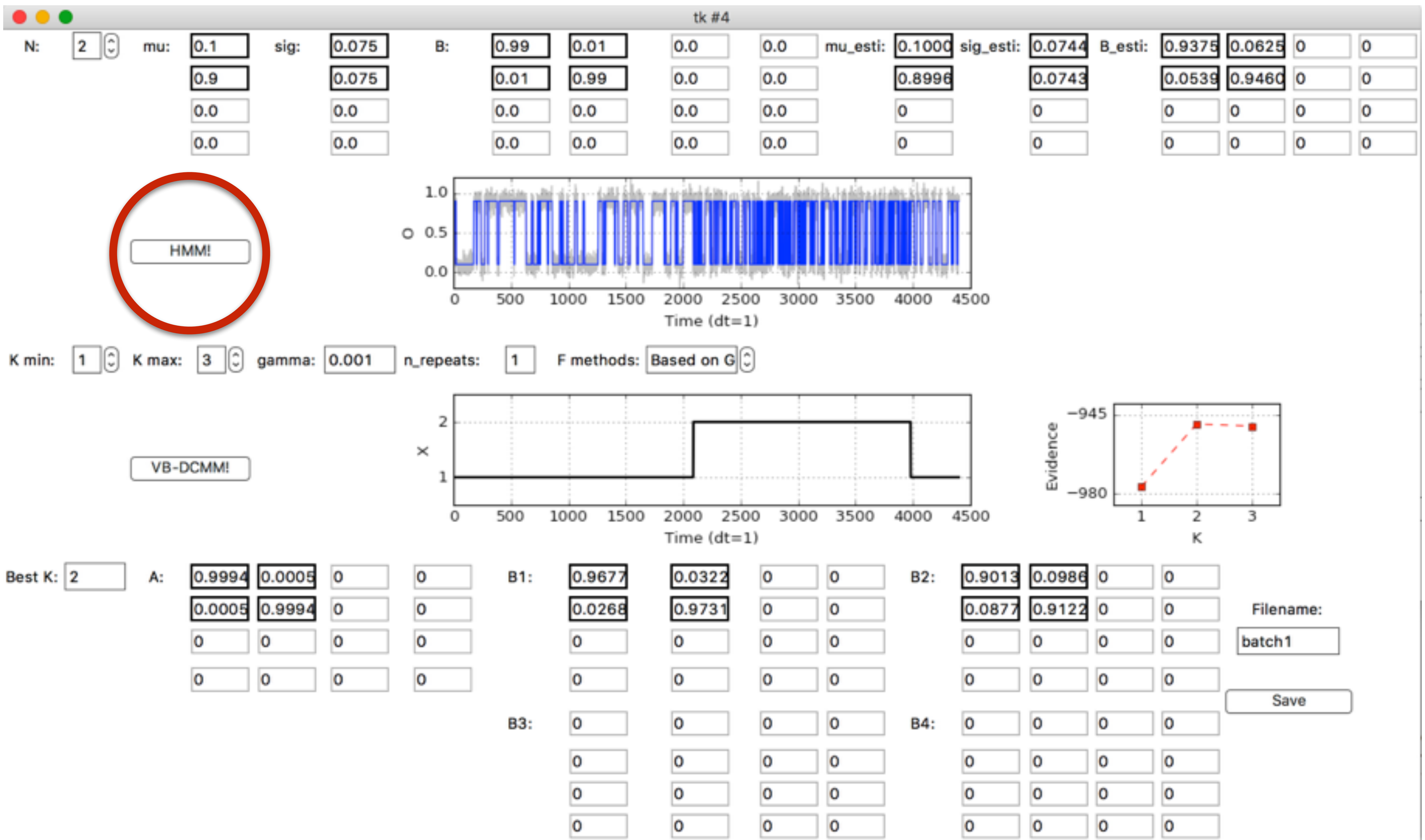
9. Result of Hidden Markov Modelling (HMM)

mu_esti: The estimated mean value of each observable state.

sig_esti: The estimated std value of each observable state.

B_esti: The estimated transition matrix (assuming the presence of one internal state).

11. HMM analysis



12. Parameters for VB-DCMM analysis

K min: Minimum number of internal states

K max: Maximum number of internal states

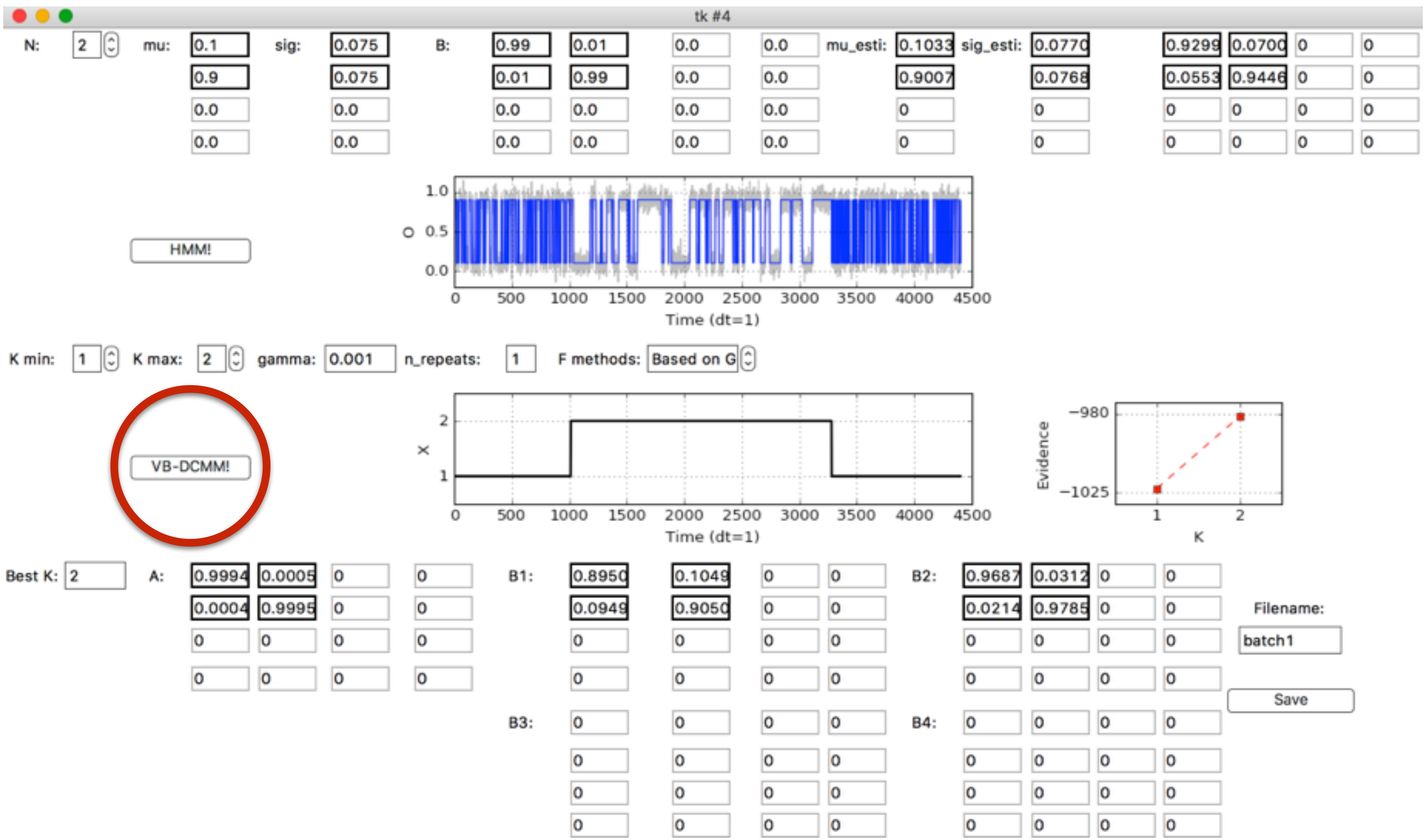
gamma: Initial guess of transition rate of internal states. “0.001” means roughly 1 transition is observed during 1000 sec observation

n_repeats: The number of repeats VB-DCMM in each model. To avoid local minimum n_repeats=20 recommended.

F methods: The method to treat model degeneracy. See the paper for the detail.

13. Find a most probable sequence of internal state

VB-DCMM!: Start VB-DCMM



14. Estimated sequence of internal state

15. Estimated lower bound of evidence.

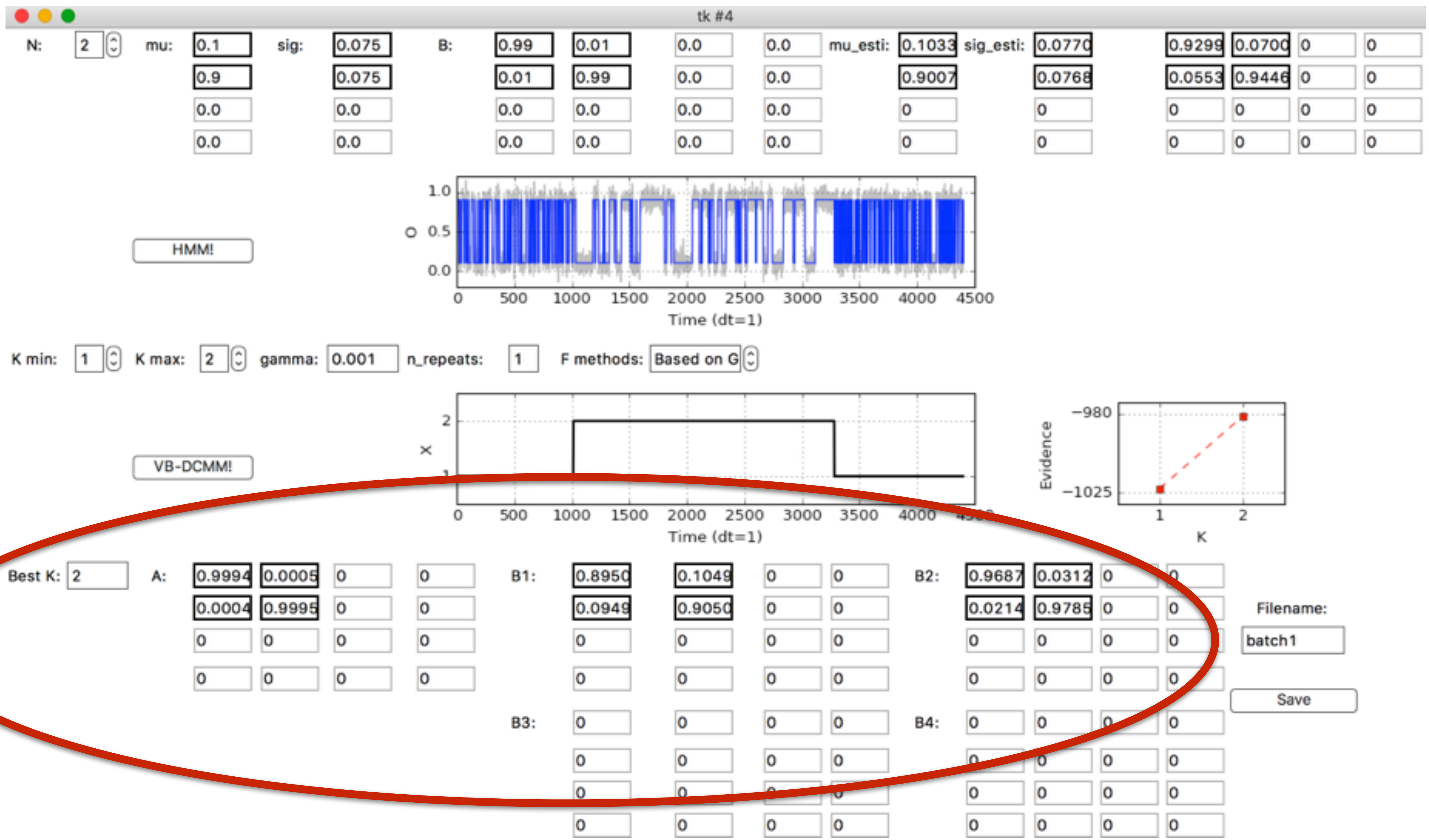
K with maximum value will be selected as a best model.

16. Estimated parameters

Best K: Best model

A: Estimated internal state transition matrix.

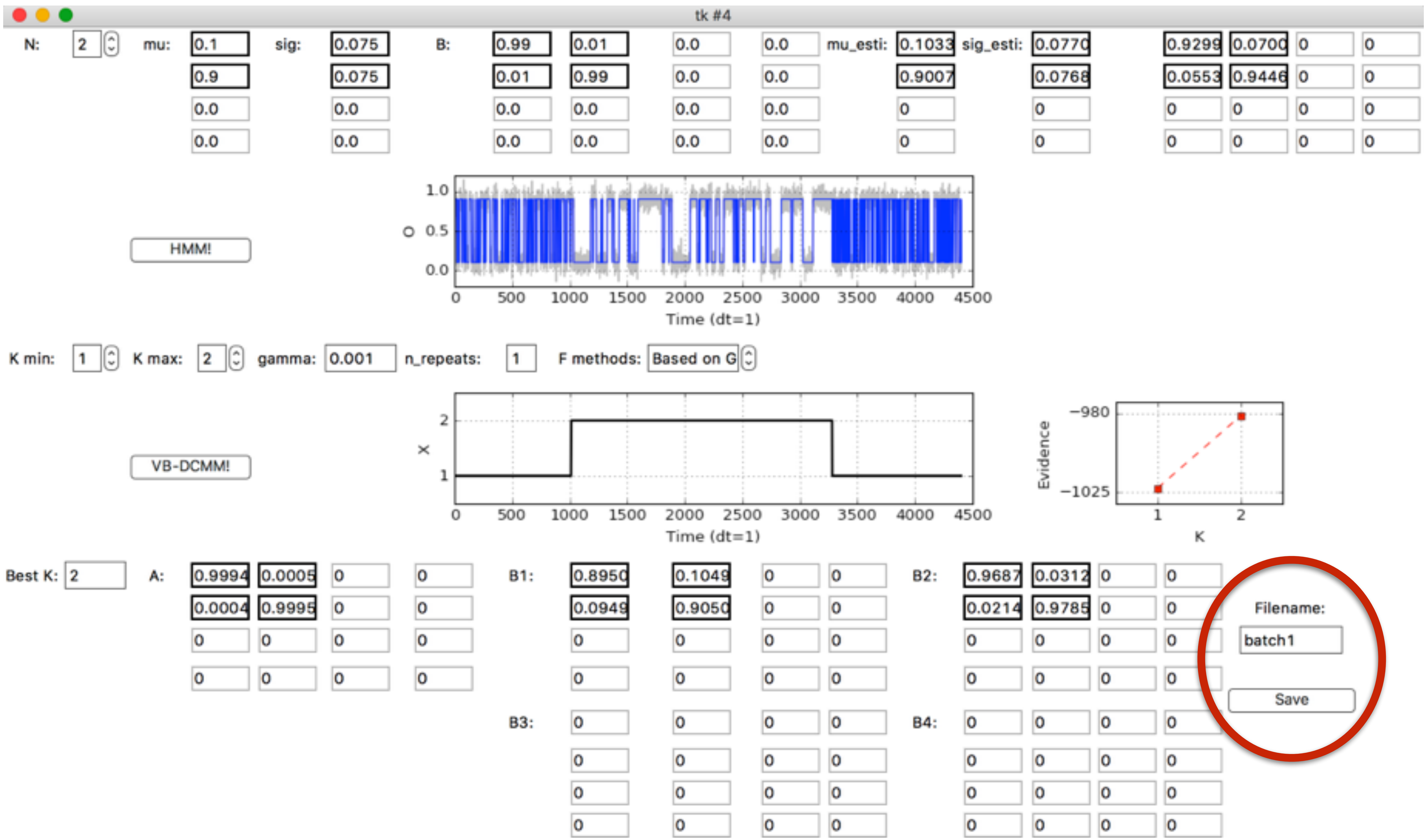
B1—4: Estimated observable transition matrices.



17. Save the results

Filename: The name for save-files.

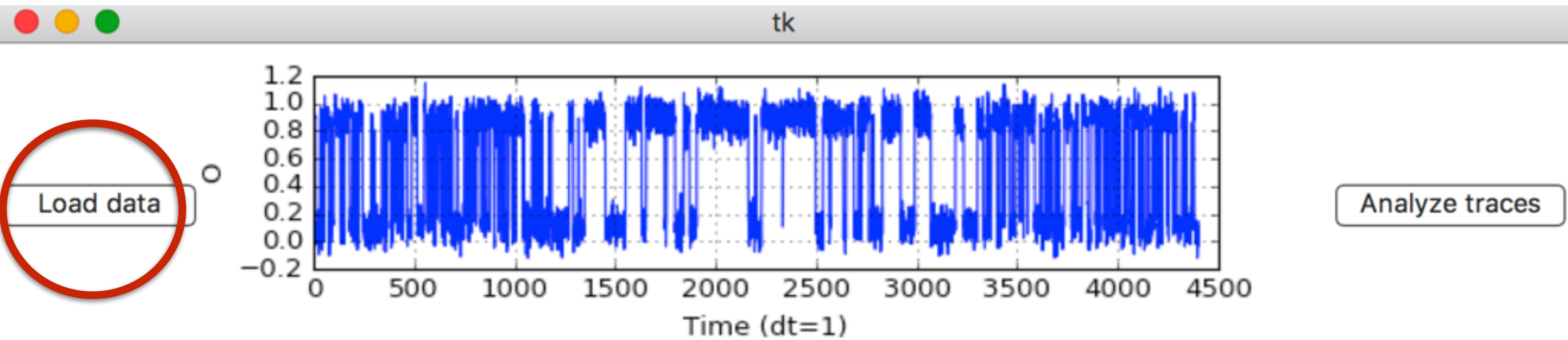
Save: Save data.



3. VBDCMM_gui_load.py (to load experimental data)

1. **Load experimental data**

Load data: Click to load data. Data should be 1-D column vector saved in text file. For example, if data contains 1000 data points, it should be **1000 x 1** matrix.



2. Analyse traces

Analyse traces: Start VB-DCMM