This code analyzes spike data using the state-space Ising model and generates the figures shown in:

Gaudreault, J., Shimazaki, H.: State-space analysis of an Ising model reveals contributions of pairwise interactions to sparseness, fluctuation, and stimulus coding of monkey V1 neurons. ICANN2018, (2018).

To generate the figures in that paper, please follow these steps:

1. Download Smith and Kohn’s data from CRCNS.org:

Smith, M.A., Kohn, A.: Spatial and temporal scales of neuronal correlation in primary visual cortex. Journal of Neuroscience 28(48), 125 91–12603 (2008).

2. Replace the Matlab data files in “preprocess” with those of the same name downloaded from CRCNS.org.

3. Run “extract_data_gratings.py”. This extracts the spike timings and the signal to noise ratio and saves them to pickle files usable with Python.

4. Run “preprocess.py”. This selects neurons with a SNR and a firing rate higher than set thresholds as suggested by Smith and Kohn and randomizes their indices.

5. Run “experiments.py” for all monkeys and populations. Do so for original and shuffled data. These options are at the beginning of “experiments.py”. This fits the state-space Ising model to the data and saves the results in the “Data” folder.

6. Run “data_fig1.py” for all monkeys and populations. Do so for original and shuffled data. These options are at the beginning of “data_fig1.py”. This computes the macroscopic properties (entropy, entropy ratio, sparseness and heat capacity) and saves them in the “Data” folder.

7. Run “quantiles.py” for all monkeys and populations. Do so for original and shuffled data. These options are at the beginning of “quantiles.py”. This computes credible intervals for the macroscopic properties and saves them in the folder “Data_quantiles”.

8. Generate the figures using “fig1.py”, ‘fig2.py”, “fig3.py” and “fig1_without_quantiles.py”. The figures will be saved in the “Figures” folder.

We acknowledge Thomas Sharp and Christian Donner respectively for providing the code for exact inference (from repository <https://github.com/tomxsharp/ssll> or <http://github.com/shimazaki/dynamic_corr> for Matlab code) and approximation methods (from repository <https://github.com/christiando/ssll_lib>).

*The code has been written with Python 3.6.5. The ssll library has been updated to work with Python 3.6.5.