Background and dependencies:

This code contains a set of functions to carry out SPUD decoding on the data from the study of Peyrache et al. (2015). Scripts are written in Python 2.7 (but should work in Python 3 after modifying a few print statements). The primary packages used are typically part of a standard install (numpy, scipy, matplotlib, scikit-learn) and can be installed, for example, using the Anaconda Distribution. The scripts to compute persistent homology on the data call Ripser, which can be installed from https://pypi.org/project/ripser/.

Instructions for use:

1) Download the Peyrache et al. (2015) data set from CRCNS (https://crcns.org/data-sets/thalamus/th-1). The session we're using for illustration is Mouse28-140313, so to get started you could just download the zipped file corresponding to that session. Make sure that the unzipped directory contains the list of measured angles in addition to the spike timing and state files. These angle files are of the form Mouse28-140313.ang. For some of the sessions on CRCNS the angle files are already in the directory with the spike information; for others the .ang files are present in a separate zipped file that's also available for download and the relevant angle files can be copied to the appropriate directory.

2) In the code folder, go to general\_params/ and edit make\_general\_params\_file.py to set the paths to point to the location of the raw data and the locations where you want various processed data files to reside (the scripts will make the directories, as necessary). Run this file, which will save the paths to a common parameter file that will be used by all the scripts.

3) Run preprocess\_raw\_data.py to (a) gather the information from the various data files into a single data structure and (b) estimate spike counts by convolving the spikes with a kernel. Steps (a) and (b) can be run separately by setting the appropriate variables to True or False in the script.

4) At this point you can start using the functions in manifold\_and\_decoding/. The simplest way to get started is to execute run\_spud\_multiple\_tests.py, which will fit the manifold and do decoding on the data a few times.

Other options/functions are:

run\_persistent\_homology.py Uses Ripser to compute the Betti numbers for the data.

run\_spud\_interactive.py Run the spline fit interactively, which will give you a bit more control and is very helpful to make sure the knots are placed correctly for noisy data.

run\_dimensionality\_reduction.py Runs the dimensionality reduction separately, both for visualization and as a reusable preprocessing step.

decode\_from\_saved\_fit Decode from a previously saved fit and plot some decoding traces.