

SNAPT ALGORITHM

Copyright (c) 2013, Adam Cohen, Daniel Hochbaum, Dougal Maclaurin
All rights reserved.

Noncommercial redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Sub-Nyquist Action Potential Timing (SNAPT) algorithms enable pixel-by-pixel determination of the timing of an action potential. The input is a movie of a neuron repetitively triggered to fire, where the neuron contains a high-speed fluorescent voltage reporter. The output is a movie at a higher frame-rate showing the propagation of the action potential through the neuron. For a detailed description of the algorithms, see: Hochbaum *et al.* "All-optical electrophysiology in mammalian neurons using engineered microbial rhodopsins", Nature Methods (2014).

The implementation given here is written for Matlab R2012a, and has been confirmed to run on a personal computer with 14 GB of RAM, running Windows 7 (64-bit). Several of the sub-routines are memory intensive. There are many adjustable parameters in the code, so it is important to run the code one piece at a time, looking at the output at each step and checking that it makes sense.

The top-level script is:

`SNAPT_PCA.m`

Run this script one section at a time (highlight a section, then press F9 on a PC).

This code uses the following helper functions:

```
apply_clicky.m  
clicky.m  
extractV.m  
mask_region.m  
pcafilt.m  
readTifFast.m  
rem_pbleach.m  
spatialfilt.m  
spikefind.m  
splinezeros.m
```

In addition, the code uses two sets of functions distributed on the MatlabCentral database:

```
grs2rgb.m (Note: the function included with this distribution has been modified  
from the version distributed at MatlabCentral)  
freezeColors.m
```

You may need to modify the code to accommodate peculiarities of your data.