SLAP Demo 2: Activity Imaging

Kaspar Podgorski – podgorskik@hhmi.org

System Requirements

MATLAB and associated toolboxes:

Signal Processing Toolbox, Image Processing Toolbox, Parallel Computing Toolbox

Ilastik version 1.2 - https://www.ilastik.org/download.html

Tested with **MATLAB R2017a and R2018b**, **Ilastik 1.2.2**, and **Windows 7 and 10.**

Getting Started

Estimated install time: 10 minutes + download time

1. Download the SLAP codebase from: https://github.com/KasparP/SLAP
2. Add the SLAP directory and any subfolders to your Matlab path
3. Download the following directory of example data:
   https://www.dropbox.com/sh/s25qqk5kbc3hhhf/AABhFFu-fxfN4SpQuit_DE9Va?dl=0
4. Edit the function `dataDirectory.m` in the SLAP repository to point to the top-level folder of the example data (…/`SLAPdemo2/`)

Introduction

Scanned Line Angular Projection Microscopy (SLAP) is a high-speed two-photon imaging technique that scans a line of illumination (in XY plane) across the sample, rather than raster scanning a single point. The resulting measurements are linear projections of the 2D sample plane along different angles. SLAP imaging is much faster than raster scanning because it collects only four projection angles of the sample. It collects each frame with just four line scans, achieving >1 kHz frame rates for megapixel fields of view.

SLAP activity imaging combines information from a segmented reference image, a dynamics model, an imaging model, and high-speed SLAP measurements, to recover the activity of the segments over time. This demo introduces you to the functions that convert a reference image into a segmentation, align SLAP data to the reference image, and solve for segment activities using the priors, data, and imaging model.
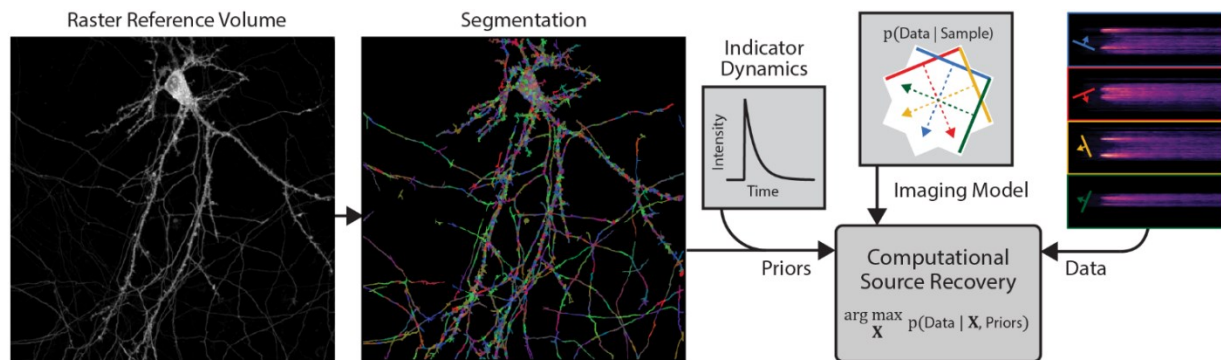

Figure: Schematic of SLAP activity reconstruction. A segmented raster volume and model of indicator dynamics act as priors to recover sources' activity from projections

The Workshop

Estimated time: 1 hr

I. Inspect the data

Data included in this demo are:

… _REF.mat :   Raster reference volumes, with metadata

… .tif: Raster reference volumes in tiff format

yGluSnFR_invivo_autocontext.ilp: an Ilastik pixel classifier

… _REDUCED.mat: SLAP recordings in scandata format

The format of the scandata files and the embedded information used to generate the P matrix is similar to that introduced in SLAP Demo 1. The .mat files contain data structures that can be explored using Matlab's built-in viewer. In particular, you can access the raw SLAP measurements as

```
y = [scandata.frames.pmtData];
```

II. Generate 'Problem Data'

Reconstructing SLAP data consists of just two top-level functions, but a large amount of underlying code.

The first function is `SLAPMi_PSXdata()`, which is used to generate "problem data" files that contain all the information needed by the solver.

Call this function without arguments. GUIs will pop up allowing you to select user-settable parameters. The first time you call these functions, accept the default values.

Start by calling `SLAPMi_PSXdata()`

This function will ask you to select the above files as inputs, and will then run Ilastik in headless mode to classify and segment your reference data. It produces several plots of the segmentation output when it is complete. It will save a new copy of the (now segmented) reference image, and ask you if you would like to manually edit the segmentation; this is useful, but not necessary. If this is your first time running the code, it's simpler not to edit your segmentation.

If you choose to edit the segmentation: a GUI will pop up. See the source code `segmentationEditor.m` for keyboard controls, and make sure you save your changes (Ctrl-S), then rerun `SLAPMi_PSXdata()` when you are done, selecting the saved segmentation as your reference image.

The first time `SLAPMi_PSXdata` is run with any given set of calibration settings, it needs to generate the **P** measurement matrix from calibration data saved in your scandata file. This will take a couple of minutes, but should only happen once. The **P** matrix will be saved with a hashed filename for future lookup.

Next, `SLAPMi_PSXdata` will align your data to the reference image, and produce an alignment plot, comparing measured to expected data.

The function saves a …_PROBLEMDATA.mat file upon completion.

III. Solve the Problem Data

The second important function is `SLAPMi_solve`. Enter the command:

```
sys_recon = SLAPMi_solve(defaultOpts_inVivoGluSnFR);
```

to solve selected saved problem data files. A dialog will pop up to select problem data files from your hard drive. If you call this function without arguments, you can edit the solver parameters via a GUI.

IV. Render a Movie

Enter the command:

```
SLAPMi_Plot(sys_recon);
```

to render a particular solution as a movie. The function will also plot the loss function and residuals of the solver. The `skipframes` option determines the interval between rendered frames in the movie; rendering every frame would be quite slow, so if you plan to render many movies with different parameters, set this to a larger number.

That's it!

If you found this interesting, have a comment, found a bug, or want to contribute, please contact me:

Kaspar Podgorski

podgorskik@janelia.hhmi.org

Useful SLAP links:

- [bioRxiv preprint on SLAP imaging](#)
- github repo for SLAP imaging code ([github.com/KasparP/SLAP](#))
- [HHMI Janelia Open Science](#) site for SLAP microscope designs