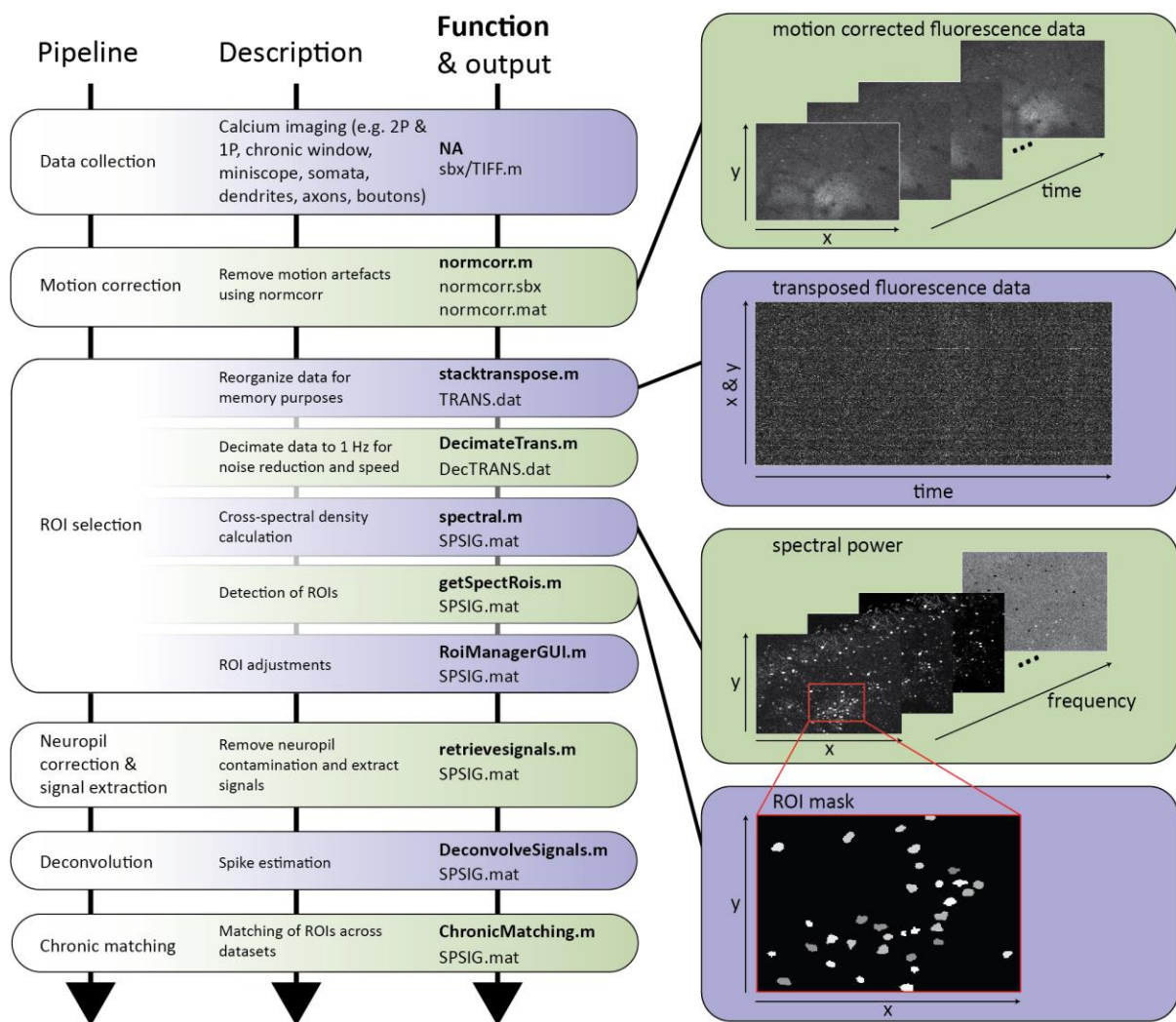


Contents

Introduction	2
Fluorescence data	3
Raw data of non-sbx format	3
Raw sbx data	3
Showsbx.m (to apply motion correction and other things on sbx data)	3
ROI selection	4
Transpose data.....	4
Decimate data.....	4
Spectral calculation.....	4
Representative fluorescence images	4
Get ROIs	5
Edit ROIs	5
Signal extraction.....	6
Fluorescence signal of ROIs	6
Spike estimation with MLspike	6
Chronic	6
Jumping into chronic without creating ROIs with spectral pipeline.	6
Chronic matching	6
References	6



Introduction

This manual covers some practical information on using the Spectral Segmentation toolbox pre-processing pipeline. For more information, see the paper ([reference](#)).

There are two easy ways to run the pipeline: Via the **SpectralPipeline** or **AutomatedAnalysis** scripts.

The script **SpectralPipeline.m** in the main folder covers all the main functions of the pre-processing pipeline. All the functions can be called without input, each function will ask for which file to analyse. This script is useful for manually processing the data.

The script **AutomatedAnalysis.m** in the main folder is the script that we use to automatically execute all the steps of the analysis for as many sbx files as you request, until the RoiManagerGUI step, which requires manual input. This script also applies motion correction.

The pipeline uses a naming convention which it uses to search for files. If you deviate from this naming convention the toolbox will be harder to use.

Transposed files should end their filename with *_Trans.dat*

Decimated transposed files should end their filename with *_DecTrans.dat*

Spectral/ROI files should end their filename with *_SPSIG.mat*

Fluorescence data

Raw data of non-sbx format

We do not have code to do motion correction on file formats other than sbx files. But normcorre, which is necessary for the sbx motion correction, can handle a lot of file formats.

Tiff files can be converted to .sbx with the script tif2sbx.m

Raw sbx data

Raw data needs to be motion corrected. We use the toolbox normcorre (Pnevmatikakis and Giovannucci 2017) for motion correction. However, normcorre cannot recognize sbx files. So we have adapted some of their code so normcorre can be used on sbx files. To be able to apply this motion correction on sbx files, download normcorre from github:

<https://github.com/flatironinstitute/NoRMCorre>. Place the normcorre code in your Matlab path **below** SpectralSegmentation folders. The motion correction can be applied by calling the GUI Showsbx.m.

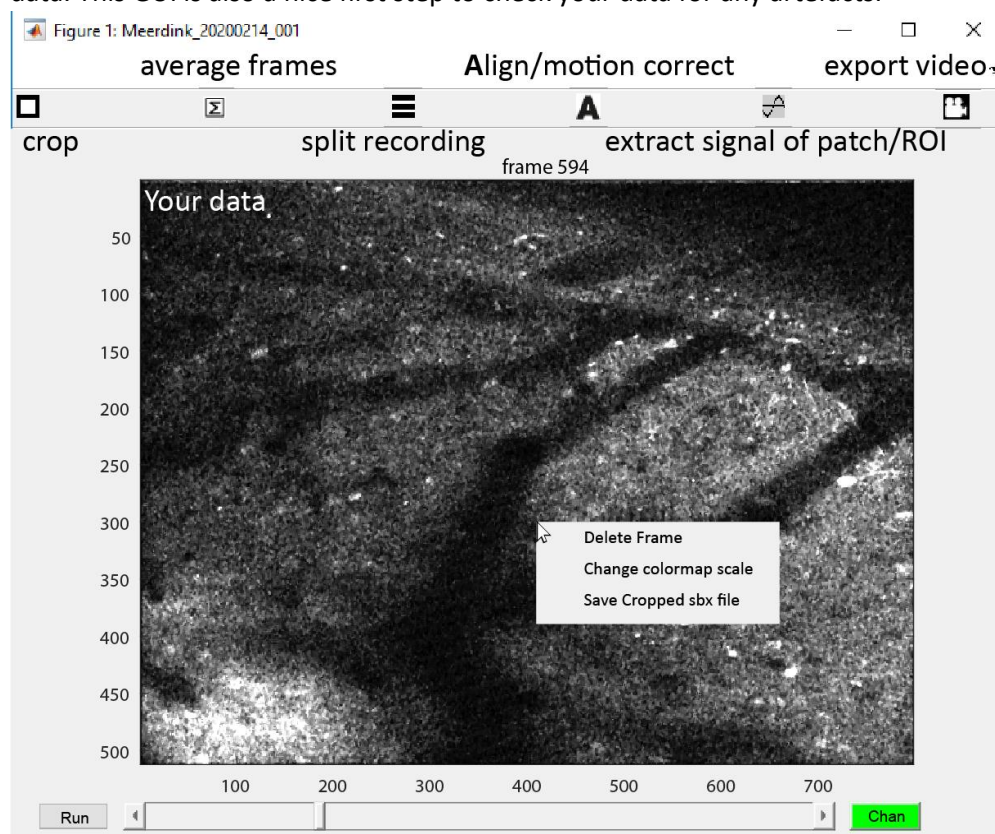
Showsbx.m (to apply motion correction and other things on sbx data)

The Showsbx.m GUI is able to play sbx files and call many functions, including motion correction.

Just start the GUI by executing *Showsbx* in the Matlab command in window. Then select which file to open in the pop up menu that appears.

The contrast of the data can be changed by right clicking in the main image (Change colormap scale).

The motion correction can be called by pressing the **A** button. If the edges of the recording contain artefacts they can be removed by cropping (square bordered button). When crop is activated, press once in the upper left corner of your data, then press in the lower right corner of your data. Then press enter. A rectangle will appear which denotes what data will be included when aligning your data. This GUI is also a nice first step to check your data for any artefacts.



The Showsbx GUI. Showing a two-photon recording of 1 depth, imaged in a mouse injected with GCaMP6f in V1 layer 2-3.

ROI selection

Transpose data

The transposed data is necessary for the retrieval of fluorescence traces.

The motion corrected sbx data is transposed with StackTranspose.m.

When calling the function without input it will ask which .sbx file to transpose, and then where to save the output.

Decimate data

To remove noise and decrease the amount of datapoints to deal with we often decimate/subsample the data to ~1Hz. Decimating data takes quite some time. So to save time decimated data is saved into a separate file with DecimateTrans.m. DecimateTrans.m creates a file with the filename of the original recording, with _DecTrans appended.

The decimated data is necessary for the spectral calculation (spectral.m). It is optional for other functions but can be used by many, speeding them up. These functions first search for the decTrans.dat file before the regular transposed file. Functions that can optionally use the decimated trans files are: RoiManagerGUI, getSpectRois, CalcSpatialCorrFun.

Spectral calculation

Run spectral.m to calculate the cross-spectral power. This function creates the SPSIG.mat file.

The spectral calculation requires the decimated transposed data.

The spectral components are stored in the variable SPic, the frequencies that correspond to the SPic 3rd dimension are stored in Sax.

Representative fluorescence images

It is also nice to have good quality fluorescence images of your data. These can be made with BackgroundImgSbx. It uses the regular motion corrected fluorescence data, not the transposed data. The projections are made by taking short average projections of multiple frames (chunks) to reduce noise, and using these chunks for a maximum and an average projection. So there are two variables that can be changed in this procedure.

- *nframes*: How many frames are averaged together to use as a picture for the average projection: how many frames in each chunk.

- *nchunks*: How many chunks are used for the average projection.

When nframes is higher, the picture will be less noisy, but very short peaks in fluorescence get averaged out more. So the max projection end result will look slightly more like an average projection.

When nchunks is higher the quality goes up, but the procedure takes longer. Also, there is no limit so you can take overlapping chunks. This means the same frames can be used more than once in the projection.

Get ROIs

getSpectRois.m is used to automatically create ROIs via the cross-spectral power images and parameters specified by the user. When calling getSpectRois you need to select the SPSIG file, and then set parameters for the ROI selection with the SpectParArm UI.

The top image shows all frequency components, with different colors for different frequencies. It also gives feedback on the selected parameters.

Parameters to set:

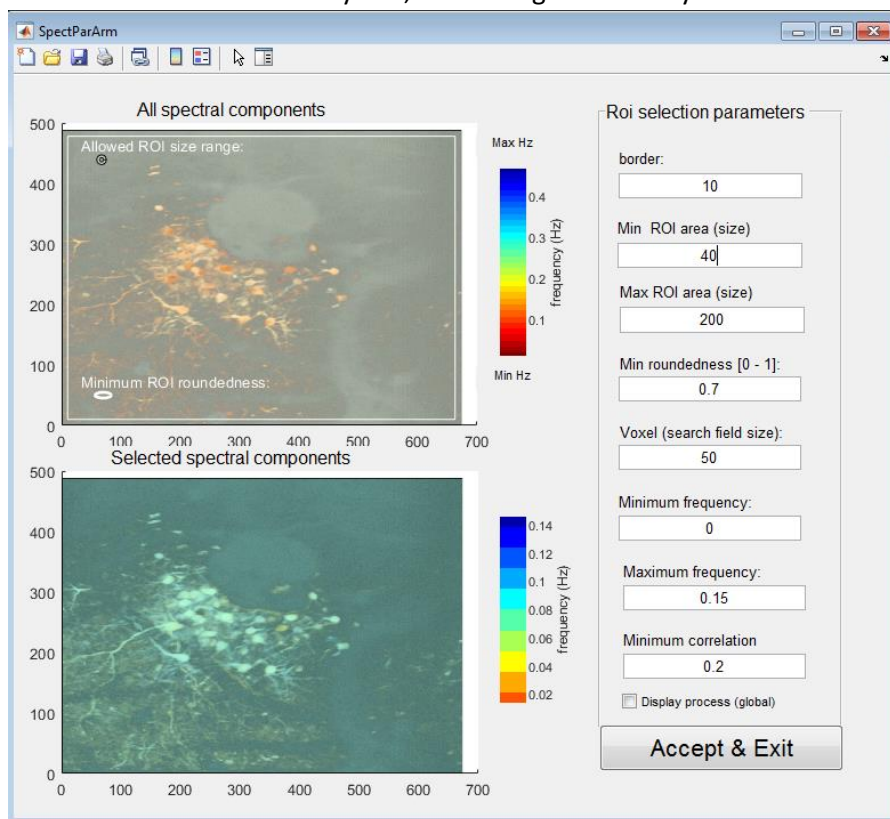
Border: How close to the edge of the images to find ROI centres. The selected border is shown with the white rectangle just inside the top image.

Min & Max ROI area (size): The size of ROIs is calculated in how many pixels of the mask/ image they occupy. Feedback is given by a donut (see top left of top image).

Min roundness: the roundness is calculated as the ratio of the area over the square of the perimeter (length of contour) of the ROI. An estimation of how not round an ROI is allowed to be is shown in the lower left corner of the top image.

Minimum & maximum frequency select which frequency components to use for the ROI search. This has a large effect on how many ROIs are going to be found. The lower image shows the spectral components that are selected, so you can estimate whether artefacts or neurons are included.

Minimum correlation: The correlation refers to the covariance of the correlation of the fluorescence signal of the pixel with the highest spectral power value in the ROI with the signal of all other pixels in the ROI. If this value is very low, the ROI signal is mainly uncorrelated noise (see paper).



SpectParArm UI. Select parameters for the ROI selection.

Edit ROIs

ROIs can be edited manually via the RoiManagerGUI. The RoiManagerGUI has a separate manual.

See the github page <https://github.com/Leveltlab/SpectralSegmentation> or the PDF file in the SpectralSegmentation\docs folder.

Signal extraction

Fluorescence signal of ROIs

RetrieveSignals.m is the script that retrieves the fluorescence signal from ROIs. It uses the regular transposed data, not the decimated data. For the neuropil correction some variables can be changed. The neuropil correction

Spike estimation with MLspike

DeconvolveSignals uses MLspike (Deneux et al., 2016) for spike estimation of the retrieved fluorescence traces. To be able to use it download the code from brick and spike at <https://github.com/MLspike>, and add it to the Matlab path.

Chronic

Jumping into chronic without creating ROIs with spectral pipeline.

The chronic matching provided in the toolbox requires some specific variables that are created in the Spectral processing pipeline. However, you can use ROIs that you have found with other toolboxes. To be able to execute the Chronic matching as provided, you need to have pre-processed the data until the spectral calculation step.

The essential variables for chronic matching are:

Mask (2D double [height x width]): An image that tells where ROIs are. Each pixel corresponds to an ROI by the number it has. 0 means no ROI present. Usually ROIs do not consist of multiple areas; they touch themselves everywhere. This variable is completely essential, and you need to provide this one yourself.

PP (struct): This struct has a lot of info about the ROIs in Mask, like size (PP.A), signal covariance (PP.Rvar), contour coordinates (PP.Con) etc. If you have a mask, the contour and center coordinates of PP can be made via the script PPfromMask.m. Then the rest of the ROI properties can be retrieved by PPModerize.m. This function requires transposed data.

SPic (3D double [height x width x frequencies]): This variable is used to create the image that is used for registration of recordings. But you could replace it by a maximum fluorescence projection.

Chronic matching

For chronic matching, execute the ChronicMatching script step by step.

References

Pnevmatikakis, E. A., & Giovannucci, A. (2017). NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *Journal of neuroscience methods*, 291, 83-94.

[10.1016/j.jneumeth.2017.07.031](https://doi.org/10.1016/j.jneumeth.2017.07.031)

Deneux, T., Kaszas, A., Szalay, G., Katona, G., Lakner, T., Grinvald, A., Rózsa, B. & Vanzetta, I. (2016) Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature Communications* DOI: [10.1038/ncomms12190](https://doi.org/10.1038/ncomms12190)