% Copyright (c) 2022

%

% Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

%

% The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

%

% THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**%% Before you start**

% Download the ephys traces and save them in the same folder as the tiff files

% Save ephys traces as ephys\_traces!!

% Add path to where the script is located on your device

Specify folder where this script is stored

% Add path to where the functions are located on your defice

Specify folder where functions used by this script are stored

% Provide the following parameters

stimulation\_start = time (in milliseconds) when stimulation of neuron starts

stimulation\_stop = time (in milliseconds) when stimulation of neuron ends

stimulation\_length = stimulation\_stop – stimulation\_start

ephys\_min = minimum value (in voltage) in electrophysiological traces according to MES

ephys\_max = maximum value (in voltage) in electrophysiological traces according to MES

AP\_threshold = threshold value (in voltage) for action potential as determined in MES file

SD\_threshold = calcium response needs to be SD\_threshold times higher than baseline SD to classify it as a

responding mitochondrion

size\_subplots = number of rows preferred for subplots

**%% Check if titles are right**

The first step of the script is to provide the names of the trials. Use the following format for the different subcellular regions:

* soma: Soma 100 pA
* dendrite: Dendrite 100 pA
* proximal axon: Axon1BO1 100 pA (hence the number between Axon and BO and the number after BO (=Branch Order))
* distal axon: Axon1enpassent3 100 pA or Axon1BOunknown 100 pA

D = current folder

S = list of .tif files from the green channel of the microscope

% Ask the user whether the titles of the recordings are right

title\_list = preallocate title\_list with 1 row and length(S) columns

prompt = preallocate prompt

definput = preallocate definput

counter j = 0

loop over the number of images to prepare a dialog box where the user can name the recordings

add 1 to the counter

set default name of image to ‘Name your recording’

store default title of image j in title\_list in column j

name the text edit field label as ‘Title subplot j’

default input values for every subplot title (j) = title\_list(j)

end

dlgtitle = title of dialog box where user can rename the recordings

dims = height and width of text edit field

answer = output of the dialog box containing the name of each recording as provided by the user (cell array)

loop over the title\_list

update title\_list values with the titles provided by the user

end

**%% Correct drift**

Update user on the next step in the analysis

% For GCaMP analysis we only use the UG file

D = current folder

S = list of .tif files from the green channel of the microscope

Get indices of correct order

Order S alphabetically based on the names of the images in S

datetime = current date and time

ymd = current date in year-month-day

mkdir = create a new folder with today’s date as its name

convert numeric ymd to a string

loop over the images (z-stacks) in S

number of rows and number of columns = dimensions of first image of z-stack

fname = name of image

separate name and extension of fname

numimgs = number of images in the z-stack

preallocate empty matrix with dimensions of image

store numimgs in array number\_images

loop over the images of the z-stack

I = read the jth image of the z-stack

Determine number of rows and columns of image I

Determine if number of rows of image j is equal to the number of rows of the first image

if image j has more rows than the first image of the stack

Delete last row of image j

elseif image j has less rows than the first image of the stack

Add missing rows at the end of the image with value zero

End

Determine if number of columns of image j is equal to the number of columns of the first image

if image J has more columns than the first image of the stack

Delete last column of image j

elseif image j has less columns than the first image of the stack

Add missing columns at the end of the image with value zero

End

Make a new z-stack (3-dimensional matrix) with the new images (matrices)

End

Delete the first layer of the matrix since it contains all zeros

% Drift correction according to the paper of Eftychios A. Pnevmatikakis & Andrea Giovannucci 2017

Convert matrix (corrected z-stack) to single

options\_rigid = change default value for bin\_width

% Calculate drift in UG stack

Calculate drift in UG z-stack using the options\_rigid

Change current folder to the analysis folder -> today’s analysis

Loop over frames in drift-corrected matrix M\_final

Isolate frame

Convert frame to uint16

Add frame to the newly created drift-corrected tif file

End

Go back one folder

Go back one folder (current folder is now the data folder)

End

Change current folder to the analysis folder -> today’s analysis

**%% Draw ROI around mitochondrion (in UG stack)**

% You can draw a maximum of 5 mitos per image!

Display instructions to drawing an ROI in the Command Window

D = current folder

S = get list of drift-corrected .tif files

Get indices of correct order

Order S alphabetically based on the names of the images in S

Set counter i to 1

Set counter j to 1

Set mito\_counter to 1

while j <= number of images in S

UG\_image\_corr = name of jth image (z-stack) in S

% Draw ROI on first frame

Read second image of z-stack

Show image with increased contrast

% Plot image 1 and draw rectangular ROI on images

Make 2-by-1 subplot in which the contrast-enhanced image at the top subplot

Title of the first subplot: ‘Select ROI around mitochondria’

Draw squared ROI around mitochondrion in the first subplot

The contrast-enhanced image appears on the bottom subplot

Title of the second subplot: ‘Move ROI to background location’

The ROI of the first subplot is copied to the second subplot

% Set up listeners for ROI moving events

Move the ROI in the second subplot to the background

Move the ROI in the second subplot to the background

Wait 5 seconds to give the user time to move the ROI to the background in the second subplot

Store number of mitochondrion in structure Stimulation

Create and store mask from mitochondrion ROI in Stimulation

Create and store mask from background ROI in Stimulation

Store number of images of stack in Stimulation

Save figure containing the two ROIs

Add 1 to counter i

Ask user if he wants to draw another ROI in this image (analyze another mitochondrion in this image)

if answer is yes

j = j (show the same image stack again to draw another mitochondrion)

add 1 to the mito\_counter

elseif answer = no

add 1 to counter j (proceed to next image stack)

reset mito\_counter to 1

end

end

**%% Apply ROIs on stacks**

Update user on the next step in the analysis

Update user on the next step in the analysis

D = current folder

S = get list of drift-corrected .tif files

Get indices of correct order

Order S alphabetically based on the names of the images in S

Set counter i to 1

Set counter j to 0

Loop over elements in Stimulation

if element has mito\_number 1

add 1 to counter j (thus, only proceed to next image in S if mito\_number is 1)

UG\_image\_corr = name of jth image in S

elseif element has mito\_number different from 1

UG\_image\_corr = name of jth image in S

End

% Assign mask to variable

Assign mitochondrion mask of current element to mask\_mito

Assign background mask of current element to mask\_background

% Apply mitochondrion mask on UG stack

Add empty field UG\_mean\_values to structure Stimulation

Loop over images in z-stack

Read image

Calculate mean value of mitochondria mask

Assign this value to the field UG\_mean\_values of structure Stimulation

End

% Apply background mask on UG stack

Add empty field UG\_mean\_background to structure Stimulation

Loop over images in z-stack

Read image

Calculate mean value of background mask

Assign this value to the filed UG\_mean\_background of structure Stimulation

End

Add 1 to counter i (proceed to next element)

End

**%% Automated start and stop times; and interval**

% The time of the first frame as well as the interval between frames is

% calculated from the information in the metadata. This information is used

% to make an x-axis in milliseconds instead of frame numbers

Go back to folder containing the data

Go back to folder containing the data

D = current folder

M = list of metadata (.txt) files in current folder

Obtain indices of order

Sort metadata (M) alphabetically based on filename

Counter k = 0

Loop over elements in Stimulation

% Search for start time

If element has mito\_number = 1

Add 1 to counter k

Open file

Return position of pointer in file

Use for loop to seek start time (time of first image in z-stack)

Move pointer to end of line 1

Obtain line containing the time

Isolate time from the line

Convert time from string to numerical

End

Assign start time to Stimulation structure

Elseif element has mito\_number different from 1

Use start time determined from mito 1

End

% search for sample interval

If element has mito\_number 1

Open file

Return position of pointer in file

Use for loop to find time interval between images in z-stack

Move pointer to end of line 1

Obtain line containing the time interval

Isolate time from the line

Convert time from string to numerical

End

Assign time interval between images to structure Stimulation

Elseif element has mito\_number different from 1

Use time interval determined from mito 1

End

% For every frame, calculate its timepoint (for x-axis of figures in milliseconds instead of frames)

If element has mito\_number 1

Preallocate x\_inseconds, the array to change the x-axis to time in milliseconds

First element in x\_inseconds is the start time

Loop over elements in x\_inseconds, starting from the second element

Time of element = start time + (number of element – 1) \* time interval between elements

End

Assign array x\_inseconds to structure Stimulation

Elseif element has mito\_number different from 1

Use x\_inseconds as calculated for mito 1

End

End

**%% Adjust titles (add the mito number after the title name)**

Reset counter j to 0

Loop over elements in Stimulation

If element has mito\_number = 1

Add 1 to counter j (thus, proceed to next element in title\_list)

Assign name of recording + mito + mito\_number to Stimulation.title

Elseif element has mito\_number different from 1

Assign name of recording + mito + mito\_number to Stimulation.title

End

End

**%% Subtract the UG background values from the UG ROI mean values**

Update user on the next step in the analysis

Loop over elements in Stimulation

Subtract the mean background value from the mean value on the mito and add to Stimulation

End

**%% Correct photobleaching until you are satisfied**

Update user on the next step in the analysis

Loop over elements in Stimulation

Calculate the frame where electrophysiological stimulation started

End

Create figure window (2)

Start infinite while loop

% Ask user to provide a value for ‘t’ (=rate of decay)

Instructions for dialog box

Title for dialog box

Dimensions of text fields in dialog box

Default input for dialog box

Ask user to provide a t value, maximum t value, and minimum t value (t = rate of decay)

Convert answer to double and store it in variable t

Convert maximum value for t to double

Convert minimum value for t to double

Loop over elements in stimulation

Assign UG\_final field of structure Stimulation to array UG\_data

Preallocate array FitTo with NaN

FitTo 1:onset stimulation = UG\_data 1:onset stimulation

xfit = array of 1 to length(FitTo) with step size 1

idxValid = values from FitTo that are not NaN

a = last value in UG\_data – first value in UG\_data

c = last value in UG\_data

ft = fit data to exponential curve

fitmethod = NonlinearLeastSquares

Startpoint = [a,c,t]

Upper values of a, c, t

Lower values of a, c, t

Fit data from before onset stimulation to an exponential curve using the options in fo

Create array BleachFitted with the intensity values after fitting

Go to the analysis folder

Determine the number mitochondria

Calculate number of rows needed for subplots

Go back to the data folder

Go back to the data folder

Draw subplot on location i (counter)

Hold on to draw multiple graphs on top of eachother

Plot UG\_data in red

Plot FitTo in yellow

Plot BleachFitted in black

Normalize BleachFitted to the first value

Switch row with column

Bleach corrected data = divide value of UG\_data by corresponding value in BleachFittedNormM

Store bleach corrected data in Stimulation structure

Calculate dFF by dividing every value of DataBC with the mean value before stimulation onset,

convert it to percentage, and store it in the structure Stimulation

Plot the dFF in percentage in blue

Title of subplot is title of recording, fontsize 5

Xlabel is ‘Frame’

Ylabel is ‘dF/F (%)’

End

Pause 5 seconds so that plots can be drawn

Open dialog box and ask user if the bleach correction worked correctly

Answer

If answer = yes

Continue with bleach corrected data

Stop infinite while loop

If no

Clear figure 2, provide different t-values now

end

end

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% dF/F using baseline averaging**

Update user on the next step in the analysis

% dF/F in percent is available under Stimulation(i).dFF\_percent

% But, we just want dF/F (not in percentage)

% Calculate F0 as the average of the frames before stimulation (mean of baseline)

Loop over elements in Stimulation

F0 = average DataBC before stimulation

Loop over each value in DataBC

Calculate dF/F by subtracting F0 from the value in DataBC and dividing this by F0

End

End

**%% Signal smoothing filter**

Update user on the next step in the analysis

% Average factor 3 (datapoint is mean of datapoint plus 2 datapoints earlier)

Loop over elements in Stimulation

First two values of smoothed dFF are equal to the first to values in dFF

Loop over values in dFF, starting from the third value

dFF smoothed is mean of the that value and the two preceding values in dFF

end

end

**%% Rest data -> dFF smoothed of Prestim (1 until 1 frame before start stimulation)**

Update user on the next step in the analysis

Loop over elements in Stimulation

Store dFF smoothed of prestimulation frames in new structure Prestim

End

Initialize baseline standard deviation by creating an empty array

% Calculate prestim SD, average, peak

Loop over elements in Prestim

Calculate prestim SD and store in structure Prestim

Store prestim SD in array baseline\_SD (for later use)

Calculate prestim average and store in structure Prestim

Calculate prestim peak and store in structure Prestim

% Calculate slope of prestim curve

Fit linear polynomial curve through x\_inseconds (x-axis), and dFF\_smoothed (y-axis)

Store slope of this curve in structure Prestim

% Calculate Area Under Curve during prestim

Loop over values in Prestim(i).dFF\_smoothed

If value is equal or larger than 0

Copy value from Prestim(i).dFF\_smoothed to Prestim(i).AUC

Else

Value in Prestim(i).AUC = 0 (dFF values below 0 do not contribute to the AUC)

End

End

AUC for recording in Prestim is calculated using the trapezoid method

End

**%% Prepare plotting**

% Calculate minimum and maximum y-value of all recordings to give all plots the same axes

Initialize ymin (minimum y value) by giving it the largest possible value (inf) so that it will always be overwritten by

values from your data

Initialize ymax (maximum y value) by giving it the lowest possible value (-inf)

Loop over recordings in Stimulation

If minimum value of recording is lower than ymin

Overwrite ymin with value of recording

End

If maximum value of recording is higher than ymax

Overwrite ymax with value of recording

End

End

**%% Frequency of Action Potentials during stimulation**

Update user on the next step in the analysis

Initialize variable AP\_frequency by creating an empty array

Create figure 12

Load electrophysiological traces recorded during the experiment

Store ephys\_traces in a structure

Reset counter j to 0

Loop over elements in Stimulation

Draw subplot in figure(12) on location i

Name the subplot by using the title stored in the Stimulation structure, fontsize = 5

Label y-axis ‘Ephys’

% Determine number of peaks only during period of stimulation

If element has mito\_number 1

Add one to counter j (proceed to next z-stack)

End

Calculate number of datapoints in electrophysiological traces per millisecond (x\_inseconds is in milliseconds)

x = ephys traces during stimulation (from 2000:3000 ms)

find and show peaks in ephys traces during stimulation, minimum height of peak is AP\_threshold

ylim for grahp = minimum and maximum value of ephys trace

Determine height and location of peaks during stimulation

AP frequency = number of peaks / duration of stimulation (in seconds)

Store number of APs found in structure Stimulation

Store frequency of APs in structure Stimulation

Store action potential frequency in vector AP\_frequency (for later use)

end

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

% If MinPeakHeight does not work properly, try MinPeakProminence. Change AP

% threshold to for example 1 and play around with this number.

**%% Find peaks in calcium response**

Update user on next step in analysis

Initialize variable peak\_values

Initialize variable peak\_loc

Initialize variable time\_to\_peak

Create figure 9

Loop over elements in Stimulation

Create subplot on location i

Plot dFF\_smoothed with time in ms on the x-axis

Name the subplot by using the title stored in the Stimulation structure, fontsize = 5

At last subplot

Label x-axis ‘Time (ms)’

End

x limits are 0 & last value in Stimulation.x\_inseconds

At first subplot

Label y-axis ‘dF/F’

End

y-limits are the earlier calculated ymin and ymax

hold on to plot multiple lines in one subplot

x = define coordinates for area that shows period of stimulation

y = define coordinates for area that shows period of stimulation

show period of stimulation with a blue area

make area transparent

remove edge of area

find maximum value and location of this value in dFF\_smoothed curve between start of stimulus and 200ms

after stimulation

store maximum value in Stimulation

store location of maximum value in Stimulation

store maximum value in peak\_values

store location of maximum value in peak\_loc

calculate the time it takes (in ms) after stimulation starts to reach the peak and store value in time\_to\_peak

store time to peak in Stimulation

if the peak exceeds the threshold to quantify the calcium uptake as ‘response’

encircle peak in green

store ‘yes’ in field Stimulation(i).response

else

encircle peak in red

store ‘no response’ in filed Stimulation(i).response

end

hold off

end

title of figure 9

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Plot Ephys traces over dF/F**

Update user on next step in analysis

Create figure 4

Reset counter j to 0

Loop over elements in stimulation

If element has mito\_number 1

Add 1 to counter j

End

Create subplot on location i

Name the subplot by using the title stored in the Stimulation structure, fontsize = 5

x1 = array from 1 : number of values in ephys traces

y1 = ephys traces

draw ephys trace

make curve transparent

create handle for axes

adjust x-axis limits to match ephys trace

color of x-axis = black

make x-axis invisible

make y-axis invisible

adjust y-axis limits using the earlier provided values

retrieve position of x-axis

create new axis on this location (in order to make the x and y axis fit to the calcium trace which has less

datapoints than the ephys trace)

make x-axis black

x-axis in milliseconds

adjust y-limits to the earlier calculated ymin and ymax

xticks at 2000 and 4000 ms

At first subplot

Label y-axis ‘dF/F’

End

At last subplot

Label x-axis ‘Time (ms)’

End

x2 = time in milliseconds

y2 = dFF\_smoothed

draw curve using coordinates in x2 and y2 on axes ax2

create handle to curve

if the peak exceeds the threshold to quantify the calcium uptake as ‘response’

change curve color to black

else

change curve color to grey

end

end

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Area under the curve**

Update user on next step in analysis

Loop over elements in Stimulation

Loop over frames of element (recording)

If dFF\_smoothed value is equal or larger than 0

Copy value from Stimulation(i).dFF\_smoothed to Stimulation(i).AUC

Else

Value in Stimulation(i).AUC = 0 (dFF values below 0 do not contribute to the AUC)

End

End

AUC for recording in Stimulation is calculated using the trapezoid method during start of stimulation until

200 ms after stimulation

End

Update user on next step in analysis

% Visualization of AUC

Create figure 5

Loop over elements in Stimulation

Create subplot on location k

Plot dFF\_smoothed of current recording

Name the subplot by using the title stored in the Stimulation structure, fontsize = 5

Use y-limits as earlier calculated

Draw vertical line to indicate start of stimulation

Hold on to keep plotting in the same subplot

x = indices over which the AUC is calculated

copy Stimulation.AUC to Stimulation.AUC\_vis

y = Stimulation.AUC\_vis values as calculated earlier

draw horizontal line at y=0

at first subplot

ylabel = ‘dF/F’

end

if the peak exceeds the threshold to quantify the calcium uptake as ‘response’

draw AUC in green

make AUC transparent

else

draw AUC in red

make AUC transparent

end

hold off

end

title of figure

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Plot sorted APfreq over peak height**

% if the cell responded, the trial is indicated by a black square

% if no response, the trial is indicated by a grey circle

Initialize variable peak\_values

Loop over elements in Stimulation

Add peak value of element in variable peak\_values

End

Sort AP frequency from low to high frequency

Put elements in peak\_values in the same order (so that the xth element in APfrequency\_sorted is from the same

recording as the xth element in peak\_values)

order baseline\_SD in the same way

create bins to categorize trials in based on action potential frequency. Bin1 = 0:89 Hz

bin2 = 90:110 Hz

bin3 = 111:130 Hz

bin4 = 131:150 Hz

bin5 = 151:400 Hz

create empty variable to store peak height data in of trials belonging to bin1

bin2

bin3

bin4

bin5

Preallocate cell variable index\_array

Loop over elements in peak\_heigt\_sorted (sorted on AP frequency)

if action potential frequency of that recording belongs to bin1

add peak height value to bin1\_data

add < 90 in index\_array

elseif action potential frequency belongs to bin2

add peak height value to bin2\_data

add 91-110 to index\_array

elseif action potential frequency belongs to bin3

add peak height value to bin3\_data

add 111-130 to index\_array

elseif action potential frequency belongs to bin4

add peak height value to bin4\_data

add 131-150 to index\_array

elseif action potential frequency belongs to bin5

add peak height value to bin5\_data

add > 150 to index\_array

end

end

create cell array ‘titles’ with name of each bin

convert cell array ‘titles’ to categorical array and store in variable ‘x’

reorder variable x in alphanumeric order (<90, 91-110, etc)

calculate the mean peak height per bin and store in variable y

Create figure 10

Create bar graph with AP frequency bins on x and average peak height per bin on y

Hold on to plot individual datapoints in the same graph

Loop over elements in peak\_height\_sorted

x\_var = create categorical array of value in bin name in index\_array (e.g. 91:110)

y\_var = value in peak\_height\_sorted

if peak height is larger than the threshold (… number of times the baseline SD)

plot peak height as black square in bar graph

else

plot peak height as grey circle in bar graph

end

end

title of figure is ‘Peak height per AP frequency’

label x-axis

label y-axis

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Plot sorted AP frequency over time to peak**

Initialize variable peak\_loc

Loop over elements in Stimulation

Use the field x\_inseconds of structure Stimulation to calculate the time of peak onset (in ms) and add this

value to peak\_loc

End

Sort values in time\_to\_peak based on AP frequency measured in that recording and store in time\_to\_peak\_sorted

Clear data in bin1\_data

bin2

bin3

bin4

bin5

Clear data in index\_array. Preallocate cell variable index\_array

Loop over elements in time\_to\_peak\_sorted (sorted on AP frequency)

if action potential frequency of that recording belongs to bin1

add time to peak value to bin1\_data

add < 90 in index\_array

elseif action potential frequency belongs to bin2

add time to peak value to bin2\_data

add 91-110 to index\_array

elseif action potential frequency belongs to bin3

add time to peak value to bin3\_data

add 111-130 to index\_array

elseif action potential frequency belongs to bin4

add time to peak value to bin4\_data

add 131-150 to index\_array

elseif action potential frequency belongs to bin5

add time to peak value to bin5\_data

add > 150 to index\_array

end

end

create cell array ‘titles’ with name of each bin

convert cell array ‘titles’ to categorical array and store in variable ‘x’

reorder variable x in alphanumeric order (<90, 91-110, etc)

calculate the mean time to peak per bin and store in variable y

Create figure 11

Create bar graph with AP frequency bins on x and average time to peak per bin on y

Hold on to plot individual datapoints in the same graph

Loop over elements in time\_to\_peak\_sorted

x\_var = create categorical array of value in bin name in index\_array (e.g. 91:110)

y\_var2 = value in time\_to\_peak\_sorted

y\_var = value in peak\_height\_sorted

if peak height is larger than the threshold (… number of times the baseline SD)

plot time to peak as black square in bar graph

else

plot time to peak as grey circle in bar graph

end

end

draw horizontal line at time point at which stimulation has stopped

title of figure is ‘Peak location per AP frequency’

label x-axis

label y-axis

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Plot sorted APfreq over AUC**

Preallocate variable AUC\_values

Loop over elements in Stimulation

Store AUC area in variable AUC\_values

End

Sort values in AUC based on AP frequency measured in that recording and store in AUC\_sorted

Clear data in bin1\_data

bin2

bin3

bin4

bin5

Clear data in index\_array. Preallocate cell variable index\_array

Loop over elements in AUC\_sorted (sorted on AP frequency)

if action potential frequency of that recording belongs to bin1

add AUC value to bin1\_data

add < 90 in index\_array

elseif action potential frequency belongs to bin2

add AUC value to bin2\_data

add 91-110 to index\_array

elseif action potential frequency belongs to bin3

add AUC value to bin3\_data

add 111-130 to index\_array

elseif action potential frequency belongs to bin4

add AUC value to bin4\_data

add 131-150 to index\_array

elseif action potential frequency belongs to bin5

add AUC value to bin5\_data

add > 150 to index\_array

end

end

create cell array ‘titles’ with name of each bin

convert cell array ‘titles’ to categorical array and store in variable ‘x’

reorder variable x in alphanumeric order (<90, 91-110, etc)

calculate the mean AUC per bin and store in variable y

Create figure 6

Create bar graph with AP frequency bins on x and average AUC per bin on y

Hold on to plot individual datapoints in the same graph

Loop over elements in AUC\_sorted

x\_var = create categorical array of value in bin name in index\_array (e.g. 91:110)

y\_var2 = value in AUC\_sorted

y\_var = value in peak\_height\_sorted

if peak height is larger than the threshold (… number of times the baseline SD)

plot AUC as black square in bar graph

else

plot AUC as grey circle in bar graph

end

end

title of figure is ‘AUC per AP frequency’

label x-axis

label y-axis

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Quantify calcium response to stimulus using the slope of the trendline**

Update user on next step in the analysis

Initialize variable trendline\_values

Create figure 7

Loop over elements in Stimulation

Create subplot on location i

If peak height exceeds threshold (… times the baseline SD)

Plot dFF\_smoothed in green

Else (no response)

Plot dFF\_smoothed in red

End

Fit linear polynomial curve to dFF\_smoothed during stimulation and store this fit under variable f

f.p1 is the slope of the fit curve, store this value in the Stimulation structure

hold on

plot fit curve on top of dFF\_smoothed graph

name the subplot

if it is the last subplot

label x-axis

end

set limits to the x-axis so that the dFF\_smoothed curve nicely fits in the subplot

set ticks at x=2000 and x=4000

format short

if it is the first subplot

label y-axis

end

use the earlier calculated ymin and ymax as y-limits

x = define coordinates for area that shows period of stimulation

y = define coordinates for area that shows period of stimulation

show period of stimulation with a blue area

make area transparent

remove edge of area

hide legend

hold off

save slope of trendline in array trendline\_values

end

title the figure

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Plot sorted AP frequency over slope trendline**

Preallocate variable trendline\_values

Loop over elements in Stimulation

Store slope of trendline in variable trendline\_values

End

Sort values in slope trendline values based on AP frequency measured in that recording and store in

trendline\_sorted

Clear data in bin1\_data

bin2

bin3

bin4

bin5

Clear data in index\_array. Preallocate cell variable index\_array

Loop over elements in trendline\_sorted (sorted on AP frequency)

if action potential frequency of that recording belongs to bin1

add slope trendline value to bin1\_data

add < 90 in index\_array

elseif action potential frequency belongs to bin2

add slope trendline value to bin2\_data

add 91-110 to index\_array

elseif action potential frequency belongs to bin3

add slope trendline value to bin3\_data

add 111-130 to index\_array

elseif action potential frequency belongs to bin4

add slope trendline value to bin4\_data

add 131-150 to index\_array

elseif action potential frequency belongs to bin5

add slope trendline value to bin5\_data

add > 150 to index\_array

end

end

create cell array ‘titles’ with name of each bin

convert cell array ‘titles’ to categorical array and store in variable ‘x’

reorder variable x in alphanumeric order (<90, 91-110, etc)

calculate the mean slope trendline value per bin and store in variable y

Create figure 8

Create bar graph with AP frequency bins on x and average slope trendline value per bin on y

Hold on to plot individual datapoints in the same graph

Loop over elements in trendline\_sorted

x\_var = create categorical array of value in bin name in index\_array (e.g. 91:110)

y\_var2 = value in trendline\_sorted

y\_var = value in peak\_height\_sorted

if peak height is larger than the threshold (… number of times the baseline SD)

plot slope trendline value as black square in bar graph

else

plot slope trendline value as grey circle in bar graph

end

end

title of figure is ‘Trendline per AP frequency’

label x-axis

label y-axis

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Repetitions (calculate the average response of repetitions) (max 5 ROIs/mitochondria per recording)**

Initialize variable title\_list

Loop over elements in Stimulation

Convert title of element to cell array of character vectors and store in title\_list\_cell

End

The first recording can never be a repetition, thus field average AUC (AvgAUC) of Stimulation remains empty

Average trendline

Average peak height

Average time to peak

Recording 1 is not a repetition

The first recording can never be a repetition, thus field average DFF (AvgDFF) of Prestim remains empty

Average slope of trendline

Average peak height

Average AUC

Recording 1 is not a repetition

Initialize variable double\_list to store names of recordings. If the name of a recording is already in this variable, it

means that the recording is a repetition

add the title of the first recording to the double\_list

loop over elements in Stimulation, starting with the second recording (first recording was not a repetition)

if element has mito\_number 1

check if title of element i is in double\_list

if true (if there is a match of the element title with any title in double\_list)

Stimulation(i).Repetition = “Yes”

Prestim(i).Repetition = “Yes”

%Calculate the average of repetitions

If i ==2 and mito\_number is 1

row\_shift = 1 (will be used later)

elseif i <= 2

empty variable temp\_array

store the mito\_number of the first three recordings in Stimulation

find the highest mito number and store under row\_shift

elseif i is not one of the first 2 or final 2 elements in Stimulation

empty variable temp\_array

store the mito\_number of 2 elements before until 2 elements after element i

find the highest mito number and store under row\_shift

elseif i is one of the 2 final elements in Stimulation

empty variable temp\_array

store the mito\_number of 2 elements before element i until the last element

find the highest mito number and store under row\_shift

end

% Now the average of the parameters of repetition trials will be calculated. Row\_shift has found how many mitochondria/ROIs are analyzed per image/z-stack. If for example 4 ROIs are drawn in an image, a parameter of element i in Stimulation needs to be averaged with the same parameter of 4 elements earlier (open Stimulation.title to visualize). Row\_shift is used to accomplish this.

Calculate average AUC during stimulation and store in field AvgAUC of structure Stimulation

Calculate average slope of trendline and store in Stimulation

Calculate average peak height and store in Stimulation

Calculate average time to peak and store in Stimulation

Calculate average dF/F during baseline and store in field AvgDFF of structure Prestim

Calculate average slope of trendline and store in Prestim

Calculate average peak height and store in Prestim

Calculate average AUC and store in Prestim

Else (if title of element is not in double\_list, thus element i is not a repetition)

Stimulation(i).Repetition = “No”

Do not calculate an average AUC and keep field AvgAUC empty

Field AvgTrendline is empty

Field AvgPeakHeight is empty

Field AvgTimeToPeak is empty

Prestim(i).Repetition = “No”

Field AvgDFF is empty

Field AvgTrendline is empty

Field AvgPeakHeight is empty

Field AvgAUC is empty

End

Elseif mito\_number of element in Stimulation is not 1

Use information of first mitochondrion/ROI of image to determine if element is a repetition or not

If element is a repetition

Prestim(i).Repetition = “Yes”

Calculate average AUC during stimulation using row\_shift and store in field AvgAUC

Calculate average slope of trendline and store in Stimulation

Calculate average peak height and store in Stimulation

Calculate average time to peak and store in Stimulation

Calculate average dF/F during baseline and store in field AvgDFF of structure Prestim

Calculate average slope of trendline and store in Prestim

Calculate average peak height and store in Prestim

Calculate average AUC and store in Prestim

Else (if element is not a repetition)

Stimulation(i).Repetition = “No”

Do not calculate an average AUC and keep field AvgAUC empty

Field AvgTrendline is empty

Field AvgPeakHeight is empty

Field AvgTimeToPeak is empty

Prestim(i).Repetition = “No”

Field AvgDFF is empty

Field AvgTrendline is empty

Field AvgPeakHeight is empty

Field AvgAUC is empty

End

Else

Stimulation(i).Repetition = “No”

Do not calculate an average AUC and keep field AvgAUC empty

Field AvgTrendline is empty

Field AvgPeakHeight is empty

Field AvgTimeToPeak is empty

Prestim(i).Repetition = “No”

Field AvgDFF is empty

Field AvgTrendline is empty

Field AvgPeakHeight is empty

Field AvgAUC is empty

End

Add title of element in double\_list

End

**%% Repetitions: Trendline over stimulation strength**

Isolate the unique titles (/stimulations/recordings) from the title list and store under dif\_titles

Determine how many unique titles there are and store under count\_titles

Preallocate cell x3

Initialize variable y3

Loop over elements in Stimulation

Convert title of recording from string to character and store in x3

Add slope of the calcium response to y3

End

Initialize variable y4

Loop over the unique titles

Calculate the mean slope of the calcium response for recordings that have the same name

End

Convert the unique titles to categorical array

Order the unique titles in alphanumeric order

Create figure 13,

Plot a bar graph with the unique titles on x and the mean slope of response on y

Hold on to plot individual datapoints in the bars

Loop over elements in Stimulation

Convert title of element to categorical and store under x\_var

y\_var = slope of response

if peak of element exceeds earlier defined threshold

plot datapoint as black square

else (no response)

plot datapoint as grey dot

end

end

name figure

name x-axis

name y-axis

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Calculate average AUC for repetitions**

Initialize variable y5

Loop over elements in Stimulation

Add AUC of element to y5

End

Initialize variable y6

Loop over unique titles

Calculate the mean AUC for recordings that have the same name

End

Create figure 14

Plot a bar graph with the unique titles on x and the mean AUC on y

Hold on to plot individual datapoints in the bars

Loop over elements in Stimulation

Convert title of element to categorical and store under x\_var

y\_var = AUC

if peak of element exceeds earlier defined threshold

plot datapoint as black square

else (no response)

plot datapoint as grey dot

end

end

hold off

name figure

name x-axis

name y-axis

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Calculate average peak height for repetitions**

Initialize variable y7

Loop over elements in Stimulation

Add peak height of element to y7

End

Initialize variable y8

Loop over unique titles

Calculate the mean peak height for recordings that have the same name

End

Create figure 15

Plot a bar graph with the unique titles on x and the mean peak height on y

Hold on to plot individual datapoints in the bars

Loop over elements in Stimulation

Convert title of element to categorical and store under x\_var

y\_var = peak height

if peak of element exceeds earlier defined threshold

plot datapoint as black square

else (no response)

plot datapoint as grey dot

end

end

hold off

name figure

name x-axis

name y-axis

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Calculate average time to peak for repetitions**

Initialize variable y9

Loop over elements in Stimulation

Add AUC of element to y9

End

Initialize variable y10

Loop over unique titles

Calculate the mean time to peak for recordings that have the same name

End

Create figure 14

Plot a bar graph with the unique titles on x and the mean time to peak on y

Hold on to plot individual datapoints in the bars

Loop over elements in Stimulation

Convert title of element to categorical and store under x\_var

y\_var = time to peak

if peak of element exceeds earlier defined threshold

plot datapoint as black square

else (no response)

plot datapoint as grey dot

end

end

hold off

name figure

name x-axis

name y-axis

plot horizontal line at y= stimulation length (stimulation stop time – start time)

paper orientation is landscape

paper units normalized

use the whole paper to draw the figure

**%% Divide data per subcellular region**

% Per subcellular region (dendrite, proximal axon, distal axon, soma), calculate the mean for every parameter using only the first recording in case of a repetition (so, exclude repetition trials). This is since responses tend to be less strong during repetitions

For every subcellular region, initialize the variables title, AUC, peak height, slope of trendline, and peak location (time to peak)

“

“

“

“

“

“

“

“

“

“

“

“

“

“

“

“

“

“

% if a stimulation is not named correctly according to the instructions, it will end up in ‘other’

“

“

“

“

“

% within cell statistics have been removed as we did not use it

Loop over elements in Stimulation

If element is not a repetition trial

If element is recorded in a dendrite (element title contains ‘endrite’, unique for dendrite recordings)

Add title of element to variable Dendrite\_titles

Add AUC of element to variable Dendrite\_AUC

Add Peak height of element to variable Dendrite\_PeakHeight

Add slope of response of element to variable Dendrite\_trendline

Add time to peak of element to variable Dendrite\_PeakLoc

Elseif element is recorded in a soma

Add title of element to variable Soma\_titles

Add AUC of element to variable Soma\_AUC

Add Peak height of element to variable Soma\_PeakHeight

Add slope of response of element to variable Soma\_trendline

Add time to peak of element to variable Soma\_PeakLoc

Elseif element is recorded in a distal axon (Branch Order Unknown of enpassant)

Add title of element to variable DistAx\_titles

Add AUC of element to variable DistAx\_AUC

Add Peak height of element to variable DistAx\_PeakHeight

Add slope of response of element to variable DistAx\_trendline

Add time to peak of element to variable DistAx\_PeakLoc

Elseif element is recorded in a proximal axon (8th character in title is a number)

Add title of element to variable ProxAx\_titles

Add AUC of element to variable ProxAx\_AUC

Add Peak height of element to variable ProxAx\_PeakHeight

Add slope of response of element to variable ProxAx\_trendline

Add time to peak of element to variable ProxAx\_PeakLoc

Else (‘other’)

Add title of element to variable Other\_titles

Add AUC of element to variable Other\_AUC

Add Peak height of element to variable Other\_PeakHeight

Add slope of response of element to variable Other\_trendline

Add time to peak of element to variable Other\_PeakLoc

End

End

End

% Calculate the mean for every parameter per subcellular region

Average AUC for dendritic mitochondria

Average slope trendline

Average peak height

Average time to peak

Average AUC for mitochondria in the proximal axon

Average slope trendline

Average peak height

Average time to peak

Average AUC for mitochondria in the distal axon

Average slope trendline

Average peak height

Average time to peak

Average AUC for somatic mitochondria

Average slope trendline

Average peak height

Average time to peak

Create matrix with on every row (region) the mean value for AUC, slope trendline, peak height & peak location

“

“

“

Convert matrix to a table and add row and column names

**%% Save data**

Update user on the next step in the analysis

Change directory to folder ‘current date’ in folder ‘Analysis’ (earlier created by the script)

Save figure 2 as PDF file, name ‘PDF Bleach Correction’

%In order to store dF/F values, preallocate a matrix with the number of columns equal to the longest recording so

that the dF/F values of all recordings fit in the matrix

initialize the variable max\_length that stores the number of dF/F values of the longest recording

loop over recordings in Stimulation

determine the number of dF/F values

if this number is larger than the largest number until now

overwrite max\_length with number of dFF values of current recording

end

end

preallocate matrix to store dFF values. Rows are the recordings, columns the dF/F values

initialize variable stim\_response

initialize average AUC

initialize average slope of trendline

initialize average peak height

initialize average time to peak

initialize repetition

initialize prestim average dFF

initialize prestim average slope of trendline

initialize prestim average peak height

initialize prestim average AUC

initialize prestim repetition

% only save the smoothed dF/F values in excel

Loop over elements in Stimulation

Loop over frames of z-stack of current element/recording

Store smoothed dF/F values in dFF\_values

End

Store response (‘yes’ or ‘no’) in stim\_response

Store frequency of imaging (number of frames recorded per second) in imaging\_freq

Store average AUC of repetitions in AvgAUC

Store average slope of trendline of repetitions in AvgTrendline

Store average peak height of repetitions in AvgPeakHeight

Store average time to peak of repetitions in AvgTimeToPeak

Store repetition (‘yes’ or ‘no’) in Repetition

Store prestim average dF/F of repetitions in PreAvgDFF

Store prestim average slope of trendline of repetitions in PreAvgTrendline

Store prestim average peak height of repetitions in PreAvgPeakHeight

Store prestim average AUC of repetitions in PreAvgAUC

Store prestim repetition (‘yes’ or ‘no’)

End

%Make a title list to name recording in excel. Since a name can only occur one (problem with repetititions), add a counter before the name

Preallocate title\_list\_for\_save

Loop over elements in Stimulation

Create array with a counter followed by the title of the element

Join content of array to create one name

Convert string to character array

Add character array to title\_list\_for\_save

End

Create table containing the values in variable stim\_response (one column, named ‘Response’)

Create table containing dFF values

Change rows with columns

Create table containing the calculated variables during stimulation for every recording

Combine the three created tables into one table

% Save prestim data in different table

Loop over elements in Prestim

Store baseline average dF/F

Store peak during baseline

Store slope of response during baseline

Store AUC during baseline

End

Create table and store Prestim parameters

Save matlab file containing all variables and data of current analysis

Save table containing data during stimulation in Excel file

Save data containing data during prestim in Excel file

% within cell statistics have been removed

Save average response per subcellular region in Excel file

Save figures created during the analysis as a PDF file with correct naming. Figures 1 (ROIs) and 2 (bleach correction) have already been saved. Figure 3 does not exist.

“

“

“

“

“

“

“

“

“

“

“

Update user that the analysis is finished

Close all figures

**%% Warn user if recording could not be classified to a subcellular region and is thus classified as ‘Other’**

If there are one ore more elements in Other\_titles

Warning: Some trials have not been divided into the right subcellular region!

End

clear workspace

**%% Create heatmap of recording. As heatmap files are large, only do this for specific recordings.**

% Create heatmap of z-stack (select lines and press ctr + t to uncomment)

Identify current folder

Make list of drift-corrected UG tif files

Get indices of alphanumeric order

Sort files so that recording one is the first item in the list, etc

% pick a trial number of which you want a heatmap-stack

trial\_numb = number of element in S you want a heatmap-stack of

trial = name of element

loop over frames of trial

read image

clear figure 1

create figure 1

create pseudocolor plot of frame

interpolate colors to create a smooth heatmap

use colormap ‘jet’ (blue – red)

color axis, adapt the lower and upper limit to the image

remove grid and get handle to current figure

remove values on x and y axis

add colorbar

add 2 ticks to the colorbar, one at a low and one at a high value (adapt this to the image)

label the low value ‘Low Ca2+’ and the high value ‘High Ca2+’

save frame

end

make list of all heatmap frames

get indices of alphanumeric order

sort heatmap files based on name (so that the first frame is the first element in the list)

loop over heatmap frames

read frame

add frame to previous frames (to create a z-stack)

end

delete heatmap frames and only keep the heatmap z-stack