## NeuroChaT GUI Use Guide

**Download and installation:**

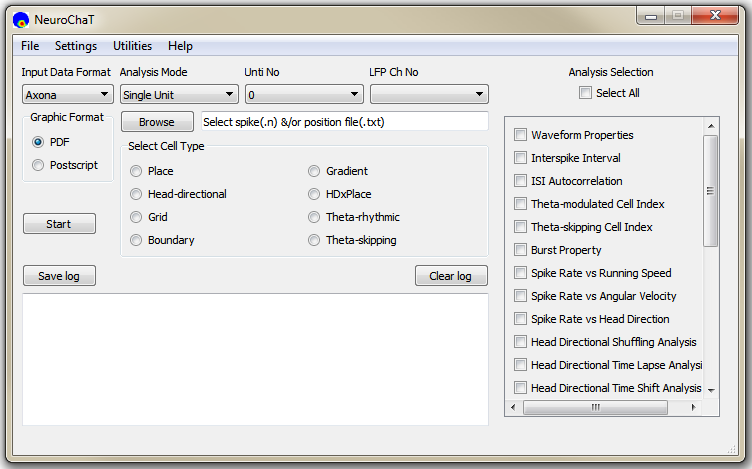
Download the codes from OSF website or GitHub

* OSF link: <https://osf.io/kqz8b>
* GitHub link: <https://github.com/mnislamraju/NeuroChaT>

**Start NeuroChaT GUI (no installation required)**

* Using command prompt:
* Change your current directory to the NeuroChaT directory
  + >cd C:\path\to\neurochat
* Execute following command
  + >python neurochat.py
* Or, >python C:\path\to\neurochat\neurochat.py
* Using Python IDE, i.e. Spyder:
* Open neurochat.py in the editor
* Press the **Run** button to start NeuroChaT GUI

Following window will pop-up:

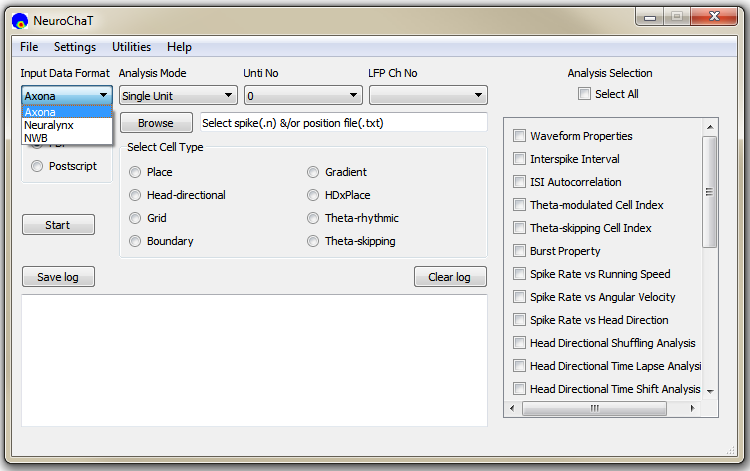


**Figure 1:** NeuroChaT graphical user interface

**Overview of the front panel:**

**Input Data Format**

Select one of the three data or file formats. NeuroChaT is currently supporting Axona, Neuralynx and NWB (HDF5) file formats.



**Figure 2:** Select the input file format from one of the three options in the 'Input Data Format' dropdown menu.

**Analysis Mode**

NeuroChaT works in three analysis modes (**Figure 3**): Single Unit, Single Session, Listed Units. Following are the descriptions of how these modes work.

**a. Single Unit:** When selected, it analyses data that belongs to a single cluster extracted from spike-sorting methods. This mode is a perfect choice when the detailed analysis using many functions is required to explore more properties of the cell.

**b. Single Session:** This mode analyses data from a single spike file. In Neuralynx system, it accepts .ntt and .nst files. For Axona system, it supports .n (n= 1, 2, 3 etc.) files. This mode looks for all the clusters that have been identified from the spike sorting methods, and analyse the selected functions for all of them. Along with appropriate choice of analysis functions, this is a powerful mode for thorough examination and characterization of units from a single electrode.

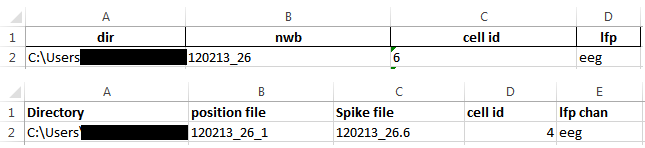
**c. Listed Units:** This mode analyses data from units those are listed in an Excel format as shown in **Figure 3**. The first column is the directory where the data is stored. Data specifications are provided by following means:

**HDF5 File:**

|  |  |  |  |
| --- | --- | --- | --- |
| Column 1 | Column 2 | Column 3 | Column 4 |
| Data directory | Name of the HDF5 file without extension | Single unit of interest | LFP channel ID |

**Axona and Neuralynx Files:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column 1 | Column 2 | Column 3 | Column 4 | Column 5 |
| Data directory | Name of the spatial data file without extension | Name of the spike data file with extension | Single unit of interest | LFP channel extension (Axona)  Or Name of the LFP data file (Neuralynx) |

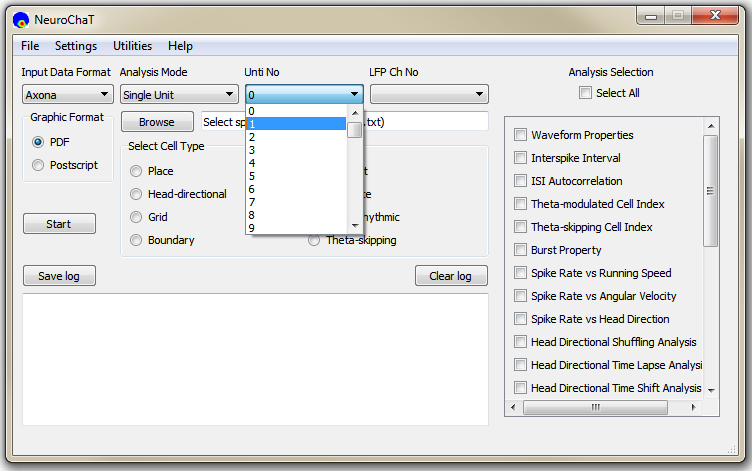


**Figure 3:** Input style in Excel files for batch mode analysis using ‘Listed Cell’. Top row shows the style for HDF5 data. Bottom row shows the style for Axona and Neuralynx systems.

Note: For Axona system, you should provide the spatial information as a .txt file format. If you use TINT for spike-sorting, you can export these information from there as .txt file.

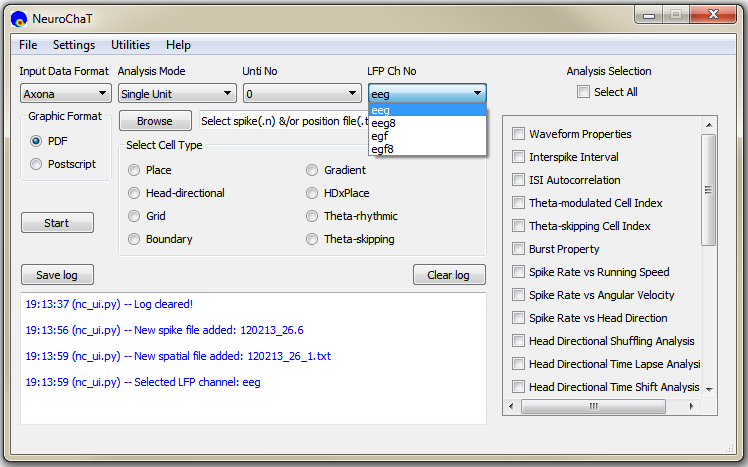
**Unit No**

Select the unit that you are interested to analyse. Although NeuroChaT lists up to 255 units, units those are identified using the clustering process can only be used.



**LFP Ch No**

Select the LFP channel from the dropdown list. When the data is browsed using the ‘Browse’ button this box fills with the potential LFP channels in the folder with recorded data. For HDF5 dataset, it shows the data groups in the directory ‘/processing/Neural Continuous/LFP’. It is always ‘ncs’ For Neuralynx data to refer to the files with .ncs extension.



**Figure 4:** Dropdown list to select the LFP channel

**Graphic Format**

Select one of the two options for the file format of graphical output from NeuroChaT analyses.

**Browse**

Clicking on this button prompts for the file or folder selection based on analysis mode. For ‘Single Unit’ and ‘Single Session’ mode, this will ask user to select .ntt/.nst file followed by .nvt file (Neuralynx) or .n file followed by .txt file (Axona). For ‘Listed Units’ mode, it will ask for specifying the .xls/.xlsx file that contains the list of the units.

**Analysis Selection**

This section provides a list of analyses that can be selected by ticking the boxes beside their names. Checking the ‘Select All’ box selects all the analyses and unchecking it removes their selection.

**Select Cell Type**

Analyses of interest can also be set by pressing one of the buttons in the ‘Select Cell Type’ section. It selects the analyses of interests essential for individual unit type. For example, characterizing units for rhythmic properties does not require spatial analyses. Therefore, only the analysis set that characterises spike trains are selected. The user can select or deselect analyses from this set.

**Log Box**

The box with white background at the bottom of the user-interface displays log of NeuroChaT actions, warnings and errors. Warnings are represented by orange texts while errors are displayed in red texts. All other NeuroChaT logs are shown as blue texts.

**Save Log**

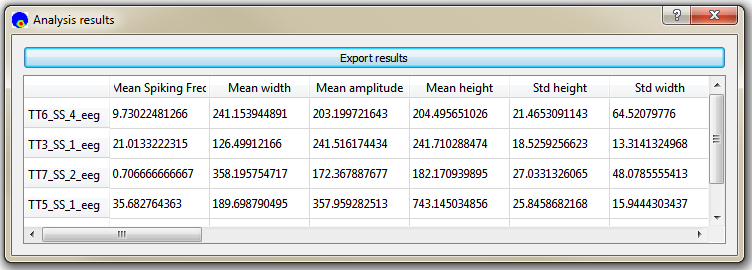
Pressing ‘Save Log’ button prompts the user to save the texts at the log box. This will export the log texts as plain ASCII texts and, therefore, there will be no colour in the output text file.

**Clear Log**

This button simply clears the log record in the log box. It is recommended that the log box is cleared at intervals so that errors are easier to find when the logs are exported

**Start**

Pressing this button starts reading the data from the specified files, and running analyses on them. As long as the data analysis keeps going, this button remains disabled to avoid unnecessary interruption of the execution. It is enabled again after the completion of the analysis or if there is a fatal error which stops the execution. At the end of the analysis of each unit, log box shows the full file directory of the output graphics which are saved in native folder of the spike data. If the selected functions have numeric results, a table appears after successful execution of all analysis functions. A sample table is shown in **Figure 5**. Clicking on the ‘Export results’ prompts the user to save the tabular data in Excel file (**Figure 5, Bottom**).



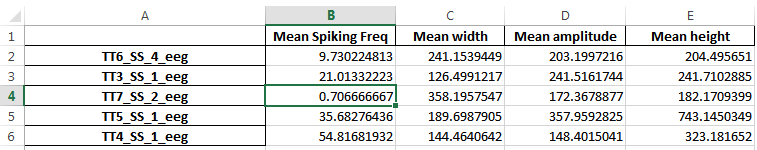


Figure 5: Sample of parametric results of NeuroChaT analyses in a tabular form. The table pops up once the analyses are complete. Bottom row shows how it looks like once it is exported to an Excel file.

**Menu items**

Current version of NC has 4 main menus items: File, Settings, Results & Help.

**File**

File menu consists of *Open*, *Save Session*, *Load Session* and *Exit* options as shown in **Figure 6** below.

**a. Open:** This acts exactly as browse button described before. It will prompt user to give appropriate input depending on the analysis mode selected. If no file is selected, it generates a warning.

**b. Save Session:** This will allow the user to save the configuration of NeuroChaT in *.ncfg* file (i.e. input and output format, analysis mode, cell no, file and directory information from user input, the state of the functions that are selected for analysis, and the parameters used for the analyses). If aborted, it creates a warning in the log box.

**c. Load Session:** This option will take the user to load configuration (.ncfg) file. If no file is selected, it will show a warning in the log box.

**d. Exit:** Terminates the NeuroChaT software.

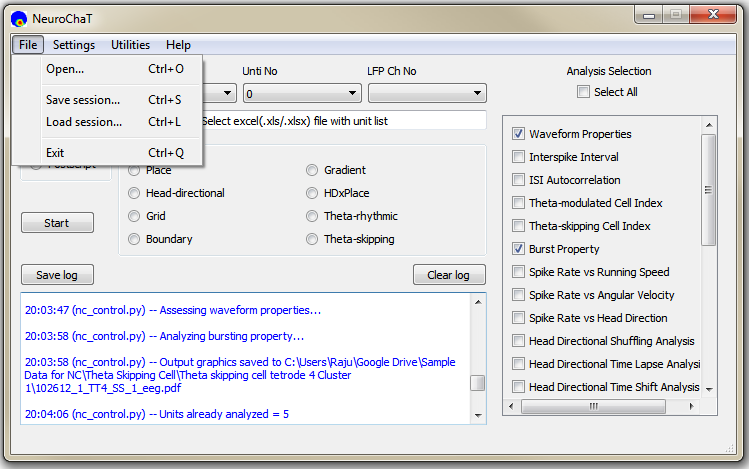
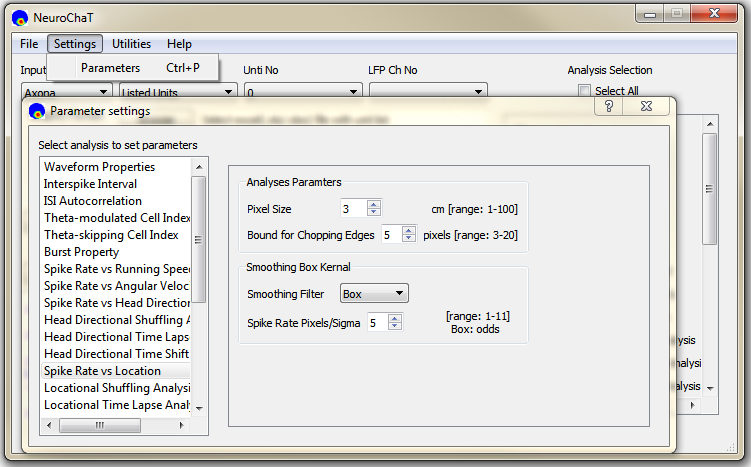


Figure 6: 'File' menu items. Shortcuts for invoking each of these actions are also shown.

**Settings**

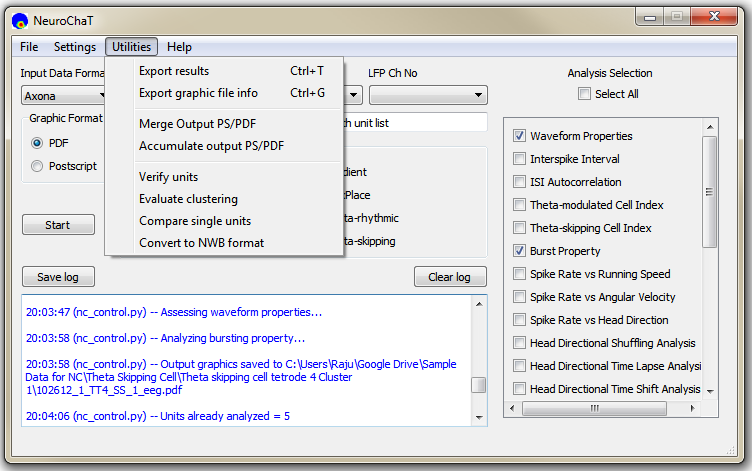
Settings menu has one item: *Parameters*. Clicking this item initiates the parameter selection box as shown in **Figure 7**. Analysis specific parameters and their accepted values are listed in **Appendix D**. Clicking on an item on the left panel will display the parameter setting panel for that analysis on the right panel.



**Figure 7:** Window for analysis specific input parameter settings.

**Utilities**

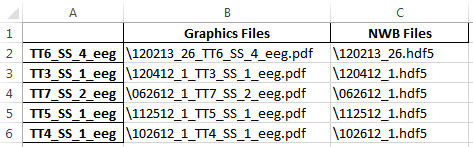
This menu comprise of useful NC utilities described below (**Figure 8**):



**Figure 8:** Items in the 'Utilities' menu.

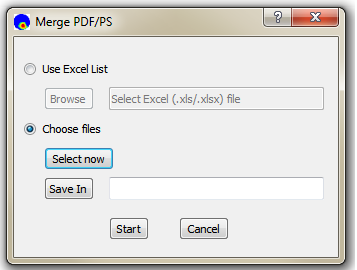
**a. Export results:** This action prompts the user to save recent analysis results in Excel file.

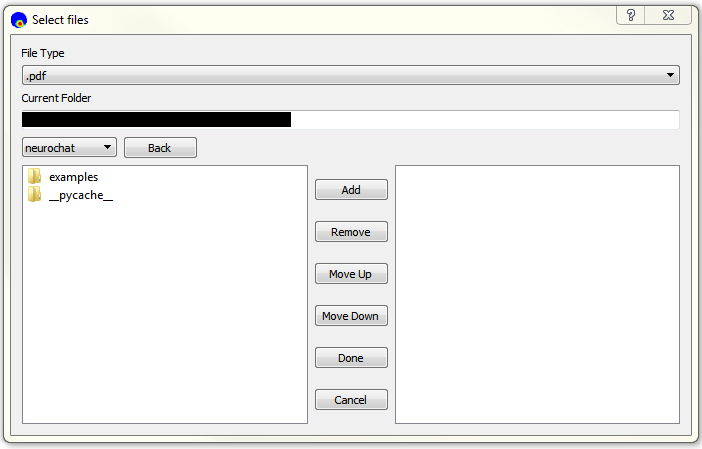
**b. Export graphics file info:** The user can export the directory and name of the output PDF/PS files containing analysis graphics and the HDF5 files associated to the data in an Excel file. The sample output looks as follows:



**Figure 9:** Sample Excel file from the export of graphics file information.

**c. Merge output PDF/PS:** This utility allows merging multiple PDF/PS files into a single PDF or PS file. The utility works in two ways, as shown on the top row of **Figure 10**: 1. By uploading an Excel list of the PDF/PS file names (full name, including directory) using the ‘Use Excel List’ option, 2. By picking the files manually using ‘Choose files’ option. ‘Select now’ button will be activated, and clicking on this will display a file-picking utility. The ‘Save In’ button opens a save file dialogue for the user to select the file. The utility executes upon pressing the ‘Start’ at the bottom of its window. Origin files remain intact.

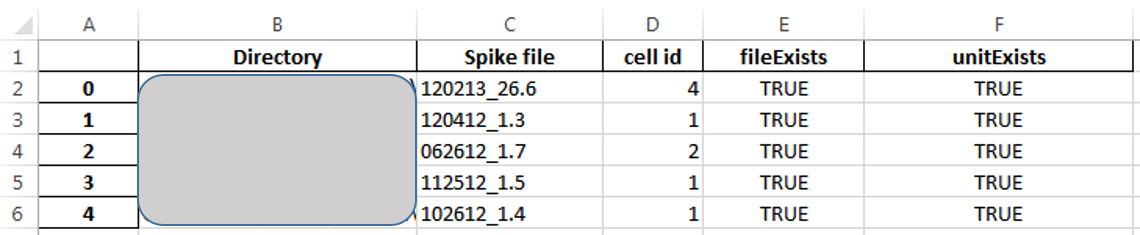




**Figure 10:** Upper row: Merge file utility showing the options of 'Use Excel List' or ‘Choose Files’. Bottom Row: The file-picker for selecting files to merge

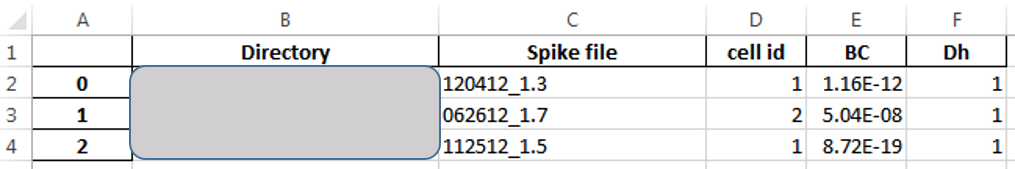
**d. Accumulate output PDF/PS:** This utility help accumulating the analysed output graphics files into a single folder. This works in the same way as merging files described above i.e., takes Excel file input or allows manual pick up. Specified files are then accumulated in the folder selected by ‘Save In’ button (**Figure 10, bottom**). This utility executes when the ‘Start’ button, located at the bottom of the window, is pressed. PDF/PS files being accumulated are not deleted from their original location.

**e. Verify units:** Clicking this item asks the user to upload an Excel file that contains the name of the directory, spike file, and the unit number that the user is verifying as shown by the sample in **Figure 11**. Output of this analysis adds the last two columns that shows if the file and the unit in that file exists.



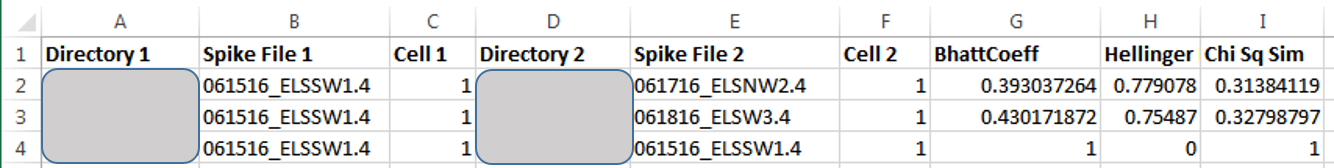
**Figure 11**: Sample input format and output in 'Verify units' utility.

**f. Evaluate clustering:** This option evaluates the quality of clustering by measuring Bhattacharyya coefficient and Hellinger distance between the clusters of a recording session. The clusters are formed using peaks, troughs and two principle components in each electrode channels. Sample input and results are shown in **Figure 12**.



**Figure 12:** Sample Excel file input for cluster evaluation. The analysis output are added at the tail of each row of data.

**g. Compare single units:** This option allows the user to compare the units in two different recordings. The analysis takes the clusters, formed as that of cluster evaluation, from two sessions, and compares their similarity by measuring their degree of overlap (Bhattacharyya coefficient and χ2-similarity) or their statistical distance (Hellinger distance). Sample input style and results are shown in **Figure 13**.



**Figure 13:** Sample Excel file input for cluster similarity measurement. Two sets of specifications are required for comparison. The analysis output are added at the tails of each set of specifications of the comparing units.

**h. Convert to NWB formats:** This option takes a list of file information in Excel format, as shown in **Figure 14**, and converts the data into HDF5 (NWB) format. The sample is for Axona recordings. Full filename (without directory) should be written in LFP data specification columns.

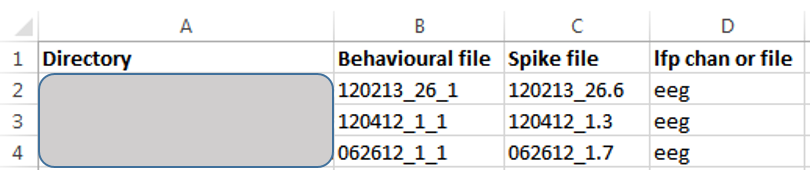


Figure 14 Sample Excel input for converting the recorded data to HDF5 format.

In all of these Excel based manipulations, columns names are not strictly defined, but the order of information is strictly followed. The first row is always considered to specify the header for the Excel data.

## API Use Guide

This API use guide produces results reported in **Chapter 5**.

In addition to the codes for verifying the place cell, head-directional cell and analyses of rhythmic units, it also shows examples of other useful methods that can be harnessed for creating simple and efficient analysis scripts and data management.

Please refer to the code-documentation for the description of each module, their classes and functions, and methods in each class.

In addition to the example units, this guide shows uses of NeuroChaT and its components in many different ways.

**Step-1: Download NeuroChaT package from** [**OSF**](https://osf.io/kqz8b) **or** [**GitHub**](https://github.com/mnislamraju/NeuroChaT)

NeuroChaT can be used without any burden of installation. You can download a local copy of NeuroChaT codes and insert the path to your system.

**Step-2 Insert NeuroChaT path to $PYTHONPATH**

import sys  
sys.path.insert(1, 'path\to\neurochat')

**Step-3 Import modules and classes**

We are importing only NSpike and NSpatial for the moment. We will add and import NLfp data for analyses that require LFP signals. nc\_plot is the module that provides with plotting functions

from neurochat.nc\_data import NData  
from neurochat.nc\_spike import NSpike  
from neurochat.nc\_spatial import NSpatial  
import neurochat.nc\_plot as nc\_plot

**Step-4 Instantiate objects**

The names C0 and S0 for for the unit and the spatial data are arbitrary

spike= NSpike(system = 'Axona')  
spike.set\_name('C0')  
  
spat= NSpatial(system = 'Axona')  
spat.set\_name('S0')

**Step-5 Add names for the data files**

data\_dir= '\full\file\directory\of\place cell\recorded\Axona\data\'  
  
spat.set\_filename(data\_dir + '040513\_1\_1.txt')  
spike.set\_filename(data\_dir + '040513\_1.6')

**For HDF5 files,**

Path of the data should also be added following a ‘+’ sign. The system argument should be changed or could be set at NSpatial(system= ‘Axona’)

spat.set\_system('NWB')  
spike.set\_system('NWB')  
  
data\_dir= '\full\file\directory\of\place\cell\HDF5\data\'  
spat.set\_filename(data\_dir + '040513\_1.hdf5+/processing/Behavioural/Position')  
spike.set\_filename(data\_dir + '040513\_1.hdf5+/processing/Shank/6')

**Step-6 Load spatial and spike data. Set the unit number**

spat.load()  
spike.load()  
  
spike.set\_unit\_no(3)

**Step-7 Instantiate NData object. Add individual data objects to NData object.**

ndata= NData()  
ndata.spike= spike  
ndata.spatial= spat

The data format, filenames for individual datasets can be set using ndata

ndata.set\_data\_format(data\_format = 'NWB')  
  
ndata.set\_spatial\_file(data\_dir + '040513\_1.hdf5+/processing/Behavioural/Position')  
ndata.set\_spike\_file(data\_dir + '040513\_1.hdf5+/processing/Shank/6')

They can be loaded using ndata

ndata.load()

Or, individually

ndata.load\_spatial()  
ndata.load\_spike()

And the unit number can be set as well

ndata.set\_unit\_no(3)

**Step-8 Perform analysis of interest**

**Analysis of place cell**

**Place cell firing map by using ndata:**

Pixel size is set 3cm. A 5x5 box filter is used for smoothing the firing map

placeData= ndata.place(pixel = 3, filter = ['b', 5])

**Similar results can be obtained by passing timestamps of the spiking unit to the spat.place() method**

NData object performs the job of connecting these two objects and simplifies the analysis

placeData= spat.place(spike.get\_unit\_stamp(), pixel = 3, filter = ['b', 5])

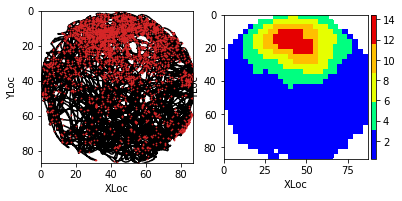
**Plotting relevant data**

Refer to neurochat.nc\_plot.py module for more plotting functions.

*Following command is used for inline display of graphics in Notebook*

%matplotlib inline

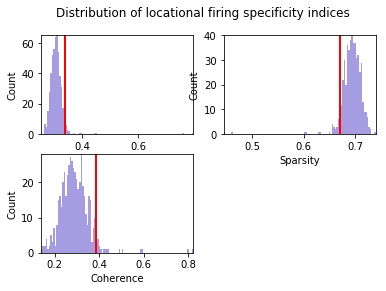
fig= nc\_plot.loc\_firing(placeData)



**Analysis and plotting of locational shuffling analysis using default parameters**

The spike timestamps are shuffled for 500 times. Pixel size is 3 cm. limit=0 implies that the spikes timestamps are randomly shuffled in (-duration, +duration) range

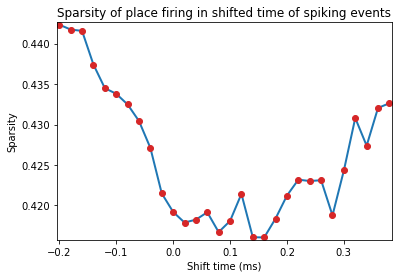
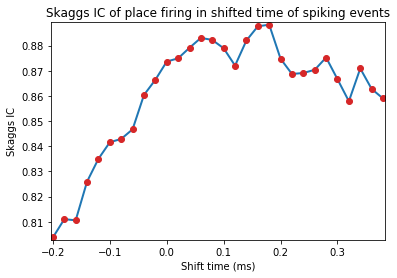
pshuffleData= ndata.loc\_shuffle(nshuff = 500, limit = 0, pixel = 3)  
fig= nc\_plot.loc\_shuffle(pshuffleData)

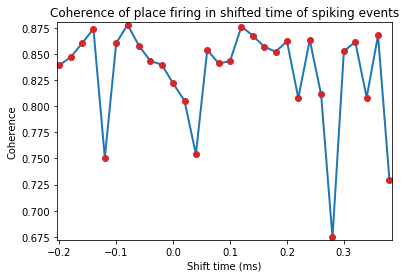


**Analysis and plotting of locational shifting analysis using shifting index from -10 to +20**

Spike timestamps are gradually shifted from -10 to +20 units of spatial time-resolution. If the video for tracking animal behviour is sampled at 50Hz, this means the spike-train is shifted from -200ms to +400ms

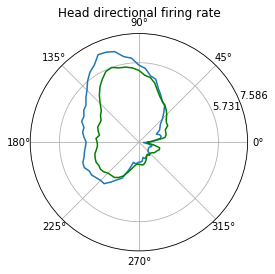
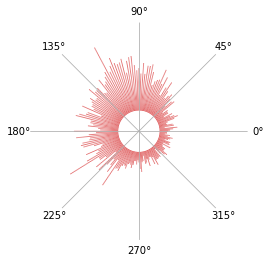
import numpy as np # numpy imported for the use of np.range  
pshiftData= ndata.loc\_shift(shift\_ind = np.arange(-10, 20))  
fig= nc\_plot.loc\_time\_shift(pshiftData)





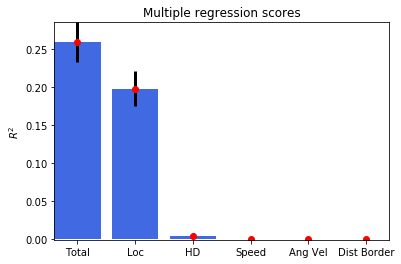
**Head directional analysis of this unit**

hdData= ndata.hd\_rate()  
fig= nc\_plot.hd\_firing(hdData)



**Multiple regression analysis and plotting**

regressData= ndata.multiple\_regression()  
fig= nc\_plot.multiple\_regression(regressData)



**If the data files are in Axona or Neuralynx format, they can be exported to HDF5 file**

ndata.save\_to\_hdf5()

**Datasets can be saved individually as well**

spike.save\_to\_hdf5()  
spat.save\_to\_hdf5()

**Parametric results of all the analysis performed can be obtained by**

results= ndata.get\_results() # Returns the results in OrderedDict  
print(results)

**Results from individual data objects can also be retrieved similarly**

spike\_results = spike.get\_results()  
spat\_results = spat.get\_results()

**Analysis of head-directional cell**

Change data filename/paths for the new unit similar to what was done for the place cell information Load new data and set the unit number. No need to reassign to ndata, as Python assignments are by reference, not by value.

ndata.set\_data\_format('NWB')  
  
data\_dir= \full\file\directory\of\head\directional\HDF5\data\'  
spat.set\_filename(data\_dir + '120412\_1.hdf5+/processing/Behavioural/Position')  
spike.set\_filename(data\_dir + '120412\_1.hdf5+/processing/Shank/3')  
  
  
spat.load()  
spike.load()  
  
spike.set\_unit\_no(1)

Reset results to omit parametric output of previously analysed unit. This can be done before loading the new datasets or at any stage of the analysis.

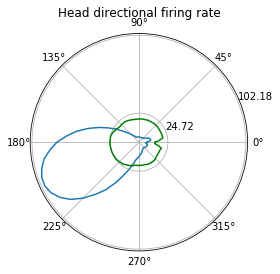
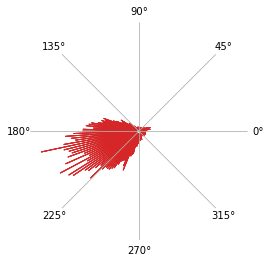
ndata.reset\_results()

Or, results can be reset using individual data objects

spat.reset\_results()  
spike.reset\_results()

**Head-directional firing rate analysis and plot**

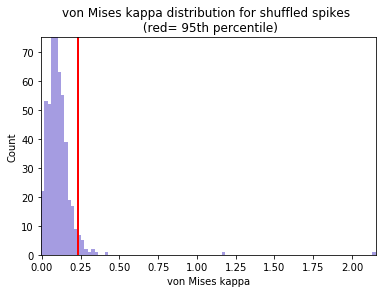
hdData= ndata.hd\_rate()  
fig= nc\_plot.hd\_firing(hdData)



**Head directional shuffling analysis and plot**

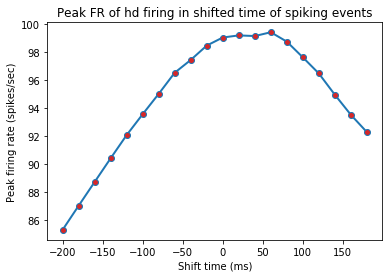
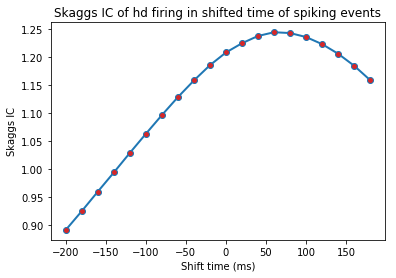
Number of bins for the histogram of the shuffled data is set to 100

hshuffleData= ndata.hd\_shuffle(nshuff = 500, limit=0, bins= 100)  
fig= nc\_plot.hd\_shuffle(hshuffleData)



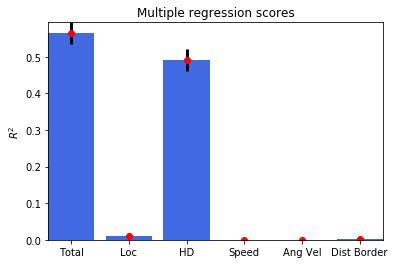
**Head directional time-shift analysis**

hshiftData= ndata.hd\_shift(shift\_ind=np.arange(-10, 10))  
fig= nc\_plot.hd\_time\_shift(hshiftData)



**Head directional multiple regression**

regressData= ndata.multiple\_regression()  
fig= nc\_plot.multiple\_regression(regressData)



**Analysis of spike-train dynamics**

**Changing the data filename/paths for the new unit**

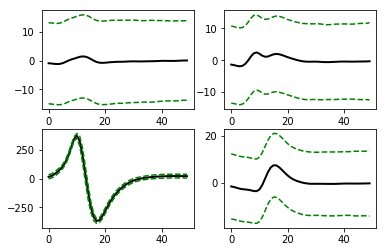
data\_dir= \full\file\directory\of\HDF5\data\'  
spat.set\_filename(data\_dir + '112512\_1.hdf5+/processing/Behavioural/Position')  
spike.set\_filename(data\_dir + '112512\_1.hdf5+/processing/Shank/5')  
  
spat.load()  
spike.load()  
  
spike.set\_unit\_no(1)

Reset results to omit parametric output of previously analysed unit

ndata.reset\_results()

**Waveform properties of the unit**

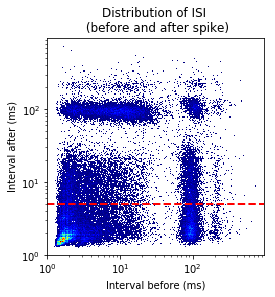
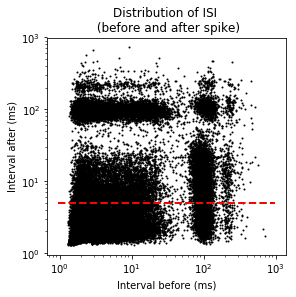
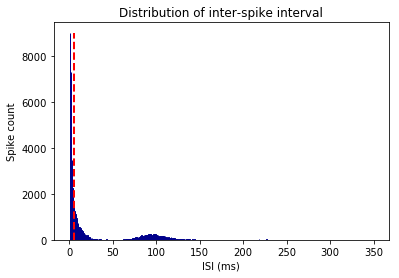
graphData= ndata.wave\_property()  
fig= nc\_plot.wave\_property(graphData, [int (spike.get\_total\_channels()/2), 2])



**Inter-spike interval (ISI) histogram**

The number of bins for histogram is 350, and the maximum ISI to bin for is 350ms. This implies each bin represents 1msec interval. ‘graphData’ term will be used repetedly from now on for reusing the memory

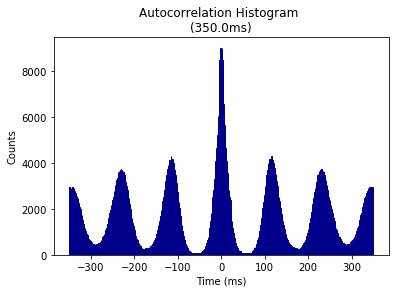
graphData= ndata.isi(bins = 350, bound = [0, 350])  
fig= nc\_plot.isi(graphData)



**ISI autocorrelation histogram for longer length**

Binsize is 1msec, and autocrrelation is performed from -350ms to +350ms

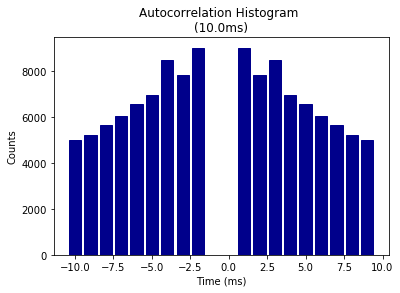
graphData= ndata.isi\_corr(bins = 1, bound = [-350, 350])  
fig= nc\_plot.isi\_corr(graphData)



**ISI autocorrelation histogram for shorter length**

Binsize is 1msec, and autocrrelation is performed from -10ms to +10ms

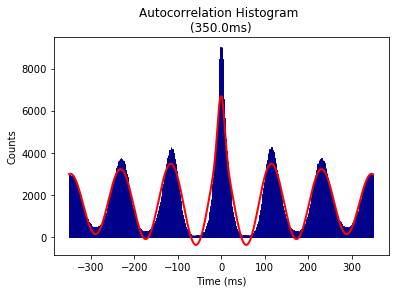
graphData= ndata.isi\_corr(bins = 1, bound = [-10, 10])  
fig= nc\_plot.isi\_corr(graphData)



**Theta modulation Index analysis**

Input paramteres are for [Frequency, tau1, tau2] and provides the starting value, lower, and upper bound for the fitted sinusoidal equation. Binsize and remporal bound are that of ISI autocorrelation histogram

graphData= ndata.theta\_index( start = [6, 0.1, 0.05], \  
 lower = [4, 0, 0], \  
 upper = [14, 5, 0.1], \  
 bins = 1, bound = [-350, 350])  
fig= nc\_plot.theta\_cell(graphData)



Above analyses can also be done using the spike data itself as it does not require information from other data object. For example,

graphData= spike.isi(bins = 350, bound = [0, 350])  
fig= nc\_plot.isi(graphData)

**Analysis of rhythmicity of LFP and spike-to-LFP phase relationships**

**Import NLfp class**

from neurochat.nc\_lfp import NLfp

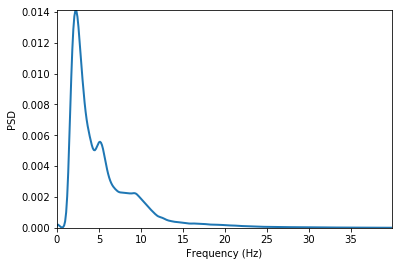
**Instatiate LFP data object, set the filename, load data, and add to ndata**

lfp= NLfp(system= 'NWB')  
  
lfp.set\_filename(data\_dir+ '\\112512\_1.hdf5+/processing/Neural Continuous/LFP/eeg')  
  
lfp.load()  
  
ndata.lfp= lfp

**LFP frequency spectrum analysis**

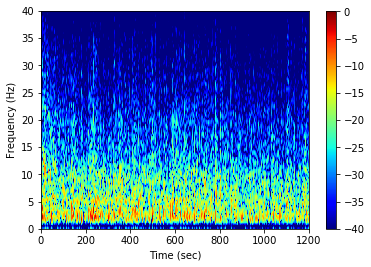
Hanning window of 2sec with 1sec overlap and number of FFT components= 2048. ptype is ‘psd’ which means power-spectral density. Other option can be ‘power’. prefilt set ‘True’ for pre-filtering the LFP signal with a bandpass filter as set by filtset. filtset= [filter order, lower cutoff frequency, higher cutoff frequency, type of filtering]. fmax defines the maximum frequency to analyse. db set to ‘True’ will convert the spectogram in decibel unit. tr set to ‘True’ creates a time-resolved spectogram with ‘window’-resolution and ‘overlap’ amount of signal overlap. tr set to ‘False’ calculates the spectogram using Welch’s method. This function can also be similarly called as ndata.spectrum()

graphData= lfp.spectrum(window = 2, noverlap = 1, nfft = 2048, ptype = 'psd', \  
 prefilt = True, filtset = [10, 1.5, 40, 'bandpass'], \  
 fmax = 40, db = False, tr = False)  
fig= nc\_plot.lfp\_spectrum(graphData)



After setting tr as True and db = True

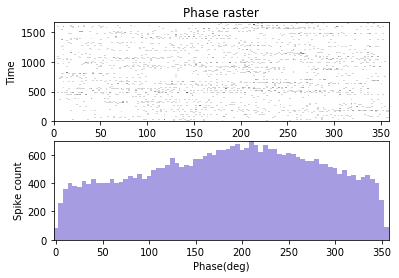
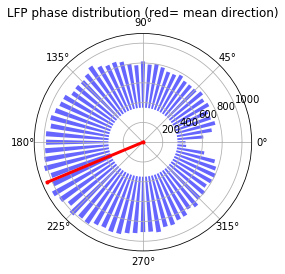
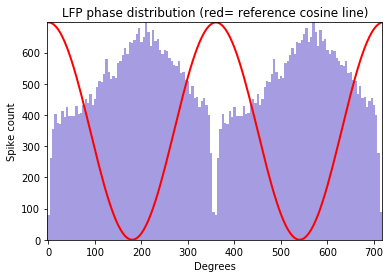
graphData= lfp.spectrum(window = 2, noverlap = 1, nfft = 2048, ptype = 'psd', \  
 prefilt = True, filtset = [10, 1.5, 40, 'bandpass'], \  
 fmax = 40, db = True, tr = True)  
fig= nc\_plot.lfp\_spectrum\_tr(graphData)



**Spike-LFP phase distribution**

fwin= [6,12] means that the phase of the spike are sought in the LFP band of 6Hz to 12 Hz. The minimum power of this band to be accepted to carry significant theta is 0.2 times the total LFP power, and that of the amplitude of the band signal is 0.15 times the amplitude of the LFP signal. The LFP signal is prefiltered using the filtset parameters.

graphData= ndata.phase\_dist(binsize = 5, rbinsize = 2, fwin = [6, 12],\  
 pratio = 0.1, aratio = 0.15, filtset = [10, 1.5, 40, 'bandpass'])  
fig= nc\_plot.spike\_phase(graphData)



**The analysis can be performed from both the NLfp() and NSpike() objects**

Using the lfp object:

graphData= lfp.phase\_dist(spike.get\_unit\_stamp(), binsize = 5, rbinsize = 2, fwin = [6, 12],\  
 pratio = 0.1, aratio = 0.15, filtset = [10, 1.5, 40, 'bandpass'])  
fig= nc\_plot.spike\_phase(graphData)

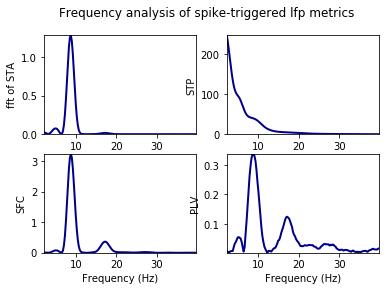
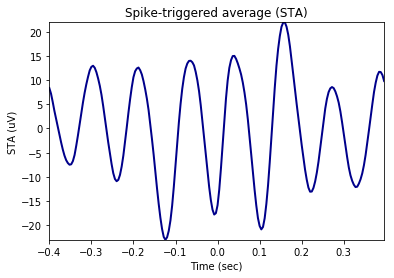
Using the spike object:

graphData= spike.phase\_dist(lfp = lfp , binsize = 5, rbinsize = 2, fwin = [6, 12],\  
 pratio = 0.1, aratio = 0.15, filtset = [10, 1.5, 40, 'bandpass'])  
fig= nc\_plot.spike\_phase(graphData)

**Analysis of phase-locking value (PLV), spike-field coherence (SFC), and spike-triggerd average (STA)**

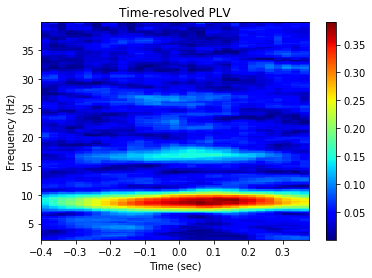
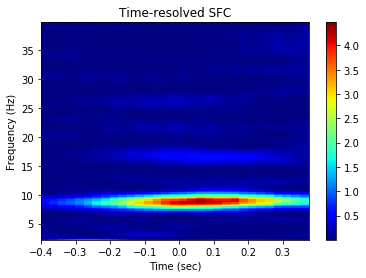
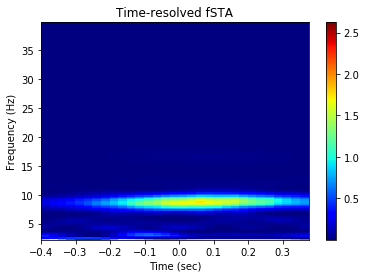
Window of the LFP chunks in reference to the spike timestamps is set to -400ms to +400ms Frequency of interest for the analysis is set as 2Hz to 30Hz

graphData= ndata.plv(window = [-0.4, 0.4], fwin = [2, 40])  
fig= nc\_plot.plv(graphData)



*Time-resolved* as set by mode= ‘tr’. nsample implies number of randomly selected spikes around which the LFP signals are cut for phase-locking analysis slide gives the time in ms by which the window is shifted from left to right to obtain the time-resolved phase-locking analysis

graphData= ndata.plv(window = [-0.4, 0.4], nfft = 1024, mode = 'tr', nsample = 2000, slide = 25, fwin = [2, 40])  
fig= nc\_plot.plv\_tr(graphData)



In most of the cases where composite information are required and ndata is not used, the spike timestamp is provided as the first argument to the methods followed by other information. Because, in such cases only information required by the analysis from the spike object is the timestamps of individual spikes in the train. For example,

graphData= ndata.plv(window = [-0.4, 0.4], fwin = [2, 40])  
fig= nc\_plot.plv(graphData)

gives the same result as the codes given below:

graphData= lfp.plv(spike.get\_unit\_stamp(), window = [-0.4, 0.4], fwin = [2, 30])  
fig= nc\_plot.plv(graphData)

**Use of Nhdf class**

**Import and instantiate Nhdf class**

from neurochat.nc\_hdf import Nhdf  
hdf= Nhdf()

**Store data using Nhdf object**

Nhdf() resolves the filename and the path for storage of the data using Nhdf().resolve\_pathname(data=data\_obj) where data\_obj can be a NSpatial(), NSpike() of NLfp() object

hdf.save\_spatial(spat)  
hdf.save\_spike(spike)  
hdf.save\_lfp(lfp)

**This can also be done using**

hdf.save\_object(obj = spat)  
hdf.save\_object(obj = spike)  
hdf.save\_object(obj = lfp)

**Graphical data from indiviudal analysis can be stored using the following codes**

*path* is the path inside HDF5 file. Analysis data are always recommended to store in the */analysis/* path. But analysis for each unit+lfp pair is stored in one path under which graphical data from individual analyses are store. The unique unit ID is established using the name resolving method Nhdf().resolve\_analysis\_path() which utilizes the filename of the recorded data, electrode/tetrode number, eeg channel ID and the unit number. *name* is the name of the analysis following the unit ID, i.e. ‘plv’ etc. *graph\_data* are the dictionary data that are plotted using the functions iin nc\_plot

unit\_id= hdf.resolve\_analysis\_path(spike = spike, lfp = lfp)  
  
hdf\_name= hdf.resolve\_hdfname(data=spike) # Resolve HDF5 filename  
hdf.set\_filename(hdf\_name) # NeuoChaT opens the file as file-object as soon as new filename is set.  
  
print(unit\_id)  
  
hdf.save\_dict\_recursive(path ='/analysis/' + unit\_id+ '/' ,  
 name = 'plv', data = graphData)

**Analysis results can be stored by**

results= ndata.get\_results()  
hdf.save\_dict\_recursive(path ='/analysis/' + unit\_id+ '/' ,  
 name = 'results', data = results)

**Apart from that data and attributes to any group or dataset can be added using**

Set *create\_group* to ‘True’ it will create the path if does not already exist

hdf.save\_dataset(path = '/path/to/group/', name = 'name\_of\_dataset', data = date\_to\_store, create\_group = True)  
hdf.save\_attributes(path= '/path/to/group/or/dataset/', attr = dict\_of\_attributes)

**Use of NeuroChaT class**

**Import NeuroChaT class and instantiate**

from neurochat.nc\_control import NeuroChaT  
nc= NeuroChaT()

**Convert files in Axona format to NWB files specified in an Excel list**

excel\_file= '\full\file\name\of\Excel\list.xlsx'  
nc.convert\_to\_nwb(excel\_file)

**Verify units provided in an Excel list before batch-mode analysis**

excel\_file= '\full\file\name\of\Excel\list.xlsx.xlsx'  
nc.verify\_units(excel\_file)

**Evaluate the quality of clustering from a list provided in an Excel file**

excel\_file= '\full\file\name\of\Excel\list.xlsx.xlsx'  
nc.cluster\_evaluate(excel\_file)

**Evaluate similarity of clusters**

The excel list contains paired list of units to be compared for similarity

excel\_file= 'C:\\Users\\Raju\\Google Drive\\Sample Data for NC\\Comparison results\_from NeuroChaT\_pawels\_data.xlsx'  
nc.cluster\_evaluate(excel\_file)

**Analysis using NeuroChaT**

Analysis using NeuroChaT class is always done with the help of Configuration class where the user specifies all the data, intended analyses, input parameters etc.

**Configuration class**

Import, instantiate, set the filename and load configuration from the file. This class uses nc\_defaults.py module for importing deafult analyses and parameters.

from neurochat.nc\_config import Configuration  
  
config= Configuration()  
  
config.set\_config\_file('\full\file\name\of\grid\_config.ncfg')  
  
config.load\_config()

Set configuration to NeuroChaT object

nc.set\_configuration(config)

Start analysis. This will ‘read’ the instructions from the config object and execute accordingly

nc.start()

Use *get\_* and *set\_* functions also known as getters and setters for accessing and setting values of interest. For example, \* Getting and setting parameters:

param\_list= config.get\_param\_list() # List of all parameters as dictionary keys  
  
params\_by\_analysis= config.get\_params\_by\_analysis(analysis= 'isi')  
print(params\_by\_analysis)  
  
param\_val= config.get\_params(name = 'isi\_length') # name is the list of parameters or the name of a single parameter'  
print(param\_val)  
  
config.set\_param(name = 'isi\_bin', value = 2)

**Getting and setting analyses**

list\_of\_analyses= config.get\_analysis\_list() # List of all analysis  
print(list\_of\_analyses)  
  
analysis\_checked= config.get\_analysis(name = 'isi') # If 'True', analysis is set to be done  
print(analysis\_checked)  
  
config.set\_analysis(name = 'theta\_skip\_cell', value = False) # Analysis of theta skippin cell turned off

Analyses can be performed in different modes, namely: 1. ‘Single Unit’-one cell at time, value ‘0’ 2. ‘Single Session’- all the cells in one recording at a time, value ‘1’ 3. ‘Listed Units’- all the cells listed in one Excel file, value ‘2’

**Getting and setting analysis mode**

print(config.get\_analysis\_mode())  
  
config.set\_analysis\_mode(analysis\_mode = 'Single Unit') # Can also set analysis\_mode = 0

What type of data file need to be specified depends on the type of mode and the format of the data Please refer to the Configuration class for more such methods. Here, we show an example of settingh Axona data and an example of batch mode analysis

**Specifying Axona files for analyses**

data\_dir= '\path\to\recorded\Axona\data\'  
  
config.set\_analysis\_mode(0) # For 'Single Unit' analysis  
  
config.set\_spatial\_file(spatial\_file = data\_dir+ '040513\_1\_1.txt')  
config.set\_spike\_file(spike\_file = data\_dir + '040513\_1.6')  
  
config.set\_unit\_no(3)

We are interested in only certain anlyses. So, we first turn off all the analyses:

config.set\_analysis(name = 'all', value = False) # 'all' for setting all the analyses

**Specify new analyses**

config.set\_analysis(name = ['loc\_rate', 'loc\_shuffle', 'loc\_time\_lapse'], value = True) # See nc\_defaults for names of the analyses

Let us use default parameters for ease of understanding. NeuroChaT() always saves the graphics in a file. Let us set the file in ‘PDF’ or ‘pdf’ format. Other option is ‘Postscript’ or ‘ps’

config.set\_graphic\_format(graphic\_format = 'PDF')

**Set this configuration for NeuroChaT’s use**

nc.set\_configuration(config)

Save this configuration to a file for future use. This file can be edited using any standard text-editing software

config.save\_config('\full\file\name\of\place\_config.ncfg')

Once the configuration file is set to NeuroChaT object, all of its methods can be uses by NeuroChaT itself. For example, the configuration can be loaded from and saved to file using the NeuroChaT object. It works this way- if NeuroChaT cannot find a method within itself, it at first searches in the Configuration object. If not found, it looks into composing object NData() for the function. This process is call delegation. The precedence for delegation is Configuration() > NData()

nc.set\_config\_file('\full\file\name\of\place\_config.ncfg')  
nc.load\_config()  
  
nc.set\_analysis\_mode(0) # Analysis mode set to 'Single Unit' in Configuration object

Once the anayses are done, NeuroChaT saves the pdf in respective data folder It always stores the NWB-converted file if the latter does not exist and stores the graphics data and the parametric results in the files. Along with that, parameteric results and names of output PDF and NWB files can be obtained by using following codes which return them in Pandas DataFrame.

results\_df= nc.get\_results()  
print(results\_df)  
output\_filename\_df= nc.get\_output\_files()  
print(output\_filename\_df)

TT6\_SS\_4\_eeg 9.730225 23.065766 21.465309 241.153945

Mean amplitude Std width Mean height Theta Index

TT6\_SS\_4\_eeg 203.199722 64.520798 204.495651 0.714889

TI fit freq Hz TI fit tau1 sec ... Mult Rsq

TT6\_SS\_4\_eeg 8.808084 0.229588 ... 0.222366

Semi Rsq Loc Semi Rsq HD Semi Rsq Speed Semi Rsq Ang Vel

TT6\_SS\_4\_eeg 0.15583 0.002322 0.03563 0.001138

Semi Rsq Dist Border DR HP DR SP DR AP DR BP

TT6\_SS\_4\_eeg 0.001403 0.085843 0.340246 0.190116 0.159364

[1 rows x 88 columns]

Graphics Files

TT6\_SS\_4\_eeg C:UsersRajuGoogle DriveSample Data for NC...

NWB Files

TT6\_SS\_4\_eeg C:UsersRajuGoogle DriveSample Data for NC...

These files can be exported for future use using DataFrame’s io utilities:

import pandas as pd  
writer= pd.ExcelWriter('\full\file\path\to\parametric\_results.xlsx') # set-up writing engine  
results\_df.to\_excel(writer, 'Sheet1') # write to file  
output\_filename\_df.to\_excel(writer, 'Sheet2')

While the graphical interface provides an easier means for performing almost all of the abovementioned functionalities, NeuroChaT and its constituent classes works as the ‘engine’ behind those tasks.

**Use NClust class**

**Import and instantiate NClust**

Athough we are initializing it with already defined spike object, we could similarly set the filename and unit and load the composing spike object as we do for any other spike object itself NClust also performs some of the analysis that spike object does, i.e. analysing waveform properties, ISI histogram, PSTH etc. See nc\_clust.py module to learn more about this aspect.

from neurochat.nc\_clust import NClust  
clust= NClust(spike= spike)

This object is intended for facilitating analysis pertaining to clustering algorithm and cluster quality measurements. Following are some of the example methods:

**Remove null channels if any**

off\_chan= clust.remove\_null\_chan()

**Resample wave by intended factor**

wave, time= clust.resample\_wave(factor= 2) # Resampling factor is 2

**Align waves by peaks for better estimation of waveform features**

clust.align\_wave\_peak()  
aligned\_wave= clust.getWaveform()

**Get the channel with highest waveform energy, peak at the channel , and the index of the peak**

peak, peak\_chan, maxInd= clust.get\_max\_wave\_chan()

**Get the Principle Components of the waveforms**

pc= clust.get\_wave\_pc(npc = 2) # 2 PC in each channel  
print(pc)

**Get features for clustering**

feat= clust.get\_feat(npc = 2) # Consist of waveform peaks, troughs and 2 PC components in each channel

**Get fetures of clustered units**

unit\_feat= clust.get\_feat\_by\_unit(unit\_no = 3)

**Get waveforms by unit number**

waves= clust.get\_unit\_waves()

**Clustering quality evaluation**

If unit\_no set to 0 all units are evaluated with a matrix output for pairwise comparison. Otherwise, maximum Bhattacharyya distance (BC) and minimum Hellinger distance (Dh) for the specified unit are returned

bc, dh = clust.cluster\_separation(unit\_no = 0)

**Evaluationg unit similarity**

clust\_1 = NClust()  
clust\_1.load(filename = '\full\file\directory\of\spike\data\_1', system = 'NWB') # An alternative approach for loading spike data   
  
clust\_2 = NClust()  
clust\_2.load(filename = '\fullfile\directory\of\spike\data\_2', system = 'NWB') # An alternative approach for loading spike data   
  
bc, dh = clust\_1.cluster\_similarity(nclust= clust\_2, unit\_1= 3, unit\_2= 3) # unit\_1 and unit\_2 are the comparable units

## Input parameter description

#### Waveform properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO PARAMETER |  |  |  |  |
|  |  |  |

#### Inter-spike interval (ISI)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| isiBin | 2 | 1-100 | ms | Bin size of the ISI histogram |
| isiLength | 350 | 10-1000 | ms | Length of ISI histogram |
|  |  |  |  |  |
| isiLogBins | 70 | 10-100 |  |  |
| isiLogLength | 350 | 10-1000 | ms |  |

#### ISI Autocorrelation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| isiCorrBinSh | 1 | 1-10 | ms | Bin size of the ISI correlation histogram obtained on short lags |
| isiCorrLenSh | 10 | 5-50 | ms | Length of the ISI correlation histogram obtained on short lags |
|  |  |  |  |  |
| isiCorrBinLong | 2 | 1-50 | ms | Bin size of the ISI correlation histogram obtained on long lags |
| isiCorrLenL | 350 | 10-1000 | ms | Length of the ISI correlation histogram obtained on long lags |

#### Theta-modulated Cell Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| thetaCellFreqMin | 6 | 1-10 | Hz | Lower limit of the theta band frequency in curve fitting |
| thetaCellFreqMax | 12 | 8-16 | Hz | Upper limit of the theta band frequency in curve fitting |
| thetaCellFreqStart | 6 | 5-10 | Hz | Starting value of the theta band frequency in curve fitting |
| thetaCellTau1Max | 5 | 0.5-10 | sec | Upper limit of the decay constant τ1 decay constant in curve fitting |
| thetaCellTau1Start | 0.1 | 0-15 | sec | Starting value of the decay constant τ1 decay constant in curve fitting |
| thetaCellTau2Max | 0.05 | 0-0.1 | sec | Upper limit of the decay constant τ2 decay constant in curve fitting |
| thetaCellTau2Start | 0.05 | 0-0.1 | sec | Starting value of the decay constant τ2 decay constant in curve fitting |

#### Theta-skipping Cell Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO PARAMETER |  |  |  | Parameters from theta-modulated cell index are be used |
|  |  |  |

#### Burst Property

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| burstThresh | 5 | 1-15 | ms | Minimum ISI between consecutive spikes in a burst |
| spikesToBurst | 2 | 2-10 | ms | Minimum number of consecutive spikes with burstThresh for a burst |
| ibiThresh | 50 | 5-1000 | ms | Minimum inter-burst interval between two bursting groups of spikes |

#### Spike Rate vs Running Speed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| speedBin | 1 | 1-10 | cm/sec | Size of the speed bin for histogram |
| speedMin | 0 | 0-10 | cm/sec | Minimum acceptable speed to analyse |
| speedMax | 40 | 10-200 | cm/sec | Maximum limit on the speed to analyse |
|  |  |  |  |  |
| speedKernLen | 3 | 1-25, odds | samples | Length of moving-average smoothing kernel of recorded speed |
| speedRateKernLen | 3 | 1-7, odds | bins | Length of moving-average smoothing kernel of the spike rate |

#### Spike Rate vs Angular Velocity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| angVelBin | 10 | 1-50 | deg/sec | Size of the angular velocity bin for histogram |
| angVelMin | -200 | -500-0 | deg/sec | Minimum acceptable angular velocity to analyse |
| angVelMax | 200 | 0-500 | deg/sec | Maximum limit on the angular velocity to analyse |
| angVelCutoff | 10 | 0-100 | deg/sec |  |
|  |  |  |  |  |
| angVelKernLen | 3 | 1-25, odds | samples | Length of moving-average smoothing kernel of calculated angular velocity |
| angVelRateKernLen | 3 | 1-5, odds | bins | Length of moving-average smoothing kernel of the spike rate |

#### Spike Rate vs Head Direction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hdBin | 5 | factors of 360 | degree | Size of the head directional bin for histogram |
| hdAngVelCutoff | 30 | 0:5:100 | deg/sec | Lower limit of the acceptable angular velocity for avoiding noise from jerking of the head |
|  |  |  |  |  |
| hdRateKernLen | 5 | 1-11 | bins | Length of moving-average smoothing kernel of the spike rate |

#### Head Directional Shuffling Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hdShuffleTotal | 500 | 100-10000 |  | Number of shuffles for head directional shuffling analysis |
| hdShuffleLimit | 0 | 0:2:500 | sec | Upper limit of the shuffled shifted time of spikes |
| hdShuffleNoBins | 100 | 10:10:200 | bins | Number of bins for displaying the distribution of specificity measures |

#### Head Directional Time Lapse Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO PARAMETER |  |  |  |  |
|  |  |  |  |

#### Head Directional Time Shift Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hdShiftMax | 10 | 1-100 | indices | Maximum number of spatial samples by which spike-events are shifted forward |
| hdShiftMin | -10 | -1 to -100 | indices | Maximum number of spatial samples by which spike-events are shifted backward |
| hdShiftStep | 1 | 1-3 |  | Steps of samples spikes are shifted with |

#### Spike Rate vs Location

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| locPixelSIze | 3 | 1-100 | cm | Pixel size (bin size) for spatial firing 2D histogram |
| locChopBound | 5 | 3-20 | pixels | Upper limit on number of empty rows and columns which are chopped off from the firing rate map |
|  |  |  |  |  |
| locRateFilter | Box | Box/Gaussian |  | Type of smoothing kernel of the firing rate map |
| locRateKernLen | 5 | 1-11, odds | pixels/ no unit | Number of pixels for box filter and standard deviation for Gaussian filter |

#### Locational Shuffling Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| locShuffleTotal | 500 | 100-10000 |  | Number of shuffles for locational shuffling analysis |
| locShuffleLimit | 0 | 0:2:500 | sec | Upper limit of the shuffled shifted time of spikes |
| locShuffleNoBins | 100 | 10:10:200 | bins | Number of bins for displaying the distribution of specificity measures |

#### Locational Time Lapse Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO PARAMETER |  |  |  |  |
|  |  |  |  |

#### Locational Time Shift Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| locShiftMax | 10 | 1-100 | indices | Maximum number of spatial samples by which spike-events are shifted forward |
| locShiftMin | -10 | -1 to -100 | indices | Maximum number of spatial samples by which spike-events are shifted backward |
| locShiftStep | 1 | 1-3 |  | Steps of samples spikes are shifted with |

#### Spatial Autocorrelation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| spatialCorrMinObs | 20 | 1-100 | pixels | Minumum number of overlapping pixels between original and the shifted firing rate map |
|  |  |  |  |  |
| rotCorrBin | 3 | factors of 360 | degree | Steps of firing rate map rotation |
|  |  |  |  |  |
| spatialCorrFilter | Box | Box/Gaussian |  | Type of smoothing kernel of the firing rate map |
| spatialCorrKernLen | 5 | 1-11, odds |  | Number of pixels for box filter and standard deviation for Gaussian filter |

#### Grid Cell Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters from spatial autocorrelation are also used | | | | |
| gridAngTol | 2 | 1-5 | degree | Accepted differences between angular measures to consider them same |
| gridAngBin | 3 | Factors of 360 less than 45 | degree | Size of angular bins to measure rotational correlation |

#### Border Cell Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| borderFiringThresh | 0.1 | 0:0.05:1 |  | Minimum firing rate to maximum firing rate ratio to define as active pixels |
| borderAngBin | 3 | Factors of 360 less than 45 |  | Size of angular bin for circular-linear firing rate map |
| borderStairSteps | 5 | 4-10 |  | Number for steps for stair plot of border firing |

#### Gradient Cell Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| gradAsympLim | 0.25 | 0.1:0.05:1 |  | Range of asymptotic parameter 'a' on Goempertz function, e.g., ±0.25 |
| gradDisplaceLim | 0.25 | 0.1:0.05:1 |  | Range of displacement parameter 'b' on Goempertz function, e.g., ±0.25 |
| gradGrowthRateLim | 0.5 | 0.1:0.05:1 |  | Range of growth rate parameter 'c' on Goempertz function, e.g., ±0.5 |

#### Multiple Regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| multiRegInterval | 0.1 | 0.1:0.1:1 | sec | Interval of spatial samples for multiple regression |
| multiRegEpisode | 120 | 60:30:300 | sec | Duration for each replication of multiple regression |
| multiRegNoRep | 1000 | 100:100:2000 |  | Number of replications for multiple regression |

#### Interdependence Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO PARAMETER |  |  |  | Parameters from other analysis should be used |
|  |  |  |

#### LFP Frequency Spectrum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| lfpPreFiltLowCut | 1.5 | 0.1:0.1:4 | Hz | LFP prefiltering lower cutoff frequency |
| lfpPreFiltHighCut | 40 | 10:5:500 | Hz | LFP prefiltering higher cutoff frequency |
| lfpPreFiltOrder | 5 | 1-20 |  | LFP prefiltering butterworth filter order |
|  |  |  |  |  |
| lfpPwelchSegSize | 2 | 0.5:0.5:100 | sec | LFP segment size for Welch's method for spectral density estimation |
| lfpPwelchOverlap | 1 | 0.5:0.5:50 | sec | Overlap between LFP segments for Welch's method for spectral density estimation |
| lfpPwelchNfft | 1024 | 128:128:8192 |  | NFFT for Welch's method for spectral density estimation |
| lfpPwelchFreqMax | 40 | 10:5:500 | Hz | Maximum frequency to display |
|  |  |  |  |  |
| lfpStftSegSize | 2 | 0.5:0.5:100 | sec | LFP segment size for short-time Fourier transform |
| lfpStftOverlap | 1 | 0.5:0.5:50 | sec | Overlap between LFP segments for short-time Fourier transform |
| lfpStftNfft | 1024 | 128:128:8192 |  | NFFT for short-time Fourier transform |
| lfpStftFreqMax | 40 | 10:5:500 | Hz | Maximum frequency to display |

#### Unit LFP-Phase Distribution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| phaseFreqMin | 6 | 1-10 | Hz | Lower frequency of LFP band for analysis of phase locking |
| phaseFreqMax | 12 | 1-10 | Hz | Higher frequency of LFP band for analysis of phase locking |
| phasePowerThresh | 0.1 | 0:0.05:1 |  | Minimum band power to overall power of acceptable LFP segments |
| phaseAmpThresh | 0.15 | 0:0.05:1 |  | Minimum segment amplitude to overall amplitude of acceptable LFP segments |
|  |  |  |  |  |
| phaseBin | 5 | Factors of 360 | degree | Size of phase bins for circular histogram of spike-phases |
| phaseRasterBin | 2 | 1-15 | degree | Size of phase bins for raster of spike-phases |

#### Unit LFP-Phase Locking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| phaseLockWinLow | -0.4 | -1:0.05:-0.1 | sec | Lower limit of the LFP segments in reference to spike-events |
| phaseLockWinUp | 0.4 | 0.1:0.05:1 | sec | Upper limit of the LFP segments in reference to spike-events |
| phaseLockNfft | 1024 | 128:128:8192 |  | NFFT for Fourier transform |
| phaseLockFreqMax | 40 | 10:5:500 | Hz | Maximum frequency to analyse and display |

## Output parameter description

#### Waveform Properties

Mean Spiking Freq= Spiking frequency over the entire trial

Mean amplitude= mean of amplitude. Amplitude is measured by the difference between the first negative peak and first positive peak,

Std amplitude= Standard deviation of amplitude

Mean height= mean of height. Height is measured by the difference between the first positive peak and the overall minimum value of the spike

Std height= Standard deviation of height

Mean width= Mean of spike width taken at 25% of the amplitude

Std width= Standard deviation of the width

#### Inter-spike Interval (ISI)

No output parameter

#### ISI Autocorrelation

No output parameter

#### Theta-modulated Cell Index

Theta Index= Ratio of the sinusoid amplitude and the constant term in the model

TI fit freq Hz= Theta frequency from the model fitting in Hz

TI fit tau1 sec= Decay constant for the exponential modulation of the sinusoidal component in the model

TI adj Rsq= Goodness of fit of the model for the ISI autocorrelation data

TI Pearse R= Goodness of fit Pearson’s R between the original and model-fit values for the ISI autocorrelation data

TI Pearse P= Goodness of fit Pearson’s P between the original and model-fit values for the ISI autocorrelation data

#### Theta-skipping Cell Index

Theta Skip Index= Theta Skipping Index

TS jump factor= Ratio of the amplitude of the theta and delta band sinusoids

TS f1 freq Hz= Fitted frequency of the faster sinusoid in Hz

TS f2 freq Hz= Fitted frequency of the slower sinusoid,

TS freq ratio= Ratio of f1 to f2

TS tau1 Hz= Decay constant for the exponential modulation of the sinusoidal component in the model

TS adj Rsq= Goodness of fit of the model for the ISI autocorrelation data

TS Pearse R= Goodness of fit Pearson’s R between the original and model-fit values for the ISI autocorrelation data

TS Pearse P= Goodness of fit Pearson’s P between the original and model-fit values for the ISI autocorrelation data

#### Burst Property

Total burst= Total number of bursts calculated

Total bursting spikes= Total number of spikes constituting bursts

Mean bursting ISI ms= Mean inter-spike-interval for the bursting spikes only (ms)

Std bursting ISI ms = Standard deviation of inter-spike-interval for the bursting spikes only (ms)

Mean spikes per burst= Average number of spikes in bursts

Std spikes per burst= Standard deviation of the number of spikes in bursts

Mean burst duration= Mean of the duration of bursts in ms

Std burst duration= Standard deviation of the duration of bursts in ms

Mean duty cycle= Mean of the duty cycles. Duty cycle is the portion of the inter-burst interval during which the burst fires (burst duration/ inter-burst interval)

Std duty cycle= Standard deviation of the duty cycles

Mean IBI= Mean of inter-burst Intervals in ms

Std IBI= Standard deviation of inter-burst intervals in ms

Propensity to burst= Total bursting spikes/total spikes in the cluster

#### Spike Rate vs Running Speed

Speed Skaggs= Skaggs information content for speed vs spiking events in bits/sec

Speed Pears R= Goodness of fit Pearson’s R between spike firing rate at different speeds and fitted straight line

Speed Pears P= Goodness of fit Pearson’s P between spike firing rate at different speeds and fitted straight line

#### Spike Rate vs Angular Velocity

Ang Vel Left Pears R= Goodness of fit Pearson’s R between spike firing rate at different counter-clockwise (-ve) angular head velocity and fitted straight line

Ang Vel Left Pears P= Goodness of fit Pearson’s P between spike firing rate at different counter-clockwise (-ve) angular head velocity and fitted straight line

Ang Vel Right Pears R= Goodness of fit Pearson’s R between spike firing rate at different clockwise (+ve) angular head velocity and fitted straight line

Ang Vel Right Pears P= Goodness of fit Pearson’s P between spike firing rate at different clockwise (+ve) angular head velocity and fitted straight line

#### Spike Rate vs Head Direction

HD Skaggs= Head directional Skaggs information content

HD Rayl Z= Rayleigh Z for the head-directional firing rate

HD Rayl P= Rayleigh P for the head directional firing rate

HD von Mises K= von Mises concentration parameter κ

HD Mean= Vector mean or preferred head direction (degree) of the unit to fire

HD Mean Rate= Firing rate in preferred direction

HD Res Vect= Resultant vector length of head-directional firing rate

HD Peak Rate= Peak firing rate in the head-directional tuning curve

HD Peak= Head-direction at which peak firing rate occurs

HD Half Width= Width of the tuning curve measured at 50% of the peak firing rate(degree)

HD Peak CW= Peak firing direction during clockwise head-directional movement

HD Peak CCW= Peak firing direction during counter-clockwise head-directional movement

HD Peak Rate CW= Peak firing rate during clockwise head-directional movement

HD Peak Rate CCW= Peak firing direction during counter-clockwise head-directional movement

HD Delta= Separation angle between peak firing direction during clockwise and counter-clockwise head-directional movement

#### Head Directional Shuffling Analysis

HD Shuff Rayl Z Per 95= 95th percentile of the distribution of Rayleigh Z parameter for the head-directional tuning curves obtained from shuffling of spike-events

HD Shuff von Mises K Per 95= 95th percentile of the distribution of von Mises concentration parameter κ for the head-directional tuning curves obtained from shuffling of spike-events

#### Head Directional Time Lapse Analysis

No Parameter

#### Head Directional Time Shift Analysis

HD ATI= Anticipatory time interval for the head-directional cells, measured as the time-shift where the counter-/clockwise head-directions are same, or the separation angle becomes zero.

HD Opt Shift Skaggs= Time shift which maximizes the information content in HD tuning of spiking events

HD Opt Shift Peak Rate= Time shift which maximizes the peak firing rate in HD tuning of spiking events

#### Spike Rate vs Location

Spatial Skaggs= Information content of spatial firing map

Spatial Sparsity= The fraction of the environment in which the cell is active (max=1, min =0)

Spatial Coherence= Measure of orderliness of the local firing pattern (max=1. Min = -1). Or, simply the correlation between raw firing map, and smoothed firing map (value at each pixel is replaced the 8 neighbouring pixels of the non-smooth map). See Muller & Kubie 1989

#### Locational Shuffling Analysis

Loc Skaggs 95= 95th percentile of the distribution of Skaggs information content for the firing rate map obtained from shuffling of spike-events

Loc Sparsity 05= 95th percentile of the distribution of sparsity for the firing rate map obtained from shuffling of spike-events

Loc Coherence 95= 95th percentile of the distribution of coherence for the firing rate map obtained from shuffling of spike-events

#### Locational Time Shift Analysis

Loc Opt Shift Skaggs= Time shift which maximizes the Skaggs information content in spatial firing map

Loc Opt Shift Sparsity= Time shift which minimizes the sparsity in spatial firing map

Loc Opt Shift Coherence= Time shift which maximizes the spatial coherence in spatial firing map

#### Spatial Autocorrelation

No Parameter

#### Grid Cell Analysis

Is Grid= Indicates if the unit is a Grid cell or not (1= yes, 0= no)

Grid Mean Alpha= Average of the angles each arm of the hexagon (formed from the peaks of the firing fields) forms with the centre of the spatial autocorrelation

Grid Mean Psi= Mean angle between the arms of the central hexagon (formed from the peaks of the firing fields) in spatial autocorrelation

Grid Spacing= Average spacing between the peak firing fields forming the grid; Obtained from spatial autocorrelation

Grid Score= Gridness score as in DOI: 10.1126/science.1125572

Grid Orientation= Inclination of the central hexagonal firing field patterns with the X-axis

#### Border Cell Analysis

Border Skaggs= Skaggs information content for border vs spike-rate

Border Ang Ext= Largest angular segment with non-zero histogram count in active pixel (>20% of maximum firing rate) vs angular distance histogram

#### Gradient cell Analysis

Grad Pearse R= Goodness of fit Pearson’s R between the calculated and model-fit rate of firing vs distance from border

Grad Pearse P= Goodness of fit Pearson’s P between the calculated and model-fit rate of firing vs distance from border

Grad adj Rsq= Goodness of adjusted R2 between the calculated and model-fit rate of firing vs distance from border

Grad Max Growth Rate= Maximum rate of growth in firing rate in the fitted Goempertz function

Grad Inflect Dist= Distance from border where the growth of firing rate is maximum

#### Multiple Regression

Mult Rsq= Goodness of fit of the linear equation with the observed spike rate. Alternatively, it is a measure of the amount of variance explained in firing rate by all the independent variables

Semi Rsq Loc= Explained variance in firing rate by the location alone

Semi Rsq HD= Explained variance in firing rate by the head direction alone

Semi Rsq Speed= Explained variance in firing rate by the running speed alone

Semi Rsq Ang Vel= Explained variance in firing rate by the angular velocity alone

Semi Rsq Dist Border= Explained variance in firing rate by the border as a variable alone

#### Interdependence Analysis

DR HP= Distributive ratio for predicting head direction tuning curve (H) from spatial firing map (P)

DR SP= Distributive ratio for predicting spike rate vs running speed curve (S) from spatial firing map (P)

DR AP= Distributive ratio for predicting spike rate vs Angular velocity curve (A) from spatial firing map (P)

DR BP= Distributive ratio for predicting spike rate vs distance from border curve (B) from spatial firing map (P)

#### LFP Frequency Spectrum

No parameter

#### Unit LFP-phase Distribution

LFP Spike Mean Phase= Average LFP phase of the spikes

LFP Spike Mean Phase= Average no of spikes with Mean Phase

LFP Spike Phase Res Vect= Resultant vector on the distribution of spike-phases on LFP waves

#### Unit LFP-phase Locking

No parameter