

An analysis platform and tools to understand the dynamics of neuronal encoding in rodents

A dissertation submitted to the University of Dublin, Trinity College

for the degree

Doctor of Philosophy

By

Md Nurul Islam

February 2019

Supervisor

Prof Shane O'Mara

The project was conducted in Institute of Neuroscience, Trinity College Dublin

“...rationality of thought imposes a limit on a person's concept of his relation to the cosmos.”

-John Forbes Nash Jr.

DECLARATION

I hereby declare that I am the sole author of this dissertation and that the work presented in it, unless otherwise referenced, is entirely my own. I also declare that the work has not been submitted, in whole or in part, to any other university as an exercise for a degree or any other qualification.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Md Nurul Islam

February 2019

ACKNOWLEDGEMENTS

Firstly, I would like to convey my deepest gratitude to my supervisor Prof Shane O'Mara for his trust in me, support and stimulating suggestions throughout the entire duration of my project.

I would also like to thank all my colleagues in Experimental Brain Research Lab led by Professor Shane O'Mara. I am personally grateful to them for their motivating suggestions and moral supports time to time throughout the entire duration of my project.

It is worth mentioning the name of Dr Charlotte Callaghan and Dr Jen Rouine who always welcome the new people in the lab and create a congenial atmosphere for a new member to assimilate well into the lab and the Irish culture.

My special thanks go to Dr Marian Tsanov and Dr Maciek Jankowski for their support since before I started my doctoral project. I will also be thankful to Dr Andrea Della-Chiesa and Prof Pawel Matulewicz for helping me understand the impact of my work and, thereby, boosting my motivation towards its completion.

All the amazing experiences in Dublin would not be possible without the support from Wellcome Trust and Science Foundation Ireland. I will remain grateful all of my life to everybody who directly or indirectly cooperated with us throughout the research program to ensure the quality doctoral training for me.

Just as many other doctoral candidates, I also felt demotivated at times. I express my most sincere gratitude to Prof Shane, Dr Paul Wynne and Prof Pawel for boosting my motivation and providing me important suggestions regarding prospects of my career. Prof Clare Kelly and Prof Tim Trimble, as my appraisers, provided useful guidelines on conducting the project and writing the thesis, I would like to thank them as well.

My gratitude to all the smiling faces in Trinity College Institute of Neuroscience. And, of course, a special thanks to Mr Ciaran Conneely, Ms Barbara Hewitt, Mr Amir Awan

Abdullah and other administrative, IT and HR colleagues to take care of our administrative aspects.

I am grateful to all my friends in Dublin who let me feel at home during my years abroad.

Finally, thanks to my family for letting me understand the challenges in life and help make me stronger every day of my life.

Md Nurul Islam

ABSTRACT

The principal goal of the proposed project is to develop a data analysis framework that incorporates analytical methods for the study of cognitive map represented by neurons *in vivo* recorded from chronically implanted freely behaving rodents.

Where and how spatial information is represented in the brain has long been studied since O'Keefe and Dostrovsky (1971) first described the spatial receptive fields in hippocampal neurons (O'Keefe and Dostrovsky, 1971). The discovery of grid cells-neurons with multiple such receptive fields arranged in a triangular grid, in entorhinal cortex (Hafting et al., 2005) led to the discovery of an internal navigation system in the brain (Moser et al., 2008). Moreover, neurons tuned to non-spatial, natural stimuli (e.g. speed-cells, etc.), have also been described, and are likely to contribute to the dynamic representations of 'self-location', e.g. for path integration (Kropff et al., 2015).

Recently, other brain areas contributing to this navigation system have also been explored with novel experiment designs in chronically implanted rodents (Jankowski et al., 2014). This approach can generate vast amounts of data, particularly if acquired over a long duration, i.e. to validate the stability of recordings and to test different experimental manipulations (Jankowski et al., 2014). Moreover, advances in the design of electrodes, i.e. increasing the density (and therefore, the number of recording sites), have, and will continue to increase the amount of generated data exponentially in the coming years (Rey et al., 2015). The process of analysing such large data sets involves first identifying the activities of single-neurons from the noisy recorded data using unsupervised machine learning techniques, second, the analysis of relationships with spatial and non-spatial variables and verifying the correlations, and finally, the computation of inferential statistics for the description of local cell population. There are some open-source software packages for studying neural codes of single-neurons, multi-neurons, and local field potential (Rey et al., 2015, Ince, 2013), however, at present, there is no toolbox, so far we know, to explore neuronal encoding of spatial and non-spatial information relevant to cognitive mapping that also incorporates batch processing of large amounts of data. Individual laboratories have been using established analyses while new ones evolve, but there is no software

that implements these algorithms within one working environment, which limits the wide applications of these methods. Moreover, it is also necessary to facilitate quick implementation and integration of new techniques along with the established ones given the challenges associated with evolution of new technology.

In this project we developed NeuroChaT (***N**euron **C**haracterization **T**oolbox*), a graphical user interface (GUI)-based open-source software suit that will bring together the existing algorithms and analysis methods in a unified framework for greater accessibility and to provide a platform for easier implementation of the context-specific techniques. I expect that it will enable the scientific community to focus more on bringing excellence in both developing novel algorithms and experimental designs with an ease of access to a widely used and standard data format much like SPM and BrainVoyager in neuroimaging.

CONTENTS

LIST OF FIGURES.....	XI
LIST OF TABLES.....	XXIII
1 THESIS OUTLINE-----	1
2 BACKGROUND-----	9
2.1 NAVIGATION	10
2.1.1 CONCEPT OF NAVIGATION-----	10
2.1.2 SPATIAL MEMORY AND COGNITIVE MAP-----	13
2.2 EXTRACELLULAR NEURONAL RECORDINGS	22
2.2.1 PRINCIPLES OF EXTRACELLULAR RECORDINGS-----	23
2.2.2 SINGLE-UNIT ACTIVITY-----	29
2.2.3 NEURONAL ENCODING SCHEMES-----	37
2.2.4 CHALLENGES IN EXTRACELLULAR DATA ANALYSIS-----	45
2.3 IMPLANTED ELECTROPHYSIOLOGY TOOLBOXES	48
2.3.1 SPIKE-SORTING-----	49
2.3.2 SPIKE-TRAIN DYNAMICS AND FIELD POTENTIALS-----	53
2.3.3 GENERAL PURPOSE TOOLBOXES-----	56
2.3.4 NEURONAL ENCODING OF SPATIAL BEHAVIOR-----	58
2.3.5 SUMMARY OF TOOLBOXES-----	58
2.4 PROJECT OBJECTIVE	65
2.5 REFERENCES	67
3 NEUROCHAT-----	75
3.1 ARCHITECTURE	76
3.2 DESIGN	79
3.2.1 TECHNOLOGIES USED-----	79
3.2.2 DATA MODEL AND STORAGE-----	80
3.2.3 CONTROLLER-----	84
3.2.4 USER INTERFACE-----	85

3.2.5	CONFIGURATION	86
3.2.6	UTILITIES	87
3.2.7	ORGANIZING THE CODES	87
3.3	ANALYSIS METHODS AND ALGORITHMS	88
3.3.1	RHYTHMIC AND WAVEFORM PROPERTIES OF SPIKES	92
3.3.2	ANALYSIS PERTAINING TO LOCATION OF ANIMAL	104
3.3.3	ANALYSES ON DIRECTIONAL CORRELATIONS OF SPIKES	122
3.3.4	NON-SPATIAL NAVIGATIONAL INFORMATION	132
3.3.5	MULTIVARIABLE ANALYSES	135
3.3.6	LOCAL FIELD POTENTIAL	137
3.3.7	SINGLE UNITS TO LOCAL FIELD POTENTIAL PHASE SYNCHRONY	138
3.4	DATA MANAGEMENT MADE EASY	147
3.4.1	NOMENCLATURE	148
3.4.2	BATCH MODE PROCESSING	150
3.4.3	VERIFICATION UTILITY	150
3.4.4	PDF MANAGEMENT UTILITY	151
3.4.5	DATA ACCESSIBILITY THROUGH NWB/ HDF5	151
3.5	REFERENCES	156
4	<u>DISCUSSION</u>	<u>159</u>
4.1	ATTAINING NEUROCHAT OBJECTIVES	160
4.1.1	INTERACTIVE TOOL	160
4.1.2	HOW CAN IT WORK AS A FRAMEWORK?	161
4.1.3	UNIFICATION OF ANALYSES METHODS	161
4.1.4	DEVELOPING PROBLEM SPECIFIC ANALYSIS	162
4.1.5	BRINGING EFFICIENCY IN RESEARCH YIELD	163
4.2	TOWARDS A CULTURE OF OPENNESS AND REPRODUCIBILITY	163
4.2.1	OPEN DATA	163
4.2.2	OPEN SOURCE	170
4.2.3	OPEN STORAGE	172
4.2.4	NEUROCHAT AND OPEN SCIENCE	175
4.3	DATA INTEGRITY AND OPEN SCIENCE IN NEUROCHAT	177
4.4	FUTURE WORKS	179
4.5	REFERENCES	180

A. NEUROCHAT GUI USE GUIDE	184
B. API USE GUIDE	199
C. INPUT PARAMETER DESCRIPTION	223
D. OUTPUT PARAMETER DESCRIPTION	234
E. EQUATIONS FOR CIRCULAR STATISTICS	242
F. REFERENCE	243

LIST OF FIGURES

FIGURE 1.1 THE SCHEMATIC OF CHAPTER 2 ----- 2

FIGURE 1.2 SCHEMATIC OF CHAPTER 3 ----- 3

FIGURE 1.3 SCHEMATIC OF CHAPTER 4 ----- 4

FIGURE 1.4 SCHEMATIC OF CHAPTER 5 ----- 5

FIGURE 1.5 SCHEMATIC OF CHAPTER 6 ----- 5

FIGURE 2.1 AN ILLUSTRATION OF TWO HYPOTHESES OF SPATIAL UPDATING. PANEL (A) ILLUSTRATES AN ALLOCENTRIC MODEL IN WHICH THE CENTRAL CROSS REPRESENTS AN EXTERNAL REFERENCE SYSTEM. THE TARGET LOCATIONS ARE REPRESENTED AS A AND B AND REMAIN THE SAME AS THE OBSERVER, REPRESENTED BY THE FILLED TRIANGLE, MOVES FROM THE OLD POSITION TO THE NEW ONE. SPATIAL UPDATING INVOLVES COMPUTATION OF THE OBSERVER’S NEW POSITION, S' , IN THE EXTERNAL REFERENCE SYSTEM ACCORDING TO THE OLD POSITION S AND THE MOVEMENT VECTOR M ($S' = S + M$). PANEL (B) ILLUSTRATES AN EGOCENTRIC MODEL IN WHICH THE TARGET POSITIONS (A AND B) ARE REPRESENTED RELATIVE TO THE OBSERVER. SPATIAL UPDATING INVOLVES COMPUTING THE NEW EGOCENTRIC POSITIONS (A' AND B') OF THE OBJECTS AS THE OBSERVER MOVES ($A' = A - M$; $B' = B - M$). (WANG ET AL., 2006)-----13

FIGURE 2.2 SCHEMATIC DIAGRAM CONCENTRATING ON THE BRAIN REGIONS AND CELL TYPES DISCUSSED IN THIS REVIEW. PLACE CELLS CAN BE FOUND IN THE HIPPOCAMPUS, NUCLEUS REUNIENS (NRE), PARATENIAL NUCLEUS (PT), ANTEROMEDIAL NUCLEUS (AM), CLAUSTRUM, MEDIAL ENTORHINAL CORTEX AND SUBICULUM. PLACE CORRELATES (I.E. WEAK SPATIAL ACTIVITY) CAN BE FOUND IN THE ORBITOFRONTAL CORTEX (OFC), POSTRHINAL CORTEX, LATERAL ENTORHINAL CORTEX AND LATERAL SEPTUM. GRID CELLS CAN BE FOUND IN THE MEDIAL ENTORHINAL CORTEX, PRE- AND PARASUBICULUM. HEAD-DIRECTION CELLS CAN BE FOUND IN THE LATERAL MAMILLARY NUCLEI (LMN), ANTERODORSAL NUCLEI (ADN), LATERODORSAL NUCLEI (LDN), RETROSPLLENIAL CORTEX (RSC), POSTSUBICULUM, NUCLEUS REUNIENS AND ANTEROMEDIAL NUCLEUS (AM). BOUNDARY CELLS CAN BE FOUND IN THE PARASUBICULUM, CLAUSTRUM, SUBICULUM, ANTERIOR CINGULATE CORTEX, PRE- AND PARASUBICULUM AND MEDIAL ENTORHINAL CORTEX. OBJECT SENSITIVE CELLS CAN BE FOUND IN THE LATERAL ENTORHINAL CORTEX, POSTRHINAL CORTEX, ORBITOFRONTAL CORTEX (OFC) AND THE LATERAL SEPTUM. GOAL CELLS CAN BE FOUND IN THE MEDIAL PREFRONTAL CORTEX (MPFC) AND PRELIMBIC AND INFRA-LIMBIC REGIONS OF THE PREFRONTAL CORTEX. SELF-MOTION OR EGOCENTRIC CELLS SUCH AS THOSE ENCODING RUNNING SPEED OR ANGULAR HEAD VELOCITY CAN BE FOUND IN THE MEC, STRIATUM, RSC, PPC, LMN AND DTN. (ADAPTED FROM GRIEVES AND JEFFERY, 2017)-----18

FIGURE 2.3 DEPICTION OF VOLUME CONDUCTOR THEORY TO MODEL CURRENT FLOWS AROUND AN AXON IN A UNIFORM, LOW-RESISTANCE EXTRACELLULAR MEDIUM CALLED VOLUME CONDUCTOR; (A) WHEN THE AXON IS AT REST, THE MEMBRANE POTENTIAL IS UNIFORM, AND NO CURRENT FLOWS; (B) CURRENT WILL FLOW IN A DEPOLARIZED SEGMENT OF THE MEMBRANE. THE FLOW IS INWARD AT THE DEPOLARIZED REGION (SINK) AND OUTWARD AT ADJACENT REGIONS (SOURCE). (ADAPTED FROM HEINRICHER, 2004) -----24

FIGURE 2.4 MODEL OF SOURCES AND SINKS PREDICTS THAT A TRIPHASIC WAVEFORM WILL BE RECORDED FROM AN ISOLATED AXON; (A) AS THE AP APPROACHES THE REGION UNDERNEATH THE ELECTRODE, THAT MEMBRANE SERVES AS A SOURCE, AND THE ELECTRODE SEES A POSITIVE POTENTIAL RELATIVE TO A DISTANT INDIFFERENT ELECTRODE; (B) WHEN THE AP REACHES THE MEMBRANE UNDERLYING THE MEMBRANE, THE ELECTRODE RECORDS A NEGATIVE POTENTIAL; (C). AS THE AP CONTINUES DOWN THE AXON, THE MEMBRANE UNDER THE ELECTRODE ONCE AGAIN ACTS AS A SOURCE, AND AS A CONSEQUENCE, THE ELECTRODE RECORDS A POSITIVE POTENTIAL. (ADAPTED FROM HEINRICHER, 2004) ----- 25

FIGURE 2.5 SIMULTANEOUS RECORDING OF THE INTRACELLULAR AND EXTRACELLULAR SPIKE. THE MONOPHASIC DEPOLARIZATION SEEN BY THE INTRACELLULAR ELECTRODE CORRESPONDS WITH A POSITIVE-NEGATIVE-POSITIVE WAVEFORM RECORDED BY THE EXTRACELLULAR ELECTRODE. NOTE DIFFERENT SCALES FOR EXTRACELLULAR AND INTRACELLULAR RECORDINGS. (ADAPTED FROM HENZE ET AL., 2000) ----- 25

FIGURE 2.6 THE SIMPLEST MODEL OF AN ISOLATED NEURON SHOWING THE BIPHASIC POTENTIAL RECORDED FROM A SOMA; (A) RECODING OF DEPOLARIZATION WHERE THE MEMBRANE WORKS AS THE SINK, AND THE RECORDED VOLTAGE IS NEGATIVE. (B) RECORDING OF REPOLARIZED MEMBRANE POTENTIAL AS THE AP MOVES DOWN THE AXON. SOMA WORKS AS THE SOURCE IN THIS CASE AND CONTRIBUTES TO A POSITIVE POTENTIAL. ----- 26

FIGURE 2.7 EXTRACELLULAR ELECTRIC FIELD MODEL OF AN EXAMPLE NEURON; (A) DIRECTION OF LINES OF CURRENT FLOW AROUND A STELLATE SHAPED NEURON DURING SOMATIC ACTION POTENTIAL; (B) DISTRIBUTION OF MEMBRANE CURRENT AND ITS RELATION TO THE POTENTIAL AT AN EXTRACELLULAR POINT (MODELLED WITH SINGLE DENDRITE). CURRENT FLOWS INWARD DURING AN ACTION POTENTIAL AT CELL'S SOMA (HATCHED), AND OUTWARD AT VARIOUS REGIONS ALONG THE DENDRITE. (ADAPTED FROM (HUMPHREY, 1979)).----- 30

FIGURE 2.8 CHANGES IN SPIKE-SHAPES WITH RESPECT TO THE ELECTRODE LOCATION; (A) RECORDED SPIKES FOR A SPHERICAL NEURON UNDERGOING A SPIKE-GENERATING CONDUCTANCE CHANGE OVER ONLY A PORTION OF A SOMA; (B) CHANGES IN THE ACTION POTENTIAL IN A MORE GENERALIZED MODEL OF NEURON-ELECTRODE INTERFACE (ADAPTED FROM OBIEN ET AL., 2015).----- 31

FIGURE 2.9 EXAMPLE OF THE EFFECT OF CELL TYPES ON RECORDED ACTION POTENTIALS; SPIKES RECORDED ACROSS (A) THE AXIS OF A PYRAMIDAL CELL; (B) DEPTH OF A STELLATE CELL (ADAPTED FROM HUMPHREY AND SCHMIDT) ----- 32

FIGURE 2.10 SCHEMATIC DIAGRAM OF FOUR-CHANNEL TETRODE AMONG PYRAMIDAL CELLS. EACH ELECTRODE RECORDS SIGNAL FROM THE NEURONS, BUT THE SHAPE OF THE WAVEFORMS FROM THE SAME NEURON RECORDED IN INDIVIDUAL CHANNELS ARE DIFFERENT (RIGHT). THEREFORE, THE ISOLATION OF THE NEURONAL ACTIVITIES BASED ON WAVEFORM SHAPES ARE MORE FACILITATED (ADAPTED FROM O'KEEFE AND RECCE, 1993) ----- 33

FIGURE 2.11 STANDARD DATA PROCESSING STEPS FOR EXTRACELLULAR RECORDING EXPERIMENTS, PARTICULARLY THOSE INVOLVE SINGLE-UNIT CHARACTERIZATION. RAW DATA IS LOWPASS FILTERED TO OBTAIN THE LFP SIGNAL. THE BANDPASS FILTERED SIGNAL UNDERGOES SPIKE-SORTING PROCESS

RESULTING IN IDENTIFYING THE WAVEFORMS AND SPIKE-TRAIN FOR EACH NEURON. (ADAPTED FROM REY ET AL., 2015B) -----34

FIGURE 2.12 SPIKE SORTING STEPS (A) THRESHOLDING ON BANDPASS FILTERED RECORDED EXTRACELLULAR POTENTIAL FOR SPIKE DETECTION. THIS IS ONE OF THE MOST POPULAR CHOICE FOR DETECTING SPIKES (SIMULATED); (B) EXTRACTING THE WAVEFORMS (LEFT) AND CREATING A FEATURE SPACE CONSTITUTING DIFFERENT FEATURES OF THE WAVEFORM (PCA, WAVELET COEFFICIENTS ETC.); (C) USING CLUSTERING ALGORITHMS TO IDENTIFY INDIVIDUAL CLUSTERS (LEFT) OR USING MANUAL CLUSTER CUTTING TECHNIQUES (MIDDLE) TO DRAW BOUNDARIES AROUND A FEATURE CLUSTER AND IDENTIFY IT AS A DISTINCT CLUSTER. WAVEFORMS FROM INDIVIDUAL NEURONS ARE SEPARATED (RIGHT) WHERE INDIVIDUAL COLOR CORRESPONDS TO WAVEFORMS BELONGING TO ONE CLUSTER; (D) THE WAVEFORMS ARE APPROPRIATE TAGGED TO IDENTIFY THEIR TEMPORAL LOCATION. THIS TIMING INFORMATION IS USED FOR SUBSEQUENT ANALYSIS. SPIKE WAVEFORMS ARE EXTRACELLULAR DATA RECORDED FROM THE RAT ANTERIOR THALAMUS AND THE FILTERED CONTINUOUS SIGNAL IS SIMULATED DATA. -----36

FIGURE 2.13 EXPERIMENT DESIGN IN NEUROSCIENCE AND THE STIMULUS TO NEURAL RESPONSE TO BEHAVIORAL PATHWAY MODEL; S : STIMULUS, B : BEHAVIOR, N: NEURAL RESPONSE (NEURAL CODING, JOHNSON, 2000)-----38

FIGURE 2.14 (A) RECORDINGS FROM A NEURON IN THE PRIMARY VISUAL CORTEX (V1) OF A CAT. A BAR OF LIGHT WAS MOVED ACROSS THE RECEPTIVE FIELD OF THE CELL AT DIFFERENT ANGLES. THE DIAGRAM ON THE LEFT SHOWS THE STIMULUS. DASHED LINE IS THE RECEPTIVE FIELD OF THE NEURON. THE BIDIRECTIONAL MOTION OF THE LIGHT BAR IS INDICATED BY THE ARROWS. (B) AVERAGE FIRING RATE AS A FUNCTION OF THE ORIENTATION ANGLE OF THE LIGHT BAR STIMULUS. THE NONLINEAR RELATION IS READILY APPARENT AS IT TAKES THE SHAPE OF A GAUSSIAN FUNCTION. THE FIRING RATE IS MAXIMUM IN THE PREFERRED DIRECTION WHICH IS 0 DEGREE RELATIVE TO RECEPTIVE FIELD ORIENTATION FOR THIS CELL. (ADAPTED FROM DAYAN AND ABBOTT, 2005) -----40

FIGURE 2.15 DEPICTION OF DIFFERENT RATE CODING STRATEGIES. (A) SPIKE COUNT RATE WHICH MEASURES FIRING RATE ON A TRIAL-BY-TRIAL BASIS BY DIVIDING THE NUMBER OF SPIKES WITH THE DURATION OF THE TRIAL. (B) TIME DEPENDENT FIRING RATE IS CALCULATED BY CALCULATING PERI-STIMULUS TIME HISTOGRAM (PSTH) AND AVERAGING OVER TRIALS OF REPEATED STIMULUS; N_{sp} =NUMBER OF SPIKES IN ONE RUN, T = DURATION OF TRIAL, Y = SPIKE-COUNT RATE, ΔT = TIME BIN FOR GENERATING THE PSTH, K = TRIAL NUMBER, N_k = NUMBER OF SPIKES IN K-TH TRIAL BETWEEN TIME T AND $T+ \Delta T$, P = TIME DEPENDENT FIRING RATE. (ADAPTED FROM GERSTNER AND KISTLER, 2002) -----40

FIGURE 2.16 PHASE PRECESSION IN PLACE CELL AND GRID CELL IN RODENT HIPPOCAMPAL FORMATION; (A) EXAMPLE OF PHASE PRECESSION IN A PLACE CELL SHOWING THE SPIKING RATE MAP (LEFT), THE SPIKES AS DOTS ON THE ANIMAL'S PATH (BLACK LINE; LEFT MIDDLE), RASTER PLOTS OF SPIKE PHASE VERSUS PDCD. RED LINE IS THE CIRCULAR LINEAR REGRESSION LINE (RIGHT MIDDLE) AND AN EXAMPLE OF THE TIMING OF SPIKES (VERTICAL TICKS) VERSUS LFP (GREY LINE, RIGHT); (B) SIMILAR EXAMPLE FOR A GRID CELL; (C) EXAMPLE OF THE TRANSFORMATION OF A RUN THROUGH A FIRING FIELD ONTO A UNIT CIRCLE WHICH REPRESENTS THE FIRING FIELD WITH THE PEAK FIRING RATE AT THE CENTER.

LOCATIONS ON THE RUN ARE TRANSFORMED RADially SO THAT THEIR PROPORTIONAL DISTANCE BETWEEN THE FIRING PEAK AND THE PERIMETER IS PRESERVED, AND THEN ROTATED ABOUT THE CENTER OF THE CIRCLE SO THAT THE MEAN DIRECTION OF THE RUN (WHITE ARROW) IS LEFT–RIGHT FOR ALL RUNS. ‘PBCD’ IS THE DISTANCE OF THE ANIMAL TO THE FIELD PEAK PROJECTED ON THE CURRENT RUNNING DIRECTION, WHICH VARIES FROM –1 TO 1 AS THE ANIMAL ENTERS THE FIELD ON THE LEFT AND EXITS ON THE RIGHT. (ADAPTED FROM JEEWAJEE ET AL., 2014)----- 42

FIGURE 2.17 EXPLANATION OF THE POPULATION ENCODING OF DIRECTIONAL STIMULI; (A) SPIKING ACTIVITY OF A NEURON FOR DIFFERENT DIRECTIONAL STIMULI, SHOWING PREFERENCE TOWARDS WEST; (B) TYPICAL TUNING CURVE IN RESPONSE TO THE DIRECTIONAL STIMULI; (C) DEPICTION OF THE OVERLAPPING SELECTIVITY. EACH BELL-CURVE SHOWS THE TUNING FUNCTION OF INDIVIDUAL NEURONS; (D) EACH BLACK LINE REPRESENTS THE PREFERRED FIRING DIRECTION OF EIGHT EXAMPLE NEURONS. THEIR LENGTH SHOWS THE FIRING RATE OF NEURON FOR A GIVEN STIMULI. THE STIMULUS DIRECTION IS RETRIEVED BY THE WEIGHTED AVERAGE OF THE BLACK LINES WHICH IS REPRESENTED BY THE BLUE LINE. (RECREATED FROM DAYAN AND ABBOTT, 2005)----- 44

FIGURE 2.18 EXPONENTIAL GROWTH IN THE NUMBER OF RECORDED NEURONS. (A) EXAMINING 56 STUDIES PUBLISHED OVER THE LAST FIVE DECADES, (STEVENSON AND KORDING, 2011) FOUND THAT THE NUMBER OF SIMULTANEOUSLY RECORDED NEURONS DOUBLED APPROXIMATELY EVERY 7 YEARS. (B) A TIMELINE OF RECORDING TECHNOLOGIES DURING THIS PERIOD SHOWS THE DEVELOPMENT FROM SINGLE-ELECTRODE RECORDINGS TO MULTI-ELECTRODE ARRAYS AND IN VIVO IMAGING TECHNIQUES. (ADAPTED FROM STEVENSON AND KORDING, 2011)----- 46

FIGURE 2.19 POLYTRODES WITH 54 RECORDING ELECTRODES (ADAPTED FROM BLANCHE ET AL., 2005)--- 47

FIGURE 3.1 THE ARCHITECTURE OF NEUROCHAT. IT SHOWS THE CONNECTIVITY OF DIFFERENT COMPONENTS AND THE FLOW OF INFORMATION BETWEEN THEM. ----- 78

FIGURE 3.2 THE GRAPHICAL-USER INTERFACE OF NEUROCHAT----- 78

FIGURE 3.3 THE WORKFLOW DIAGRAM IN NEUROCHAT AND ACTION SPACE OF DIFFERENT COMPONENTS OF THE PROGRAM. ----- 79

FIGURE 3.4 CLASS DIAGRAM FOR THE DATA CLASSES IN NEUROCHAT. THE REGULAR ARROWS SHOW THE INHERITANCE AND THE DIAMOND ARROWS SHOW THE COMPOSITION. INHERITANCE OR COMPOSITION ARE ONE-TO-ONE IN ALL THE CASES. TOP DIAGRAM SHOWS THE MULTIPLICITY (NUMBERS) OF ASSOCIATION, AND THE BOTTOM DIAGRAM SHOWS THE COMPOSITE OBJECTS (GREEN TEXTS). THE LATTER IS GENERATED FROM THE CODE USING THE PYREVERSE PACKAGE.----- 83

FIGURE 3.5 CLASS DIAGRAM OF THE NEUROCHAT CLASS, THE CONTROLLER OF THE NEUROCHAT SOFTWARE. TOP DIAGRAM SHOWS THE MULTIPLICITY OF ASSOCIATION, AND THE BOTTOM DIAGRAM SHOWS THE COMPOSITE OBJECTS (GREEN TEXTS). THE LATTER IS GENERATED FROM THE CODE USING PYREVERSE PACKAGE. THE NAME PRECEDES THE DOT OF EACH CLASSES ARE THE NAME OF THE PYTHON MODULES WHERE THEY RESIDE.----- 85

FIGURE 3.6 CLASS DIAGRAM FOR THE NEUROCHAT_UI CLASS THAT DEFINES THE GRAPHICAL USER INTERFACE OF THE SOFTWARE. ----- 86

FIGURE 3.7 CATEGORY OF ANALYSES IN NEUROCHAT AND RELEVANT COUNT OF THE METHODS ----- 89

- FIGURE 3.8** THE MEAN \pm STD WAVEFORM OF A UNIT ISOLATED IN A TETRODE CONFIGURATION FROM DATA RECORDED IN RAT ANTERIOR THALAMUS. CHANNEL NUMBERS ARE, 1= UPPER LEFT, 2= UPPER-RIGHT, 3= LOWER-LEFT, 4= LOWER-RIGHT. IT SHOWS THAT THE VARIABILITY IS MUCH LESS IN CHANNEL 3 WHERE THE WAVEFORM IS MUCH STRONGER- INDICATING THAT A TINY SMALL PROPORTION OF NOISE-SPIKES (SPIKES NOT BELONGING TO THIS UNIT) WERE INCLUDED DURING THE SPIKE-SORTING. AS EXPECTED, THE VARIABILITY IS MUCH HIGHER IN OTHER CHANNELS. -----93
- FIGURE 3.9** DEPICTION OF HOW DIFFERENT WAVEFORM PROPERTIES ARE MEASURED. THE RED SOLID LINE IS THE WIDTH OF THE WAVEFORM WHICH IS MEASURED AS THE 25% OF THE PEAK VOLTAGE OF THE WAVEFORM AS SHOWN IN RED DOTTED LINE. -----94
- FIGURE 3.10** DISTRIBUTION OF INTER-SPIKE INTERVAL (ISI); X-AXIS GIVES THE ISI BINS, AND Y-AXIS GIVES THE COUNT OF ISI FOR A PARTICULAR BIN. THE RED-DOTTED LINE IS THE USER-SPECIFIED THRESHOLD FOR THE BURSTING SPIKES. THIS UNIT SHOWS A LARGE COUNT UNDER THE BURSTING THRESHOLD TIME AS WELL AS A HIGHER COUNT AT APPROXIMATELY 100MS (REPRESENTING RHYTHMICITY AT AROUND 10HZ). SO, THIS IS BOTH A BUSTING UNIT WITH A HIGHER BURST PROPENSITY AND THETA-RHYTHMICITY (THESE CONCEPTS WILL BE EXPLAINED IN THE FOLLOWING SECTIONS) -----95
- FIGURE 3.11** THE PLOT OF ISI AFTER VS ISI BEFORE; A) SCATTER PLOT, B) TWO-DIMENSIONAL HISTOGRAM. THE RED DOTTED LINE IS THE THRESHOLD FOR BURST. THE COLORED REGION DISPLAYS A HIGH DENSITY OF SPIKES AT INTERVALS LOWER THAN THE BURST THRESHOLD- IMPLYING THAT THIS IS A BURSTING UNIT. THE HORIZONTAL AND VERTICAL STRIP AT (A) AND (B) REPRESENTS A REPETITION OF ISI AT AROUND 100MS- IMPLYING THAT THE UNIT IS RHYTHMIC AT AROUND 10HZ.-----95
- FIGURE 3.12** ALGORITHM FOR CALCULATING THE CORRELATION BETWEEN INTER-SPIKE INTERVAL OF TWO SPIKING TRAINS. ONE OF THEM ACT AS REFERENCE, AND THE OTHER AS TARGET- AS THAT OF IN PERI-STIMULUS HISTOGRAM. THE SPIKING EVENTS IN THE REFERENCE TRAIN CAN BE COMPARED TO THE EVENTS AND THE HISTOGRAM WITH RESPECT TO EACH SPIKE IS CALCULATED ON THE TARGET SPIKE TRAIN. THE SUMMATION OF ALL SUCH HISTOGRAM RESULTS IN THE ISI CORRELATION. IF THE TARGET AND REFERENCE SPIKE TRAINS ARE FROM THE SAME UNIT, THIS IS CALLED AUTO-CORRELATION HISTOGRAM.-----96
- FIGURE 3.13** THE AUTOCORRELATION HISTOGRAM OF THE ISI OF A UNIT. THE PEAKS-TROUGH RHYTHMIC PATTERN IS MORE APPARENT THAN THE ISI HISTOGRAM (INSET) OF THE UNIT.-----97
- FIGURE 3.14** THE ISI AUTOCORRELATION HISTOGRAM FOR SHORT LAGS. THIS PARTICULAR EXAMPLE REVEALS THAT THERE IS NO COUNT OF SPIKES WITHIN 2MS INTERVAL, WHICH IS AN INDICATION THAT, IF WE CONSIDER A 2MS REFRACTORY PERIOD FOR THE UNIT, SPIKES FROM OTHER UNITS ARE NOT INCORPORATED INTO THIS UNIT. THIS IS AN ALTERNATIVE ASSESSMENT OF QUALITY OF SPIKE SORTING. -----97
- FIGURE 3.15** ISI AUTOCORRELATION HISTOGRAM OF A THETA-MODULATED UNIT. THE RED ENVELOPE SHOWS THE FITTED CURVE FROM THE EQUATION 3.1. THE INDEX FOR THETA MODULATION IS DERIVED FROM THIS FITTED CURVE. -----99
- FIGURE 3.16** ISI AUTOCORRELATION HISTOGRAM OF A THETA-SKIPPING UNIT (BLUE HISTOGRAM) AND THE CORRESPONDING FITTED CURVE FROM EQUATION 3.2 (RED ENVELOPE).-----100

FIGURE 3.17 ILLUSTRATION OF BURSTING SPIKE TRAINS AND THE CONCEPT OF INTER-BURST INTERVAL (IBI) ALONG WITH THE DURATION OF BURST. THE RATIO OF BURST DURATION TO THE IBI IS DEFINED AS THE DUTY CYCLE OF BURSTING (ADAPTED FROM [HTTP://WWW.SCHOLARPIDIA.ORG/ARTICLE/BURSTING](http://www.scholarpedia.org/article/bursting)).

-----102

FIGURE 3.18 DEPICTION OF THE USE OF ISI HISTOGRAM TO IDENTIFY THE BURSTING UNITS; TOP: REGULAR SPIKING NEURON, BOTTOM: SPIKE WITH BURSTING SHOWING THE BIMODALITY OF DISTRIBUTION WITH CHARACTERISTIC HIGH COUNTS OF SPIKES AT LOW INTERVALS (ADAPTED FROM [HTTP://WWW.SCHOLARPIDIA.ORG/ARTICLE/BURSTING](http://www.scholarpedia.org/article/bursting)).

-----103

FIGURE 3.19 ILLUSTRATION OF HOW LOCATIONAL FIRING PLACE IS CREATED FROM THE LOCATION DATA OF THE ANIMAL AND THE SPIKE TIMES. A VISIT COUNT MAP IS CREATED (UPPER LEFT) FROM THE TWO-DIMENSIONAL HISTOGRAM OF THE LOCATION WHICH SHOWS THE NUMBER OF TIMES ANIMAL VISITED IN A PARTICULAR PIXEL IN A FORAGING ARENA. PIXELS OF 2CM SIZES ARE PAIRED AND COLOR CODED IN THE RAW LOCATIONAL DATA IN THE MIDDLE. IT IS THEN MULTIPLIED BY THE SAMPLING INTERVAL OF THE LOCATION DATA TO OBTAIN THE TOTAL AMOUNT OF TIME THE ANIMAL HAS TRAVERSED IN DIFFERENT PIXELS OF THE ARENA (UPPER-MIDDLE, TIME MAP). ANOTHER MAP IS CREATED FOR THE SPIKE COUNT AT THOSE LOCATIONS (UPPER RIGHT) BY COUNTING THE NUMBER OF SPIKES IN THE PAIRED PIXELS. THE MAP IN THE RIGHT IS DIVIDED BY THE TIME-MAP TO OBTAIN THE FIRING-RATE MAP OF A UNIT IN THE ARENA.

-----105

FIGURE 3.20 SAMPLE OUTCOME OF LOCATIONAL FIRING RATE ANALYSIS; (A) FIRING RATE AS A FUNCTION OF LOCATION OF THE ANIMAL; (B) BLACK LINES SHOW THE PATH OF THE ANIMAL TRAVERSED, THE RED DOTS REPRESENT SPIKING EVENTS.

-----106

FIGURE 3.21 DISTRIBUTION OF SPECIFICITY MEASURES IN LOCATIONAL SHUFFLING ANALYSIS. RED LINE SHOWS THE 95TH-PERCENTILE. THE SKAGGS INFORMATION CONTENT (IC), COHERENCE OF THE FIRING MAP FROM THE ORIGINAL SPIKE-TRAIN IS COMPARED TO THE RED LINE TO TEST FOR SIGNIFICANCE AT 5% LEVEL AT THE UPPER TAIL AS THEY ARE SUPPOSED TO BE HIGHER IN ORIGINAL SPIKE-TRAIN IF THE SPIKING INCIDENTS WITH RESPECT TO LOCATION ARE EXPECTED NOT TO BE RANDOM (A, B). AS THE SPARSITY IS EXPECTED TO BE MINIMUM, THE LOWER TAIL AT 5% SIGNIFICANCE LEVEL IS CONSIDERED FOR THE COMPARISON (C).

-----107

FIGURE 3.22 THE TIME-LAPSE SPIKE-PLOT (TOP TWO ROWS) AND FIRING RATE MAP (BOTTOM TWO ROWS). THE CUMULATIVE TIME-LAPSE ARE 0-1MIN, 0-2MIN, 0-4MIN, 0-8MIN, 0-16MIN, 0-END OF THE RECORDING. THE PROGRESSIVE TIME-LAPSE ARE 1-2MIN, 2-4MIN, 4-8MIN, 8-16MIN, 0-END OF THE RECORDING.

-----108

FIGURE 3.23 CONSTRUCTION OF FIRING RATE MAP FROM THE SPIKE TRAIN OF THE UNIT AND SHIFTED SPIKE TRAIN; (A) VISIT COUNT, TIME MAP, AND SPIKE-COUNT MAP FOR THE UNCHANGED SPIKE-TRAIN OF THE UNIT; (B) MAPS SIMILAR TO (A), BUT WITH A SHIFT IN SPIKE TRAIN AS SHOWN IN (C); (C) TOP ROW SHOWS THE COORDINATES OF THE LOCATION, MIDDLE ROW DEPICTS THE SPIKE TRAIN, AND THE BOTTOM ROW SHOWS THE SPIKE TRAIN WHEN IT IS SHIFTED TO MATCH WITH A PAST LOCATION OF THE ANIMAL. THE PIXELS FORMED BY PAIRING THE X-, AND Y-COORDINATES ARE COLOR-CODED. AS

WE CAN SEE, THE COUNT IN SPIKE CHANGES IN THE SHIFTED SPIKE TRAIN, AND A NEW MAP IS FORMED OUT OF IT.-----109

FIGURE 3.24 ILLUSTRATION OF THE RATIONALE BEHIND TIME-SHIFT ANALYSIS. THE TRAJECTORY IS CONSIDERED ONE-DIMENSIONAL FOR EASE OF EXPLANATION. THE UNIT IS ASSUMED TO SIGNAL THE LOCATION Y. SOLID ARROW REPRESENTS A TRAJECTORY FROM LEFT TO RIGHT ACROSS THE PLACE FIELD, THE DOTTED ARROWS SHOWS THE SIMILAR BUT IN OPPOSITE (RIGHT TO LEFT) TRAJECTORY; (A) EXPECTED FIRING RATE CURVE IF THE UNIT FIRES SIMULTANEOUSLY AS THE LOCATION OF THE ANIMAL CHANGES; (B) AN EXAMPLE OF CASE-A WHERE THE FIRING RATE IS HIGHER AT A LOCATION (X OR Z) AFTER THE ANIMAL TRAVERSES THE SIGNALING OR PREFERRED LOCATION Y (LEFT); THE FIRING RATE MAPS OVERLAPS AND THE PLACE FIELD BECOMES NARROWER WHEN THE SPIKE TRAIN IS PAIRED WITH A PAST LOCATION (RIGHT); (C) AN EXAMPLE OF CASE-B WHERE THE FIRING RATE IS HIGHER AT A LOCATION (X OR Z) BEFORE THE ANIMAL TRAVERSES THE SIGNALING OR PREFERRED LOCATION Y (LEFT); THE FIRING RATE MAPS OVERLAPS AND THE PLACE FIELD BECOMES NARROWER WHEN THE SPIKE TRAIN IS PAIRED WITH A FUTURE LOCATION (RIGHT); -----111

FIGURE 3.25 SPECIFICITY MEASURES FOR LOCATIONAL FIRING IN A UNIT WITH PLACE FIELD; (A) SKAGGS INFORMATION CONTENT (IC); (B) COHERENCE; (C) SPARSITY. THE SKAGGS IC AND THE COHERENCE ARE EXPECTED TO ATTAIN THE MAXIMUM VALUE WHEN THE UNIT IS PAIRED WITH IT S PREFERRED LOCATION WHERE THE SPARSITY WILL REACH TO ITS MINIMUM AS PLACE FIELD WILL BE MORE COMPACT. AS THEY REACH THE OPTIMUM VALUE AT A +VE TIME SHIFT, IT CAN BE SPECULATED THAT THE UNIT IS ANTICIPATORY.-----112

FIGURE 3.26 CONSTRUCTION OF AUTOCORRELATION MAP- CONCEPT OF LAGS AND HOW THE SPATIAL AUTOCORRELATION IS CALCULATED. THE SHADED AREA SHOWS THE OVERLAPPING PIXELS BETWEEN A LAGGED MAP AND THE ORIGINAL MAP. (A) OVERLAP AT $T_x = -3$ AND $T_y = -2$; (B) OVERLAP AT $T_x = +5$ AND $T_y = +2$. CORRELATION BETWEEN THE SHADED AREAS FOR A PARTICULAR LAG REPRESENTS THE AUTOCORRELATION FOR THAT LAG.-----113

FIGURE 3.27 PATTERNED FIRING FIELDS IN A GRID CELL; (A) PATTERN OF TESSELLATING TRIANGLES AMONG THE FIRING FIELDS; (B) HEXAGONAL PATTERN AROUND THE CENTRAL PEAK OF THE SPATIAL AUTOCORRELATION.-----114

FIGURE 3.28 FLOW CHART FOR THE DETECTION OF GRID CELL AND MEASURING THE GRIDNESS SCORE. -115

FIGURE 3.29 ILLUSTRATION OF IDENTIFYING GRID CELL AND CALCULATING GRIDNESS SCORE; (A) MAXIMA LOCATIONS IN SPATIAL AUTOCORRELATION; (B) PEAKS FORMING THE CENTRAL HEXAGON; (C) SHADED ZONE SHOWS THE $(1 \pm 0.5) * \text{MEAN PEAK DISTANCE}$; (D) ROTATIONAL CORRELATION MEASURED WITH THE SHADED ZONE IN (C). RED DOTS SHOW THE MAXIMA LOCATIONS, AND THE GREEN DOT SHOWS THE MINIMA IN THE CURVE. GRIDNESS SCORE IS THE DIFFERENCE IN BETWEEN THE ROTATIONAL CORRELATIONS AT PEAKS AND TROUGHS; (E) DEFINITION OF CENTRAL ANGLE (LEFT), INTERIOR ANGLE (MIDDLE) AND ORIENTATION OF THE GRID (RIGHT).-----116

FIGURE 3.30 SPIKE PLOT (A) AND FIRING RATE MAP (B) OF A UNIT WITH PREFERENTIAL FIRING ALONG THE BORDER OF THE ENVIRONMENT.-----117

FIGURE 3.31 STEPWISE DEPICTION OF BORDER DETECTION PROCEDURE; (A) BLACK LINES SHOW THE PATH OF THE ANIMAL TRAVERSED IN THE ENVIRONMENT; (B) PATHS ARE PIXELATED TO COUNT THE AMOUNT OF TIME ANIMAL VISITED A PIXEL; BLACK AREA SHOWS AREAS CAPTURED BY THE CAMERA BUT NOT VISITED BY THE ANIMAL; (C) BINARY PICTURE OF THE MAP IN (B). UNVISITED AREA IS REPRESENTED BY '1' AND VISITED AREA BY '0' (COLORMAP IS INVERSED); (D) BINARY MAP IS REGULARIZED TO OBTAIN A SMOOTHER BOUNDARY; (E) BLACK PIXELS ARE THE ONES FORMING THE BOUNDARY; (F) MAP OF THE DISTANCE OF EACH PIXEL FROM THE BOUNDARY PIXELS.-----118

FIGURE 3.32 BORDER CELL ANALYSIS; (A) SPATIAL FIRING RATE MAP SHOWING A CHARACTERISTIC CIRCULAR RING OF HIGH FIRING RATE ZONE NEAR THE ENVIRONMENTAL BOUNDARY; (B) HISTOGRAM OF NORMALIZED ACTIVE PIXEL COUNT FROM THE BORDER AT A PARTICULAR DISTANCE. ACTIVE PIXELS ARE COUNTED AS THE TOTAL PIXELS WITH FIRING RATE HIGHER THAN THE USER SPECIFIED THRESHOLD, I.E., 20% NORMALIZED BY THE TOTAL PIXEL AT THAT DISTANCE; (C) FIRING RATE CALCULATED ACCORDING TO THE RATE CODING SCHEME; (D) STAIR PLOT SHOWING A LOW MEAN DISTANCE FROM BORDER FOR PIXELS WITH HIGH FIRING RATES, AND SUDDEN INCREASE IN DISTANCE FOR PIXELS WITH LOW FIRING RATES, INDICATING THAT FIRING RATE IS HIGH NEAR THE BORDER AND ALMOST DIMINISHES NEAR THE CENTER OF THE ARENA.-----119

FIGURE 3.33 LINEARIZATION OF FIRING RATE MAP; (A) NUMBER OF ACTIVE PIXELS (ABSOLUTE COUNT) AT DIFFERENT ANGULAR DISTANCE OF THE ENVIRONMENT FROM THE CENTER OF THE ARENA; (B) NUMBER OF ACTIVE PIXEL AT DIFFERENT DISTANCE FROM THE BORDER ALONG PARTICULAR CIRCULAR DISTANCE. -----120

FIGURE 3.34 GRADIENT CELL ANALYSIS; (A) FIRING RATE MAP SHOWING THAT THE ACTIVITY OF THE UNIT INCREASES GRADUALLY AS THE ANIMAL MOVES FROM BORDER TOWARDS THE CENTER, FORMING BANDS OF ACTIVITY BELONGING TO A PARTICULAR FIRING RATE PERCENTAGE. HERE, 5 COLORS HAVE BEEN USED REPRESENTING SPIKING ACTIVITY OF 0-100% IN STEP OF 20%; (B) STAIR PLOT LIKE THE ONE CALCULATED IN BORDER CELL. THE STAIRS ARE INCREASING GRADUALLY, AND THERE IS NOT ABRUPT CHANGE IN THE PLOT, FURTHER PROVIDING SUPPORT FOR THE GRADIENT CELL; (C) THE SPIKING RATE VS DISTANCE OF THE ANIMAL FROM THE BORDER IS CALCULATED LIKE THAT IN BORDER CELL (BLUE). THIS CURVE IS FITTED WITH GOMPERTZ FUNCTION (GREEN); (D) DIFFERENTIAL FIRING RATE VS DISTANCE SHOWING THAT THE INCREASE IN FIRING RATE MAXIMIZES AT SHORT DISTANCE FURTHER FROM THE BORDER WITHIN THE ARENA. THE POINT WHERE THIS RATE OF CHANGES IN FIRING RATE MAXIMIZES IS CALLED DEFLECTION POINT. -----122

FIGURE 3.35 DEPICTION OF CALCULATION OF THE SPIKING RATE WITH RESPECT TO THE HEAD DIRECTION OF THE ANIMAL. TOP ROW- LEFT: VISIT COUNT OF THE ANIMAL IN DIFFERENT DIRECTIONAL BINS, MIDDLE: AMOUNT OF TIME SPENT IN SUCH DIRECTIONS, RIGHT: NUMBER OF SPIKES PAIRED WITH SUCH DIRECTION; MIDDLE ROW: HEAD-DIRECTION OF THE ANIMAL AT DIFFERENT TRACKER SAMPLES.; BOTTOM ROW- SPIKE-TRAIN OF THE UNIT DURING 9 SAMPLES SHOWN. HEAD DIRECTIONS FALLING IN DIFFERENT BINS ARE COLOR-CODED DIFFERENTLY. -----123

FIGURE 3.36 TUNING CURVES FROM A RECORDED HEAD-DIRECTIONAL UNIT; (A) FIRING RATE WITH RESPECT TO DIRECTIONAL HEADING OF THE ANIMAL (BLUE) AND THE PREDICTED FIRING RATE (GREEN) TO

OBSERVE THE HOMOGENEITY IN SAMPLING OF THE DIRECTIONAL BINS; (B) DEPICTION OF THE MEASURE OF HALF-WIDTH, A CHARACTERISTIC OF THE TUNING CURVE TO DESCRIBE HOW WIDE THE DIRECTIONAL RECEPTIVE FIELD IS; (C) MEASUREMENT OF SEPARATION ANGLE, THE DIFFERENCE BETWEEN THE DIRECTIONS WHERE PEAK FIRING OCCURS DURING CW(RED) AND CCW(BLUE) HEAD MOVEMENT.-----125

FIGURE 3.37 ILLUSTRATION OF THE DIRECTIONAL SPLIT AT A PIXEL. THE TOTAL TIME AT DIRECTION Θ_1 IS THE SUM OF TIME SPENT IN ALL PIXELS J, K, ... OR $T_j(\Theta_1) + T_k(\Theta_1) + \dots$ WHERE TOTAL SPIKES (AVERAGE OR PREDICTED) AT DIRECTION Θ_1 IS THE SUM OF THE RATES AT EACH PIXELS MULTIPLIED BY THE TIME AT THAT PIXEL OR $R_{pj}T_j(\Theta_1) + R_{pk}T_k(\Theta_1) + \dots$. THE PREDICTED RATE IS CALCULATED BY DIVIDING THE PREDICTED SPIKES BY TOTAL TIME. -----126

FIGURE 3.38 SAMPLE PLACE UNIT WITH APPARENT DIRECTIONAL TUNING RECORDED IN A PELLET CHASING TASK IN AN OPEN ARENA. THE BLUE LINE SHOWS THAT THE UNIT PREFERENTIALLY FIRES ALONG THE NORTH-NORTHWEST DIRECTION. GREEN LINE, THE PREDICTED RATE, ALMOST RESEMBLES THE OBSERVED FIRING RATE AND, THEREFORE, THE DR IS VERY SMALL (0.14). ONSET SHOWS THE BEHAVIOR OF THE UNIT WITH RESPECT TO LOCATION. THE PLACE FIELD IS IN THE NORTH-NORTHEAST WALL OF THE CYLINDRICAL ARENA AND MAY CAUSE THE UNIT APPARENTLY FIRES MORE ALONG THAT DIRECTION. WE CAN ALSO CONCLUDE THAT THERE IS AN INHOMOGENEITY IN SAMPLING OF THE HEAD DIRECTION.-----127

FIGURE 3.39 SCATTER PLOT OF THE HEAD DIRECTION CORRESPONDING TO EACH SPIKE OF (A) THE HEAD DIRECTIONAL UNIT WITH THE FIRING RATE SHOWN IN FIGURE 3.36 AND (B) A NON-HEAD DIRECTIONAL UNIT-----128

FIGURE 3.40 SAMPLE OUTPUT OF HEAD-DIRECTIONAL SHUFFLING ANALYSIS; (A) DISTRIBUTION OF RAYLEIGH Z SCORES (B) DISTRIBUTION OF THE VON MISES K. THE RED LINE SHOWS THE 95TH-PERCENTILE IN THE DISTRIBUTION WHICH IS THEN COMPARED TO THE RAYLEIGH $Z = 930$ AND $K = 2.44$ TO TEST FOR SIGNIFICANCE AT 5% LEVEL.-----129

FIGURE 3.41 THE TIME-LAPSE DIRECTIONAL SPIKE-PLOT (TOP TWO ROWS) AND TUNING CURVES (BOTTOM TWO ROWS). THE CUMULATIVE TIME-LAPSE ARE 0-1MIN, 0-2MIN, 0-4MIN, 0-8MIN, 0-16MIN, 0-END OF THE RECORDING. THE PROGRESSIVE TIME-LAPSE ARE 1-2MIN, 2-4MIN, 4-8MIN, 8-16MIN, 0-END OF THE RECORDING. -----130

FIGURE 3.42 CONSTRUCTION OF A TUNING CURVE FROM THE SPIKE TRAIN OF A UNIT AND IT SHIFTED SPIKE TRAIN; (A) VISIT COUNT, TIME MAP, AND SPIKE-COUNT MAP FOR THE UNCHANGED SPIKE-TRAIN OF THE UNIT; (B) MAPS SIMILAR TO (A), BUT WITH A SHIFT IN SPIKE TRAIN AS SHOWN IN (C); (C) TOP ROW SHOWS THE HEAD DIRECTION, MIDDLE ROW DEPICTS THE SPIKE TRAIN, AND THE BOTTOM ROW SHOWS THE SPIKE TRAIN WHEN IT IS SHIFTED TO MATCH WITH A PAST LOCATION OF THE ANIMAL. DIRECTIONS FALLING INTO DIFFERENT BINS ARE COLOR CODED. AS WE CAN SEE, THE COUNT IN SPIKE CHANGES IN THE SHIFTED SPIKE TRAIN, AND A NEW RATE IS FORMED OUT OF IT. -----131

FIGURE 3.43 SAMPLE RESULTS OF THE HEAD-DIRECTION TIME-SHIFT ANALYSIS FOR A UNIT THAT HAS A DIRECTIONAL FIRING FIELD; (A) THE SKAGGS INFORMATION CONTENT GRADUALLY MAXIMIZES TOWARDS THE PREFERRED DIRECTION AND DECLINES AFTERWARDS AS EXPECTED FOR A UNIT WITH

DIRECTIONAL FIRING FIELD. IT MAXIMIZES AT A +VE TIME SHIFT, WHICH IMPLIES THAT THE UNIT IS ANTICIPATING A PREFERRED DIRECTION. (B) SIMILAR RESULTS CAN BE OBTAINED BY USING THE PEAK FIRING RATE OF THE TUNING CURVES AT EACH SHIFTED TIME. (C) PLOT OF SEPARATION ANGLE Δ VS SHIFT OF TIME. SEPARATION OF THE CW VS CCW CURVES ARE EXPECTED TO BE ZERO AT THE PREFERRED DIRECTION. THEREFORE, THE ANTICIPATORY TIME INTERVAL IS MEASURED AS THE SHIFT OF TIME WHERE THE FITTED Δ LINE INTERSECTS X-AXIS (INTERSECTION OF THE BLUE AND BLACK LINES).

-----132

FIGURE 3.44 GRAPHICAL OUTPUT OF A UNIT WITH A HIGH CORRELATION ITS FIRING RATE TO THE SPEED OF THE ANIMAL (PEARSON'S $R= 0.9$, $P< 0.001$)-----133

FIGURE 3.45 ANGULAR HEAD VELOCITY ANALYSIS; (A) CONCEPT OF CLOCKWISE (CW) AND COUNTER-CLOCKWISE (CCW) HEAD DIRECTIONAL MOVEMENT (ADAPTED FROM BLAIR AND SHARP, 1995); (B) SAMPLE FIRING RATE OF A HEAD DIRECTION CELL THAT FIRES AT A DECREASING RATE AS THE MAGNITUDE OF AHV INCREASES, BUT DOES NOT DEPEND ON THE DIRECTIONALITY OF THE HEAD MOVEMENT.-----134

FIGURE 3.46 MULTIPLE REGRESSION CORRELATION PARAMETERS (MEAN \pm STD) FOR DIFFERENT INDEPENDENT VARIABLES. THE INSET SHOWS THE APPARENT DIRECTIONAL PREFERENCE (BLUE LINE) FOR A UNIT WITH LOCATIONAL FIRING PREFERENCE. AS EXPECTED, THE SEMIPARTIAL CORRELATION DOMINANTLY EXPLAINS THE VARIABILITY IN SPIKING ACTIVITY, AS THE DIRECTIONAL PREFERENCE IS AN ARTEFACT EXPLAINED BY THE GREEN LINE- THE PREDICTED FIRING RATE- WHICH IS VERY SIMILAR TO THE OBSERVED RATE (DR= 0.14)-----136

FIGURE 3.47 RESULT OF THE LFP ANALYSIS; (A) THE PSD (MV^2/HZ) CALCULATED USING THE WELCH'S METHOD; (B) THE POWER SPECTROGRAM OF THE LFP BAND-LIMITED TO 40HZ AND CONVERTED INTO DECIBEL IN THIS EXAMPLE. IT IS APPARENT THAT THERE ARE STRONGER THETA COMPONENTS AS IT PEAKS AT AROUND 8HZ AND 10HZ. -----138

FIGURE 3.48 ALGORITHM TO OBTAIN THE PHASES OF A UNIT LOCKED TO THE LFP SIGNAL. -----140

FIGURE 3.49 STEPWISE ILLUSTRATION OF CALCULATING THE SPIKE TO LFP PHASE LOCKING WITHIN A PARTICULAR BAND OF LFP SIGNAL IN THE FREQUENCY RANGE $[F_{MIN}, F_{MAX}]$; (A) THE BLUE LINE SHOWS THE HIGH-PASS FILTERED LFP SIGNAL L. RED LINE SHOWS THE BAND-PASS FILTERED SIGNAL B; (B) BLUE LINE IS THE B SIGNAL. RED LINE SHOWS THE AMPLITUDE OF THE SIGNAL DERIVED FROM HILBERT TRANSFORM. BLACK DOTTED LINE IS THE CROSSING-LINE TO DETERMINE SEGMENTS FOR FURTHER ANALYSIS. GREEN DOTS ARE CROSSING POINTS, WHERE GREEN ARROW SHOWS A SEGMENT THAT IS LONGER THAN $1/F_{MIN}$ OR THE MINIMUM TEMPORAL LENGTH TO REPRESENT THE SLOWEST COMPONENT OF THE B SIGNAL; (C) POWER SPECTRAL DENSITY MEASUREMENT FOR THE L SIGNAL. THE POWER WITHIN THE B-BAND IS SUM OF THE POWER BETWEEN $[F_{MIN}, F_{MAX}]$ AND MUST BE GREATER THAN A SET PERCENTAGE (PRATIO) OF THE TOTAL POWER; (D) ONCE THE SEGMENT IS VERIFIED TO CARRY SIGNIFICANT BAND-POWER AND HAVING PEAK-TO-PEAK (P2P) AMPLITUDE COMPARABLE TO THE OVERALL P2P AMPLITUDE OF THE L SIGNAL, PHASES OF THE SPIKES WITHIN THAT SEGMENT (RED STEMS) ARE CALCULATED FROM THE PHASES (BLUE DOTTED LINES) OF THE B SIGNAL USING HILBERT TRANSFORM (RED DOTS); (E) THE RASTER (TOP ROW) SHOWS THE PHASES OF THE SPIKES AT EACH

SEGMENT (Y-AXIS), AND THE BOTTOM ROW SHOWS THE COUNTS AT DIFFERENT PHASE-ANGLES; (F) THE BOTTOM ROW IN (E) IS SHOWN ALONG WITH A COSINE CURVE TO DISTINGUISH THE PORTION OF THE DISTRIBUTION WITH PHASE-PREFERENCE. THE INSET IN THE FIGURE SHOWS A CIRCULAR PLOT OF THE SAME DISTRIBUTION ALONG WITH THE MEAN PHASE (RED LINE). -----	141
FIGURE 3.50 FLOW CHART FOR CALCULATION OF SPIKE-FIELD COHERENCE (SFC) AND PHASE-LOCKING VALUE (PLV) -----	143
FIGURE 3.51 STEPWISE ILLUSTRATION OF MEASURING THE PHASE-LOCKING METRICS; (A) A SEGMENT OF THE LFP IS CUT OUT, CALLED SPIKE-TRIGGERED LFP, FOR A WINDOW OF, FOR EXAMPLE, [-0.5, 0.5] SEC CENTERED AROUND THE SPIKING TIME. THE RED STEMS SHOW THE SPIKES AND THE BLUE TRACES ARE THE LFP SEGMENTS; (B) EACH SEGMENT IS MULTIPLIED BY A HANNING WINDOW AND THE FOURIER TRANSFORMATIONS ARE OBTAINED. THESE GRAPHS SHOW THE POWER SPECTRAL DENSITY (PSD) OBTAINED FROM THE FOURIER TRANSFORM; (C) THE LFP SEGMENTS ARE AVERAGED TO OBTAIN SPIKE-TRIGGERED AVERAGE (STA); (D) POWER SPECTRUM OF THE SEGMENTS ARE AVERAGED TO OBTAIN THE AVERAGE SPECTRUM (STP); (E) THE POWER SPECTRUM OF THE STA; (F) THE SPIKE-FIELD COHERENCE (SFC) MEASURED FROM FSTA AND STP. -----	144
FIGURE 3.52 RESULTS OF DIFFERENT PHASE-LOCK METRICS; (A) POWER SPECTRUM OF THE SPIKE-TRIGGERED AVERAGE. INSET SHOWS THE STA; (B) AVERAGE OF THE POWER SPECTRUM OF INDIVIDUAL LFP TRACES; (C) SPIKE-FIELD COHERENCE; (D) PHASE-LOCKED VALUE. ALL THESE GRAPHS SHOW THAT THERE IS A STRONG COUPLING OF THE SPIKING ACTIVITY TO THE UNDERLYING LFP SIGNAL AT 9HZ. -----	145
FIGURE 3.53 TIME-RESOLVED PHASE-LOCK METRICS; (A) POWER SPECTRUM OF THE SPIKE-TRIGGERED AVERAGE; (B) THE SPIKE-FIELD COHERENCE; (C) THE PHASE-LOCKING VALUE;-----	146
FIGURE 3.54 PHASE-LOCK METRICS USING BOOTSTRAP TECHNIQUE; THE MEAN (DOTTED) \pm SEM OF THE (A) POWER SPECTRUM OF THE SPIKE-TRIGGERED AVERAGE; (B) AVERAGE OF THE POWER SPECTRUM OF INDIVIDUAL TRACES; (C) SPIKE-FIELD COHERENCE; (D) PHASE-LOCKING VALUE; IN THIS PARTICULAR EXAMPLE, THE SEM VALUES ARE VERY SMALL, AND THEIR SHADED PLOTS ARE NOT RECOGNIZABLE. -----	147
FIGURE 3.55 POPULAR FILE FORMATS USING THE HDF5 FORMAT. -----	153
FIGURE 3.56 THE HIERARCHY USED FOR DATA STORAGE IN THE HDF5 FILE FORMAT -----	154
FIGURE 4.1 THE SPATIOTEMPORAL DOMAIN OF NEUROSCIENCE AND OF THE MAIN METHODS AVAILABLE FOR THE STUDY OF NERVOUS SYSTEM IN 2014. EACH COLORED REGION SHOWS THE EXTENT OF SPATIAL AND TEMPORAL RESOLUTION FOR A METHOD. FILLED BOXES SHOW PERTURBATION TECHNIQUES FOR ALTERNATING THE FUNCTIONS OF NEURONAL CIRCUITRY AND THE UNFILLED BOXES SHOW THE MEASURING TECHNIQUES. INSET SHOWS A CARTOON RENDERING OF THE METHODS AVAILABLE IN 1988. (ADAPTED FROM SEJNOWSKI AND CHURCHLAND., 2014). EEG= ELECTROENCEPHALOGRAPHY; MEG= MAGNETOENCEPHALOGRAPHY; PET= POSITRON EMISSION TOMOGRAPHY; VS= VOLTAGE-SENSITIVE DYE; TMS= TRANSCRANIAL MAGNETIC STIMULATION; 2-DG, 2-DEOXYGLUCOSE. (ADAPTED FROM SEJNOWSKI ET AL., 2014) -----	165
FIGURE 4.2 SUMMARY OF WHY OPEN DATA INITIATIVE IS NECESSARY-----	166
FIGURE 4.3 LIST OF OPEN DATA TECHNICAL REQUIREMENTS -----	168

FIGURE 4.4 LIST OF OPEN DATA ECOSYSTEM REQUIREMENTS -----	168
FIGURE 4.5 LANDSCAPE OF DATA SHARING IN NEUROSCIENCE AND THE ISSUES TO ADDRESS. THESE ISSUES CAN BE GROUPED INTO THREE CLUSTERS- RED, BLUE AND GREEN- EACH FOR THREE DIFFERENT CATEGORIES: 1. CULTURAL, TECHNICAL AND PRACTICAL, 2. NATURE OF THE DATA AND RESOURCES, AND 3. MARKETPLACE OR ECOSYSTEM. (ADAPTED FROM WIENER ET AL., 2016)-----	170
FIGURE 4.6 STANDARD PIPELINE FOR THE PROCESSING OF BEHAVIORAL NEUROPHYSIOLOGY DATA FROM RECORDING TO PUBLICATION.-----	179
FIGURE 0.1: NEUROCHAT GRAPHICAL USER INTERFACE-----	186
FIGURE 0.2: SELECT THE INPUT FILE FORMAT FROM ONE OF THE THREE OPTIONS IN THE 'INPUT DATA FORMAT' DROPDOWN MENU.-----	187
FIGURE 0.3: 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.-----	188
FIGURE 0.4 DROPDOWN LIST FOR THE SELECTION OF CELL NUMBER.-----	189
FIGURE 0.5: DROPDOWN LIST TO SELECT THE LFP CHANNEL-----	190
FIGURE 0.6 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.-----	192
FIGURE 0.7 'FILE' MENU ITEMS. SHORTCUTS FOR INVOKING EACH OF THESE ACTIONS ARE ALSO SHOWN.-----	193
FIGURE 0.8: WINDOW FOR ANALYSIS SPECIFIC INPUT PARAMETER SETTINGS.-----	194
FIGURE 0.9 ITEMS IN THE 'UTILITIES' MENU.-----	195
FIGURE 0.10 SAMPLE EXCEL FILE FROM THE EXPORT OF GRAPHICS FILE INFORMATION.-----	195
FIGURE 0.11: 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-----	196
FIGURE 0.12: SAMPLE INPUT FORMAT AND OUTPUT IN 'VERIFY UNITS' UTILITY.-----	197
FIGURE 0.13: SAMPLE EXCEL FILE INPUT FOR CLUSTER EVALUATION. THE ANALYSIS OUTPUTS ARE ADDED AT THE TAIL OF EACH ROW OF DATA.-----	197
FIGURE 0.14: SAMPLE EXCEL FILE INPUT FOR CLUSTER SIMILARITY MEASUREMENT. TWO SETS OF SPECIFICATIONS ARE REQUIRED FOR COMPARISON. THE ANALYSIS OUTPUTS ARE ADDED AT THE TAILS OF EACH SET OF SPECIFICATIONS OF THE COMPARING UNITS.-----	198
FIGURE 0.15 SAMPLE EXCEL INPUT FOR CONVERTING THE RECORDED DATA TO HDF5 FORMAT.-----	198

LIST OF TABLES

TABLE 2.1 HIERARCHY IN THE BEHAVIORAL MECHANISMS OF SPATIAL NAVIGATION (FRANZ AND MALLOT, 2000)-----	12
TABLE 2.2 SUMMARY OF THE SCOPES AND FUNCTIONALITIES OF AVAILABLE TOOLS AND SOFTWARE FOR SINGLE-UNIT DATA ANALYSIS-----	58
TABLE 3.1 LIST OF CLASSES IN NEUROCHAT SOFTWARE AND THE MODULES THEY BELONG TO.-----	87
TABLE 3.2 SUMMARY OF THE ANALYSES DEVELOPED IN NEUROCHAT; HD= HEAD DIRECTION OF THE ANIMAL, BORDER= DISTANCE FROM THE BORDER OF THE FORAGING ENVIRONMENT. -----	89
TABLE 3.3 NAMING CONVENTIONS OF DATA FORMATS SUPPORTED IN NEUROCHAT. -----	148
TABLE 3.4 NEUROCHAT-GENERATED HDF5 FILE DATA PATHS AND ITEMS COMPARABLE TO THE NWB FORMATS. -----	154

1 Thesis Outline

Chapter 2 provides the contextual framework for the thesis with a description of the different stages on our way to build our project objectives. Chapter 2.1 is an overview of different concepts of navigation. It defines navigation, different mechanisms that animal follows to perform the navigational tasks and the concept of referencing in navigation. It also describes the concept of spatial memory and cognitive map- how the spatial information is represented in the nervous system for storage, maintenance, and updating etc. Chapter 2.2 outlines the principles of neurophysiology and recording of such data as well as the detection and identification of single units- putatively the activity of individual neurons. Chapter 2.3 introduces different toolboxes used in the analyses of neurophysiology data across research fields. We provide a logical grouping of the toolboxes and briefly describe the scopes, usage or limitations of such toolboxes. At the end of this chapter, we describe the objectives that we pursued throughout the project. The schematic of the chapter is given below:

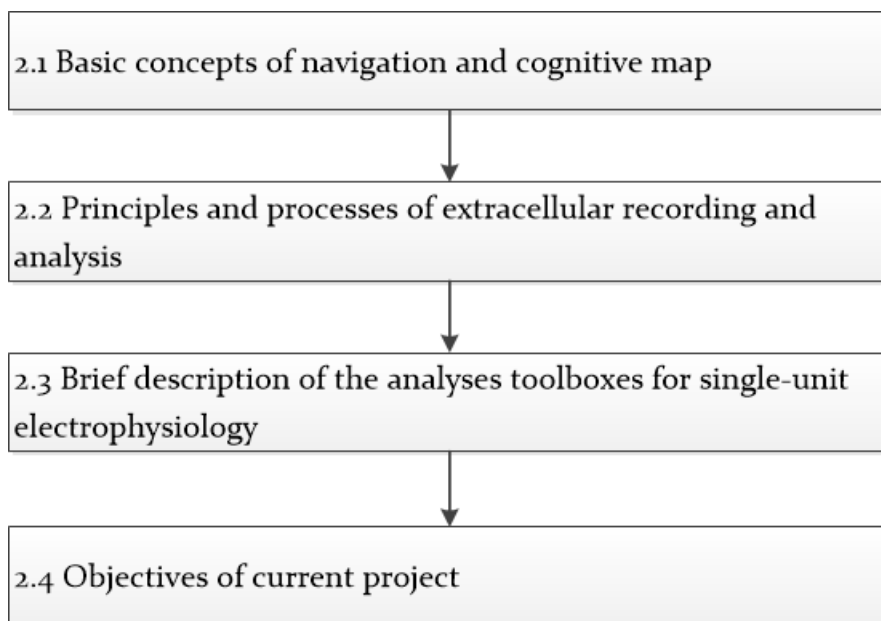


Figure 1.1 The schematic of Chapter 2

Chapter 3 discusses the ***Neuron Characterization Toolbox***, or NeuroChaT- the software and the application programming interface that enables us to achieve the project objectives in a systematic and user-friendly way. Chapter 3.1 describes the basic architecture of the toolbox, connectivity between different components of the

toolbox, and the flow of information among them. Chapter 3.2 provides a more detail explanation of the design concepts, mostly from software development perspectives. In this section, we have shown class diagrams and explained the design patterns in different layers of the programming. In section 3.3, we described the analysis methods-grouping them into logical subsections. Chapter 3.4 explains different utilities that help make the data management easier through NeuroChaT and attain the objectives of bringing efficiency, controlling the quality at certain aspects and ensuring integrity in data analysis pipeline. The schematic of the chapter is given below:

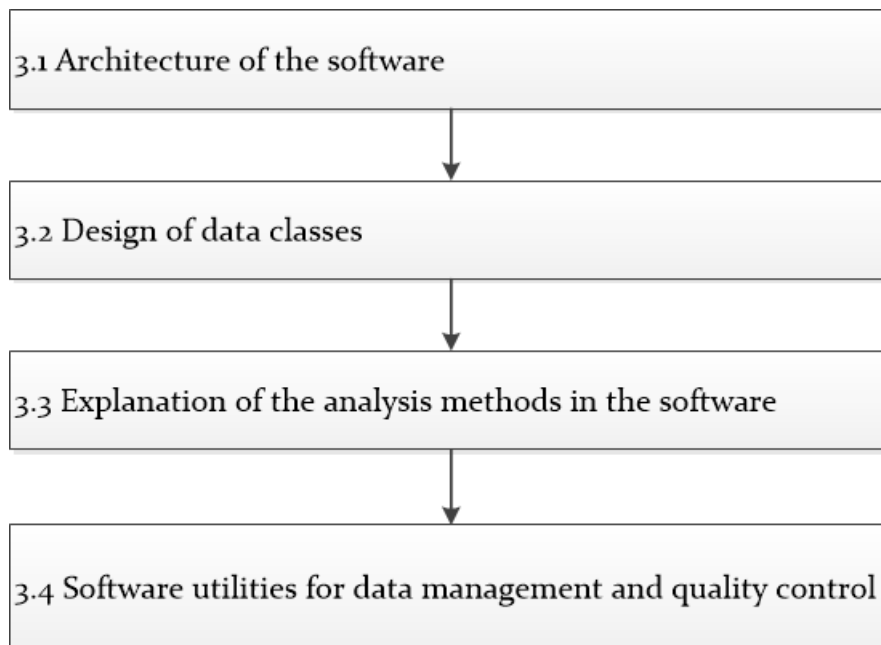


Figure 1.2 Schematic of Chapter 3

Chapter 4 discusses a project that we undertook to further reduce the cognitive efforts for analyzing data by automatizing the clustering or isolation of individual units or neurons. Chapter 4.1 briefly explains why automated spike sorting algorithm is necessary and the reasoning behind studying a new clustering algorithm. Chapter 4.2 explains the algorithm that laid the foundation of the algorithm we developed. Chapter 4.3 provides a detail description of the algorithm and its evaluation on simulated and real datasets. Chapter 4.4 discusses different quality assessment

metrics used for spike-sorting and their use in assessing the stability of chronic recording of single units. The chapter ends with a brief description of future works and relevant aspects of spike sorting. The schematic of the chapter is given below:

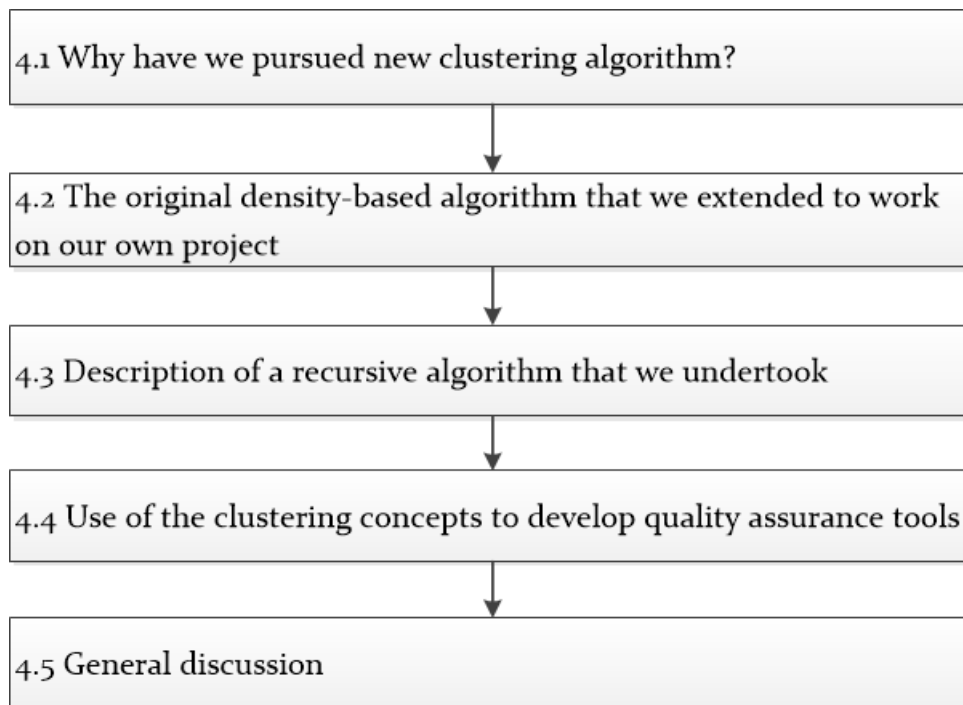


Figure 1.3 Schematic of Chapter 4

Chapter 5 presents different use cases of NeuroChaT as an electrophysiology toolbox. Chapter 5.1 shows how a single-unit can be identified and verified to be responsive to the spatial information of the animal in the foraging environment. Chapter 5.2 gives an example of how NeuroChaT can be used to assess the rhythmicity of neuronal activity of a single unit and the relationship to another neurophysiological measurement called local field potential (LFP). Chapter 5.3 describes the use of NeuroChaT in an experiment where the nature of correlation of the single-unit physiology to that of the spatial behavior of the foraging rodents were modulated by the external experimental manipulations. The schematic of the chapter is given below:

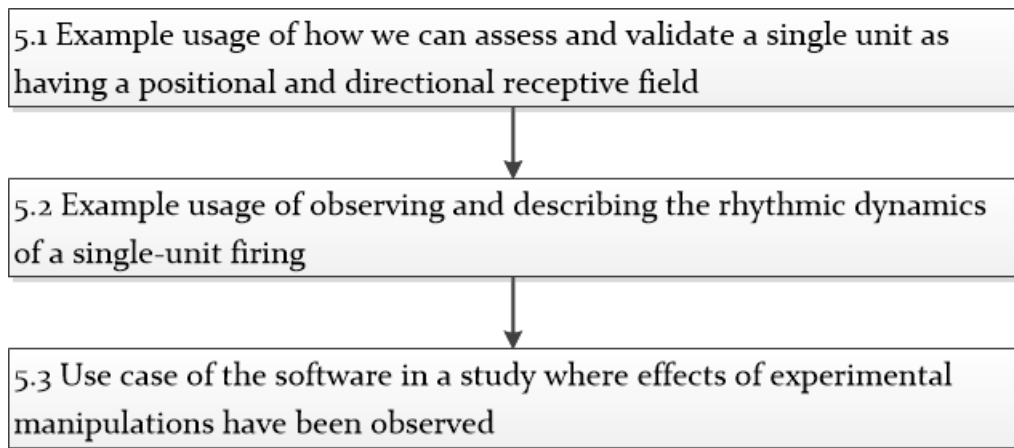


Figure 1.4 Schematic of Chapter 5

Chapter 6 provides a general discussion on NeuroChaT and its different aspects. Chapter 6.1 explains how NeuroChaT successfully attained the project objectives which were defined to address the problem domains. Chapter 6.2 discusses the requirements and challenges of openness initiative in research and how NeuroChaT aligns itself with the concept of open research. Chapter 6.3 explains the data-integrity and practice of open science in NeuroChaT. This chapter ends with a short description of potential developments of NeuroChaT in future. The schematic of the chapter is given below:

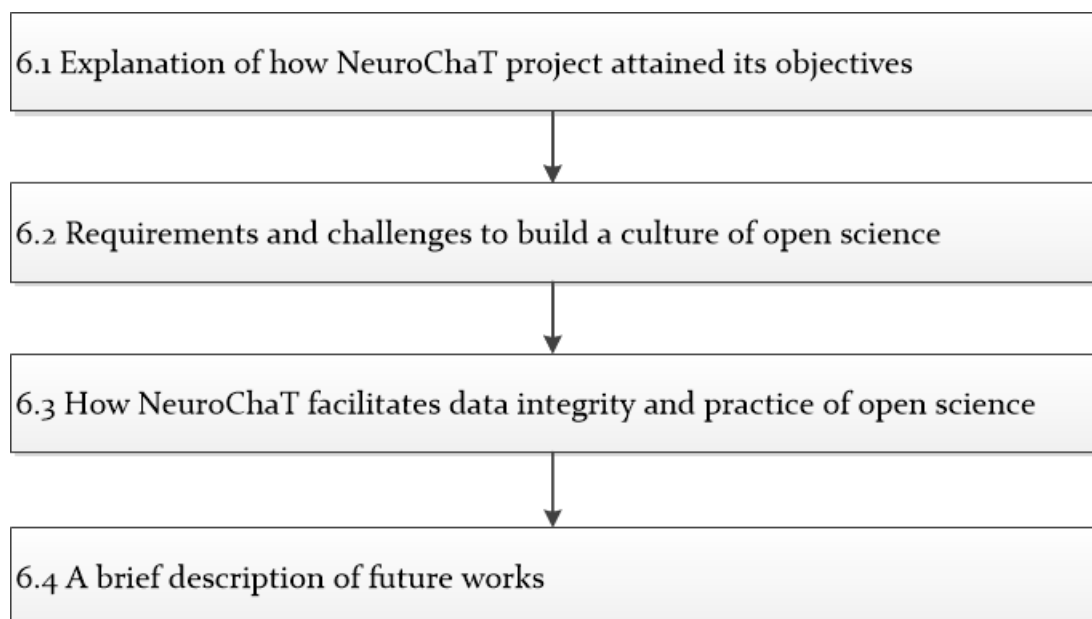


Figure 1.5 Schematic of Chapter 6

NeuroChaT has been extensively used by the neurophysiologists in Experimental Brain Research Lab led by Prof Shane O'Mara and the laboratory of Dr Marian Tsanov. The software has been tested and used by at least 11 researchers in these two laboratories. The feedback from the users have been crucial in designing the software, its graphical interface, which analysis and parameter to include, and how the data and results should be presented. The prototype or the proof of concept has been developed in MATLAB® before the extensive development in Python programming language. The software underwent evolution as new requirements arose in terms of analysis and user experiences. The user-friendly software succeeded in analyzing data very quickly which resulted in a number of publications that are listed at the end of this chapter. NeuroChaT or the methods developed in the software were used for data analysis in these publications. The author of this thesis contributed by tailoring the analysis to be used in a particular experiment while also contributing to interpreting the results and writing the papers. Specific contributions can be found in relevant section of each of these publications.

List of relevant publications and presentations:

* Equal contributions by authors

Journal Publications:

- 1 Jankowski MM*, **Islam MN***, Wright NF, Vann SD, Erichsen JT, Aggleton JP, **O'Mara SM** (2014). Nucleus reuniens of the thalamus contains head direction cells. *eLife*, doi: **10.7554/eLife.03075**
- 2 Jankowski MM, Passecker J, **Islam MN**, Vann SD, Erichsen JT, Aggleton JP and **O'Mara SM**. (2015) Evidence of spatially responsive neurons in rostral thalamus. *Frontiers in Behavioral Neuroscience* 9:256, doi: **10.3389/fnbeh.2015.00256**
- 3 Jankowski MM*, **Islam MN*** and **O'Mara SM**. (2017) Dynamics of spontaneous local field potentials in the anterior claustrum of freely moving rats. *Brain Research*, doi: **10.1016/j.brainres.2017.09.021**

- 4 Mamad O, **Islam MN**, Cunningham C, Tsanov M. (2018) Differential Response of hippocampal and prefrontal oscillations to systematic LPS application. *Brain Research*, doi: [10.1016/j.brainres.2017.12.036](https://doi.org/10.1016/j.brainres.2017.12.036)
- 5 Passecker J, **Islam MN**, Hok V and **O'Mara SM**. (2018) Influence of photic stress on postsubicular head-directional processing (Accepted for publication in *European Journal of Neuroscience*)
- 6 Matulewicz PM, Ulrich K, **Islam MN**, Mathiasen ML, Aggleton JP, and **O'Mara SM**. (2018) Proximal perimeter encoding in the rat rostral thalamus (submitted for peer review)
- 7 **Islam MN**, **O'Mara SM** (2018) Recursive fast-search and find of density peaks algorithm for spike sorting from extracellular neuronal recordings (In preparation)
- 8 **Islam MN**, **O'Mara SM** (2018) NeuroChaT: An interactive analysis toolbox for behavioral correlates of neural ensembles (In preparation)
- 9 Wynne PJ, **Islam MN**, Jankowski MM and **O'Mara SM**. (2018) Boundary and Head directional encoding in anterior retrosplenial cortex (In preparation)

Conference presentations:

- 1 **Islam MN**, **O'Mara SM** (2017) Recursive fast-search and find of density peaks algorithm for spike sorting from extracellular neuronal recordings. In *Proceedings of the British Neuroscience Association Festival of Neuroscience*, Birmingham, United Kingdom, April 10-13, 2017.
- 2 Dillingham DM, **Islam MN**, Chandra R, Jankowski MM, **O'Mara SM** (2017) High-voltage spindle oscillation episodes in the rat claustrum. In *Proceedings of the British Neuroscience Association Festival of Neuroscience*, Birmingham, United Kingdom, April 10-13, 2017.

- 3 **Islam MN, O'Mara SM** (2016) Spike-sorting using density-based clustering around fuzzy cores. In *Proceedings of the Young Neuroscientists Symposium 2016*, Dublin, Ireland, September 1, 2016.
- 4 **Islam MN, O'Mara SM** (2016) NeuroChaT: An interactive analysis toolbox for behavioral correlates of neural ensembles. In *Proceedings of the 10th FENS Forum of Neuroscience Meeting*, Copenhagen, Denmark, July 2-6, 2016.
- 5 Ulrich K, Matulewicz PM, **Islam MN**, Aggleton JP, **O'Mara SM** (2016) Are cells within rostral thalamus sensitive to environmental changes? In *Proceedings of the 10th FENS Forum of Neuroscience Meeting*, Copenhagen, Denmark, July 2-6, 2016.
- 6 Matulewicz P, Ulrich K, **Islam MN**, Aggleton JP, **O'Mara SM** (2016) Spatial properties of perimeter units in the parataenial nucleus of the thalamus. In *Proceedings of the 10th FENS Forum of Neuroscience Meeting*, Copenhagen, Denmark, July 2-6, 2016.
- 7 Wynne PJ, **Islam MN**, Jankowski MM, **O'Mara SM** (2016) Spatial and theta-modulated units are present in the rodent anterior retrosplenial cortex. In *Proceedings of the 10th FENS Forum of Neuroscience Meeting*, Copenhagen, Denmark, July 2-6, 2016.

2 Background

This chapter provides a systematic review of the research concepts relevant to the development of NeuroChaT software. A brief description of the concept of navigation and how the spatial information is presented in the neuronal ensembles is provided in Chapter 2.1. This section, therefore, presents the importance and aspects of the cognitive mapping as a fundamental area of research for broadening our understanding of how the brain works. It also provides with information about the scopes of cognitive map research. Chapter 2.2 illustrates the principles and concepts of neurophysiology recordings that enable us to examine the activity of the brain in neuronal level. Understanding this activity in relation to the navigational information formed the basis set of what NeuroChaT should put emphasis on. This chapter also explains different challenges in such recordings and sheds light on the importance of developing a software like that of NeuroChaT. Chapter 2.3 summarizes the ongoing efforts by computational neuroscientists to develop toolboxes that address the challenges and need of the modern-day neurophysiology research. The review of the toolboxes also provides us the opportunity to identify the necessity of developing NeuroChaT and its contribution in mitigating the capacity gap in relevant researches. At the end of this chapter, the reader will have an overall understanding of the area of research we are interested in where the representations of the brain's global positioning system are studied. This also follows the understanding of the relevant concepts, principles, challenges and opportunities to explore.

2.1 Navigation

2.1.1 Concept of navigation

What is Navigation

Mobile behavior such as exploration, or obstacle avoidance are primitive survival skills that animals must possess to ensure survival. One such behavior is navigation which is, in a formal definition, 'the process of determining and maintaining a course or directory to a goal direction' (Franz and Mallot, 2000). Therefore, the minimal capabilities to navigate are to move in space and to determine whether or not the goal has been reached. It is distinguished from other spatial behaviors like foraging, exploration, body orientation by the factor that it requires a goal location.

The behavioral actions driving navigation follow a hierarchy based on the availability of the information. The hierarchy is summarized in **Table 2.1** (Franz and Mallot, 2000). Local navigation requires one particular goal location perceived by sensory information, while way finding involves recognition of several places and the representation of the relations between these places to reach a goal. Local navigation is further divided into four categories: search, direction following, aiming, and guidance. On the other hand, the way-finding incorporates three different levels: recognition-triggered response, topological and survey/metric navigation. These behavioral classifications are based on the level of task complexity. For example, in a 'search' of a goal, the only competence required by the animal is to perceive the goal, while in a 'direction-following', the animal knows the direction of the goal, but not the exact location of where the goal can be perceived (see Franz and Mallot, 2000, Trullier et al., 1997 for details)

Mechanism of Navigation

While navigating, the animal needs to move from one place to another and may change its direction as well. In such actions, the spatial relationship between the animal and its environment changes constantly. Spatial updating is a cognitive process that computes the spatial relationships between an animal and its surrounding environment as it moves based on the perception of its own motion or

'self-motion' (Wang et al., 2006). The basic form of spatial updating is the *path integration* or dead-reckoning where the location of an origin relative to the animal's current position and orientation are computed. Such path integration mechanism has been observed in a study by (Mittelstaedt and Mittelstaedt, 1980) where they used the strong drive of the mother rat to retrieve a pup displaced from the nest in darkness. A pup was removed from the nest (starting location) to a location within the adjacent arena. The mother left the nest and searched for the pup giving rise to a complex outgoing path involving many turns. Once it found the pup, it returned directly to the nest. The experiment was repeated and when the mother left the nest, the arena was rotated slowly. This resulted in a return trajectory error of the same magnitude of the arena rotation. (Mittelstaedt and Mittelstaedt, 1980) argued that, the rat, in its outgoing journey, calculated its distance and orientation from the origin to take the shortcut in its return journey. Spatial updating, in its advanced form, keeps track of multiple targets in the environment and estimates the new relationship to the environment as the animal moves (Collett et al., 1999).

Concept of Egocentric and allocentric navigation

In a navigation task (or spatial updating), animals ascertain their own location (distance, orientation, etc.), plans for a destination and follow the route. To perform the process successfully, animals need to possess knowledge of the space that they belong (spatial knowledge). The self-localization and the planning of the spatial actions generally follows measuring the distances and directions (Poucet, 1993) with respect to a reference frame. Based on the selection of this reference frame, there are two systems of spatial representations defining two approaches for spatial navigation (Wang et al., 2006)-

- **Egocentric or self-to-object:** The distance and orientation of an object are measured with respect to oneself (**Figure 2.1a**). This approach is essential in local navigation for body alignment and to reach the goal.
- **Allocentric or object-to-object:** The location of one object is measured with respect to the location of another object (**Figure 2.1b**). For example, exploring the environment involves learning how the landmarks are placed with respect to each other, and thus, updating the perspective as the animal moves.

Table 2.1 Hierarchy in the behavioral mechanisms of spatial navigation (Franz and Mallot, 2000)

Action	Behavioral prerequisite	Navigation competence
	Local Navigation	
Search	Goal recognition	Finding the goal without active goal orientation
Direction-following	Align course with local direction	Finding the goal from one direction
Aiming	Keep goal in front	Finding a salient goal from a catchment area
Guidance	Attain spatial relation to the surrounding objects	Finding a goal defined by its relation to the surroundings
	Way finding	
Recognition-triggered response	Association sensory pattern-action	Following fixed routes
Topological navigation	Route integration, route planning	Flexible concatenation of route segments
Survey navigation	Embedding into a common reference frame	Finding paths over novel terrain

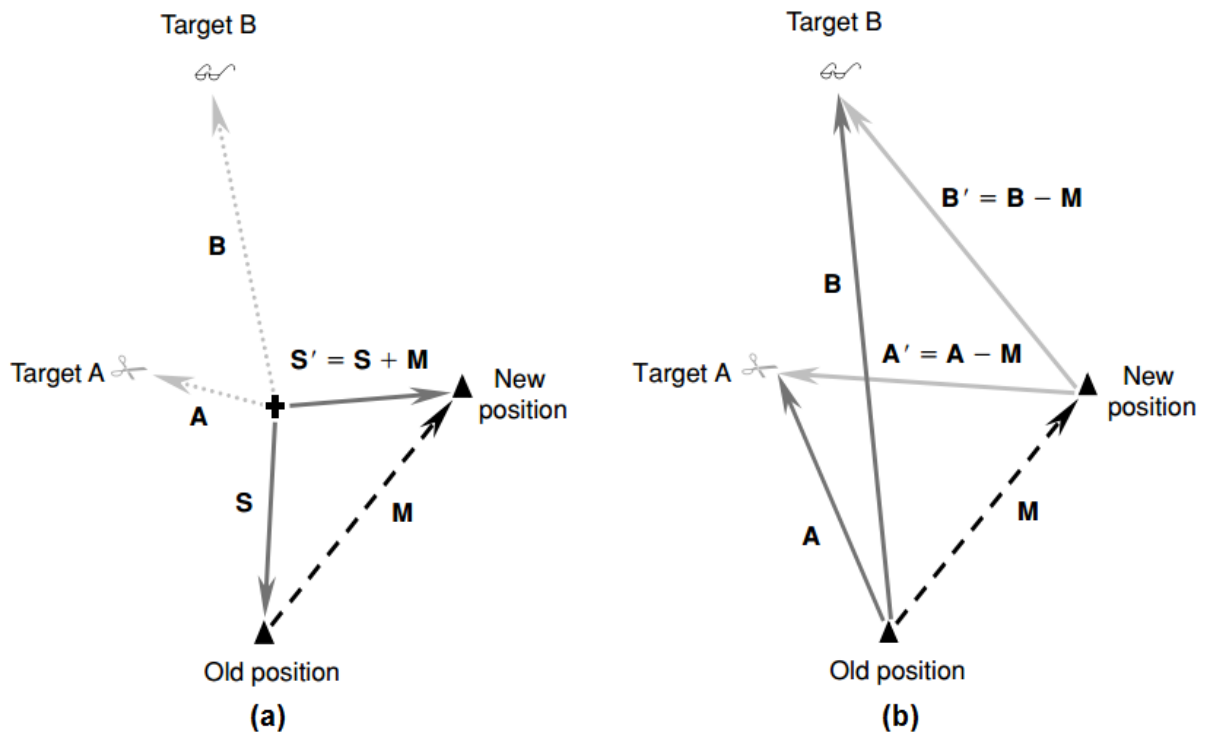


Figure 2.1 An illustration of two hypotheses of spatial updating. Panel (a) illustrates an allocentric model in which the central cross represents an external reference system. The target locations are represented as A and B and remain the same as the observer, represented by the filled triangle, moves from the old position to the new one. Spatial updating involves computation of the observer's new position, S' , in the external reference system according to the old position S and the movement vector M ($S' = S + M$). Panel (b) illustrates an egocentric model in which the target positions (A and B) are represented relative to the observer. Spatial updating involves computing the new egocentric positions (A' and B') of the objects as the observer moves ($A' = A - M$; $B' = B - M$). (Wang et al., 2006)

2.1.2 Spatial memory and cognitive map

As the spatial navigation mechanisms, i.e. path integration, depends on the storage of the spatial information, or requires that a representation of the spatial relationships be present *a priori* (e.g. finding shortcuts and detours; Trullier et al., 1997), it intrigued the idea of the presence of spatial memory and a map in the brain. Such an idea was first suggested by (Tolman, 1948) where he studied how animals learn to navigate. Although the concurrent behaviorist approach suggested that the chains of 'stimulus-response' relationships form complex behaviors, Tolman argued that the exploration of the environment gradually resulted in the formation of a cognitive map that enabled animals to navigate and find the optimal path through the environment. For example, in one experiment, two groups of rats were trained to find food reward (reinforcement) at two locations respectively in the arms of a T-maze, where the starting location was placed at the other end of the T-arm across a round table. After

the training session, the table and the T-maze arms were rotated 180-degree. Tolman showed that the rats learned to turn right to obtain the reward now preferentially turned left and vice versa, implying that the rats had created a cognitive map of the maze during the training session, and used the map to retrieve the reward when the maze orientation was altered. The presence of spatial memory and cognitive maps has also been explained by other experimental approaches. For example, in the Morris water maze task (Morris, 1981), rodents are placed in a large circular pool of water containing a hidden platform under the water. The animal needs to escape from the water onto the platform whose location is randomized with respect to the start location of the animal. The decrease in escape time (time to get into the platform) over trials, finally leading to a directed path navigation, which does not prevail in lesion studies (Morris et al., 1982) indicates that the special localization does not require presence of local sensory cues for navigation.

However, where and how in the brain these maps are being represented were not addressed until the advent of techniques that enabled neuroscientists to record from brains of freely moving animals using chronically implanted microwires (Strumwasser, 1958) to obtain cellular level activity of individual neurons in response to the spatial stimuli (e.g. location, head direction etc.). The following section introduces such discoveries that led to the winning of Nobel Prize in Medicine and Physiology in 2014.

Neuronal encoding of spatial information

Neuronal cells in different brain areas discretely represent spatial information. Some of them represent a specific location in the environment and some of them represent a particular direction. Information regarding egocentric movement, such as running speed, are also represented. A brief description of such encoding paradigms is described below.

Place cell and hippocampus

The experimental evidence of the spatial representation of brain was first described in (O'Keefe and Dostrovsky, 1971). They reported the presence of spatial receptive fields, called place fields, in hippocampal neurons. When a rat forages in an open environment, such neurons, known as place cells, show spiking activity only when the animal is residing in a certain portion of the environment. The neighboring place cells fire at different locations as if the local cell population, throughout the environment, represents the entire environment, and it is possible to estimate the position of a rat using the activity of these cells (Wilson and McNaughton, 1993). The pattern of spiking activity remains highly correlated when recorded in consecutive days in the same environment, but a 'remapping' occurs when moved to a new environment (O'Keefe and Conway, 1978). The activity of these cells are also highly modulated by idiothetic, e.g. vestibular, cues (Sharp et al., 1995) or other sensory cues apart from the visual stimulus (Etienne et al., 2000). Inspired by the suggestion of (Tolman, 1948), that local navigation is guided by an internal 'cognitive map' that flexibly represents the overall spatial relationships between landmarks in the environment, (O'Keefe and Nadel, 1978) proposed that place cells are the basic elements of the distributed and non-centered map-like representation, where hippocampus is the locus of the brain's internal map of the spatial environment. Place cells provide a continuously updated representation of the allocentric space and the animal's own position in that space. Although this suggests that the place cells contribute to forming the spatial working memory, longitudinal studies like (Alme et al., 2014), interestingly, report recurring place cell activities in the hippocampal neurons. This finding also suggests that a number of spatial representations are 'stored' in the brain region. Place-like cells have also been found in bat (Yartsev and Ulanovsky, 2013; 3D spatial cells), primate (Rolls and O'Mara, 1995; Spatial view cells) and human (Ekstrom et al., 2003).

Encoding of head-directional information

(Taube et al., 1990) later discovered the presence of cells in postsubiculum of subicular complex, adjacent to hippocampus, that discharge when an animal is

heading towards a particular direction of the environment, regardless of its location. These cells are called head-direction (hd) cells and can be found in many structures throughout the rodent brain (Taube, 2007). Apparently, they are similar to a compass, but unlike a magnetic compass that depends on the geomagnetism, such cells depend on landmarks, and self-motion represented by vestibular and proprioceptive cues (Yoder and Taube, 2014). These cells use the environment as the reference coordinate system using visual landmarks or other internal idiothetic cues i.e. vestibular signal. Their contribution in the spatial behavior is to dynamically maintain and correct spatial orientation of the animal in path integration task even in the absence of visual information (Valerio and Taube, 2012), although, in the presence of conflict between idiothetic cues and visual cues, these cells preferentially fire with respect to the visual cues (Zugaro et al., 2000).

Grid cell in entorhinal cortex

The idea of the presence of cognitive map was further strengthened by the fact that lesioning of the hippocampus resulted in the loss of spatial recognition in both human (Scoville and Milner, 1957) and rodents (Hollup et al., 2001, Broadbent et al., 2004). Interestingly, some other brain regions, such as entorhinal cortex, also participate to form a network for such representation. A new type of cell, called Grid cell, was discovered by (Fyhn et al., 2004), where the neurons have preferential activation in multiple place-fields. The place-fields of each neuron formed a periodic triangular array, or grid, that tiled the entire environment explored by the animal. Each grid is characterized by spacing (distance between place-fields), orientation (tilt of major axis with respect to an external reference), and phase (xy displacement relative to an external reference point). Grids of neighboring cells share a common orientation and spacing, but their vertex locations (their phases) differ. The spacing and size of individual fields increase from dorsal to ventral dorso-caudal medial entorhinal cortex. The map is anchored to external landmarks, but persists in their absence, suggesting that grid cells may be part of a generalized, path-integration-based map of the spatial environment (Hafting et al., 2005) and may contribute to the formation of a metric system for spatial navigation (Moser et al., 2008).

Discrete representation to spatial navigation

Place cells are predominantly found in the hippocampus, but were also identified in the brain areas like rostral thalamus (Jankowski et al., 2015) and in claustrum (Jankowski and O'Mara, 2015). Head directional information recruits even larger network of distributed interconnected brain regions (Taube, 2007). Grid cells are predominantly found in layer II of the medial entorhinal cortex (mEC). But the deep cortical layers provides a conjunctive representation of grid cells, head direction cells and grid-by-head directional cells, all of them correlating to the running speed of the animal, and projects to superficial layer II (Sargolini et al., 2006). Boundary cells, responsive to environment boundaries, either in particular allocentric direction (boundary vector cells or BVC) (Barry et al., 2006) or along all environment boundaries (perimeter or annulus cells) (Weible et al., 2012), are also found in number of brain areas (Grieves and Jeffery, 2017). This cell type is also accompanied by the cells that fire only in the center of an environment (boundary-off cells) in mouse anterior cingulate cortex (Weible et al., 2009). A schematic diagram of the cell-types and the brain regions where they are found is shown in **Figure 2.2**.

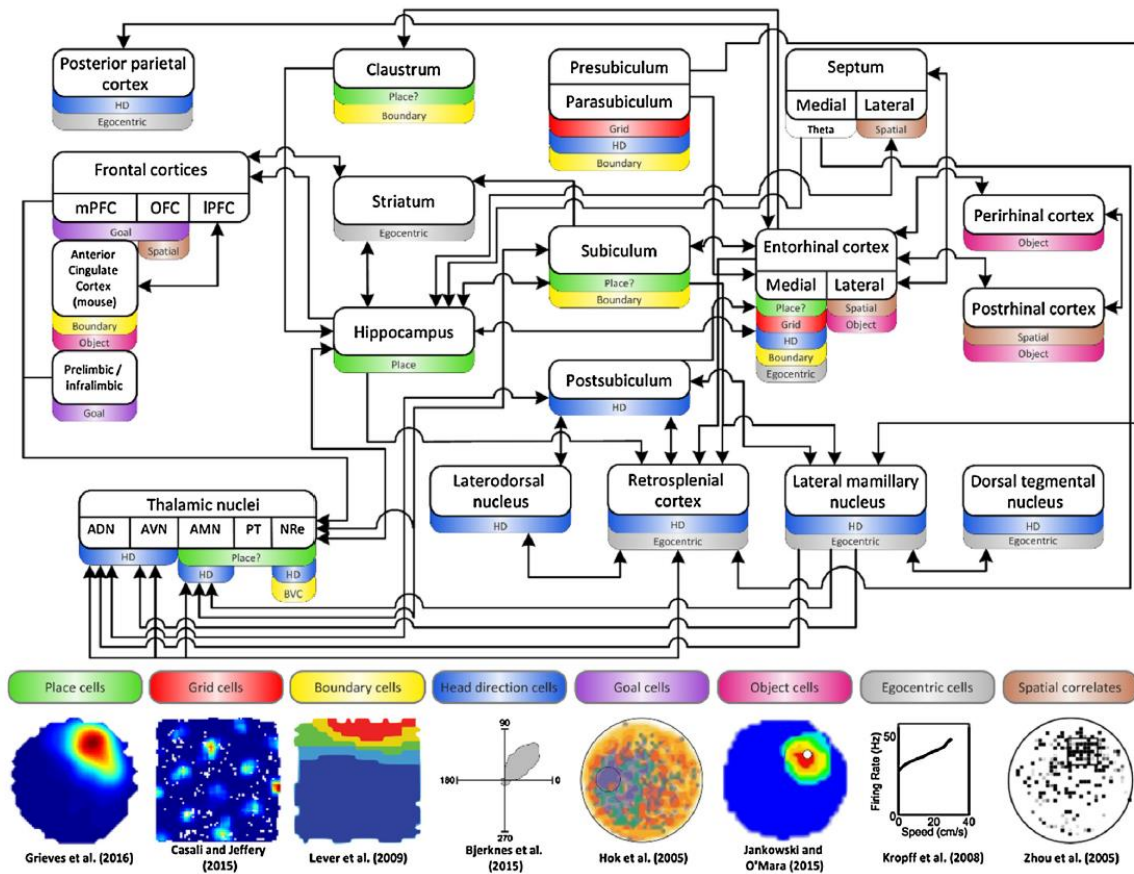


Figure 2.2 Schematic diagram concentrating on the brain regions and cell types discussed in this review. Place cells can be found in the hippocampus, nucleus reuniens (NRe), paratenial nucleus (PT), anteromedial nucleus (AM), claustrum, medial entorhinal cortex and subiculum. Place correlates (i.e. weak spatial activity) can be found in the orbitofrontal cortex (OFC), postrhinal cortex, lateral entorhinal cortex and lateral septum. Grid cells can be found in the medial entorhinal cortex, pre- and parasubiculum. Head-direction cells can be found in the lateral mamillary nuclei (LMN), anterodorsal nuclei (ADN), laterodorsal nuclei (LDN), retrosplenial cortex (RSC), postsubiculum, nucleus reuniens and anteromedial nucleus (AM). Boundary cells can be found in the parasubiculum, claustrum, subiculum, anterior cingulate cortex, pre- and parasubiculum and medial entorhinal cortex. Object sensitive cells can be found in the lateral entorhinal cortex, postrhinal cortex, orbitofrontal cortex (OFC) and the lateral septum. Goal cells can be found in the medial prefrontal cortex (mPFC) and prelimbic and infralimbic regions of the prefrontal cortex. Self-motion or egocentric cells such as those encoding running speed or angular head velocity can be found in the mEC, striatum, RSC, PPC, LMN and DTN. (Adapted from Grieves and Jeffery, 2017)

It is not, however, clear where the origins of these cells are or what are the inputs to their discharges. (Chersi and Burgess, 2015) suggested that self-motion and environmental sensory inputs are driving force for these spatially responsive cells which were supported by the evidence from cue-control studies of the cells. For example, BVC activity is driven by the environment walls, ridges or vertical drops, but is unaffected by the color, texture or odor cues present in the environment (Lever et al., 2009)- implying that these cells are responsive to more primitive environmental

features and are basic to survival strategy. In contrary, place cell activity is affected by distal cues and landmarks surrounding the environment and changes in completely novel environment (Anderson and Jeffery, 2003), and, therefore, is particularly important for allocentric navigation (Ekstrom et al., 2014), although a very recent study suggests that distance from a salient cue is sufficient to form place fields in preweaning rats even before the grid-cell system has matured (Bjerknes et al., 2018). Recognizing novel or familiar stimulus in the environment is also crucial for survival. Memory system required for navigation is comprised of spatial ('where') and nonspatial ('what') information (Knierim et al., 2014). There are cells which are active near specific objects in an environment, called object cells. Although it was assumed that these cells are encoding visual or tactile information (Burke et al., 2012), there are small population of cells in the rat anterior claustrum which fire persistently in darkness and if the object is replaced with another one, implying that they do not represent visual or other sensory features (Jankowski and O'Mara, 2015), and rather encode the spatial location of the objects. The allocentric navigation requires establishing a relationship between the environmental objects recognized by their location and features i.e., shape, size, color etc. requiring both location-related and view-related coding. On the other hand, the egocentric relationship requires that a representation of their relative movement to the environment be present for updating the distance or direction to the goal. Speed cells and angular head velocity cells provide processing of such egocentric information (Kropff et al., 2015, Bassett and Taube, 2001) and are very often present conjunctively with other cell types. For example, grid cells in mEC and head direction cells in rat lateral mammillary nucleus (LMN) are modulated respectively by the speed and angular head velocity of the animal (Sargolini et al., 2006, Stackman and Taube, 1998).

But how these discrete representations interpret to the navigation mechanisms, i.e. path integration, is yet to understand. (McNaughton et al., 2006) attempted to explain that an attractor dynamics in the medial entorhinal-hippocampal may contribute to path integration, but the model is questioned by the fact that grid cells form stable firing pattern quite late in the development phase compared to the place cell activity (Wills et al., 2010, Langston et al., 2010). Goal-directed spatial navigation recruits prefrontal-thalamo-hippocampal circuit, suggesting that the long-range

communication between cortical regions, beyond the local interaction in the hippocampus and entorhinal cortex, are involved in the navigation (Ito et al., 2015). Moreover, it is little known how the allocentric information combines to give rise to the navigation. It is complicated by the fact that there is no specific distinction between allocentric and egocentric information processing networks, and they are most often not mutually exclusive in higher order species (Ekstrom, 2015, Ekstrom et al., 2014). Mammalian allocentric strategy measures the direction and distances from the local objects and such vector representations are also implemented in the neural representations of space in object-vector cells (Hoydal et al., 2018). These cells are not results of the tactile sensation or blockade of trajectory and are elicited from the sensory landmark properties. In rodents, the lesion of Nucleus Reunions, a thalamic nucleus representing egocentric information (Jankowski et al., 2014), alters the allocentric hippocampal place cell stability (Cholvin et al., 2018).

The rhythmic properties of the neural signals also contribute to the spatial information processing. The head-directional cells are segregated in time by alternating theta cycles according to their directional preference, which indicates that the theta oscillation facilitate the segregation of information (Brandon et al., 2013). The location specific phase-segregation in hippocampus place cells reflects the distance representation by time-compression and is dependent on the speed of the animal (Geisler et al., 2007). Distinct environmental representations and the changes in context, i.e., location of reward, are also segregated across separate theta cycles (Jezek et al., 2011). These results support the hypothesis that there is a spatial information packaging by theta rhythms (Colgin, 2013). Inactivation of the medial septum, the pacemaker of theta rhythm, eliminates the periodicity of grid cells and reduces the number of theta-modulated cells in the entorhinal cortex (Brandon et al., 2011). Septal inactivation does not affect formation of place fields in novel environments when recorded in rodent hippocampus (Brandon et al., 2014) and the place cells in flying bats exhibit little to no such theta-modulation (Yartsev and Ulanovsky, 2013). These findings, along with other studies (Colgin, 2016), suggest that theta may not be required for the formation of spatial memory representations in single cell level, but is essential in sensory processing, coordination of neuronal ensembles participating in the distributed process of memory, coupling between

different brain regions, and formation of long-term memory. Travelling at a certain velocity for a certain time translates an animal by a particular distance. Therefore, timing is important for the spatial coding, and given that the theta rhythm varies in frequency with running speed (Shin and Talnov, 2001), and artificially altering the theta frequency affects the running speed of mice (Bender et al., 2015), there is also an specific locomotor role of this rhythmic component, particularly associated with the active movements (Vanderwolf, 1969).

Understanding the mechanistic principles of spatial navigation requires studying what other brain regions may contribute to the spatial circuitry, how the spatial cells receive information or what are their origin, are there any functional integration or specialization among the participating brain regions, and how the rhythmic components couple the information flow among different brain areas or represent spatial information processing, how different navigation mechanisms are represented in the brain, or whether there are any mechanistic dependency between such mechanisms etc. Although most of the studies have been conducted on a horizontal plane, real world space itself is a three-dimensional entity and presents a complex scenario of different known and novel object, views, perspective, optic flow, and– for species that can swim or fly– large volumetric spaces (Jeffery et al., 2015a). Therefore, recent studies have also shifted focus on how 3D spatial information and their temporal relationship are represented in neural ensembles (Page et al., 2018, Wilson et al., 2015). Studies also include use of virtual reality to mimic different navigational strategy and information presentation to the animal (Chen et al., 2013). Exploration of long-range navigation, three dimensional maps and compass, and factors affecting such representations are also being studied (Finkelstein et al., 2016, Geva-Sagiv et al., 2016).

The following section presents the concept of how the activities of the brain in such neuronal level are recorded and identified for individual neurons. It also briefly describes different schemes of how the neuronal encoding is studied. The challenges in such recordings are also outlined providing an idea of why the development of software like NeuroChaT is very much essential and timely for cognitive map research.

2.2 Extracellular neuronal recordings

Behavioral events are represented at cellular level in the nervous system (Allen et al., 2017, Hartley et al., 2014). Therefore, we need to understand the activity of individual neurons and the way these activities contribute to the neuronal circuitry. Neurophysiological techniques are used to record the bioelectrical/biophysical activities of individual neurons or ensembles. These methods record the activities intracellularly (patch clamp, voltage clamp etc.), or extracellularly from external matrix (single-unit and local field potential recording, electrocorticography etc.) by penetrating micropipettes, metal electrodes, or patches of microfabricated electrodes into the brain. A metal recording electrode may consist of a single insulated wire or a group of two or four wires called stereotrodes or tetrodes respectively. Such recordings can be done either *in vivo* from the living animal or *in vitro* using brain slices. Intracellular recordings are usually used to explore cell physiology and are mostly performed *in vitro*. Extracellular recordings can be both *in vivo* or *in vitro*.

Extracellular recording method allows recording the discharges or action potentials (AP) of a single neuron without impaling it as it records the AP from the very vicinity of a cell. Therefore, the activity of a single cell can be isolated and studied chronically and can thus be characterized comprehensively by correlating with the behaviors or physiological events and by studying their connectivity. Electrical, chemical or optical stimulation of identified cell populations at the recording site can also be used to determine the effects of cell activity on behavior or physiology. The neuronal APs of individual neurons are considered as the code used by the nervous system for information transmission. A number of coding schemes are used across different research areas in Neuroscience which are described in **Section 2.1.2**.

Extracellular electrodes record activity of a relatively small number of neurons in the implanted area and can provide information with high spatial and temporal resolution. But they are not capable of capturing the subthreshold events, such as synaptic potentials that may influence cell excitability but does not produce an AP. Intracellular or membrane level approaches with *in vitro* preparations are used for recording the subthreshold potentials.

2.2.1 Principles of extracellular recordings

(This section is based on (Heinricher, 2004) and (Logothetis, 2003) unless otherwise specified)

All the neuronal coding schemes involve the task of detecting whether or not an AP has occurred at a specific time. The waveforms and their characteristic changes depending on the morphology of recording sites or distance of recording electrodes form the basis for spike identification. It is, therefore, of particular interest to consider the basis for the APs obtained with an extracellular microelectrode.

Extracellular microelectrodes record the APs or *spikes* produced by the currents that flow in the extracellular space around an active neuron. This is explained by the volume conduction theory (Rall, 1962) where the extracellular medium surrounding a neuron works as low uniform resistance or a volume conductor. The specific impedance of this conductor is higher ($\sim 200\text{--}400\ \Omega/\text{cm}$, depending on neural site) (Ranck, 1966, Nicholson and Freeman, 1975, Mitzdorf, 1985) than that measured in a saline bath ($\sim 65\ \Omega/\text{cm}$, 1 Hz-10 kHz; Logothetis, 2003). This is because ions move around the cellular processes in a very limited space. A static (or quasi-static) electric field can be assumed surrounding a neuron as the inductive, magnetic, and propagative effects of the bioelectrical signals in the extracellular space can be neglected for the frequency range that is of interest in physiology studies (0 to ~ 2 kHz). Ideally, the membrane potential is uniform along the entire length of an axon at rest, and there is no current flowing inside or outside the neuron (**Figure 2.3a**). When the AP propagates through the axon, it depolarizes part of the membrane as the inflow of Na^+ into the active sites of a neuron (**Figure 2.3b**). The active region is called sink and the inactive region near the sink is called source. By Ohm's law, the potential difference between the depolarized and resting parts causes the current to flow. Current flows inward in the sink and outward in the source. If an electrode is placed adjacent to the axonal membrane at this point, it will record negative voltage with respect to a distant indifferent electrode. But if records from a source, the potential will be positive with respect to a distant indifferent electrode as the current flows outward in the source. Because of the resistance of the extracellular medium, such currents generate so-called extracellular field potentials (EFPs).

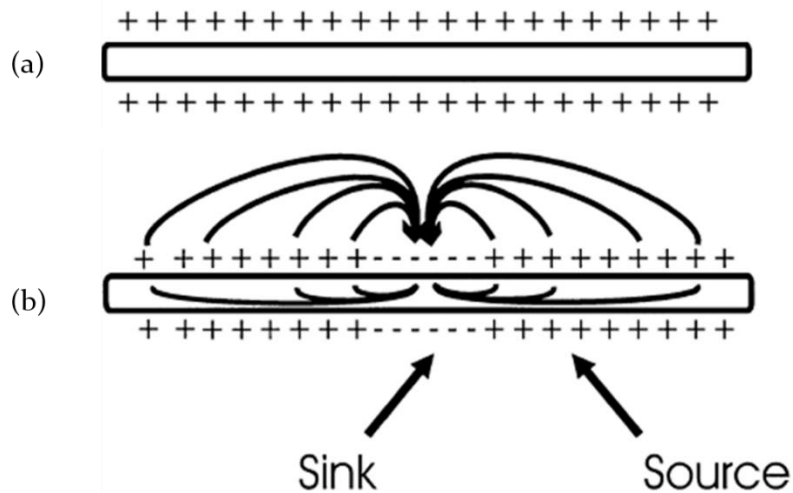


Figure 2.3 Depiction of volume conductor theory to model current flows around an axon in a uniform, low-resistance extracellular medium called volume conductor; (a) When the axon is at rest, the membrane potential is uniform, and no current flows; (b) Current will flow in a depolarized segment of the membrane. The flow is inward at the depolarized region (sink) and outward at adjacent regions (source). (Adapted from Heinricher, 2004)

Let us consider the simplest model of extracellular recording consisting of an isolated neuron and an electrode recording somewhere along the axon (**Figure 2.4**). The electrode registers positive potential while the membrane underneath acts as a source for the membrane depolarized at a distance. As the AP approaches the electrode, the difference in potential between the source and sink decrease, causing the decrease of positive potential. When the AP reaches the region underlying the electrode, the depolarized membrane acts as a sink, and the electrode records a negative potential. Finally, as the AP moves away from the electrode, membrane is repolarized, and resume serving as a current source. The electrode records the positive potential once again.

Therefore, the AP recorded by an electrode adjacent to an isolated axon should be triphasic in theory. This theoretical prediction has been experimentally verified by simultaneous intra- and extracellular recording (Henze et al., 2000), as shown in the example in **Figure 2.5**. The negative phase of the extracellular spike coincides with the depolarization seen by the intracellular electrode. The late positive phase of the extracellular potential corresponds to the repolarization of the membrane recorded intracellularly.

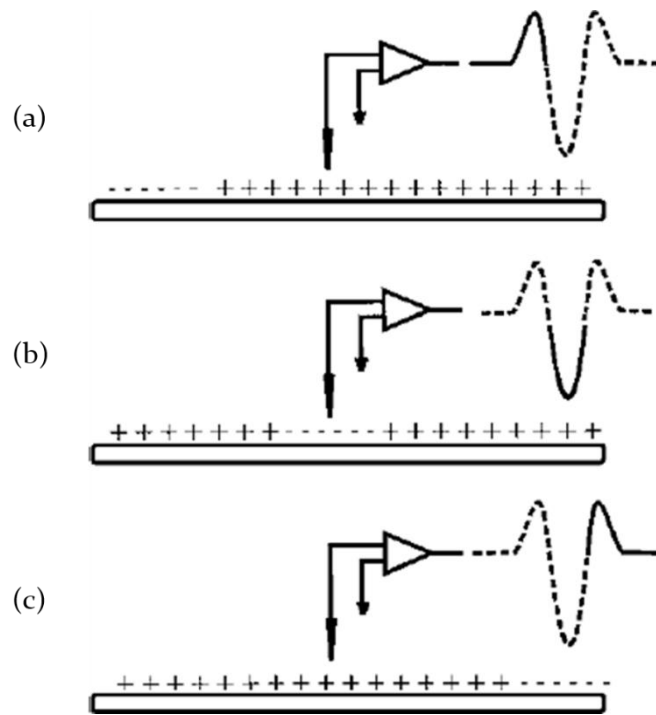


Figure 2.4 Model of sources and sinks predicts that a triphasic waveform will be recorded from an isolated axon; (a) As the AP approaches the region underneath the electrode, that membrane serves as a source, and the electrode sees a positive potential relative to a distant indifferent electrode; (b) When the AP reaches the membrane underlying the membrane, the electrode records a negative potential; (c). As the AP continues down the axon, the membrane under the electrode once again acts as a source, and as a consequence, the electrode records a positive potential. (Adapted from Heinricher, 2004)

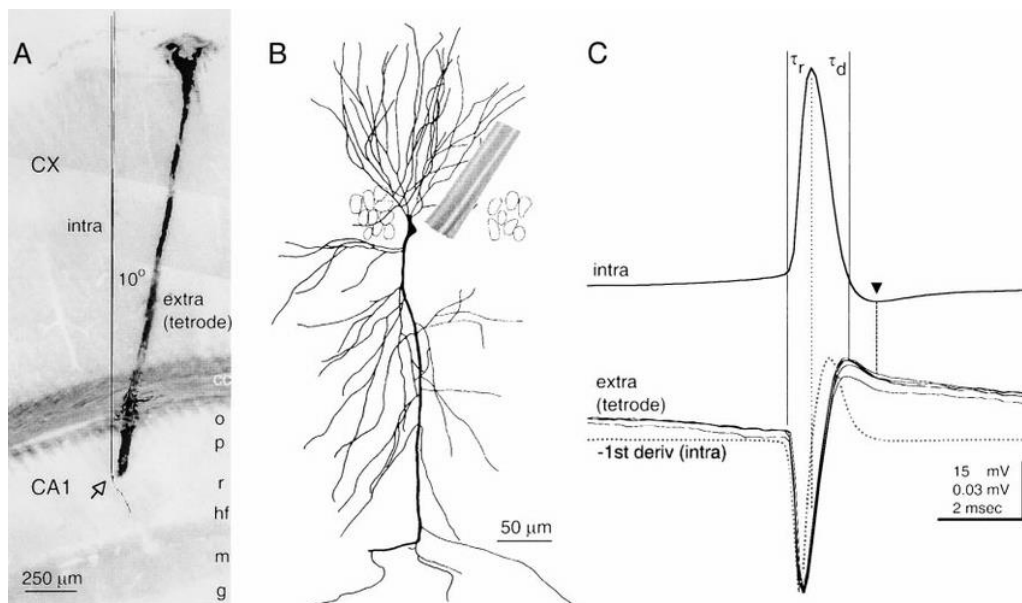


Figure 2.5 Simultaneous recording of the intracellular and extracellular spike. The monophasic depolarization seen by the intracellular electrode corresponds with a positive-negative-positive waveform recorded by the extracellular electrode. Note different scales for extracellular and intracellular recordings. (Adapted from Henze et al., 2000)

Similarly, when an electrode records AP close to the cell soma, soma depolarization is recorded as negative potential at the electrode. Consider a simple neuron model in **Figure 2.6**. As the AP passes through the axon, the soma membrane becomes positive (source). The waveform recorded with an extracellular electrode near the soma, therefore, should be biphasic in theory, with an initial negative component followed by a positive potential. However, actual neurons are more complex, and cell morphology and distribution of active conductance on the somatic and dendritic membrane, the location of the electrode relative to the cell body and the state of cell excitability causes a deviation to the ideal waveform obtained using volume conduction theory. Recordings from soma vary in the relative amplitude of the negative and positive phases of the waveform showing the distinct inflection points on either phase.

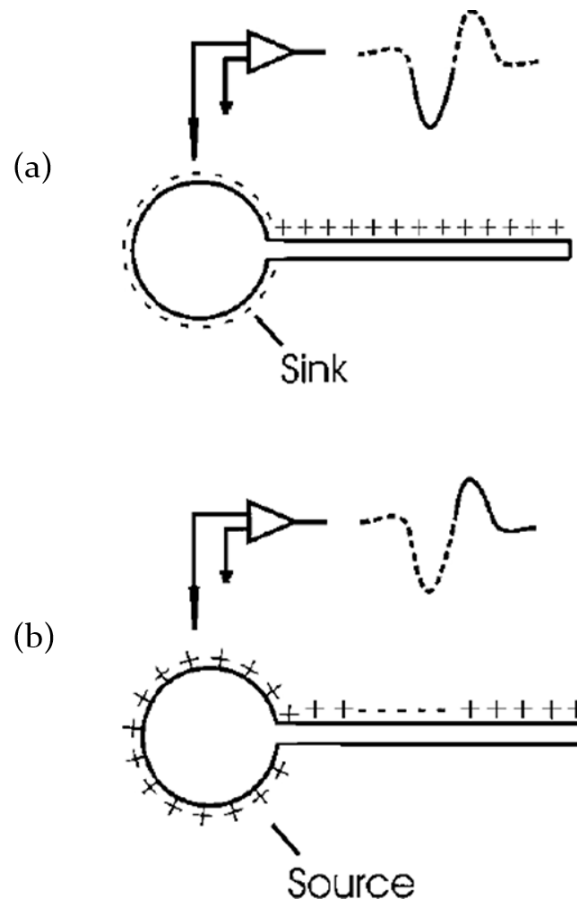


Figure 2.6 The simplest model of an isolated neuron showing the biphasic potential recorded from a soma; (A) Recording of depolarization where the membrane works as the sink, and the recorded voltage is negative. (B) Recording of repolarized membrane potential as the AP moves down the axon. Soma works as the source in this case and contributes to a positive potential.

Practically, an electrode placed at a neural site records the mean EFP (mEFP) signal from the weighted sum of all sinks and sources along multiple cells (Logothetis, 2003). The superposition principle dictates that the EFPs from different cells add up linearly throughout the volume conductor. The lipid layers in neuronal membranes create a low pass filter in conjunction with the volume conduction which creates a low pass filter attributable to the weighting- making the interpretation of mEFP difficult. For the cells oriented opposite to each other, information is lost as the currents of equal magnitude, but opposite polarity generates potentials that cancel each other. The volume conductor itself is strongly anisotropic because of the varying alignment of the neural elements. The orientation-dependent conductivity means that the detailed anatomical and geometrical information are necessary to ensure the correct interpretation of the mEFP signal.

Whether the electrode records APs of a neuron or the ensemble properties of the neuronal mass of the recording site depends on the choice of electrodes, their exact positioning, and the recording sites. Microelectrode with a small tip, if placed near the soma or axon of a neuron, measures the mEFP directly and reports the spike traffic of that neuron and that of its immediate neighbors as well. Simultaneous intra- and extracellular recordings of pyramidal cells in rat hippocampus by placing tetrodes within 50 μm of it show that extracellular potentials provide accurate information on number of parameters such as latency, amplitude, and shape of the APs (Harris et al., 2000, Henze et al., 2000).

The discharge of a large neuron generates a greater flow of membrane current for an equivalent transmembrane AP- creating a bias in recording due to cell types (Stone, 1973) and/or sizes (Towe and Harding, 1970). The resulting extracellular field remains above the recording noise levels over a greater distance. Larger neurons with diameter greater than 20–30 μm diameter are estimated to generate a potential larger than 100 μV within a 100 μm -diameter sphere with the electrode tip at its center (Rall, 1962) which decreases rapidly with increasing the distance from the electrode tip because of the low-pass properties of recording medium. Spikes are not distinguishable anymore from background noise at a distance larger than \sim 140 μm (Henze et al., 2000). Spikes from the small neurons do not propagate that far and, therefore, their recordings for longer duration are less stable.

Electrodes placed at a distance from the spike-generating neurons and with low impedance monitors the totality of the potentials in that region as the APs do not predominate. The EFPs recorded under these conditions are attributed to both the integrative processes (dendritic events) and to the spikes generated by hundreds of cells (Bishop and O'Leary, 1942)– known as the local field potential (LFP) and multi-unit activity (MUA) respectively. A high-pass filter cutoff of ~300–400Hz is used in most recordings to obtain multiple-unit spiking activity (MUA), and a low-pass filter cutoff of ~300Hz is used to obtain the so called local field potentials (LFPs) (Logothetis, 2003).

Depending on the recording sites and the electrode properties, the MUA most likely represents a weighted sum of the extracellular APs of all neurons within a sphere of ~140–300 μ m radius, with the electrode at its center (Gray et al., 1995). Synchronous firings of many cells can enhance the spikes by summation and are detected over a larger distance (Chi-ming and Buchwald, 1977). The MUA EFPs are the function of cell and axon size, and, therefore, is site specific (Buchwald and Grover, 1970). It varies considerably from one brain region to another but remains relatively constant for a particular site (e.g., hippocampus vs neocortex). Sites where MUA of large amplitude fast activity are found also contain homogenous population of large cells, and the size of the transmitting axon also correlates the magnitude of the axonal spikes (Logothetis, 2003).

The LFPs reflect cooperative activity in neural populations as they represent the slow events. They are suggested to reflect the weighted average of synchronized dendrosomatic components of the synaptic signals of neuronal ensemble within 0.5–3mm of the recording sites (Mitzdorf, 1987). Unlike the multiunit activity, their magnitude is not correlated with the cell size. Instead, they depend on the extent and geometry of dendrites in the recording sites. If the cells are in open field geometrical arrangement- i.e., dendrites face in one direction and somata in another- they produce strong dipoles between dendrite and soma when activated by synchronous synaptic input. Neurons oriented horizontally contribute less efficiently or not at all to the sum of potentials. There is also evidence of the existence of other types of slow activity unrelated to synaptic events, including voltage-dependent membrane

oscillations (Kamondi et al., 1998) and spike afterpotentials (Buzsaki et al., 1988) that contribute to the LFP generation.

It is essential to identify the activities of individual neurons to explore their relation to spatial encoding. The following section provides a description of how this is achieved using the principles that facilitates the isolation of their activities from signals recorded in electrodes with thousands of neurons in their vicinity.

2.2.2 Single-unit activity

Principles of single unit isolation

(This section is adapted from (Humphrey, 1979) unless otherwise specified)

Action potentials do not vary in amplitude and duration throughout the nervous system and across the species, although the conduction velocity dramatically varies depending on the size and myelination of the axons (Hartline and Colman, 2007). But the recorded action potential depends on number of different factors which causes to vary the shapes of the waveform. Given that the recording is stable during the entire duration of the experiment, the waveforms and their properties allow discriminating the groups of action potential belonging to a putative neuron, also known as ‘single unit’.

The extracellular field theory described before states that the electrodes positioned near the neurons record changes in electric potential in the extracellular field which is generated from the volume conductance and the current flow. The membrane current consists of both capacitive and resistive components given by the equation,

$$J_m = J_{cap} + J_{res} = C_m(dV_m/dt) + G_mV \quad (2.1)$$

Where, V_m is the transmembrane potential, C_m is the membrane capacitance, and G_m is the membrane conductance.

Let us consider a stellate-shaped neuron. Current flows radially outward from the soma (sink) into the dendritic tree during an action potential in its soma, which is extracellularly viewed as a single sink in the soma and distributed sources at dendrites (**Figure 2.7a**). The recorded potential at an electrode in the extracellular field is then

given by the net effect from the distributed dipole model or compartmental neuron model based on Hodgkin-Huxley membrane model (Moffitt and McIntyre, 2005), which is given by,

$$V_e = \frac{1}{4\pi\sigma} \sum_i (J_{m,i}/r_i) \Delta S_m \quad (2.2)$$

$J_{m,i}$ is the membrane current density at segment i , σ is the volume conductance, r_i is the distance of segmented dipole to an external point (**Figure 2.7b**), and ΔS_m is the membrane surface area for the elemental membrane dipole. In a more complex model, the compartments may comprise the soma, the axon hillock, and segments of both axons and dendrites.

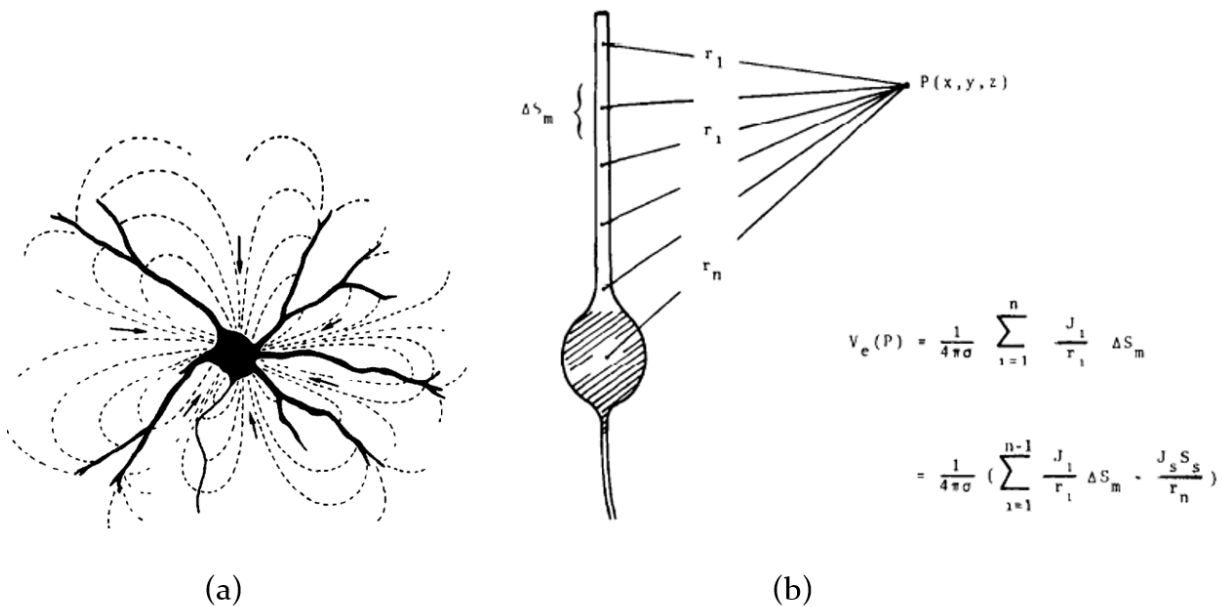


Figure 2.7 Extracellular electric field model of an example neuron; (a) Direction of lines of current flow around a stellate shaped neuron during somatic action potential; (b) Distribution of membrane current and its relation to the potential at an extracellular point (modelled with single dendrite). Current flows inward during an action potential at cell's soma (hatched), and outward at various regions along the dendrite. (Adapted from (Humphrey, 1979)).

As the model suggests, spike configuration and polarity may markedly vary depending on the location of the recording electrode with respect to the cell geometry, or alternatively, to the spatial distribution of the sources and sinks. As explained in **Section 2.2.1**, phases of the spike changes as the recording site moves from soma to axon- altering the shape of the waveform (**Figure 2.8**).

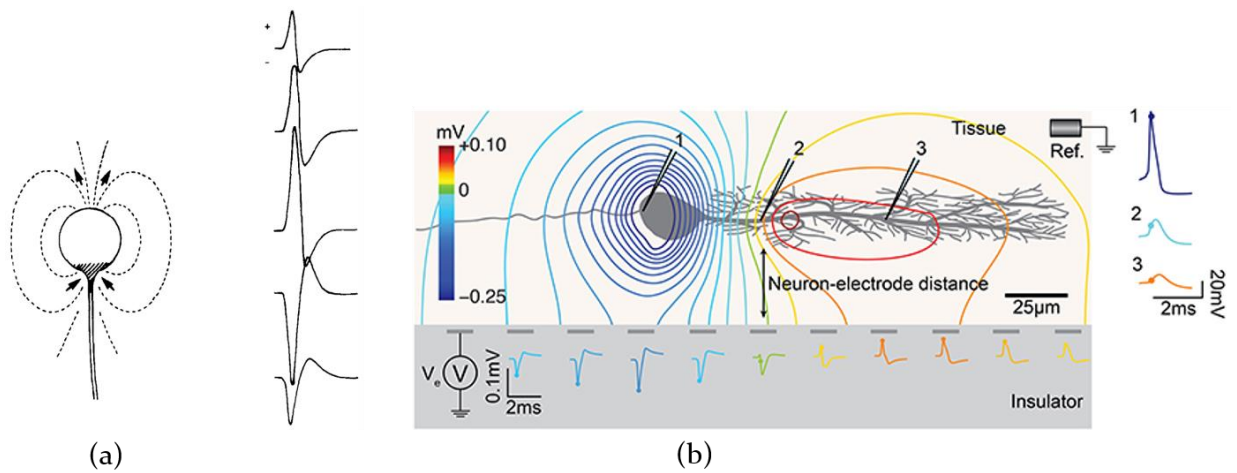


Figure 2.8 Changes in spike-shapes with respect to the electrode location; (a) Recorded spikes for a spherical neuron undergoing a spike-generating conductance change over only a portion of a soma; (b) Changes in the action potential in a more generalized model of neuron-electrode interface (Adapted from Obien et al., 2015).

Spike shape is influenced by both the cell geometry and cell size. Spike shape can vary depending on the structure of the dendritic trees even under the simple assumption of passive dendrites. The cell size determines the cell's input impedance, which in turn affects the magnitude of the membrane currents during depolarization. For a given change of transmembrane potential, large cells can induce membrane currents I_m that are larger than for small cells. Here, $I_m = \Delta V_m / Z_{in}$, where ΔV_m is the transmembrane potential and Z_{in} is the neuron's input impedance. An alternative way to think of the issue is that the larger cells have greater capacitance and hence greater current must flow to discharge and then recharge the membrane. Larger membrane currents imply stronger field potentials based on **Equation 2.1**.

Initial models (Nelson and Frank, 1964) assert that the differences in spike polarity and configuration result from the degree of active invasion of the dendrites as the action potential propagates, like what is explained in **Section 2.2.1**. Although this is applicable for Purkinje cells, the differences in spike shapes for pyramidal and stellate cells can be accounted for based on the cell geometry (**Figure 2.9**). The axon hillock in pyramidal cells act as sink, whereas the apical and basal dendrites appear as the distributed current sources. Therefore, when recorded at a distant point from the dendrites, the sink dominates, and the spike is negative in polarity. As the recording site approaches the apical dendrites, outwardly directed membrane current

dominates and the spike becomes biphasic (positive-negative) (**Figure 2.9a, 2.8b**). At points ventral to soma, the contribution from the sink dominates and the spikes are negative. The soma of the stellate cells acts as the sink, but the dendritic sources are more diffused and distributed- creating a more complex scenario of the spike-shapes depending on where the recordings are performed.

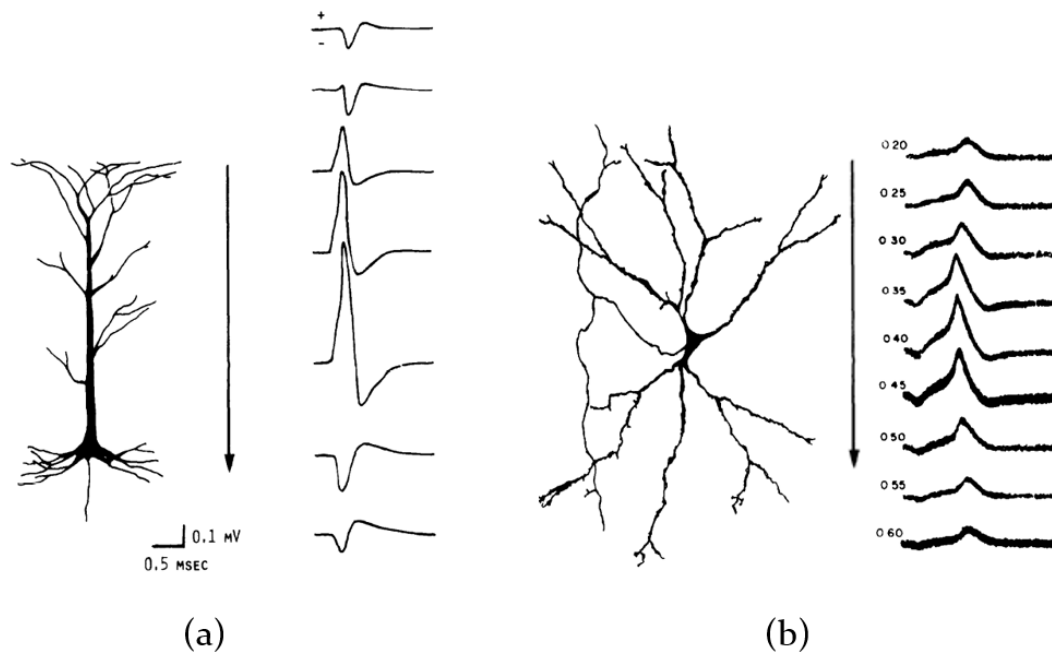


Figure 2.9 Example of the effect of cell types on recorded action potentials; Spikes recorded across (a) the axis of a pyramidal cell; (b) depth of a stellate cell (Adapted from Humphrey and Schmidt)

The cell size also alters the spike shape. The input impedance of a cell is a function of the cell size and affects the magnitude of the membrane current during depolarization. Large cells induce more transmembrane potential than the small ones owing to their low impedance. This can be attributed to the fact that larger cells have greater transmembrane capacitance. This implies that a large amount of current must flow to discharge and recharge the membrane. The extracellular potential is a function of the membrane current, and, therefore, causes a larger field potential according to **Equation 2.1**. In effect, this also poses a challenge in recording single-units. Majority of the CNS cells are small with soma diameters ranging from 10 to 20 μ m. Their spike amplitudes are in order of 0.05-0.1mV unless the recording is done in the immediate vicinity of the cell body where they generate spikes of amplitude ranging from 0.2 to 0.3mV (Humphrey and Corrie, 1978). These spikes are detectable

only over a short distance and cannot be held for longer periods as they are lost, particularly, if the electrode moves. This tempts the experimenters to identify only the easily obtainable large-amplitude spikes that survive for required periods of the recording.

The variability in spiking waveforms due to the relative location of the electrode, cell types, and sizes makes it feasible to identify the event times of action potentials from individual putative neurons. The use of multiple electrodes, such as tetrode, which is a group of four electrodes bundled together (**Figure 2.10**), provides a better discrimination of the waveforms and improves the quality of isolation of activities generated by individual neurons (Harris et al., 2000). The distance and orientation of each cell with respect to the individual electrodes are different- creating more features for separation of the waveforms, e.g., electrode closer to one neuron will record stronger signal than the one located further.

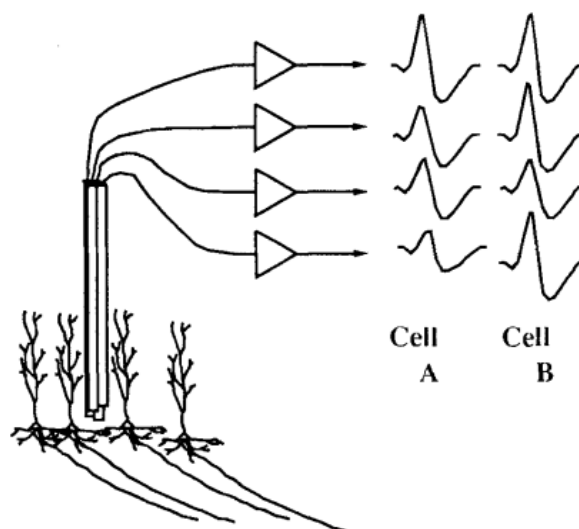


Figure 2.10 Schematic diagram of four-channel tetrode among pyramidal cells. Each electrode records signal from the neurons, but the shape of the waveforms from the same neuron recorded in individual channels are different (right). Therefore, the isolation of the neuronal activities based on waveform shapes are more facilitated (Adapted from O'Keefe and Recce, 1993)

Processing of implanted electrode data

The recording and data processing steps are shown in **Figure 2.11**. LFP data reflect the dynamics of neuronal population surrounding the electrode and are obtained by low pass filtering (cutoff frequency: ~300Hz, (Logothetis, 2003)). The recorded signal

is band-pass filtered to obtain the activity of neurons near the electrodes. Activities from the neurons located very far away give rise to the background activity or noise signal (black trace in the bottom panel of **Figure 2.11**). The bandpass filtered signal comprises action potentials from different neurons superimposed. Spike sorting techniques are then used to identify the waveforms generated from individual putative neurons and their temporal information are obtained.

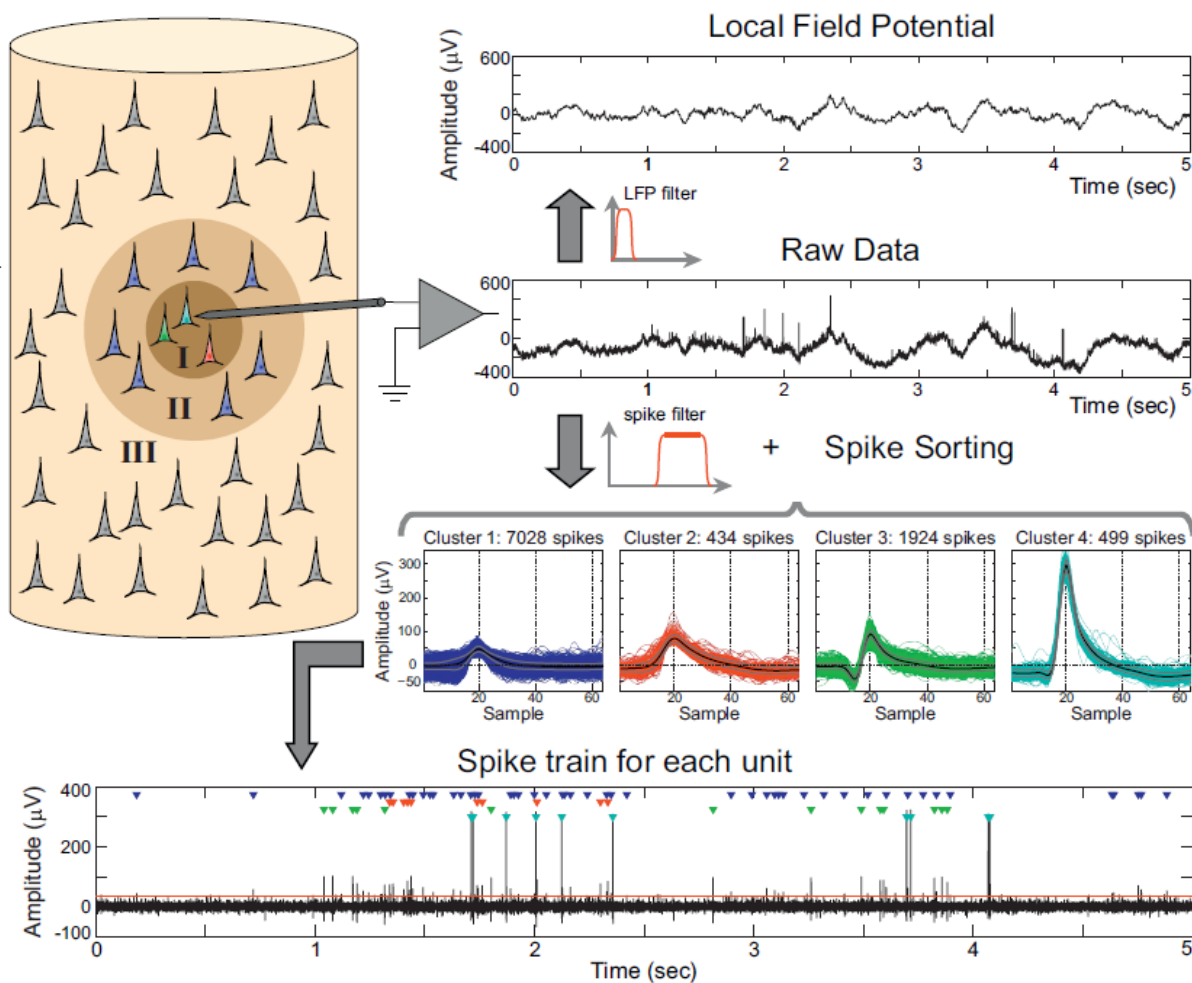


Figure 2.11 Standard data processing steps for extracellular recording experiments, particularly those involve single-unit characterization. Raw data is lowpass filtered to obtain the LFP signal. The bandpass filtered signal undergoes spike-sorting process resulting in identifying the waveforms and spike-train for each neuron. (Adapted from Rey et al., 2015b)

Figure 2.12 depicts the spike sorting process. The waveforms of individual neurons are used to identify the spikes and their timing with an underlying generative model that spike-waveforms of similar shapes belong to the same neuron. The next logical

step is to identify the spike waveforms from the noise-corrupted low-pass filtered signal. This step is called 'spike detection'. There are number of spike detection techniques (Bestel et al., 2012). One of the most common approaches is to use a voltage threshold to determine the presence of spiking waveforms. The threshold can be manually set by the experimenter during the recording session, or an automated threshold measurement method can be used (Quiroga et al., 2004) (**Figure 2.12a**). Usually, a signal chunk of 1msec is taken around the threshold to obtain the waveforms (**Figure 2.12b**). Different waveform features like wavelet coefficients (Hulata et al., 2002, Quiroga et al., 2004), principle component analysis (PCA) coefficients (Shoham et al., 2003), spike-shapes (width, peak, troughs)(Vogelstein et al., 2004) are used to describe individual waveforms. These descriptions or features form groups or clusters, where each cluster represents waveforms belonging to each putative neuron (**Figure 2.12c**). Clustering algorithms or manual cluster cutting techniques (Lewicki, 1998) are used to identify the groups and the spike-sorting process is complete with assigning appropriate tags to each cluster and corresponding waveforms (**Figure 2.12d**). Manual clustering involves cutting the clusters by drawing ellipsoids or polygons around the blob at feature space, usually in two-dimensional projection of the features, e.g., plotting against the amplitude vs width of the waves, and delimiting the spikes of different neurons. This approach is very time-consuming and subjective. There are attempts for automated spike sorting algorithms (Rey et al., 2015b) that employ unsupervised machine learning techniques and statistical approaches (Kadir et al., 2014, Quiroga et al., 2004). Spike-sorting results in identifying the sequence of spike times, spike shapes and the neurons they putatively belong to. Shapes that cannot be separated due to low signal to noise ratio leads to a cluster of multiunit activity (MUA). Neurons contributing to MUA are relatively close from the electrodes for their spikes being detected, but not close enough for their shapes to be isolated.

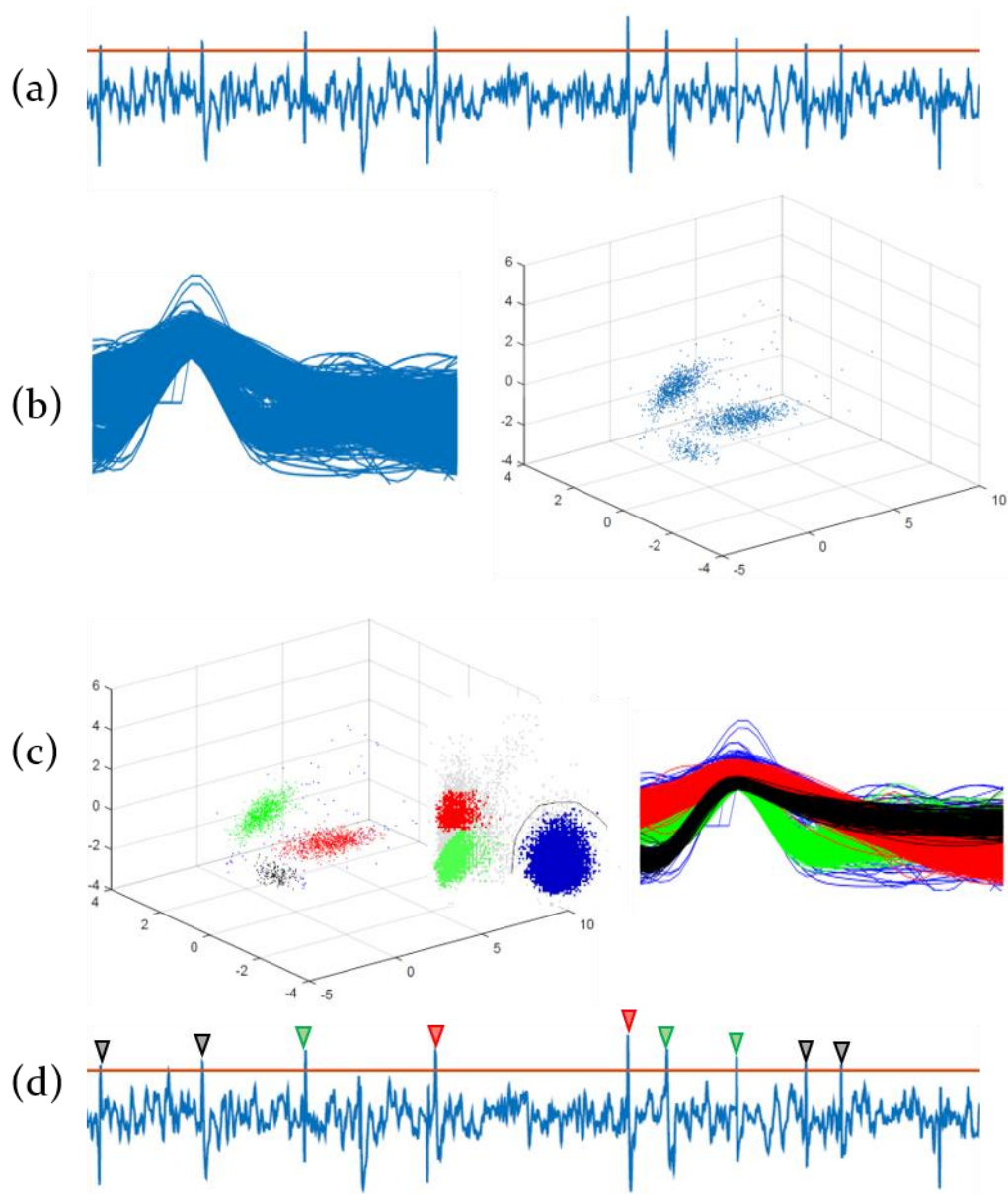


Figure 2.12 Spike sorting steps (a) Thresholding on bandpass filtered recorded extracellular potential for spike detection. This is one of the most popular choice for detecting spikes (simulated); (b) Extracting the waveforms (left) and creating a feature space constituting different features of the waveform (PCA, wavelet coefficients etc.); (c) Using clustering algorithms to identify individual clusters (left) or using manual cluster cutting techniques (middle) to draw boundaries around a feature cluster and identify it as a distinct cluster. Waveforms from individual neurons are separated (right) where individual color corresponds to waveforms belonging to one cluster; (d) The waveforms are appropriately tagged to identify their temporal location. This timing information is used for subsequent analysis. Spike waveforms are extracellular data recorded from the rat anterior thalamus and the filtered continuous signal is simulated data.

Spike sorting is essential to characterize individual neurons in response to the experimental manipulations. Close-by neurons can be picked up by the same electrode but can fire in response to different information item. For example, in the

human or rat hippocampus, nearby neurons fire to unrelated people in the first case and to distant place fields in the latter (Quiari Quiroga, 2012). Individual neurons have their characteristic tuning and firing properties, relationship with other neurons, and dynamics with the local field potentials.

It is indeed also a matter of concern that how the isolated single-units or putative neurons encode the information. The following section provides an overview of different encoding schemes which also contain the description of ‘Spike-count rate’ and the ‘Temporal encoding’ that are used in the NeuroChaT software.

2.2.3 Neuronal encoding schemes

The study of nervous system generally follows a standard procedure of presenting stimuli and observing the behavior as the response (**Figure 2.13**). It is then observed how the behavior and the stimulus are correlated to the underlying neural elements. The focus of neural encoding is to identify how different neural components, namely single-units, LFP and MUA responds to a wide variety of stimuli and to develop models to predict the response to other stimuli. Neural decoding refers to the process of reconstructing a stimulus from neural responses. A train of spikes can contain information based on different coding schemes. It may be the firing rate of the spikes (rate code) that represents the strength at which innervated muscles contract. Neurons firing in preference to a specific location in the environment also encodes the location information by the timing of firing (temporal code). Schemes to study neural encoding by neural spike trains can be described in 4 categories:

Neural Coding Studies

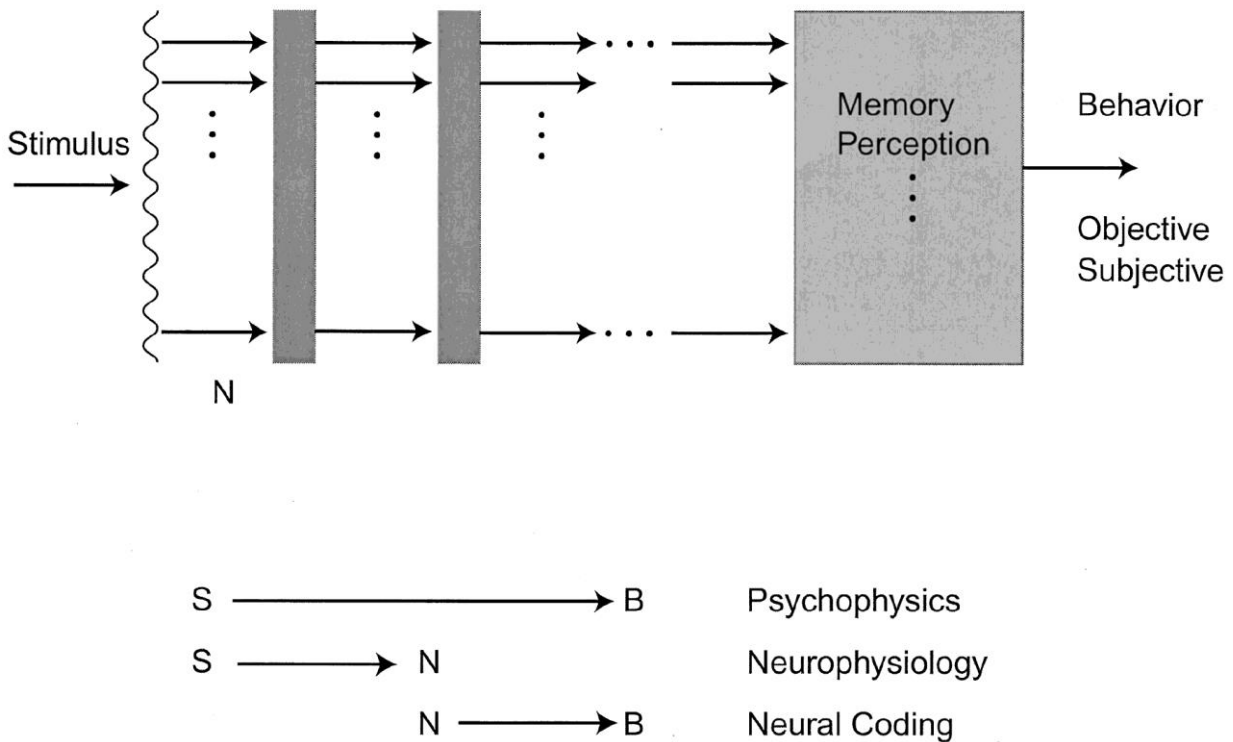


Figure 2.13 Experiment design in neuroscience and the stimulus to neural response to behavioral pathway model; *S* : Stimulus, *B* : Behavior, *N*: Neural response (Neural Coding, Johnson, 2000)

Rate coding

The rate coding scheme states that the frequency or rate of spikes increases as the stimulus intensity increases. This is based on the idea that the information about the stimulus is contained in the firing rate of the neuron. In most the sensory systems, there is a nonlinear relationship between the increasing stimulus intensity (Kandel et al., 1991) and spike firing rate calculated by counting the number of spikes in a given time interval. For example, the direction sensitive visual neurons have higher preferential firing rates to certain directional rotation of the visual stimulus (**Figure 2.14**) (Dayan and Abbott, 2005).

Firing rates in this scheme can be defined in two different ways based on the averaging procedure followed such as an average over time or an average over several repetitions of experiment. Rate as “spike count”, also called “temporal average”, is calculated by dividing the number of spikes in a given trial by the duration of the trial

(**Figure 2.15a**). Trial length T is set by the experimenter which is heuristically determined depending on the neuron type and the stimulus (Gerstner and Kistler, 2002). Although rate can be determined using a single trial, the temporal resolution of the variation in neural response is lost in this approach. The rate is calculated for a set of stimuli to observe the effect of a particular event (e.g. changes in direction of a light bar in the receptive field to observe direction selectivity; **Figure 2.14**).

The time-dependent firing rate is the number of spikes averaged over trials within short intervals. It can be viewed as splitting the entire trial into small time windows and calculating the spike-count rate over many trials (**Figure 2.15b**). It is useful for both impulsive stimulus and the time-dependent one. In this approach, the same stimulus is repeated several times and the response of the neuron is reported in a Peri-Stimulus-Time Histogram (PSTH). As it counts the discrete random events within short term interval, it is also mathematically equivalent to calculating the point density estimates at each short interval Δt . Therefore, it is a measure of probability that a spike occurs during interval t and $t + \Delta t$. The number of occurrences of spikes n_K within t and $t + \Delta t$ summed over all repetitions of the experiment is divided by the number of repetitions K to obtain the average number of spikes. Division by the interval length Δt gives the time dependent firing rate $r(t)$ of the neuron. This approach neglects all the information possibly contained in the exact timing of the spikes.

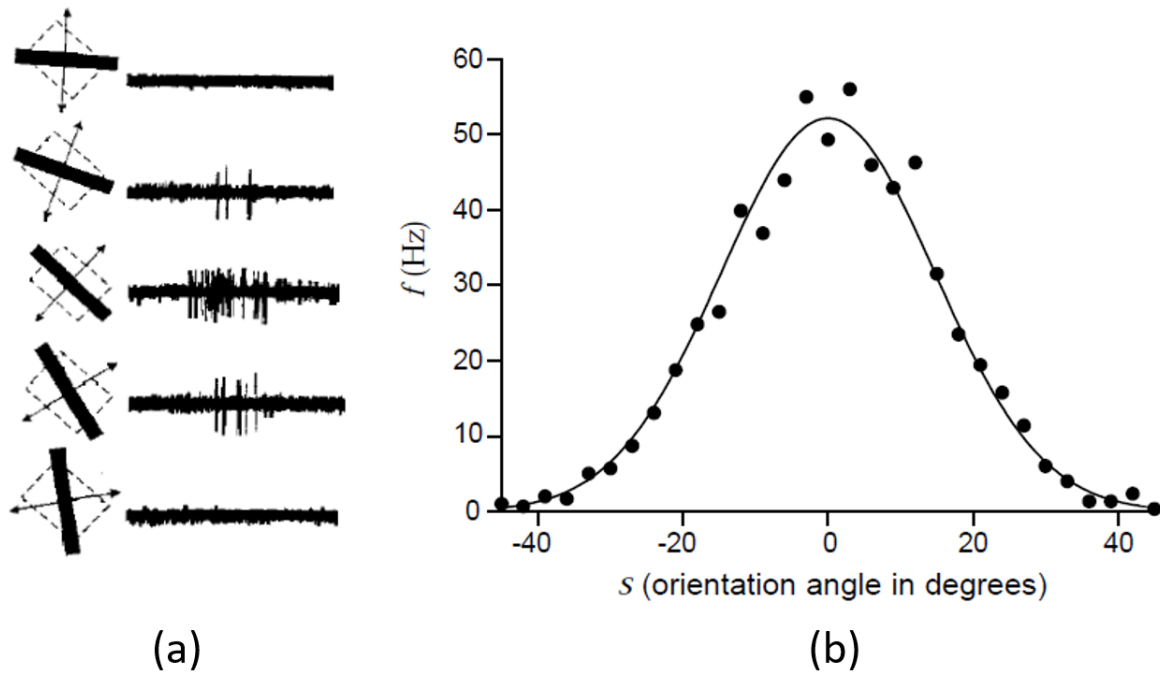


Figure 2.14 (a) Recordings from a neuron in the primary visual cortex (VI) of a cat. A bar of light was moved across the receptive field of the cell at different angles. The diagram on the left shows the stimulus. Dashed line is the receptive field of the neuron. The bidirectional motion of the light bar is indicated by the arrows. (b) Average firing rate as a function of the orientation angle of the light bar stimulus. The nonlinear relation is readily apparent as it takes the shape of a Gaussian function. The firing rate is maximum in the preferred direction which is 0 degree relative to receptive field orientation for this cell. (Adapted from Dayan and Abbott, 2005)

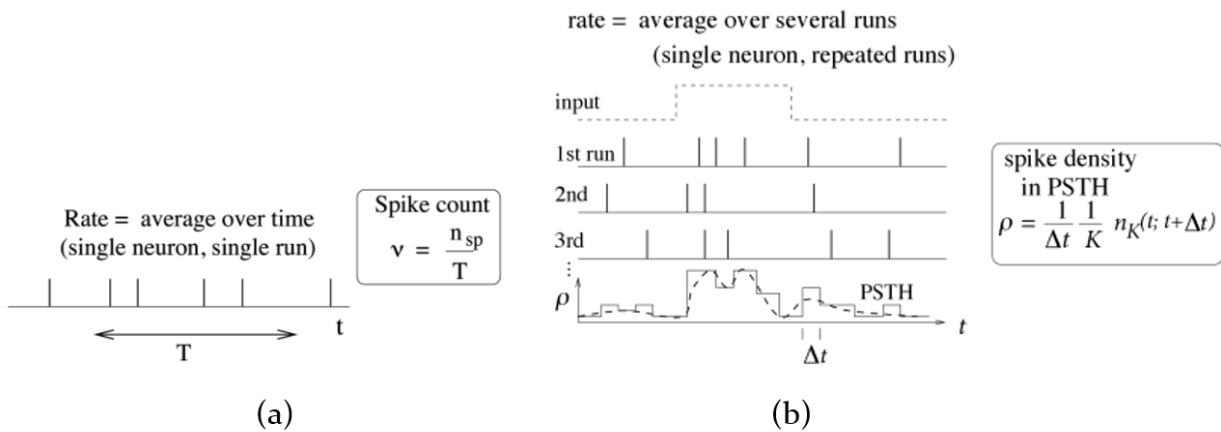


Figure 2.15 Depiction of different rate coding strategies. (a) Spike count rate which measures firing rate on a trial-by-trial basis by dividing the number of spikes with the duration of the trial. (b) Time dependent firing rate is calculated by calculating peri-stimulus time histogram (PSTH) and averaging over trials of repeated stimulus; n_{sp} =number of spikes in one run, T = duration of trial, v = spike-count rate, Δt = time bin for generating the PSTH, K = trial number, n_K = number of spikes in K -th trial between time t and $t+\Delta t$, ρ = time dependent firing rate. (Adapted from Gerstner and Kistler, 2002)

Temporal coding

Neuronal spiking rates exhibit high frequency fluctuations. The rate coding assumes that such irregularities are noise on the Poisson process of spike generation. The temporal coding implies that the timing-interval between individual spikes encode information of the underlying stimulus (Dayan and Abbott, 2005). The spiking activity is modelled '0' (no spike) or '1' (spike) constituting a binary sequence. Spike trains 00110100 is different from 01010110 and carry different features of a stimulus according to the temporal coding scheme, while the rate coding implies that these two trains carry the same information. Experiments with similar but not identical stimuli shows that the neural responses are significantly different for the stimuli set- suggesting that the distinctness in the patterns of spikes contain more information than can be expected in rate code (Stevens and Zador, 1995). There are also features of neuronal activity that cannot be explained in terms of the rate code. For example, the rhythmicity in spiking activity of many neurons in rodent Hippocampus reflects the synchrony with underlying LFP theta signal which is observed by the inter-spike interval distribution of the spike train (Cacucci et al., 2004). Other features, i.e., time to spike after stimulus onset, or temporal patterns or groups of spikes precisely timed cannot be identified without exploring the temporal codes (Kostal et al., 2007). Given that there is no reference frame to represent time in the nervous system, temporal coding uses either the relative timing of the spikes in a neuronal population or ongoing brain oscillation as reference (Stein et al., 2005). The latter is more commonly known as 'phase of firing' code that uses the time reference based on the phase of oscillations and counts the spikes occurring at particular phases of the underlying LFP activities or its sub-bands, i.e. LFP theta band. Place cells and grid cells in hippocampus are shown to encode the animal's progression through a place field by the advancement of phase with respect to the underlying theta cycle (O'Keefe and Recce, 1993, Jeewajee et al., 2014) (**Figure 2.16**). The cell fires in synchrony with the wave cycle outside of the place field. As the animal proceeds through the place field, the phase at which action potential occurs increases and finally aligns to the trough of the wave cycle at the center of the place field where maximum firing rate occurs. This phase advancement keeps going as the animal moves away along the place field and finally aligns in synchrony with the next theta cycle making it in-phase again. Thus, the advancement in phase, or the temporal sequence of the action

potential, through the place field represents animal's position in the field. This phenomenon is also called phase precession.

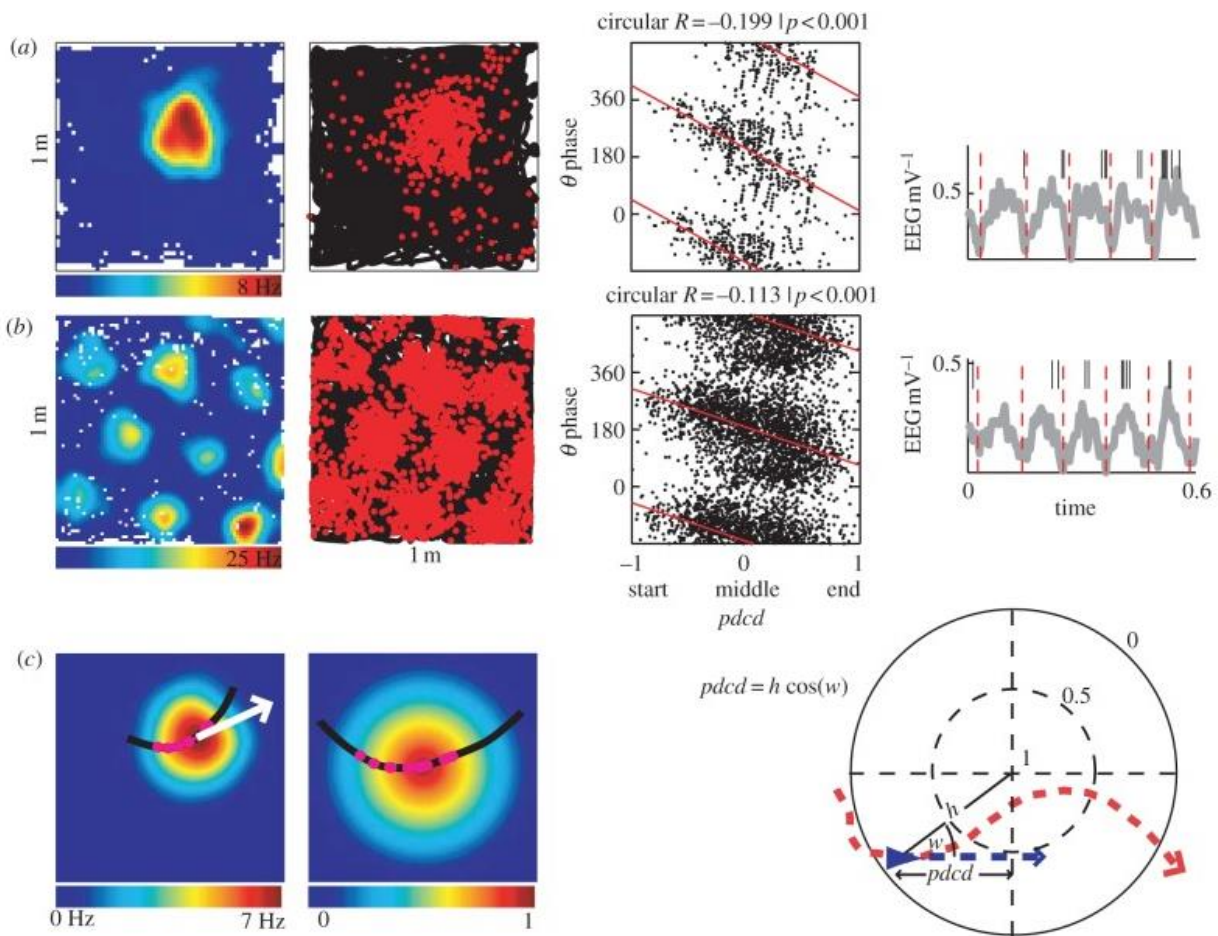


Figure 2.16 Phase precession in place cell and grid cell in rodent hippocampal formation; (a) Example of phase precession in a place cell showing the spiking rate map (left), the spikes as dots on the animal's path (black line; left middle), raster plots of spike phase versus pdc . Red line is the circular linear regression line (right middle) and an example of the timing of spikes (vertical ticks) versus LFP (grey line, right); (b) Similar example for a grid cell; (c) Example of the transformation of a run through a firing field onto a unit circle which represents the firing field with the peak firing rate at the center. Locations on the run are transformed radially so that their proportional distance between the firing peak and the perimeter is preserved, and then rotated about the center of the circle so that the mean direction of the run (white arrow) is left-right for all runs. 'pbc' is the distance of the animal to the field peak projected on the current running direction, which varies from -1 to 1 as the animal enters the field on the left and exits on the right. (Adapted from Jeewajee et al., 2014)

Population coding

While the rate coding dictates the behavior of individual neurons, population coding explains how these responses are combined by many neurons to represent the stimuli. This coding paradigm is widely applicable in the sensory and motor systems as

revealed by the experimental studies. The neurons from visual area MT are tuned to different moving directions of an object (Maunsell and Essen, 1983). The neurons together fire in a noise corrupted Gaussian activity pattern across the population. The direction of the object is, therefore, retrieved from the population activity. This is performed using the population vector coding model (Georgopoulos et al., 1986) where a monkey was trained to reach its arm to a series of directions. Neuronal activities were then recorded from the primary motor cortex of the monkey. Neurons have been found to preferentially fire for the arm-reach at a particular direction (**Figure 2.17a**). The response, again, forms a characteristic bell-curve like the one shown in **Figure 2.17b**. The population vector coding suggests that spikes from each neuron can be viewed as the votes for the motion in its preferred movement direction. When a movement occurs, votes are combined, and movement occurs in the average direction. Mathematically, the weighted average of individual neurons' preferred direction retrieves the stimuli- where each direction is a vector and the firing rates act as the weights. As it calculates the vector summation of the directional firing rates, the resulting average is called population vector code (**Figure 2.17d**).

Alternatively, we can explain it in terms of linear algebra. Each directional tuning curve form the basis set where the firing rates are the weights. The linear combination of the responses from all the neurons gives rise to the stimuli that triggered the neural response. Therefore, the collective response, or the population response, encode the stimuli information. Although individual neurons are too noisy to encode the stimuli faithfully, the population code ensure fidelity and precision as it reduces the neuronal variability. Individual neurons have overlapping selectivity (**Figure 2.17c**), so that, if not all, many neurons respond to a given stimulus. Given that the tuning curves of each neuron are known, population coding can reflect the stimulus almost instantly (Hubel and Wiesel, 1959) and is much faster than the rate coding.

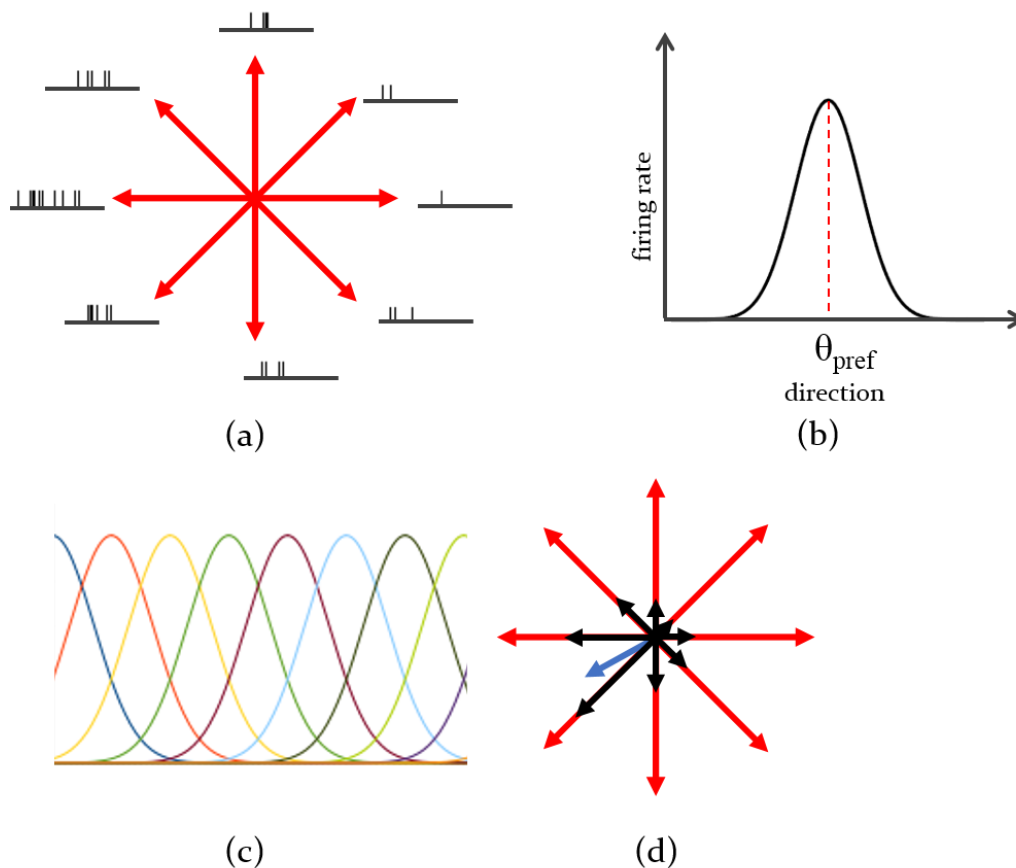


Figure 2.17 Explanation of the population encoding of directional stimuli; (a) Spiking activity of a neuron for different directional stimuli, showing preference towards west; (b) Typical tuning curve in response to the directional stimuli; (c) Depiction of the overlapping selectivity. Each bell-curve shows the tuning function of individual neurons; (d) Each black line represents the preferred firing direction of eight example neurons. Their length shows the firing rate of neuron for a given stimuli. The stimulus direction is retrieved by the weighted average of the black lines which is represented by the blue line. (Recreated from Dayan and Abbott, 2005)

Sparse coding

Information in the brain are represented by the pattern of activation of the neurons. But strong activation of a small set of neurons represents an information item at any given time, and the subset of all available neurons are different for each item to be encoded. This philosophy of neural computation is called sparse coding. The sparseness may be on temporal sequence of spiking activities, i.e. a relatively small number of time periods are active or may be on the sparseness on the activated population of neurons (Kloppenburg and Nawrot, 2014).

Sparse encoders attempt to find a small set of basis functions or patterns to represent a large set of input patterns. These patterns can reproduce the original input patterns

when they are combined in the right proportions and are called the sparse codes for the input. For example, the letters, numbers, punctuations etc. are the sparse codes for a written language. When these codes are combined in a particular order, they can produce words and sentences

Sensory representation of information is experimentally observed in the sensory and memory systems. Vision, touch, audition and olfaction (Vinje and Gallant, 2000, Crochet et al., 2011, Hromádka et al., 2008, Ito et al., 2008) sensations are coded accordingly, e.g., sparse coding of odors by Kenyon cells in *Drosophila* is thought to generate large number of odor specific memory locations which are precisely addressable (Wood, 2014). Theoretical explanation of sparse distributed memory suggests that sparse coding reduces the overlap between representations and, therefore, increases the capacity of associative memory (Palm, 2013).

The above description of different neuronal encoding schemes provides us with the idea of how information carried by neuronal encoding are studied. But such studies also follow a number of challenges as outlined in the following section.

2.2.4 Challenges in extracellular data analysis

With the progress of the electrode design technology, increase in the number of electrode sites, lack of a universally suitable spike sorting algorithm retaining the need for manual cluster cutting techniques, and the need to perform complex and multi-faceted experimentation to observe the characteristic tuning of the same neurons pose the challenges to analyze of a large volume of data for the neuroscientists. Here, we give a very brief description of each of these aspects and a tentative idea of how big the data associated to each recording and for each neuron can be.

A. Increase in the number of electrode sites:

Since (Strumwasser, 1958) first invented the microwire technology to chronically record the electrical activity in the brain, there has been a tremendous growth in the number of simultaneously recorded neurons. With current multi-electrode technology, spikes from hundreds of neurons can be recorded simultaneously. A study by (Stevenson and Kording, 2011) reports that the number of neurons recorded

simultaneously approximately doubles in every 7 years (**Figure 2.18**). This improvement is principally attributed to the development of cutting-edge recording techniques. The use of tetrodes (4 electrodes) allowed recording from four sites simultaneously. The advent of silicon technologies, i.e., advanced micromachining techniques, allows developing the polytrodes or multiple columns with 8-64 recording sites or channels in each (Fig-3; Blanche et al., 2005). As the electronics industry flourishes, data acquisition systems provides robust and stable, hence low-noise, recording- availing further the observation of higher number of neurons simultaneously (Alivisatos et al., 2013). Most recent developments in nanoscale analysis tools, and in the design and synthesis of nanomaterials have made it possible to deploy nearly one thousand recording sites (Alivisatos et al., 2013). In addition to that, the introduction of wireless transmitter has enabled recording for a longer duration from the animal, thus increasing the amount of data to be processed to a great extent (Schwarz et al., 2014).

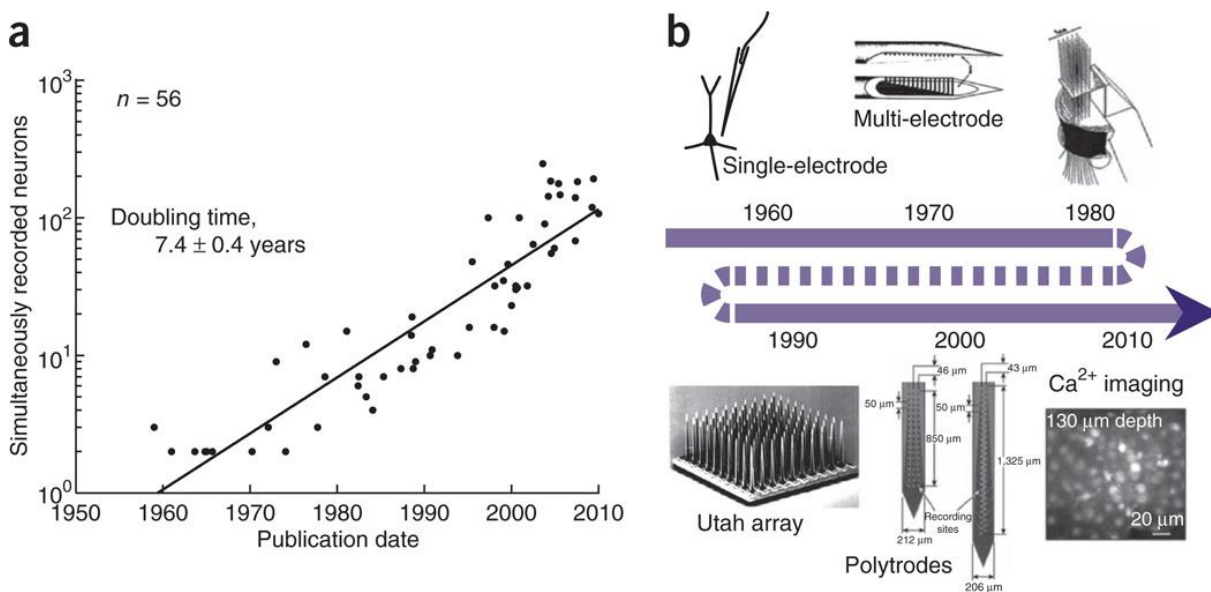


Figure 2.18 Exponential growth in the number of recorded neurons. (a) Examining 56 studies published over the last five decades, (Stevenson and Kording, 2011) found that the number of simultaneously recorded neurons doubled approximately every 7 years. (b) A timeline of recording technologies during this period shows the development from single-electrode recordings to multi-electrode arrays and in vivo imaging techniques. (Adapted from Stevenson and Kording, 2011)

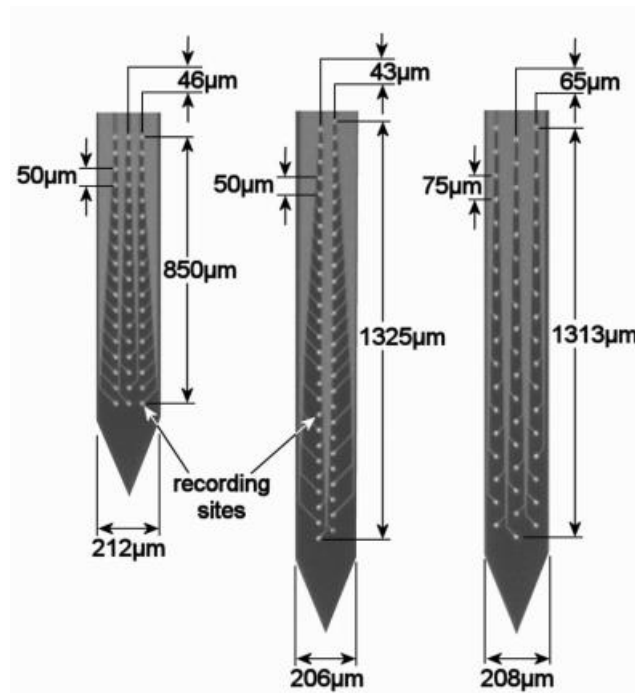


Figure 2.19 Polytrodes with 54 recording electrodes (adapted from Blanche et al., 2005)

B. Lack of a universal automated spike sorting algorithm:

As identifying the firing pattern from individual neurons is a crucial step for their proper characterization, the increase in the recorded neurons poses another big challenge- suitable spike sorting algorithm. Although manual cluster cutting methods (Gray et al., 1995) have been used for years, it has become increasingly difficult and time consuming to cut the cluster one by one basis as number of recorded neurons increases. This is a substantial bottleneck for large scale multi-electrode recording. There are a number of efforts to develop automated spike-sorting algorithms to lessen the need for human intervention (Rey et al., 2015a, Kadir et al., 2014). But the lack of a properly validated spike sorting utility to process such a high volume of data efficiently and reliably (Einevoll et al., 2012, Rey et al., 2015a) further exaggerates the need for both sophisticated spike-sorting tools and, till then, a platform to minimize the human-efforts for analysis of such identified neurons.

C. The demand of more experimental manipulations:

As new research questions are being conceived for a deep-understanding of the navigation mechanisms in animal brain, complex experimental manipulations are

being performed to observe the effect of continuous changes in the spatial information while the animal navigates. For example, in a study by (Alme et al., 2014), it was shown that a spatially-tuned single neuron responded differently to eleven different experiment rooms, where it has completely different receptive fields as the environment changes. Although this is an example of a very simple manipulation, neural recording of large-scale three-dimensional space are also being performed (Jeffery et al., 2015b). The feasibility of recording in a more complex setup poses the challenge that more such manipulations be performed to understand the navigation mechanism in a high-level processing perspective. Therefore, neuroscientists should have an appropriate tool to properly manage thousands of analysis outcomes in a systemic way to reduce both of their manual and cognitive efforts.

In summary, there are challenges owing to the increased number of recorded neurons, lack of universal software to isolate their activities requiring a lot of manual efforts, and the necessity of a toolbox for managing large volume of analysis outcomes. There is also a lack of tools containing standardized and peer-reviewed analyses. These challenges were the driving forces in conceiving the idea of developing the NeuroChaT software in the first place which then also followed how the software was designed.

The following section provides an overview of different neurophysiology toolboxes and their scopes of data analysis. The review of the toolboxes shows that there is no toolbox that can accomplish the domain specific needs of the cognitive map research that also addresses the challenges as outlined above.

2.3 Implanted electrophysiology toolboxes

Over the years, computational neuroscientists leveraged their efforts to make data handling and analysis fast, interactive and user friendly by developing open-source software tools for the wider neuroscientist community. This section introduces the current state of the art of these software packages and toolboxes for semi- or automated processing and analysis of extracellular neuronal recordings. Such toolboxes can be categorized based on their scopes of analyses:

- a) Toolboxes for Spike-sorting
- b) Toolboxes for spike-trains and field potential analysis
- c) Toolboxes for simulating the extracellular electrophysiology
- d) General purpose toolboxes

2.3.1 Spike-sorting

With the increasing capability to record simultaneously from a growing number of neurons, there is a continuous effort to develop new toolboxes to address the challenges of sorting of spikes from thousands of electrodes (Harris et al., 2016, Rey et al., 2015b). Following are the toolboxes currently available for spike-sorting for various electrode configurations.

WaveClus

WaveClus (Quiroga et al., 2004) is a MATLAB package that implements an unsupervised algorithm for spike detection and sorting. The features for clustering are selected as the wavelet coefficients of spike waveforms and are clustered with super-paramagnetic clustering (Blatt et al., 1996).

Link: https://github.com/csn-le/wave_clus, or
https://vis.caltech.edu/~rodri/Wave_clus/Wave_clus_home.htm

MClust

MClust (Redish, 2014) is a toolbox with graphical user interface developed in MTALAB®. It enables both manual and automated spike sorting on single-electrode, stereotrodes and tetrode recordings. It facilitates the selection of a range of user-defined features and uses expectation maximization (EM) algorithm for the clustering. It can also be used in a batch-mode using the MATLAB® scripts.

Link: <http://redishlab.neuroscience.umn.edu/MClust/MClust.html>

OSort

OSort (Rutishauser et al., 2006) uses spike waveform templates and sorts them using unsupervised clustering by template matching. The template matching algorithm

uses the distance based on residual sum of squares and custom thresholds. The spike-sorting can be performed online using this MATLAB-base toolbox.

Link: <http://www.urut.ch/new/serendipity/index.php?/pages/osort.html>

Spyke

This Python package allows data visualization, navigation and sorting of extracellular waveform data from multi-channel high density recordings (Spacek et al., 2008). The feature extraction uses principal component analysis (PCA) and the sorting uses modified gradient ascent clustering (Swindale and Spacek, 2014) to classify the features.

Link: <http://spyke.github.io/>

KlustaKwik

KlustaKwik (Harris et al., 2000) is written in C++ and provides the automatic clustering using a mixture of Gaussian and masked EM algorithm (Rossant et al., 2015). It is also integrated in the NeuroScope toolbox and can also be used stand-alone.

Link: <https://github.com/klusta-team/klustakwik>

SAC

SAC (Shoham et al., 2003) is a MATLAB package with graphical user interface that uses principal components (PC) of spike waveforms as features and clusters them using a mixture of t-distributions. An EM-based competitive mixture decomposition is used to identify the clusters.

EToS

Efficient Technology of Spike-sorting or EToS is a command-line based spike-sorting program (Takekawa et al., 2012). It uses weighted-PCA based feature extraction and clusters using robust variational Bayes on a mixture of t-distributions. It is supported only in UNIX-like operating systems.

Link: <http://etos.sourceforge.net/>

SpikeOMatic

Developed in R, SpikeOMatic implements spike sorting using Gaussian Mixture and hidden Markov model as the generative model, and estimates using EM and Markov chain Monte Carlo methods respectively (Pouzat et al., 2004).

Link: <http://www.biomedicale.parisdescartes.fr/SpikeOMatic/>

UltraMegaSort2000

This MATLAB toolbox implements the hierarchical clustering algorithm for spike sorting using the similarities of spike shape and spike timing statistics (Fee et al., 1996, Hill et al., 2011). The quality metrics are provided by false-positive and false-negative errors.

Link: <http://neurophysics.ucsd.edu/software.php>

NEV2lkit

The objective of NEV2lkit is to supply a friendly user interface that links the experimental data to a basic set of routines for analysis, visualization and classification of spikes in a consistent framework (Bongard et al., 2014). It implements different clustering algorithms where features are extracted using the PCA. It is developed in C++ and is available as an executable file.

Link: <http://nev2lkit.sourceforge.net/>

WIToolbox

This toolbox is developed in MATLAB® and uses wavelet transform and information theory on the temporal structure of spike-train for better classification of spike (Lopes-dos-Santos et al., 2015).

Link: <https://www2.le.ac.uk/centers/csn/software/WI>

JRCLUST

Jenelia Rocket Cluster (JRCLUST) is a software for real-time spike sorting using fast-search of density peak clustering (Rodriguez and Laio, 2014) with a linear-scaling of 1000 channel recording in single workstation and addresses the issue of probe drift in recorded signal (Jun et al., 2017). The core clustering algorithm is developed in

hardware-specific C-language for NVIDIA processors called CUDA that provides a faster processing by parallel computing in graphical processing unit (GPU).

Link: <http://www.jrclust.org/> or <https://github.com/JaneliaSciComp/JRCLUST/wiki>

Kilosort

Kilosort is another spike sorting platform to cluster units in real time from large-scale in vivo multielectrode recordings (Pachitariu et al., 2016). It models the recorded voltage as a sum of template waveforms triggered on the spike times that allows overlapping spikes to be identified and resolved. It also harnesses the computational power of GPU if CUDA-enabled GPU is installed. The software is developed in MATLAB®.

Link: <https://github.com/cortex-lab/KiloSort>

MountainSort

MountainSort is developed in Python to provide a means for automated spike sorting (Chung et al., 2017) using density based clustering called ISO-SPLIT based on the representation of the spikes in low dimensional feature space with an assumption that each cluster arises from a density function that, when projected onto any line, is unimodal. The clustering part of the algorithm uses the parallel computations on CPU cores.

Link: <https://github.com/magland/mountainlab>

spyKING CIRCUS

This is another spike sorting toolbox that works with large silicone probes with thousands of recording channels (Yger et al., 2018). It extracts a dictionary of ‘templates’ from the recording waveforms and uses them to decompose the signal with a template-matching algorithm for clustering. Developed in Python, it provides parallelization over distributed computers.

Link: <https://github.com/spyking-circus/spyking-circus>

YASS (Yet Another Spike Sorter)

YASS is another spike-sorting tool for sorting spikes in dense multielectrode array (MEA) (Lee et al., 2017). The algorithm uses the Dirichlet Process Gaussian Mixture Models (DP-GMM) and non-parametric Bayesian estimate (Wood and Black, 2008) for clustering on subset of feature data-points called 'core set'. The mean of clustered waveforms is used as templates for collided or missed spikes using the matching pursuit approach.

Link: <https://github.com/paninski-lab/yass>

2.3.2 Spike-train dynamics and field potentials

NeuroQuest

NeuroQuest is a MATLAB®-based toolbox that incorporates a number of signal processing algorithms including LFP pre-processing, spike detection, and spike sorting (Kwon et al., 2012). Blind source separation (BSS) is used for the pre-processing of the spiking data, and the PCA, peak-to-peak value and discrete wavelet transformations are used as features. The clustering algorithms include Fuzzy c-means, EM, k-means, linkage, and manual cluster cutting. It also implements basic analysis measuring e.g., inter-spike interval (ISI), peristimulus time histogram (PSTH), cross-correlogram between two spiking units etc.

nSTAT

nSTAT (Cajigas et al., 2012) performs spike train analysis in time-domain using Kalman filtering, frequency domain using multi-taper spectral estimation, and point-process generalized linear model (PP-GLM) for analyzing the causality between spiking trains. It is developed in MATLAB®.

Link: <http://www.neurostat.mit.edu/nstat/>

Spike Train Analysis Toolkit (STAToolkit)

This tool implements information theoretic methods to analyze spike-train behavior with respect to stimuli (Goldberg et al., 2009). The backend computations are

developed in C for better performance and speed but are interfaced with MATLAB® using the MEX framework.

Link: <http://neuroanalysis.org>

FIND toolbox

The MATLAB® based FIND toolbox provides unified data import function from various proprietary formats and implements the analysis techniques for multiple-neuron recordings and network simulations (Meier et al., 2008)

Link: <http://find.bccn.uni-freiburg.de/>

sigTOOL

sigTOOL is written in MATLAB® which provides the analysis for neural spike trains, i.e., ISI distribution, Poincare plots, spike-triggered averaging etc., and frequency-domain analysis, i.e., power-spectral estimation or coherence estimation (Lidierth, 2009).

Link: <http://sigtool.sourceforge.net/>

STAR

Spike Train Analysis with R (STAR) is a tool to visualize spike trains and fit, test, and compare models for discharge applied to data (Pouzat and Chaffiol, 2009). As it explains, the toolbox is developed in R programming language.

Link: <https://sites.google.com/site/spiketrainanalysiswithr/>

Neuropy:

Neuropy implements analyses of spike trains in relation to stimuli (Spacek et al., 2008). The analyses include ISI histograms, instantaneous firing rates and their distributions, auto- and cross-correlogram and spike-triggered average. Other analyses include binary code of population spike trains, their correlation coefficient distribution, maximum entropy using modelling of such codes etc. The toolbox is written in Python.

Link: <http://www.swindale.ecc.ubc.ca/neuropy>

MEA-Tools

Micro-Electrode Array Tools (MEA-Tools), a graphical user interface based tool, implements shape-based spike sorting, visualization of continuous data, and extracting characteristic LFP segments (Egert et al., 2002). It is developed in MATLAB®.

Link: <http://material.brainworks.uni-freiburg.de/research/meatools/>

Data-MEAns

This is an executable software developed in Delphi 7 and MATLAB® with basic analyses like rate histograms, correlograms etc. and spatiotemporal visualization of unit activity in an MXN electrode configuration (Bonomini et al., 2005). It also provides clustering using the nearest-neighbor or k-means algorithms.

Link: <http://cortivis.umh.es/>

Brain System for Multivariate Autoregressive Time Series (BSMART)

Based on the multivariate auto-regressive model (MAR), BSMART is intended to establish causal relationship and direction of driving between multiple neural signals using the Granger causality analysis (Cui et al., 2008). The MAR analysis also facilitates computing the spectral quantities such as spectral power, partial directed coherence and directed transfer function etc. It is developed in MATLAB® and C.

Link: <http://www.brain-smart.org/>

CSDPlotter

This software is developed to visualize the current source density (CSD) using a graphical user interface (Pettersen et al., 2006). It is implemented in MATLAB® and incorporates the standard CSD method or the inverse CSD method. It is a pure GUI application, and the parameters for the CSD plots can be changed from the user-interface. It is primarily designed to plot extracellular recording from electrodes perpendicular to cortex at different cortical depths.

Link: <https://github.com/espenhgn/CSDplotter>

iCSD 2D

iCSD 2D (Leski et al., 2011) is a CSD plotter for two-dimensional data. It is a GUI toolbox primarily developed in MATLAB®, but later migrated to Python.

Link: <http://www.neuroinf.pl/Members/szleski/csd2d/toolbox>

SigMate

It is a MATLAB-based software primarily for the extensive pre-processing of LFP data (baseline correction, artefact removal etc.) and the visualization of CSD (Mahmud et al., 2010). It also integrates EEGLAB, a popular tool for electroencephalography data analysis (Delorme and Makeig, 2004), and Wave_Clus for spike analysis.

Link: <https://sites.google.com/site/muftimahmud/codes>

QSpikeTools

This is a cloud-computing framework to facilitate the parallel processing and analysis of extracellular multi-unit activity (Mahmud et al., 2014). The software follows a client-server architecture and is developed using MATLAB® and Bash scripts. It works by delegating the processing of CPU-intensive operations for electrode channels to a multi-core computer or to a computer cluster.

Link: <https://sites.google.com/site/qspiketool/>

2.3.3 General purpose toolboxes

Chronux

Chronux is a MATLAB® library providing routines for the analysis of point process and continuous neural data (Bokil et al., 2010). It provides the functionalities for pre-processing, filtering, spectral analysis and visualization of the recorded signal.

Link: <http://chronux.org/>

pyEntropy

This is a Python module for the estimation of the entropy and information theoretic quantities from the discrete signals using bias correction methods (Ince et al., 2009). It also includes an algorithm for estimating the marginal constraint maximum

entropy distributions which is used for quantitative assessment of the higher order interactions between the neural signals treated as stochastic communication channels.

Link: <https://code.google.com/archive/p/pyentropy/>

Information Breakdown ToolBox (ibTB)

This software measures different bias-corrected estimates of the information theoretic quantities for both continuous and discrete signals i.e., LFP and spike trains (Magri et al., 2009). It is written in C and MATLAB®. It provides information breakdown technique for exploring the encoding of sensory stimuli by different groups of neurons.

Link: <http://static-content.springer.com/esm/art%3A10.1186%2F1471-2202-10-81/MediaObjects/1471-2202-10-81-SI.zip>

OpenElectrophy

This is a GUI and command-line based Python module to facilitate the management and visualization of the data (Garcia and Fourcaud-Trocme, 2009). It stores all experimental and meta-data in a single central MySQL database. It incorporates offline spike-sorting methods using k-means and super-paramagnetic clustering algorithms (Blatt et al., 1996). The non-stationary oscillations in LFP are also detected using the scalogram to extract individual oscillations with ridge extraction methods. This toolbox maintains a very standard workflow and data-integration and very modular in architecture.

Link: <http://neuralensemble.org/OpenElectrophy/>

Relacs

Relaxed Electrophysiology data acquisition, Control, and Stimulation (Relacs) is a data acquisition platform for closed-loop experiments (Benda and Grewe, 2009). It is developed in C++ and performs online analysis and visualization. It can generate stimuli (e.g. offset, variance) in a closed-loop fashion to control the running experiment. The platform also includes some basic analysis pertaining to firing rates (ISI, PSTH etc.) and mutual information content etc.

Link: <http://relacs.sourceforge.net/>

2.3.4 Neuronal encoding of spatial behavior

As we can see that the software was intended for one of the following purposes,

- Identification and isolation of single-unit action-potentials (spike-sorting) from electrophysiological recordings
- Analysis of spike-train dynamics
- Analysis of stimulus-triggered average spike-firing and implementation of relevant information theoretic approaches
- Analysis of frequency spectrum, current source density or stimulus-triggered dynamics of the LFP data

Although Neuroscope implements visualization of continuously recorded signals, spiking activity, and behavioral events, it does not provide any data analysis. There is no comprehensive toolbox for studying the single-unit characteristics with respect to the spatial information. Although it is understandable that each research lab may have custom-written codes or software that suits the type of methods and data processing pipeline, it is not very clear why there no such open-source toolbox regarding the study of cognitive mapping of neurons. We speculate that the pattern of analysis methods that can be accepted as ‘standard’ to be included in an open-source software grew over many years of research in this field. This may result in the researchers being used to or being trained to the usual practices of the individual laboratories. It may also be historic to study one neuron at a time given that the earlier recording electrodes allowed studying only a very limited number of neurons (Stevenson and Kording, 2011).

2.3.5 Summary of toolboxes

The software for analyzing single neuron and LFP data from implanted electrodes are summarized in the table below.

Table 2.2 Summary of the scopes and functionalities of available tools and software for single-unit data analysis

Name of toolbox	Reference	Development environment	Functionality	Scope
WaveClus	(Quiroga et al., 2004)	MATLAB	Implementation of unsupervised spike sorting using wavelet coefficients of spike waveforms as features followed by super-paramagnetic clustering	Spike sorting
MClust	(Fraley and Raftery, 2002)	MATLAB & R	Manual or automated clustering (using Expectation-Maximization) with a selection of or user-defined features using graphical user interface or batch mode.	
OSort	(Rutishauser et al., 2006, Spacek et al., 2008)	MATLAB	Online spike sorting using spike waveform templates. The sorting is unsupervised	
Spyke	(Spacek et al., 2008)	Python	Python package for data visualization and navigation of extracellular waveform data, and spike sorting using template matching approach	
KlustaKwik	(Harris et al., 2000)	C++	Spike sorting using masked EK algorithm on a selected subset of high-dimensional features from high-density electrode array	
SAC	(Shoham et al., 2003)	MATLAB	Spike sorting using principal components of spike waveforms as features and clustering using a mixture of t-distributions algorithm. It includes a graphical user interface.	

EToS	(Takekawa et al., 2012)	Standalone suite of command-line interface	Weighted-PCS based feature extraction and clustering using robust variation Bayes for a mixture of t-distributions
SpikeOMatic	(Pouzat et al., 2004)	R	R package for spike sorting using a generative model (hidden Markov model)
UltraMegaSort2000	(Fee et al., 1996)	MATLAB	hierarchical clustering algorithm for spike sorting using similarities of spike shape and spike timing statistics
NEV2lkit	(Bongard et al., 2014)	C++	GUI executable for analysis, visualization and classification of spikes using number of clustering algorithms. Discrimination of neural spikes is based on PCA
WIToolbox	(Lopes-dos-Santos et al., 2015)	MATLAB	wavelet transform and information theory on the temporal structure of spike-train for better classification of spikes
JRCLUST	(Jun et al., 2017)	CUDA	Real-time spike sorting using fast-search of density peak clustering. It can scale up to 1000 recording channels in single workstation and addresses the issue of probe drift in recorded signal. CUDA provides a faster processing by parallel computing in GPU.
Kilosort	(Pachitariu et al., 2016)	MATLAB	Spike sorting platform to cluster units in real time from large-scale in vivo

			<p>multielectrode recordings. Provides a means for identifying and resolving overlapping spikes. Uses computational power of GPU if CUDA-enabled GPU is installed.</p>
Mountainsort	(Chung et al., 2017)	Python	<p>automated spike sorting using density-based clustering called ISO-SPLIT. The clustering part of the algorithm uses the parallel computations on CPU cores.</p>
SpykingCircus	(Yger et al., 2018)		<p>Spike sorting toolbox that works with large silicone probes with thousands of recording channels. It extracts a dictionary of 'templates' from the recording and uses these templates to decompose the signal with a template-matching algorithm for clustering. It provides parallelization over distributed computers.</p>
Yet Another Spike Sorter (YASS)	(Lee et al., 2017)		<p>Spike-sorter for dense MEA using Dirichlet Process Gaussian Mixture Models (DP-GMM) and non-parametric Bayesian estimate for clustering. The means of clusters are used as templates for collided or missed spikes using matching pursuit approach.</p>

NeuroQuest	(Kwon et al., 2012)	MATLAB	LFP pre-processing, spike detection, spike-sorting using blind source separation (BSS) as pre-processing, PCA and peak-to-peak value along with discrete wavelet transform coefficients as the features, and Fuzzy c-means, Expectation-Maximization, k-means, linkage and manual cluster cutting as the clustering options. It also avails basic analysis i.e., interspike interval (ISI), peristimulus time histogram (PSTH), cross-correlogram between two separate units etc.	Spike sorting and spike train analysis
nSTAT	(Cajigas et al., 2012)	MATLAB	Implementation of point process-generalized linear model algorithms (PP-GLM)	Spike train analysis
Spike Train Analysis Toolkit	(Goldberg et al., 2009)	MATLAB	Information theoretic analysis of spike train	
FIND toolbox	(Meier et al., 2008)	MATLAB	Toolbox providing analysis techniques for multiple-neuron recordings and network simulations	
sigTOOL	(Lidierth, 2009)	MATLAB	Analysis of neural spike train (inter-spike interval distribution, Poincare plots, spike-triggered averaging etc.), and frequency analysis (power spectral analysis, coherence estimation etc.)	
STAR	(Pouzat and Chaffiol, 2009)	R	Tools to visualize spike trains and fit, test and compare	

			models of discharge applied to actual data	
NeuroPy	(Spacek et al., 2008)	Python	Analysis of spike trains and their relation to stimuli	
MEA tools	(Egert et al., 2002)	MATLAB	Implements shape-based spike sorting, visualization of continuous data, extracting characteristic LFP segments	Spike sorting LFP
DATA-MEAns	(Bonomini et al., 2005)	Delphi 7, MATLAB	Executable software with basic analyses like rate histograms, correlograms etc. and spatiotemporal visualization of unit activity in an MXN electrode configuration. It also provides clustering using nearest-neighbor or k-means algorithms.	Spike sorting and spike train analysis
BSMART	(Cui et al., 2008)	MATLAB, C	Exploring causal relationship using multivariate autoregressive model and Granger causality. It also computes other spectral quantities such as spectral power, partial directed coherence and directed transfer function etc.	
CSDPlotter	(Pettersen et al., 2006)	MATLAB	Graphical user interface-based toolbox for estimation and visualization of current source density (CSD) from implanted electrode recordings. The estimation techniques include both standard CSD and inverse CSD (iCSD)	Field potential
iCSD 2D	(Leski et al., 2011)	MATLAB	Implementation of iCSD method in two-dimensional	

			electrode distribution (i.e., Utah array)	
SigMate	(Mahmud et al., 2010)	MATLAB	Tool for extensive pre-processing of LFP data (baseline correction, artefact removal etc.) and visualization of CSD. It also integrates EEGLAB and Wave_Clus.	
QSpikesTools	(Mahmud et al., 2014)		Cloud-computing, client-server framework to facilitate parallel processing and analysis of extracellular multi-unit activity. It delegates the processing of CPU-intensive operations for electrode channel to a multi-core computer or to a computer cluster.	Cloud computing
Chronux	(Bokil et al., 2010)	MATLAB	A general neurobiological data analysis toolbox including pre-processing and spectral analysis functionalities for continuous (e.g., LFP) and point-process signals (e.g., spike-occurrence).	General purpose
lbtb	(Magri et al., 2009)	MATLAB	Implementation of bias corrected estimates of information theoretic quantities	
pyEntropy	(Ince et al., 2009)	Python	Implementation of bias corrected estimates of information theoretic quantities. Similar to <i>infoToolbox</i>	

OpenElectrophy	(Garcia and Fourcaud-Trocme, 2009)	Python	GUI-, and command line-based python module to facilitate data management and visualization. Incorporates an offline spike-sorter and time-frequency analysis tools along with detection of transient oscillations in LFP	
Relacs	(Benda and Grewe, 2009)	C++	Platform for closed loop experiments facilitating both online analysis and stimulus generation, and integration of experiments and model simulations	Experimentation, data acquisition and stimulus-triggered spike data analysis

2.4 Project objective

The neuronal mechanisms that drive the cognitive ability to perceive spatial information and enable navigation are yet to explore. New experimental methods are being employed in for improved understanding of the mechanism. There is also an exponential increase in the number of recording sites as cutting-edge technologies evolve that enables recording a large number of single-unit data from many sites of the brain simultaneously and helps better understanding the functions of the brain in the cellular level and in network level. New analysis techniques are also being developed simultaneously. The novel experiments and the new electrode technologies poses the challenge of analyzing an ever-growing amount of data. But there is no suitable toolbox to bring all these analysis methods required for investigating spatial information processing ability of the brain into one platform in a managed standard procedure using a friendly and interactive interface. Therefore, the objective of the current research project is to develop such software with a parallel development and inclusion of new analysis methods in an agile development framework. The basic features or objectives that the project will pursue are:

- ❑ To create a framework for in-vivo single-unit data analysis
- ❑ To provide a unified platform for established methods
- ❑ To provide an easy-to-use interactive tool for neuroscientists
- ❑ To enable easy implementation of problem-specific analysis techniques
- ❑ To save time to analyze data, bringing efficiency and exceling in research
- ❑ To provide an open-source ecosystem for neurophysiology data analysis

Neuroimaging already has matured and standardized open-source tools such as SPM, BrainVoyager, AFNI etc. those are widely accepted among the scientific community. In contrast to that, behavioral neurophysiology is rapidly evolving, and still in research and development phase owing to the diverse technologies involving recording and stimulation. The situation in behavioral neurophysiology now is about the same as in neuroimaging two decades ago requiring a revolution in how we address the challenges. NeuroChaT is an effort to that pursuit.

2.5 References

- ALIVISATOS, A. P., ANDREWS, A. M., BOYDEN, E. S., CHUN, M., CHURCH, G. M., DEISSEROTH, K., DONOGHUE, J. P., FRASER, S. E., LIPPINCOTT-SCHWARTZ, J., LOOGER, L. L., MASMANIDIS, S., MCEUEN, P. L., NURMIKKO, A. V., PARK, H., PETERKA, D. S., REID, C., ROUKES, M. L., SCHERER, A., SCHNITZER, M., SEJNOWSKI, T. J., SHEPARD, K. L., TSAO, D., TURRIGIANO, G., WEISS, P. S., XU, C., YUSTE, R. & ZHUANG, X. 2013. Nanotools for Neuroscience and Brain Activity Mapping. *ACS Nano*, 7, 1850-1866.
- ALLEN, W. E., KAUVAR, I. V., CHEN, M. Z., RICHMAN, E. B., YANG, S. J., CHAN, K., GRADINARU, V., DEVERMAN, B. E., LUO, L. & DEISSEROTH, K. 2017. Global Representations of Goal-Directed Behavior in Distinct Cell Types of Mouse Neocortex. *Neuron*, 94, 891-907.e6.
- ALME, C. B., MIAO, C., JEZEK, K., TREVES, A., MOSER, E. I. & MOSER, M.-B. 2014. Place cells in the hippocampus: Eleven maps for eleven rooms. *Proceedings of the National Academy of Sciences*, 111, 18428-18435.
- ANDERSON, M. I. & JEFFERY, K. J. 2003. Heterogeneous Modulation of Place Cell Firing by Changes in Context. *The Journal of Neuroscience*, 23, 8827-8835.
- BARRY, C., LEVER, C., HAYMAN, R., HARTLEY, T., BURTON, S., O'KEEFE, J., JEFFERY, K. & BURGESS, N. 2006. The boundary vector cell model of place cell firing and spatial memory. *Reviews in the neurosciences*, 17, 71-97.
- BASSETT, J. P. & TAUBE, J. S. 2001. Neural correlates for angular head velocity in the rat dorsal tegmental nucleus.
- BENDA, J. & GREWE, J. 2009. Closed-loop electrophysiological experiments and metadata management with RELACS and LabLog. *BMC Neuroscience*, 10, 1-1.
- BENDER, F., GORBATI, M., CADAVIECO, M. C., DENISOVA, N., GAO, X., HOLMAN, C., KOROTKOVA, T. & PONOMARENKO, A. 2015. Theta oscillations regulate the speed of locomotion via a hippocampus to lateral septum pathway. *Nature Communications*, 6, 8521.
- BESTEL, R., DAUS, A. W. & THIELEMANN, C. 2012. A novel automated spike sorting algorithm with adaptable feature extraction. *J Neurosci Methods*, 211, 168-78.
- BISHOP, G. H. & O'LEARY, J. L. 1942. Factors determining the form of the potential record in the vicinity of the synapses of the dorsal nucleus of the lateral geniculate body. *Journal of Cellular and Comparative Physiology*, 19, 315-331.
- BJERKNES, T. L., DAGSLOTT, N. C., MOSER, E. I. & MOSER, M. B. 2018. Path integration in place cells of developing rats. *Proc Natl Acad Sci U S A*, 115, E1637-E1646.
- BLANCHE, T. J., SPACEK, M. A., HETKE, J. F. & SWINDALE, N. V. 2005. Polytrodes: high-density silicon electrode arrays for large-scale multiunit recording. *J Neurophysiol*, 93, 2987-3000.
- BLATT, M., WISEMAN, S. & DOMANY, E. 1996. Superparamagnetic Clustering of Data. *Physical Review Letters*, 76, 3251-3254.
- BOKIL, H., ANDREWS, P., KULKARNI, J. E., MEHTA, S. & MITRA, P. P. 2010. Chronux: a platform for analyzing neural signals. *J Neurosci Methods*, 192, 146-51.
- BONGARD, M., MICOL, D. & FERNANDEZ, E. 2014. NEV2lkit: a new open source tool for handling neuronal event files from multi-electrode recordings. *Int J Neural Syst*, 24, 1450009.
- BONOMINI, M. P., FERRANDEZ, J. M., BOLEA, J. A. & FERNANDEZ, E. 2005. DATA-MEAns: An open source tool for the classification and management of neural ensemble recordings. *Journal of Neuroscience Methods*, 148, 137-146.
- BRANDON, M. P., BOGAARD, A. R., LIBBY, C. P., CONNERNEY, M. A., GUPTA, K. & HASSELMO, M. E. 2011. Reduction of Theta Rhythm Dissociates Grid Cell Spatial Periodicity from Directional Tuning. *Science*, 332, 595-599.
- BRANDON, M. P., BOGAARD, A. R., SCHULTHEISS, N. W. & HASSELMO, M. E. 2013. Segregation of cortical head direction cell assemblies on alternating theta cycles. *Nat Neurosci*, 16, 739-48.
- BRANDON, MARK P., KOENIG, J., LEUTGEB, JILL K. & LEUTGEB, S. 2014. New and Distinct Hippocampal Place Codes Are Generated in a New Environment during Septal Inactivation. *Neuron*, 82, 789-796.

- BROADBENT, N. J., SQUIRE, L. R. & CLARK, R. E. 2004. Spatial memory, recognition memory, and the hippocampus. *Proc Natl Acad Sci U S A*, 101, 14515-20.
- BUCHWALD, J. S. & GROVER, F. S. 1970. Amplitudes of background fast activity characteristic of specific brain sites. *Journal of Neurophysiology*, 33, 148-159.
- BURKE, S. N., MAURER, A. P., HARTZELL, A. L., NEMATOLLAHI, S., UPRETY, A., WALLACE, J. L. & BARNES, C. A. 2012. Representation of three-dimensional objects by the rat perirhinal cortex. *Hippocampus*, 22, 2032-2044.
- BUZSAKI, G., BICKFORD, R., PONOMAREFF, G., THAL, L., MANDEL, R. & GAGE, F. 1988. Nucleus basalis and thalamic control of neocortical activity in the freely moving rat. *The Journal of Neuroscience*, 8, 4007-4026.
- CACUCCI, F., LEVER, C., WILLS, T. J., BURGESS, N. & O'KEEFE, J. 2004. Theta-modulated place-by-direction cells in the hippocampal formation in the rat. *J Neurosci*, 24, 8265-77.
- CAJIGAS, I., MALIK, W. Q. & BROWN, E. N. 2012. nSTAT: open-source neural spike train analysis toolbox for Matlab. *J Neurosci Methods*, 211, 245-64.
- CHEN, G., KING, J. A., BURGESS, N. & O'KEEFE, J. 2013. How vision and movement combine in the hippocampal place code. *Proc Natl Acad Sci U S A*, 110, 378-83.
- CHERSI, F. & BURGESS, N. 2015. The Cognitive Architecture of Spatial Navigation: Hippocampal and Striatal Contributions. *Neuron*, 88, 64-77.
- CHI-MING, H. & BUCHWALD, J. S. 1977. Interpretation of the vertex short-latency acoustic response: A study of single neurons in the brain stem. *Brain Research*, 137, 291-303.
- CHOLVIN, T., HOK, V., GIORGI, L., CHAILLAN, F. A. & POU CET, B. 2018. Ventral Midline Thalamus Is Necessary for Hippocampal Place Field Stability and Cell Firing Modulation. *J Neurosci*, 38, 158-172.
- CHUNG, J. E., MAGLAND, J. F., BARNETT, A. H., TOLOSA, V. M., TOOKER, A. C., LEE, K. Y., SHAH, K. G., FELIX, S. H., FRANK, L. M. & GREENGARD, L. F. 2017. A Fully Automated Approach to Spike Sorting. *Neuron*, 95, 1381-1394 e6.
- COLGIN, L. L. 2013. Mechanisms and functions of theta rhythms. *Annu Rev Neurosci*, 36, 295-312.
- COLGIN, L. L. 2016. Rhythms of the hippocampal network. *Nat Rev Neurosci*, 17, 239-49.
- COLLETT, M., COLLETT, T. S. & WEHNER, R. 1999. Calibration of vector navigation in desert ants. *Curr Biol*, 9, 1031-4.
- CROCHET, S., POULET, JAMES F. A., KREMER, Y. & PETERSEN, CARL C. H. 2011. Synaptic Mechanisms Underlying Sparse Coding of Active Touch. *Neuron*, 69, 1160-1175.
- CUI, J., XU, L., BRESSLER, S. L., DING, M. & LIANG, H. 2008. BSMART: a Matlab/C toolbox for analysis of multichannel neural time series. *Neural Netw*, 21, 1094-104.
- DAYAN, P. & ABBOTT, L. F. 2005. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, The MIT Press.
- DELORME, A. & MAKEIG, S. 2004. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134, 9-21.
- EGERT, U., KNOTT, T., SCHWARZ, C., NAWROT, M., BRANDT, A., ROTTER, S. & DIESMANN, M. 2002. MEA-Tools: an open source toolbox for the analysis of multi-electrode data with MATLAB. *J Neurosci Methods*, 117, 33-42.
- EINEVOLL, G. T., FRANKE, F., HAGEN, E., POUZAT, C. & HARRIS, K. D. 2012. Towards reliable spike-train recordings from thousands of neurons with multielectrodes. *Curr Opin Neurobiol*, 22, 11-7.
- EKSTROM, A. D. 2015. Why vision is important to how we navigate. *Hippocampus*, 25, 731-5.
- EKSTROM, A. D., ARNOLD, A. E. & IARIA, G. 2014. A critical review of the allocentric spatial representation and its neural underpinnings: toward a network-based perspective. *Front Hum Neurosci*, 8, 803.
- EKSTROM, A. D., KAHANA, M. J., CAPLAN, J. B., FIELDS, T. A., ISHAM, E. A., NEWMAN, E. L. & FRIED, I. 2003. Cellular networks underlying human spatial navigation. *Nature*, 425, 184.
- ETIENNE, S., LUDEK, N. & BRUNO, P. 2000. Contribution of multiple sensory information to place field stability in hippocampal place cells. *Hippocampus*, 10, 64-76.

- FEE, M. S., MITRA, P. P. & KLEINFELD, D. 1996. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-Gaussian variability. *Journal of Neuroscience Methods*, 69, 175-188.
- FINKELSTEIN, A., LAS, L. & ULANOVSKY, N. 2016. 3-D Maps and Compasses in the Brain. *Annu Rev Neurosci*, 39, 171-96.
- FRALEY, C. & RAFTERY, A. E. 2002. Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association*, 97, 611-631.
- FRANZ, M. O. & MALLOT, H. A. 2000. Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30, 133-153.
- FYHN, M., MOLDEN, S., WITTER, M. P., MOSER, E. I. & MOSER, M. B. 2004. Spatial representation in the entorhinal cortex. *Science*, 305, 1258-64.
- GARCIA, S. & FOURCAUD-TROCME, N. 2009. OpenElectrophy: An Electrophysiological Data- and Analysis-Sharing Framework. *Front Neuroinform*, 3, 14.
- GEISLER, C., ROBBE, D., ZUGARO, M., SIROTA, A. & BUZSÁKI, G. 2007. Hippocampal place cell assemblies are speed-controlled oscillators. *Proceedings of the National Academy of Sciences*, 104, 8149-8154.
- GEORGOPOULOS, A., SCHWARTZ, A. & KETTNER, R. 1986. Neuronal population coding of movement direction. *Science*, 233, 1416-1419.
- GERSTNER, W. & KISTLER, W. M. 2002. *Spiking Neuron Models, Single Neurons, Populations, Plasticity*, Cambridge University Press.
- GEVA-SAGIV, M., ROMANI, S., LAS, L. & ULANOVSKY, N. 2016. Hippocampal global remapping for different sensory modalities in flying bats. *Nat Neurosci*, 19, 952-8.
- GOLDBERG, D. H., VICTOR, J. D., GARDNER, E. P. & GARDNER, D. 2009. Spike train analysis toolkit: enabling wider application of information-theoretic techniques to neurophysiology. *Neuroinformatics*, 7, 165-78.
- GRAY, C. M., MALDONADO, P. E., WILSON, M. & MCNAUGHTON, B. 1995. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *J Neurosci Methods*, 63, 43-54.
- GRIEVES, R. M. & JEFFERY, K. J. 2017. The representation of space in the brain. *Behav Processes*, 135, 113-131.
- HAFTING, T., FYHN, M., MOLDEN, S., MOSER, M.-B. & MOSER, E. I. 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436, 801-806.
- HARRIS, K. D., HENZE, D. A., CSICSVARI, J., HIRASE, H. & BUZSAKI, G. 2000. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *J Neurophysiol*, 84, 401-14.
- HARRIS, K. D., QUIROGA, R. Q., FREEMAN, J. & SMITH, S. L. 2016. Improving data quality in neuronal population recordings. *Nat Neurosci*, 19, 1165-74.
- HARTLEY, T., LEVER, C., BURGESS, N. & O'KEEFE, J. 2014. Space in the brain: how the hippocampal formation supports spatial cognition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369.
- HARTLINE, D. K. & COLMAN, D. R. 2007. Rapid Conduction and the Evolution of Giant Axons and Myelinated Fibers. *Current Biology*, 17, R29-R35.
- HEINRICHER, M. M. 2004. Principles of extracellular single unit recording. In: ISRAEL, Z. & BURCHIEL, K. (eds.) *Microelectrode Recording in Movement Disorder Surgery*. 1 ed. Stuttgart: Thieme.
- HENZE, D. A., BORHEGYI, Z., CSICSVARI, J., MAMIYA, A., HARRIS, K. D. & BUZSÁKI, G. 2000. Intracellular Features Predicted by Extracellular Recordings in the Hippocampus In Vivo. *Journal of Neurophysiology*, 84, 390-400.
- HILL, D. N., MEHTA, S. B. & KLEINFELD, D. 2011. Quality metrics to accompany spike sorting of extracellular signals. *J Neurosci*, 31, 8699-705.
- HOLLUP, S. A., MOLDEN, S., DONNETT, J. G., MOSER, M.-B. & MOSER, E. I. 2001. Accumulation of Hippocampal Place Fields at the Goal Location in an Annular Watermaze Task. *The Journal of Neuroscience*, 21, 1635-1644.

- HOYDAL, O. A., SKYTOEN, E. R., MOSER, M.-B. & MOSER, E. I. 2018. Object-vector coding in the medial entorhinal cortex. *bioRxiv*.
- HROMÁDKA, T., DEWEESE, M. R. & ZADOR, A. M. 2008. Sparse Representation of Sounds in the Unanesthetized Auditory Cortex. *PLOS Biology*, 6, e16.
- HUBEL, D. H. & WIESEL, T. N. 1959. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148, 574-591.
- HULATA, E., SEGEV, R. & BEN-JACOB, E. 2002. A method for spike sorting and detection based on wavelet packets and Shannon's mutual information. *Journal of Neuroscience Methods*, 117, 1-12.
- HUMPHREY, D. 1979. Extracellular single unit recording methods. In: HUMPHREY, E. (ed.) *Electrophysiological Techniques*. Bethesda, MD: Society for Neuroscience.
- HUMPHREY, D. R. & CORRIE, W. S. 1978. Properties of pyramidal tract neuron system within a functionally defined subregion of primate motor cortex. *Journal of Neurophysiology*, 41, 216-243.
- INCE, R. A. A., PETERSEN, R. S., SWAN, D. C. & PANZERI, S. 2009. Python for Information Theoretic Analysis of Neural Data. *Frontiers in Neuroinformatics*, 3, 4.
- ITO, H. T., ZHANG, S.-J., WITTER, M. P., MOSER, E. I. & MOSER, M.-B. 2015. A prefrontal-thalamo-hippocampal circuit for goal-directed spatial navigation. *Nature*, 522, 50-55.
- ITO, I., ONG, R. C.-Y., RAMAN, B. & STOPFER, M. 2008. Sparse odor representation and olfactory learning. *Nature Neuroscience*, 11, 1177.
- JANKOWSKI, M. M., ISLAM, M. N., WRIGHT, N. F., VANN, S. D., ERICHSEN, J. T., AGGLETON, J. P. & O'MARA, S. M. 2014. Nucleus reuniens of the thalamus contains head direction cells. *Elife*, 3.
- JANKOWSKI, M. M. & O'MARA, S. M. 2015. Dynamics of place, boundary and object encoding in rat anterior claustrum. *Front Behav Neurosci*, 9, 250.
- JANKOWSKI, M. M., PASSECKER, J., ISLAM, M. N., VANN, S., ERICHSEN, J. T., AGGLETON, J. P. & O'MARA, S. M. 2015. Evidence for Spatially-Responsive Neurons in the Rostral Thalamus. *Frontiers in Behavioral Neuroscience*, 9.
- JEEWAJEE, A., BARRY, C., DOUCHAMPS, V., MANSON, D., LEVER, C. & BURGESS, N. 2014. Theta phase precession of grid and place cell firing in open environments. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369.
- JEFFERY, K. J., WILSON, J. J., CASALI, G. & HAYMAN, R. M. 2015a. Neural encoding of large-scale three-dimensional space-properties and constraints. *Front Psychol*, 6, 927.
- JEFFERY, K. J., WILSON, J. J., CASALI, G. & HAYMAN, R. M. A. 2015b. Neural encoding of large-scale three-dimensional space – properties and constraints. *Frontiers in Psychology*, 6.
- JEZEK, K., HENRIKSEN, E. J., TREVES, A., MOSER, E. I. & MOSER, M.-B. 2011. Theta-paced flickering between place-cell maps in the hippocampus. *Nature*, 478, 246.
- JOHNSON, K. O. 2000. Neural Coding. *Neuron*, 26, 563-566.
- JUN, J. J., MITELUT, C., LAI, C., GRATIY, S., ANASTASSIOU, C. & HARRIS, T. D. 2017. Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction. *bioRxiv*.
- KADIR, S. N., GOODMAN, D. F. & HARRIS, K. D. 2014. High-dimensional cluster analysis with the masked EM algorithm. *Neural Comput*, 26, 2379-94.
- KAMONDI, A., ACSADY, L., WANG, X. J. & BUZSAKI, G. 1998. Theta oscillations in somata and dendrites of hippocampal pyramidal cells in vivo: activity-dependent phase-precession of action potentials. *Hippocampus*, 8, 244-61.
- KANDEL, E., SCHWARTZ, J. & JESSEL, T. M. 1991. *Principles of Neural Science*, Elsevier.
- KLOPPENBURG, P. & NAWROT, MARTIN P. 2014. Neural Coding: Sparse but On Time. *Current Biology*, 24, R957-R959.
- KNIERIM, J. J., NEUNUEBEL, J. P. & DESHMUKH, S. S. 2014. Functional correlates of the lateral and medial entorhinal cortex: objects, path integration and local-global reference frames. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369.

- KOSTAL, L., LANSKY, P. & ROSPARS, J.-P. 2007. REVIEW ARTICLE: Neuronal coding and spiking randomness. *European Journal of Neuroscience*, 26, 2693-2701.
- KROPFF, E., CARMICHAEL, J. E., MOSER, M. B. & MOSER, E. I. 2015. Speed cells in the medial entorhinal cortex. *Nature*, 523, 419-24.
- KWON, K. Y., ELDAWLATLY, S. & OWEISS, K. 2012. NeuroQuest: a comprehensive analysis tool for extracellular neural ensemble recordings. *J Neurosci Methods*, 204, 189-201.
- LANGSTON, R. F., AINGE, J. A., COUEY, J. J., CANTO, C. B., BJERKNES, T. L., WITTER, M. P., MOSER, E. I. & MOSER, M.-B. 2010. Development of the Spatial Representation System in the Rat. *Science*, 328, 1576-1580.
- LEE, J., CARLSON, D., SHOKRI, H., YAO, W., GOETZ, G., HAGEN, E., BATTY, E., CHICHILNISKY, E., EINEVOLL, G. & PANINSKI, L. 2017. YASS: Yet Another Spike Sorter. *bioRxiv*.
- LESKI, S., PETTERSEN, K. H., TUNSTALL, B., EINEVOLL, G. T., GIGG, J. & WOJCIK, D. K. 2011. Inverse current source density method in two dimensions: inferring neural activation from multielectrode recordings. *Neuroinformatics*, 9, 401-25.
- LEVER, C., BURTON, S., JEEWAJEE, A., O'KEEFE, J. & BURGESS, N. 2009. Boundary vector cells in the subiculum of the hippocampal formation. *J Neurosci*, 29, 9771-7.
- LEWICKI, M. S. 1998. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network*, 9, R53-78.
- LIDIERTH, M. 2009. sigTOOL: A MATLAB-based environment for sharing laboratory-developed software to analyze biological signals. *J Neurosci Methods*, 178, 188-96.
- LOGOTHETIS, N. K. 2003. The Underpinnings of the BOLD Functional Magnetic Resonance Imaging Signal. *The Journal of Neuroscience*, 23, 3963-3971.
- LOPES-DOS-SANTOS, V., PANZERI, S., KAYSER, C., DIAMOND, M. E. & QUIROGA, R. Q. 2015. Extracting information in spike time patterns with wavelets and information theory. *Journal of Neurophysiology*, 113, 1015-1033.
- MAGRI, C., WHITTINGSTALL, K., SINGH, V., LOGOTHETIS, N. K. & PANZERI, S. 2009. A toolbox for the fast information analysis of multiple-site LFP, EEG and spike train recordings. *BMC Neurosci*, 10, 81.
- MAHMUD, M., BERTOLDO, A., GIRARDI, S., MASCHIETTO, M. & VASSANELLI, S. SigMate: A MATLAB-based neuronal signal processing tool. 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, 31 Aug.-4 Sept. 2010. 1352-1355.
- MAHMUD, M., PULIZZI, R., VASILAKI, E. & GIUGLIANO, M. 2014. QSpoke tools: a generic framework for parallel batch preprocessing of extracellular neuronal signals recorded by substrate microelectrode arrays. *Front Neuroinform*, 8, 26.
- MAUNSELL, J. H. & ESSEN, D. C. V. 1983. Functional properties of neurons in middle temporal visual area of the macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. *Journal of Neurophysiology*, 49, 1127-1147.
- MCNAUGHTON, B. L., BATTAGLIA, F. P., JENSEN, O., MOSER, E. I. & MOSER, M. B. 2006. Path integration and the neural basis of the 'cognitive map'. *Nat Rev Neurosci*, 7, 663-78.
- MEIER, R., EGERT, U., AERTSEN, A. & NAWROT, M. P. 2008. FIND — A unified framework for neural data analysis. *Neural Networks*, 21, 1085-1093.
- MITTELSTAEDT, M. L. & MITTELSTAEDT, H. 1980. Homing by path integration in a mammal. *Naturwissenschaften*, 67, 566-567.
- MITZDORF, U. 1985. Current source-density method and application in cat cerebral cortex: investigation of evoked potentials and EEG phenomena. *Physiological Reviews*, 65, 37-100.
- MITZDORF, U. 1987. Properties of the Evoked Potential Generators: Current Source-Density Analysis of Visually Evoked Potentials in the Cat Cortex. *International Journal of Neuroscience*, 33, 33-59.
- MOFFITT, M. A. & MCINTYRE, C. C. 2005. Model-based analysis of cortical recording with silicon microelectrodes. *Clinical Neurophysiology*, 116, 2240-2250.
- MORRIS, R. G. M. 1981. Spatial localization does not require the presence of local cues. *Learning and Motivation*, 12, 239-260.

- MORRIS, R. G. M., GARRUD, P., RAWLINS, J. N. P. & O'KEEFE, J. 1982. Place navigation impaired in rats with hippocampal lesions. *Nature*, 297, 681.
- MOSER, E. I., KROPFF, E. & MOSER, M.-B. 2008. Place Cells, Grid Cells, and the Brain's Spatial Representation System. *Annual Review of Neuroscience*, 31, 69-89.
- NELSON, P. G. & FRANK, K. 1964. EXTRACELLULAR POTENTIAL FIELDS OF SINGLE SPINAL MOTONEURONS. *Journal of Neurophysiology*, 27, 913-927.
- NICHOLSON, C. & FREEMAN, J. A. 1975. Theory of current source-density analysis and determination of conductivity tensor for anuran cerebellum. *Journal of Neurophysiology*, 38, 356-368.
- O'KEEFE, J. & CONWAY, D. H. 1978. Hippocampal place units in the freely moving rat: Why they fire where they fire. *Experimental Brain Research*, 31, 573-590.
- O'KEEFE, J. & DOSTROVSKY, J. 1971. The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34, 171-175.
- O'KEEFE, J. & NADEL, L. 1978. *The hippocampus as a cognitive map*, Oxford University Press.
- O'KEEFE, J. & RECCE, M. L. 1993. Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus*, 3, 317-330.
- OBIEN, M. E. J., DELIGKARIS, K., BULLMANN, T., BAKKUM, D. J. & FREY, U. 2015. Revealing neuronal function through microelectrode array recordings. *Frontiers in Neuroscience*, 8.
- PACHITARIU, M., STEINMETZ, N., KADIR, S., CARANDINI, M. & HARRIS, K. D. 2016. Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *bioRxiv*.
- PAGE, H. J. I., WILSON, J. J. & JEFFERY, K. J. 2018. A dual-axis rotation rule for updating the head direction cell reference frame during movement in three dimensions. *J Neurophysiol*, 119, 192-208.
- PALM, G. 2013. Neural associative memories and sparse coding. *Neural Networks*, 37, 165-171.
- PETTERSEN, K. H., DEVOR, A., ULBERT, I., DALE, A. M. & EINEVOLL, G. T. 2006. Current-source density estimation based on inversion of electrostatic forward solution: Effects of finite extent of neuronal activity and conductivity discontinuities. *Journal of Neuroscience Methods*, 154, 116-133.
- POUCET, B. 1993. Spatial cognitive maps in animals: new hypotheses on their structure and neural mechanisms. *Psychol Rev*, 100, 163-82.
- POUZAT, C. & CHAFFIOL, A. 2009. Automatic spike train analysis and report generation. An implementation with R, R2HTML and STAR. *J Neurosci Methods*, 181, 119-44.
- POUZAT, C., DELESCLUSE, M., VIOT, P. & DIEBOLT, J. 2004. Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: a Markov chain Monte Carlo approach. *J Neurophysiol*, 91, 2910-28.
- QUIAN QUIROGA, R. 2012. Spike sorting. *Curr Biol*, 22, R45-6.
- QUIROGA, R. Q., NADASDY, Z. & BEN-SHAUL, Y. 2004. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput*, 16, 1661-87.
- RALL, W. 1962. Electrophysiology of a Dendritic Neuron Model. *Biophysical Journal*, 2, 145-167.
- RANCK, J. B. 1966. Electrical impedance in the subicular area of rats during paradoxical sleep. *Experimental Neurology*, 16, 416-437.
- REDISH, A. D. 2014. *MClust free-ware spike sorting* [Online]. Available: <http://redishlab.neuroscience.umn.edu/MClust/MClust.html> [Accessed].
- REY, H. G., PEDREIRA, C. & QUIAN QUIROGA, R. 2015a. Past, present and future of spike sorting techniques. *Brain Research Bulletin*, 119, Part B, 106-117.
- REY, H. G., PEDREIRA, C. & QUIAN QUIROGA, R. 2015b. Past, present and future of spike sorting techniques. *Brain Res Bull*, 119, 106-17.
- RODRIGUEZ, A. & LAIO, A. 2014. Clustering by fast search and find of density peaks. *Science*, 344, 1492-1496.
- ROLLS, E. T. & O'MARA, S. M. 1995. View-responsive neurons in the primate hippocampal complex. *Hippocampus*, 5, 409-424.
- ROSSANT, C., KADIR, S. N., GOODMAN, D. F. M., SCHULMAN, J., BELLUSCIO, M., BUZSAKI, G. & HARRIS, K. D. 2015. Spike sorting for large, dense electrode arrays. *bioRxiv*.

- RUTISHAUSER, U., SCHUMAN, E. M. & MAMELAK, A. N. 2006. Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo. *J Neurosci Methods*, 154, 204-24.
- SARGOLINI, F., FYHN, M., HAFTING, T., MCNAUGHTON, B. L., WITTER, M. P., MOSER, M. B. & MOSER, E. I. 2006. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312, 758-62.
- SCHWARZ, D. A., LEBEDEV, M. A., HANSON, T. L., DIMITROV, D. F., LEHEW, G., MELOY, J., RAJANGAM, S., SUBRAMANIAN, V., IFFT, P. J., LI, Z., RAMAKRISHNAN, A., TATE, A., ZHUANG, K. Z. & NICOLELIS, M. A. L. 2014. Chronic, wireless recordings of large-scale brain activity in freely moving rhesus monkeys. *Nat Meth*, 11, 670-676.
- SCOVILLE, W. B. & MILNER, B. 1957. Loss of recent memory after bilateral hippocampal lesions. *J Neurol Neurosurg Psychiatry*, 20, 11-21.
- SHARP, P., BLAIR, H., ETKIN, D. & TZANETOS, D. 1995. Influences of vestibular and visual motion information on the spatial firing patterns of hippocampal place cells. *The Journal of Neuroscience*, 15, 173-189.
- SHIN, J. & TALNOV, A. 2001. A single trial analysis of hippocampal theta frequency during nonsteady wheel running in rats. *Brain Research*, 897, 217-221.
- SHOHAM, S., FELLOWS, M. R. & NORMANN, R. A. 2003. Robust, automatic spike sorting using mixtures of multivariate t-distributions. *Journal of Neuroscience Methods*, 127, 111-122.
- SPACEK, M., BLANCHE, T. & SWINDALE, N. 2008. Python for large-scale electrophysiology. *Front Neuroinform*, 2, 9.
- STACKMAN, R. W. & TAUBE, J. S. 1998. Firing Properties of Rat Lateral Mammillary Single Units: Head Direction, Head Pitch, and Angular Head Velocity. *The Journal of Neuroscience*, 18, 9020-9037.
- STEIN, R. B., GOSSEN, E. R. & JONES, K. E. 2005. Neuronal variability: noise or part of the signal? *Nature Reviews Neuroscience*, 6, 389.
- STEVENS, C. F. & ZADOR, A. 1995. Neural Coding: The enigma of the brain. *Current Biology*, 5, 1370-1371.
- STEVENSON, I. H. & KORDING, K. P. 2011. How advances in neural recording affect data analysis. *Nat Neurosci*, 14, 139-42.
- STONE, J. 1973. Sampling properties of microelectrodes assessed in the cat's retina. *Journal of Neurophysiology*, 36, 1071-1079.
- STRUMWASSER, F. 1958. Long-Term Recording from Single Neurons in Brain of Unrestrained Mammals. *Science*, 127, 469-470.
- SWINDALE, N. V. & SPACEK, M. A. 2014. Spike sorting for polytrodes: a divide and conquer approach. *Front Syst Neurosci*, 8, 6.
- TAKEKAWA, T., ISOMURA, Y. & FUKAI, T. 2012. Spike sorting of heterogeneous neuron types by multimodality-weighted PCA and explicit robust variational Bayes. *Front Neuroinform*, 6, 5.
- TAUBE, J. S. 2007. The head direction signal: origins and sensory-motor integration. *Annu Rev Neurosci*, 30, 181-207.
- TAUBE, J. S., MULLER, R. U. & RANCK, J. B., JR. 1990. Head-direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations. *J Neurosci*, 10, 436-47.
- TOLMAN, E. C. 1948. Cognitive maps in rats and men. *Psychological Review*, 55, 189-208.
- TOWE, A. L. & HARDING, G. W. 1970. Extracellular microelectrode sampling bias. *Experimental Neurology*, 29, 366-381.
- TRULLIER, O., WIENER SI FAU - BERTHOZ, A., BERTHOZ A FAU - MEYER, J. A. & MEYER, J. A. 1997. Biologically based artificial navigation systems: review and prospects.
- VALERIO, S. & TAUBE, J. S. 2012. Path integration: how the head direction signal maintains and corrects spatial orientation. *Nat Neurosci*, 15, 1445-53.
- VANDERWOLF, C. H. 1969. Hippocampal electrical activity and voluntary movement in the rat. *Electroencephalography and Clinical Neurophysiology*, 26, 407-418.

- VINJE, W. E. & GALLANT, J. L. 2000. Sparse Coding and Decorrelation in Primary Visual Cortex During Natural Vision. *Science*, 287, 1273-1276.
- VOGELSTEIN, R. J., MURARI, K., THAKUR, P. H., DIEHL, C., CHAKRABARTTY, S. & CAUWENBERGHS, G. Spike sorting with support vector machines. The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1-5 Sept. 2004 2004. 546-549.
- WANG, R. F., CROWELL, J. A., SIMONS, D. J., IRWIN, D. E., KRAMER, A. F., AMBINDER, M. S., THOMAS, L. E., GOSNEY, J. L., LEVINTHAL, B. R. & HSIEH, B. B. 2006. Spatial updating relies on an egocentric representation of space: Effects of the number of objects. *Psychonomic Bulletin & Review*, 13, 281-286.
- WEIBLE, A. P., ROWLAND, D. C., MONAGHAN, C. K., WOLFGANG, N. T. & KENTROS, C. G. 2012. Neural Correlates of Long-Term Object Memory in the Mouse Anterior Cingulate Cortex. *The Journal of Neuroscience*, 32, 5598-5608.
- WEIBLE, A. P., ROWLAND, D. C., PANG, R. & KENTROS, C. 2009. Neural Correlates of Novel Object and Novel Location Recognition Behavior in the Mouse Anterior Cingulate Cortex. *Journal of Neurophysiology*, 102, 2055-2068.
- WILLS, T. J., CACUCCI, F., BURGESS, N. & O'KEEFE, J. 2010. Development of the Hippocampal Cognitive Map in Prewearling Rats. *Science*, 328, 1573-1576.
- WILSON, J. J., HARDING, E., FORTIER, M., JAMES, B., DONNETT, M., KERSLAKE, A., O'LEARY, A., ZHANG, N. & JEFFERY, K. 2015. Spatial learning by mice in three dimensions. *Behav Brain Res*, 289, 125-32.
- WILSON, M. & MCNAUGHTON, B. 1993. Dynamics of the hippocampal ensemble code for space. *Science*, 261, 1055-1058.
- WOOD, F. & BLACK, M. J. 2008. A nonparametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, 173, 1-12.
- WOOD, J. 2014. A sparse memory is a precise memory. *Oxfrod Science Blog* [Online]. Available from: <http://www.ox.ac.uk/news/science-blog/sparse-memory-precise-memory> [Accessed 28 Feb.
- YARTSEV, M. M. & ULANOVSKY, N. 2013. Representation of Three-Dimensional Space in the Hippocampus of Flying Bats. *Science*, 340, 367-372.
- YGER, P., SPAMPINATO, G. L., ESPOSITO, E., LEFEBVRE, B., DENY, S., GARDELLA, C., STIMBERG, M., JETTER, F., ZECK, G., PICAUD, S., DUEBEL, J. & MARRE, O. 2018. A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. *Elife*, 7.
- YODER, R. M. & TAUBE, J. S. 2014. The vestibular contribution to the head direction signal and navigation. *Front Integr Neurosci*, 8, 32.
- ZUGARO, M. B., TABUCHI, E. & WIENER, S. I. 2000. Influence of conflicting visual, inertial and substratal cues on head direction cell activity. *Exp Brain Res*, 133, 198-208.

3 NeuroChaT

To achieve the objectives of the project systematically, we have developed a graphical user interface-based toolbox called ***Neuron Characterization Toolbox*** or NeuroChaT. In this chapter, we described the design of the toolbox and the application programming interface, the architecture and the layers of interaction between the user, data and analyses. We also explain, in groups, different types of analyses that are core to the NeuroChaT and explain the underlying algorithms as easily as possible considering that the targeted user group may span from biologists to expert computational neuroscientists.

3.1 Architecture

NeuroChaT is built on the concept of object-oriented programming (OOP). Interested users can read any standard programming book to get acquainted with the OOP. The architecture of the software is depicted in **Figure 3.1**. The user starts providing input through the graphical user interface (GUI) shown in **Figure 3.2**. The interface interacts with the user using the log-screen at the bottom of the GUI. At this stage (specification phase shown in **Figure 3.3**), user selects the data, specify analyses and parameters- collectively called configuration- required for the analyses which is saved in a custom Python object named Configuration. The details of how to specify the configuration through the interface will be shown in **Appendix A** and **Appendix B**. The information is then passed to a 'NeuroChaT' object which acts as the controller class or the façade that provides a unified interface to different components of the entire system. We will discuss the design pattern of the software in **Section 3.2**. NeuroChaT then connects to a data structure called NData which itself is composed of three different data types

- NSpike: The object of this class contains information about the spiking activity of the single-unit, the waveform and the clustering information that are derived from spike-sorting techniques. Apart from providing information to other data objects, it also contains some analyses which are described in **Section 3.3**.
- NSpatial: This object contains spatial information regarding animal position, head direction, speed, angular head velocity etc. and performs relevant analyses using the information from NSpike object which is passed through NData object.
- NLfp: It is a placeholder for LFP data and contains relevant analyses borrowing necessary information from other objects i.e. spiking information of a unit from the NSpike object.

NeuroChaT then passes the relevant parameters to the NData and asks to perform the analyses, on a cell-by-cell basis, based on the user input in the specification phase. This is the execution phase of the software (**Figure 3.3**). The analysis results are sent back to the NeuroChaT which then uses the nc_plot-a custom module or Python file-

to plot the graphical outcome of the analyses, stores the parametric results in a tabular format, and converts the data in a standardized format called the Hierarchical Data File format (HDF5) (www.hdfgroup.org). This is the visualization and export phase in **Figure 3.3**. The user can perform statistical analysis on the parametric results if required, and this step is the Inference phase of the data analysis workflow using NeuroChaT. The graphical output and the tabular listings follow a nomenclature for making the post-processing tracking easier. This will be discussed in **Section 3.4**. The specifications of the data, analyses, and input parameters can be saved for future use in *.ncfg* (NeuroChaT configuration) file format, which is basically a YAML (Ain't Markup Language) format, a human readable data-serialization format commonly used for configuration files (yaml.org).

The software enables managing graphical files through the GUI. Once the graphical output of the analyses is saved in PDF or PS file format, and the relevant information are provided to the user in a tabular form for record and future use, the user can use it to collect all the output graphics into one place so that the unit of similar properties can be grouped together or they can be merged to store them into one file. There are other utilities in the software like unit verification, cluster quality evaluation, conversion to the HDF5 file format from the proprietary formats etc. which uses other types of data objects namely NClust and Nhdf. NClust itself is a derived class of the NSpike class. These features of the software will be discussed in **Section 3.4** and **Chapter 4.5**.

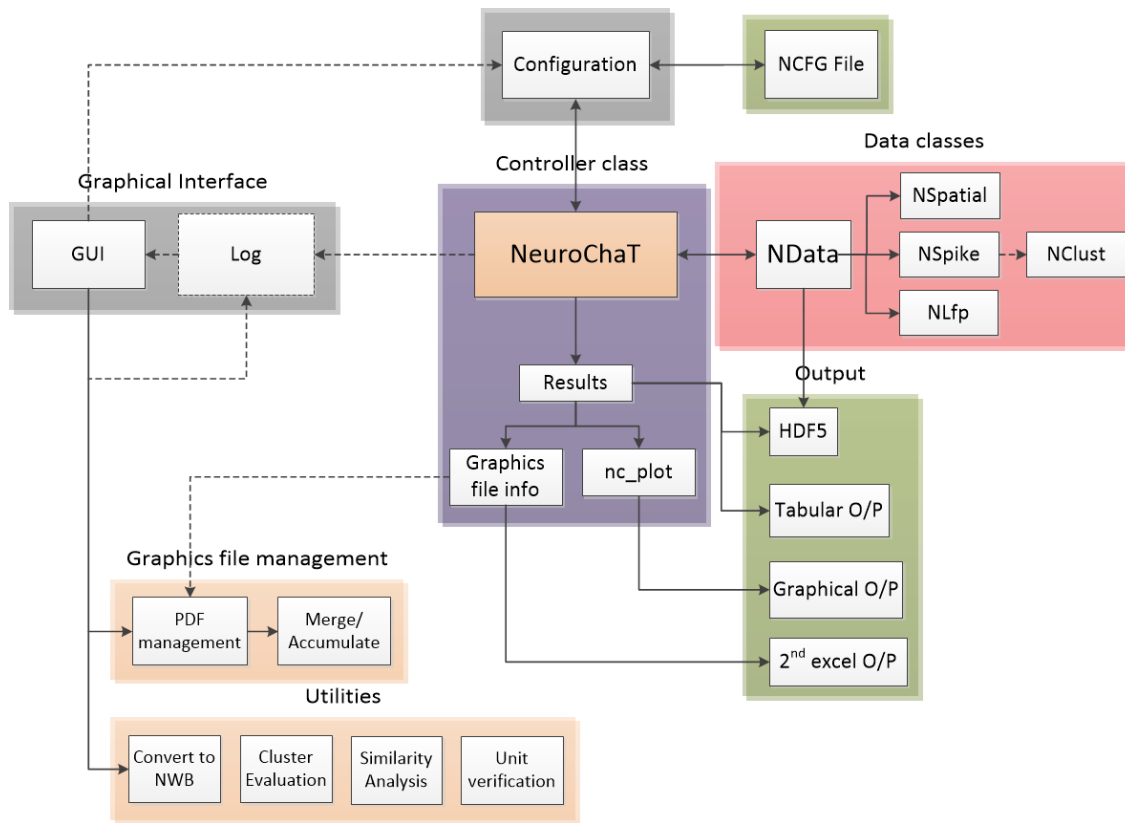


Figure 3.1 The architecture of NeuroChaT. It shows the connectivity of different components and the flow of information between them.

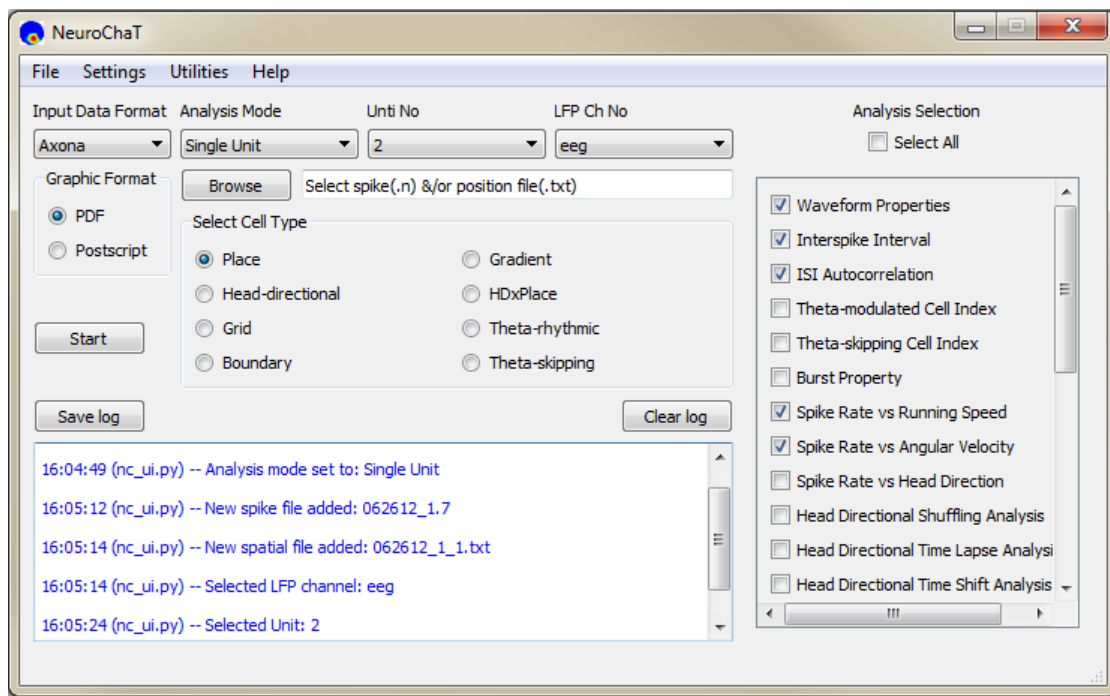


Figure 3.2 The graphical-user interface of NeuroChaT

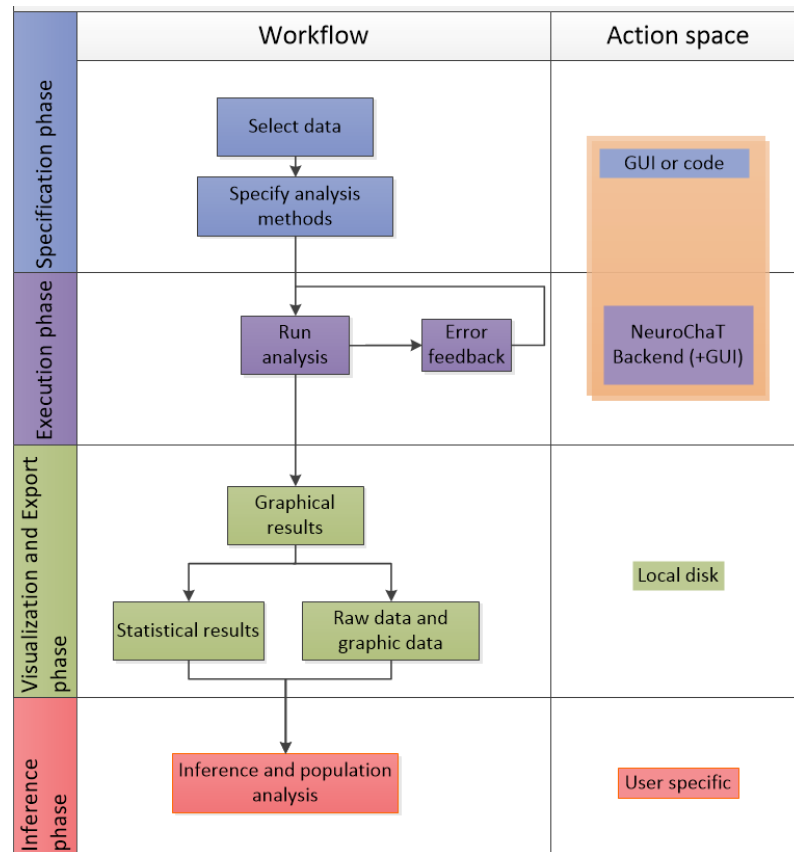


Figure 3.3 The workflow diagram in NeuroChaT and action space of different components of the program.

3.2 Design

N.B: The description of the methods and attributes for each class defined in NeuroChaT can be found in the API documentation section in <https://osf.io/kqz8b/>.

3.2.1 Technologies used

The software is developed using the latest version of Python 3.5. The GUI is developed using PyQt5, a cross-platform framework that enables Python to be used as an application development language by providing graphical elements and necessary functionalities.

The development is done using *NumPy* (from *Numeric Python*) package (www.scipy.org) which provides MATLAB®-like matrix manipulation facilities and is

widely used for scientific computing. The project also has use of *SciPy* (*Scientific Python*) library that provides linear algebra functionalities and mathematical optimizations i.e. curve fitting etc. NeuroChaT also uses *SciPy* for standard signal processing techniques like filtering, Fourier Transform etc. There is a brief use of *pandas* library for the storage of parametric results. The DataFrame object of pandas facilitates storing data in a tabular form with non-numeric indexing facilities, along with basic numeric operations across rows or columns of the tabular data. It also provides the functionalities to import/export (IO tools) the data to commonly used formats like CSV, Microsoft Excel or plain text files.

The manipulation of HDF5 binary file format through Nhdh object is performed using the h5py package (www.h5py.org) that enables manipulating the data using NumPy. The configuration data are exported or imported using PyYAML (pyyaml.org), which parses the text-editable *.ncfg* file and loads the information to the Configuration object.

We used matplotlib (matplotlib.org) for generating graphical output.

3.2.2 Data model and storage

The NData object, as shown in **Figure 3.4**, is composed of data objects of different kinds, and is built upon the composite structural object pattern (Gamma et al., 1995). The regular arrows show class inheritance and the diamond arrows show object composition. In NeuroChaT, NAbstract and NBase form the parent classes with basic and common methods and attributes across different data types. Each data class representing the neural data (NSpike, NClust, NLfp), along with the event class NEvent inherits the NBase, where NBase itself inherits NAbstract and extends its capabilities. The NSpatial class inherits the NAbstract class. The NData class gets one instance or object for each NSpatial, NSpike and NLfp class as its attribute. The rationale behind this design is to provide an encapsulation of the interaction among the behavioral and neural data types, i.e. how the peers like NSpatial and NSpike would know each other. Either they will need to have a reference to each other, which increases their coupling, or they need to be cooperated using another object, which,

in our design, is the NData object. The similar design principle is also followed in other composite classes of NeuroChaT. The getter (get_*) and setter (set_*) methods of the composite class instance then allow dynamically changing the objects or retrieving it. For example, the spatial data does not need to be changed for a single recording while analyzing for multiple single units recorded in the same session. Therefore, NSpatial object remains same, but the data in the NSpike object changes with changing the units. Now, creating one instance of NData for every pair of spatial and single unit data is not very memory efficient. Instead, we can replace the data in the NSpike by reloading the spike file while it is still a member of the NData object and optimize the reuse of data objects, save memory, and increase the performance of the software.

In another situation, if the NData class was formed using multiple inheritance from NSpatial, NSpike and NLfp data, it would create a tight coupling between the classes. If the users need to add a new analysis method, i.e. involving LFP and single-unit data, they can inherit the existing NLfp class, add the new analysis on it, create an instance of the new class, i.e., NLfp2 and set it to the NData class without much changes in the NData class. It would not be that straightforward for the inherited NData class, as its inheritance would need to be redefined with the new NLfp2 class, along with the NSpatial and NSpike class or their child classes (classes inheriting another class). Over the time, the coupling would be difficult to maintain as the data classes evolve to include new methods. Thus, the type of design pattern used in NeuroChaT creates a modular structure and allows the objects to alter dynamically without intense refactoring of the codes.

NSpike class contains analyses involving spiking activity of the single-units i.e. inter-spike interval autocorrelation histogram etc. along with implementing the decoders for the copyright data formats. NSpatial class contains methods for analyzing the spatial correlation of the single units and requires the information about the timing of the unit activity only which it receives either directly as an input to its methods (**Appendix B**, API use guide) or through NData. When used with the NData class, it receives the information through that class instead of coupling directly to the NSpike data. Therefore, it creates another layer of independence between the data classes which are highly susceptible to changes but requires less efforts to couple them and

to get the expected results. Therefore, it enables quick integration of problem-specific data analysis. This design also applies to the methods in NLfp class. It represents analyses that involves use of the LFP data, possibly along with the spiking information. NData also contains Nhdf data object to provide read/write access of HDF5 (hierarchical data file, version 5) data formats containing spatial or neural data within the class without trying to decode the native binary file formats every time the data is loaded. As the HDF5 file contains all the data, it makes storage more manageable through a readable format. Nhdf contains methods to read and write what is called groups and datasets in HDF5 file format. It also contains methods that are specific to storing individual NSpike, NLfp, and NSpatial data to their common HDF5 container for a recording session. NeuroChaT creates one such file for each recording session, not for individual units or electrodes as will be discussed in **Section 3.4**.

NClust class manages and implements the spike-sorting information and provides the waveform features, unit spiking activity, measures of cluster separation for quality assessment of spike-sorting and measuring the cluster similarity with a unit in another NClust object. It contains a NSpike object and adds on its functionalities that are very specific for the clustering and relevant analyses.

NEvent class implements event-related data management and basic analyses, i.e., peri-stimulus time histogram or psth, and analyses pertaining to locking of the LFP signals to the events. But there is no decoder for proprietary formats, and, therefore, can only be used when the event information is added in it.

The analyses are mainly implemented based on the exclusiveness of the data among the data classes, and the methods are written in a style that intrinsically minimizes the dependence on other data structures in the software. Therefore, whenever applicable, the spike-timing of the unit under consideration is given as an input to an analysis implementation involving spatial and LFP data classes instead of passing the entire NSpike object as its input parameter. This is because, the way we developed the algorithms, as explained in **Section 3.3**, the waveform information, for example, or the recording parameters are not useful to conduct the analyses.

We cross-validated the relationship between classes or objects using Pyreverse, a python utility for reverse-engineering the Python code (<https://pypi.org/project/pyreverse/>). Each designed class diagram is accompanied by a code-generated class diagram- the later also revealing the composing objects in the code.

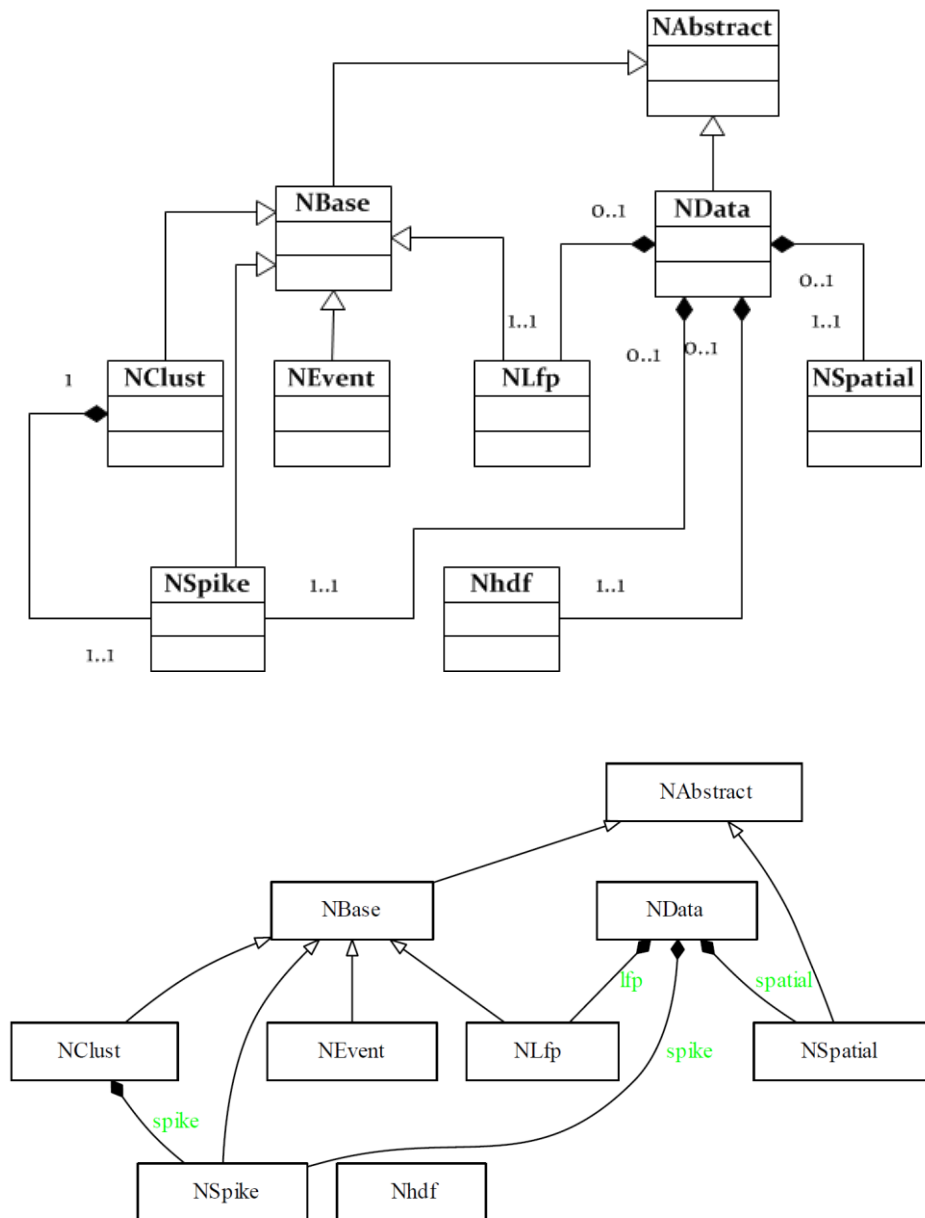
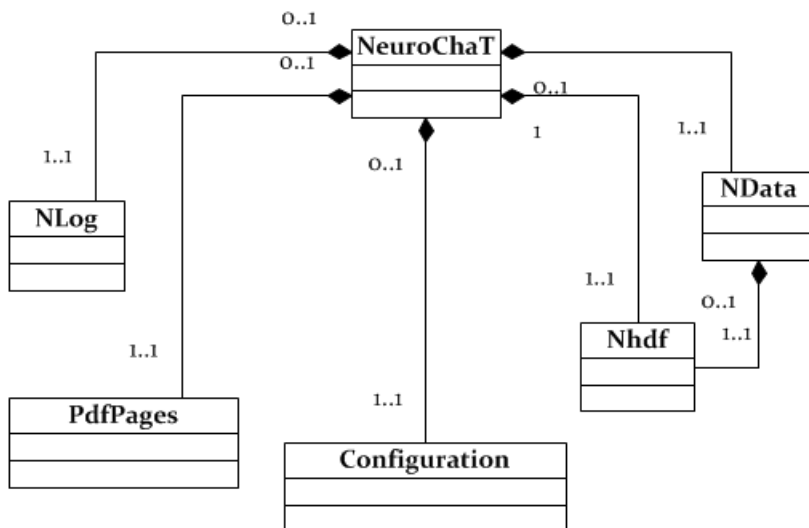


Figure 3.4 Class diagram for the data classes in NeuroChaT. The regular arrows show the inheritance and the diamond arrows show the composition. Inheritance or composition are one-to-one in all the cases. Top diagram shows the multiplicity (numbers) of association, and the bottom diagram shows the composite objects (green texts). The latter is generated from the code using the pyreverse package.

3.2.3 Controller

NeuroChaT, the controller class in NeuroChaT software, is the top layer application programming interface and provides encapsulation of the configuration object, data object, object for creating pdf or ps file from matplotlib graphics using the PdfPages class, HDF5 file object, and the logger object. We used NLog for logging which is a wrapper of the native 'logging' class to create formatted logs of the events or statements. The relevant class diagram is shown in **Figure 3.5**. The design of the NeuroChaT class is what is formally called façade design pattern, because an instance of this object represents an entire subsystem. It carries out responsibilities and exchanges information between the data class and the configuration class, which describes the analyses to perform, information about the data, and the input parameters. It can be used standalone (see **Appendix B**), but when used with the GUI, it is the entry point to the backend analytic platform. All the commands, and configurations that are requested through the GUI is forwarded through this class to its components. It also deciphers the configuration and updates the NData object to perform the analyses on a unit by unit basis.



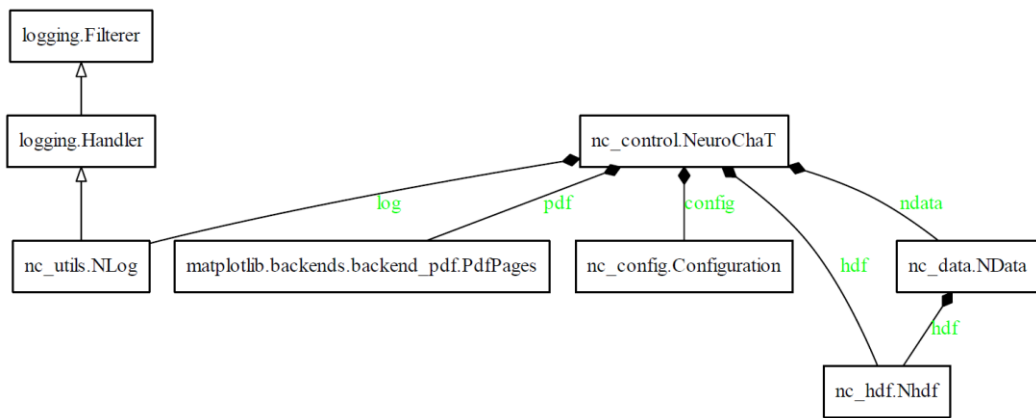


Figure 3.5 Class diagram of the NeuroChaT class, the controller of the NeuroChaT software. Top diagram shows the multiplicity of association, and the bottom diagram shows the composite objects (green texts). The latter is generated from the code using pyreverse package. The name precedes the dot of each classes are the name of the Python modules where they reside.

3.2.4 User interface

NeuroChaT_Ui is the graphical element of the NeuroChaT software that provides an interface for the users to specify the type of analyses they want to perform, to select the data for that analyses, to provide the input for those analyses, and the type of graphical file format they want to store their results in. As it is explained in **Appendix A**, this is a simple-to-use interface with lots of features that enables setting the commands and information to be forwarded to the NeuroChaT object. Its composing objects are all graphical elements except the NeuroChaT object (**Figure 3.6**). NeuroChaT API documentation (<https://osf.io/kqz8b/>) describes each of these classes. Although built in a composite structure, this class is static, in the sense that its components cannot be altered dynamically using commands outside of the class itself. Therefore, the coupling between these classes to others is considered tight, and any changes required must involve changing within the code file where the class is defined.

NLogBox is an editable graphical widget that is sub-classed from QTextEdit of the QtWidgets of PyQt5 to format the logged messages into html format. ParamBoxLayout is inherited from QVBoxLayout which is used for arranging the parameter definition in a vertical layout in the Settings menu of the interface (**Appendix A**). ScrollableWidget provides a container of listed items so that the user

can scroll through the items if the list takes more space than the widget, they are located in. UiParameters define and add the graphical elements to the interface. NOut replaces the standard output texts of Python or IPython (**print** command) into texts that are received by the logger of the system. UiResults is a sub-class of QDialog of QtWidget that displays the results of the analysis in a tabular format along with an option to export them in an Excel file (**Appendix A**). UiMerge is a graphical window that asks the user to select a list of pdf filename to either merge them into one file or to transfer them to one folder. The user can also select the pdf files manually using an interactive window (**Appendix A**) that is built in UiGetFiles class (**Figure 3.6**).

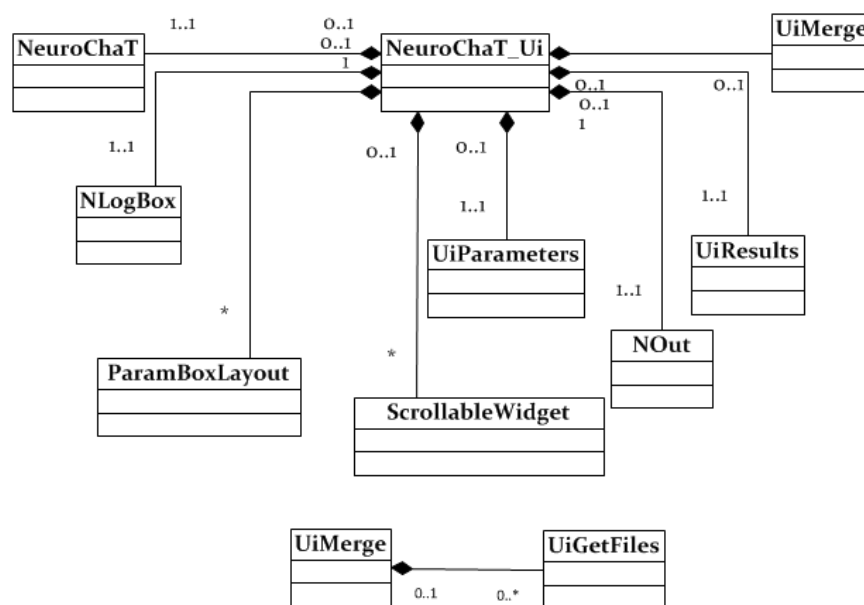


Figure 3.6 Class diagram for the NeuroChaT_Ui class that defines the graphical user interface of the software.

3.2.5 Configuration

Configuration class is basically a container for all the information that are required for NeuroChaT to execute the analyses, or equivalently the information that are set through the GUI. It includes the type of analyses user is interested to perform, definition of the data, parameters that are required by each analysis type etc. It also contains methods that are required for accessing or setting the information. It has

methods to export the information into an YAML file with `ncfg` extension, and to import from the same using the PyYAML parser.

3.2.6 Utilities

There are other classes and modules those are required at different stages of the software. They assist other classes or contain static functions. `CircStat` class encapsulates circular data and provides methods to calculate circular statistics i.e., mean, standard deviation, Rayleigh statistics, von Mises distribution parameters, circular histogram, circular smoothing filter etc. The `nc_utils` module contains static utility functions that are not part of any classes or can be used by different data classes. The `nc_plot` module is another utility module that creates plots for different analyses used in the NeuroChaT. The plotting functions in this module can be considered as thin wrappers on the matplotlib plotting functions. The `nc_defaults` module defines the default analyses parameters in name-value pairs for individual analyses type.

3.2.7 Organizing the codes

Python uses modules to organize the codes. A module is a file that contains the definitions and statements. A module is named by the file name and can contain the definition of classes and functions. The codes in NeuroChaT are organized in 18 modules. **Table 3.1** shows which module the classes belong.

Table 3.1 List of classes in NeuroChaT software and the modules they belong to.

Class names	Module name
NAbstract, NBase	<code>nc_base.py</code>
CircStat	<code>nc_circular.py</code>
NClust	<code>nc_clust.py</code>
Configuration	<code>nc_config.py</code>

NeuroChaT	nc_control.py
NData	nc_data.py
NEvent	nc_event.py
Nhdf	nc_hdf.py
NLfp	nc_lfp.py
NSpatial	nc_spatial.py
NSpike	nc_spike.py
NeuroChaT_Ui, UiResults, UiParameters, ParamBoxLayout	nc_ui.py
UiGetFiles	nc_uigetfiles.py
UiGetMerge,	nc_uimerge.py
ScrollableWidget, NLogBox, NOut, PandasModel	nc_uiutils.py

3.3 Analysis methods and algorithms

There are 25 different types of analysis already developed in NeuroChaT. The types of analysis are summarized in **Figure 3.7** and a brief description of these analyses are provided in **Table 3.2**. Rest of the subsections describe the algorithms.

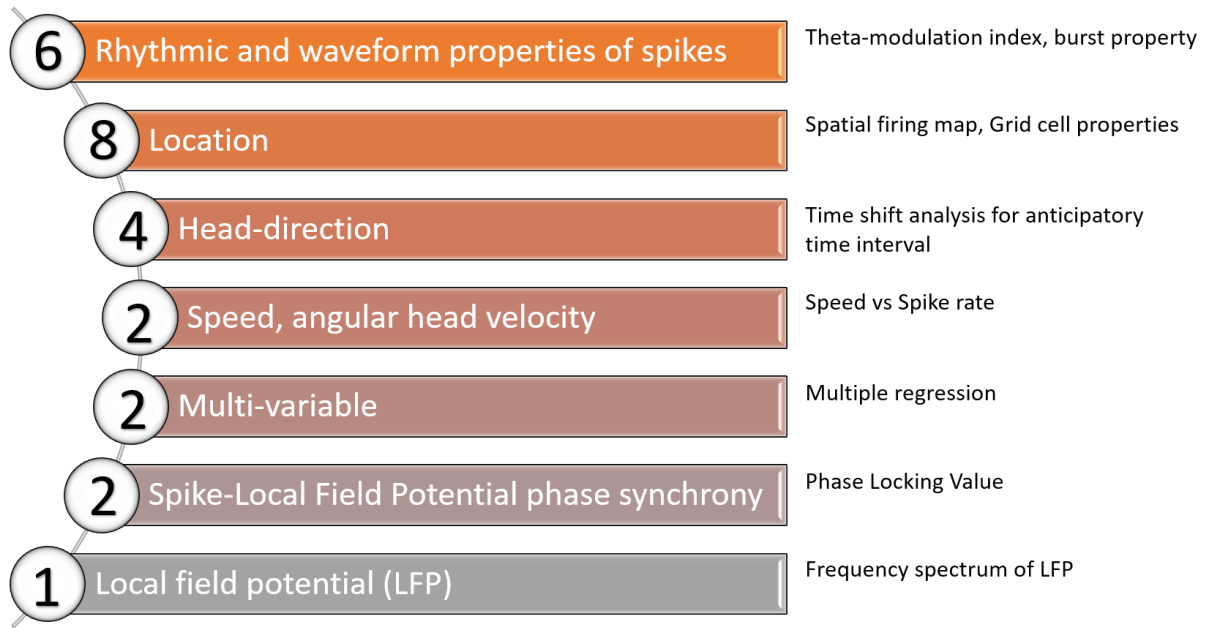


Figure 3.7 Category of analyses in NeuroChaT and relevant count of the methods

Table 3.2 Summary of the analyses developed in NeuroChaT; *hd*= head direction of the animal, *border*= distance from the border of the foraging environment.

Name of function	Category of analysis	Input data file	Category of information required
Waveform properties	Spike waveform	Spike	Spike waveform
Inter-spike Interval	Spike firing	Spike	Spike time
ISI Autocorrelation	Spike firing	Spike	Spike time
Theta-modulated Cell Index	Spike firing	Spike	Spike time
Theta-skipping Cell Index	Spike firing	Spike	Spike time
Burst Property	Spike firing	Spike	Spike time
Spike Rate vs Running Speed	Speed	Spike, spatial	Spike time, speed
Spike Rate vs Angular Velocity	Angular velocity	Spike, spatial	Spike time, hd, angular velocity,
Spike Rate vs Head Direction	Head directional	Spike, spatial	Spike time, hd, angular velocity
Head Directional Shuffling Analysis	Head directional	Spike, spatial	Spike time, hd
Head Directional Time Lapse Analysis	Head directional	Spike, spatial	Spike time, hd
Head Directional Time Shift Analysis	Head directional	Spike, spatial	Spike time, hd

Spike Rate vs Location	Locational	Spike, spatial	Spike time, position, angular velocity, hd
Locational Shuffling Analysis	Locational	Spike, spatial	Spike time, Position
Locational Time Lapse Analysis	Locational	Spike, spatial	Spike time, position
Locational Time Shift Analysis	Locational	Spike, spatial	Spike time, position
Spatial Autocorrelation	Locational	Spike, spatial	Spike time, position
Grid Cell Analysis	Locational	Spike, spatial	Spike time, position
Border Cell Analysis	Locational	Spike, spatial	Spike time, border
Gradient Cell Analysis	Locational	Spike, spatial	Spike time, border
Multiple regression	Multivariable	Spike, spatial	Spike time, position, border, direction, speed, angular velocity
Interdependence Analysis	Multivariable	Spike, spatial	Spike time, border, position, direction, speed, angular velocity
LFP frequency Spectrum	LFP	LFP	LFP
Unit LFP-phase Distribution	Phase synchrony	Spike, LFP	Spike time, LFP

Unit	LFP-phase	Phase synchrony	Spike, LFP	Spike time, LFP
Locking				

3.3.1 Rhythmic and waveform properties of spikes

Waveform Properties

It provides graphical output of mean±std waveform for the unit (**Figure 3.8**). The graphical representation provides further information on the particular electrode where the unit was detected and the variability of the waveform, as an alternative indication of the signal-to-noise ratio, measured by the standard deviation of the waveform from the mean. This function also calculates the summary statistics of basic spike features i.e. mean & standard deviation of spike width, amplitude, and height in the channel where the spike-waveforms have the largest peaks.

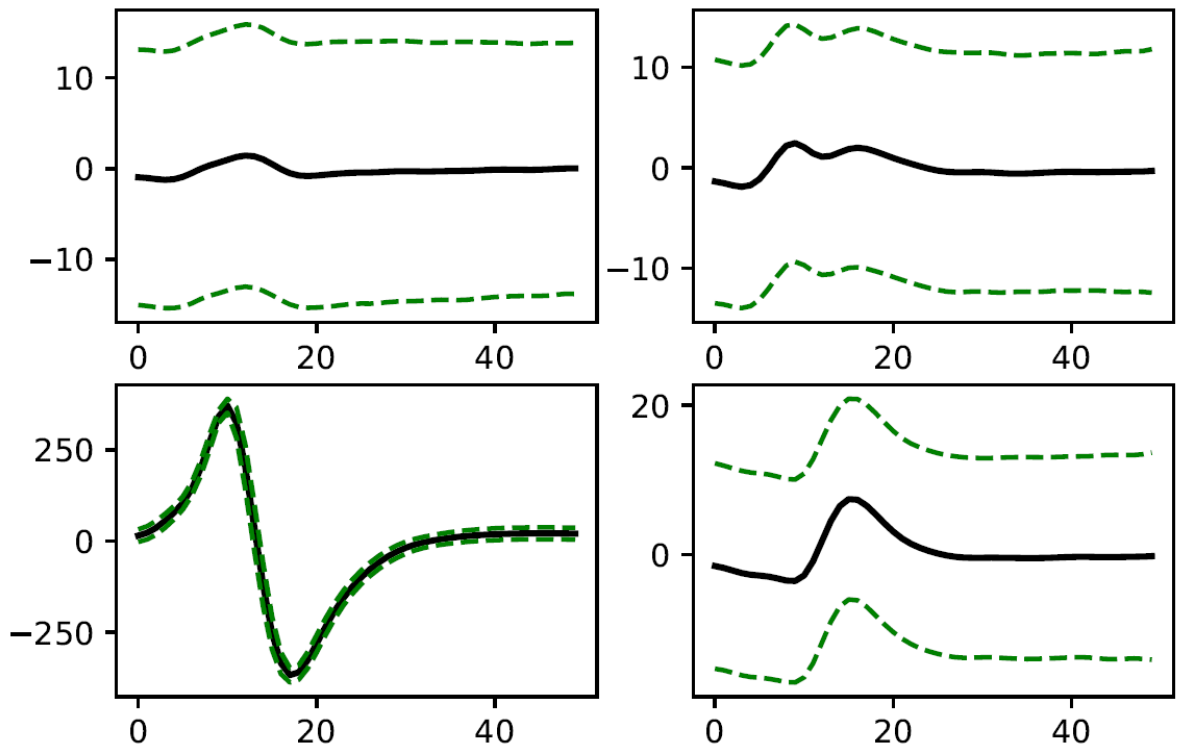


Figure 3.8 The mean \pm std waveform of a unit isolated in a tetrode configuration from data recorded in rat anterior thalamus. Channel numbers are, 1= upper left, 2= upper-right, 3= lower-left, 4= lower-right. It shows that the variability is much less in channel 3 where the waveform is much stronger- indicating that a tiny small proportion of noise-spikes (spikes not belonging to this unit) were included during the spike-sorting. As expected, the variability is much higher in other channels.

Definition of the features are as follows and is depicted in **Figure 3.9**:

Spike width: Width of the spike waveform at the 25% of the peak amplitude (**Figure 3.9**; red solid line).

Amplitude: The amplitude of the spike is the first negative peak of the spike to the first positive peak of the spike. If there is no negative peak before the positive peak, the amplitude refers to the peak voltage.

Height: Height is defined as the difference between the peak amplitude and the minimum waveform voltage of the spikes.

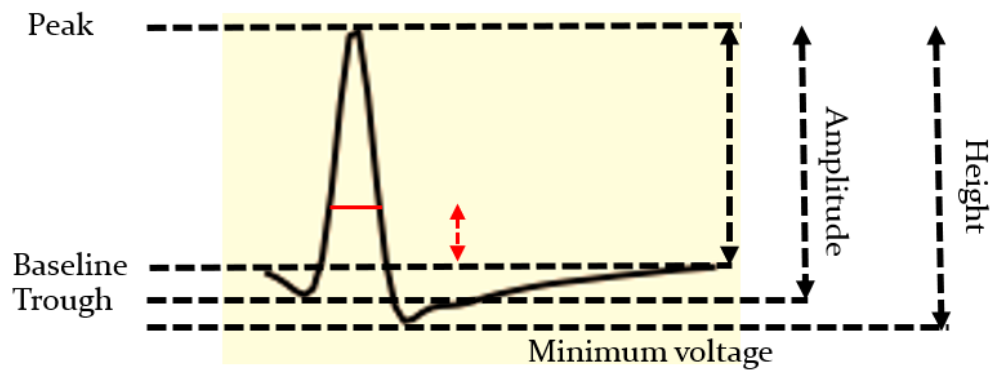


Figure 3.9 Depiction of how different waveform properties are measured. The red solid line is the width of the waveform which is measured as the 25% of the peak voltage of the waveform as shown in red dotted line.

Inter-spike Interval

This function gives the distribution of inter-spike-interval (ISI) of spikes of a unit. ISI is calculated by taking the time difference between spikes of particular unit. A histogram is then calculated as shown in **Figure 3.10**. We do also present an alternative visualization of the ISI where the interval before a spiking event is plotted against the interval after a spike of a unit (**Figure 3.11**). This analysis, along with the IS autocorrelation histogram described later, is very useful for observing bursting and/or rhythmic properties of the spikes, example of which will be given in the following sections.

Distribution of inter-spike interval

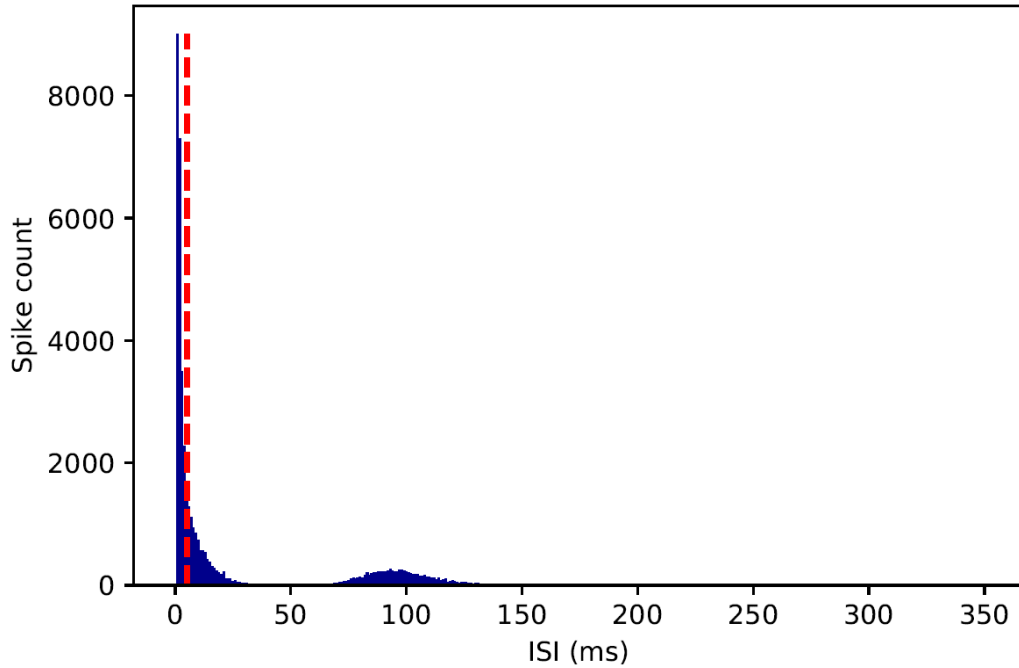


Figure 3.10 Distribution of inter-spike interval (ISI); x-axis gives the ISI bins, and y-axis gives the count of ISI for a particular bin. The red-dotted line is the user-specified threshold for the bursting spikes. This unit shows a large count under the bursting threshold time as well as a higher count at approximately 100ms (representing rhythmicity at around 10Hz). So, this is both a bursting unit with a higher burst propensity and theta-rhythmicity (these concepts will be explained in the following sections)

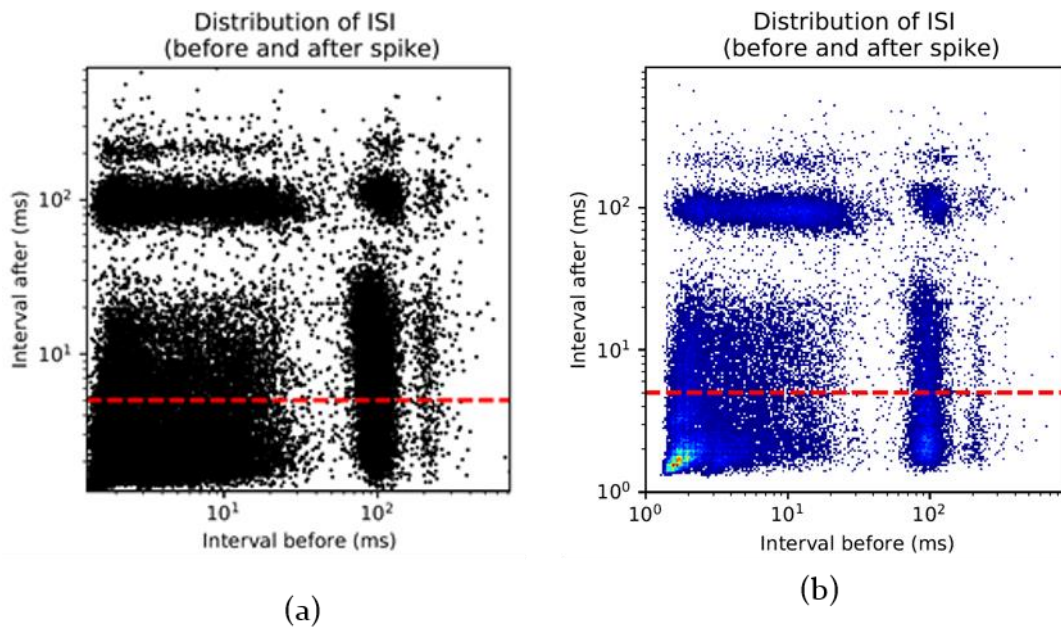


Figure 3.11 The plot of ISI after vs ISI before; a) scatter plot, b) two-dimensional histogram. The red dotted line is the threshold for burst. The colored region displays a high density of spikes at intervals lower than the burst threshold- implying that this is a bursting unit. The horizontal and vertical strip at (a) and (b) represents a repetition of ISI at around 100ms- implying that the unit is rhythmic at around 10Hz.

Autocorrelation of inter-spike interval

The idea of cross-correlation is depicted in **Figure 3.12**. One of the units is considered as the reference and the other one as target. It is based on the concept of peri-stimulus time histogram (PSTH), a very popular rate-coding technique in event-related single unit analysis (Shinomoto, 2010). But instead of the event, the spike-timing of the unit is used for making spike count around it at binned intervals. All such histogram count for each spiking-event is then summed up to obtain the cross-correlation. The autocorrelation histogram is basically a cross-correlation of a unit to itself. A sample autocorrelation histogram is shown in **Figure 3.13**.

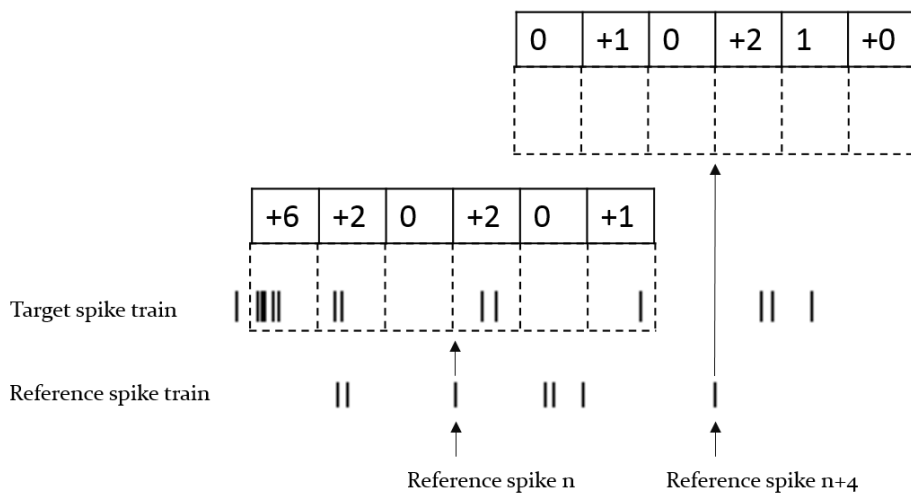


Figure 3.12 Algorithm for calculating the correlation between inter-spike interval of two spiking trains. One of them act as reference, and the other as target- as that of in peri-stimulus histogram. The spiking events in the reference train can be compared to the events and the histogram with respect to each spike is calculated on the target spike train. The summation of all such histogram results in the ISI correlation. If the target and reference spike trains are from the same unit, this is called auto-correlation histogram.

The autocorrelation histogram is powerful in exhibiting the rhythmicity in the spiking activity of the neurons. Sometimes, the rhythmicity is not stronger enough or hidden under the scaling of high counts in low ISI, as shown in the inset of **Figure 3.13**, but is clearly revealed in the autocorrelation histogram. It is also useful in assessing the quality of spike-sorting while evaluated for a small autocorrelation lag (**Figure 3.14**). It is expected to have a zero-count for $t < T_r$, where t = histogram bins in ms and T_r = Refractory period of the units.

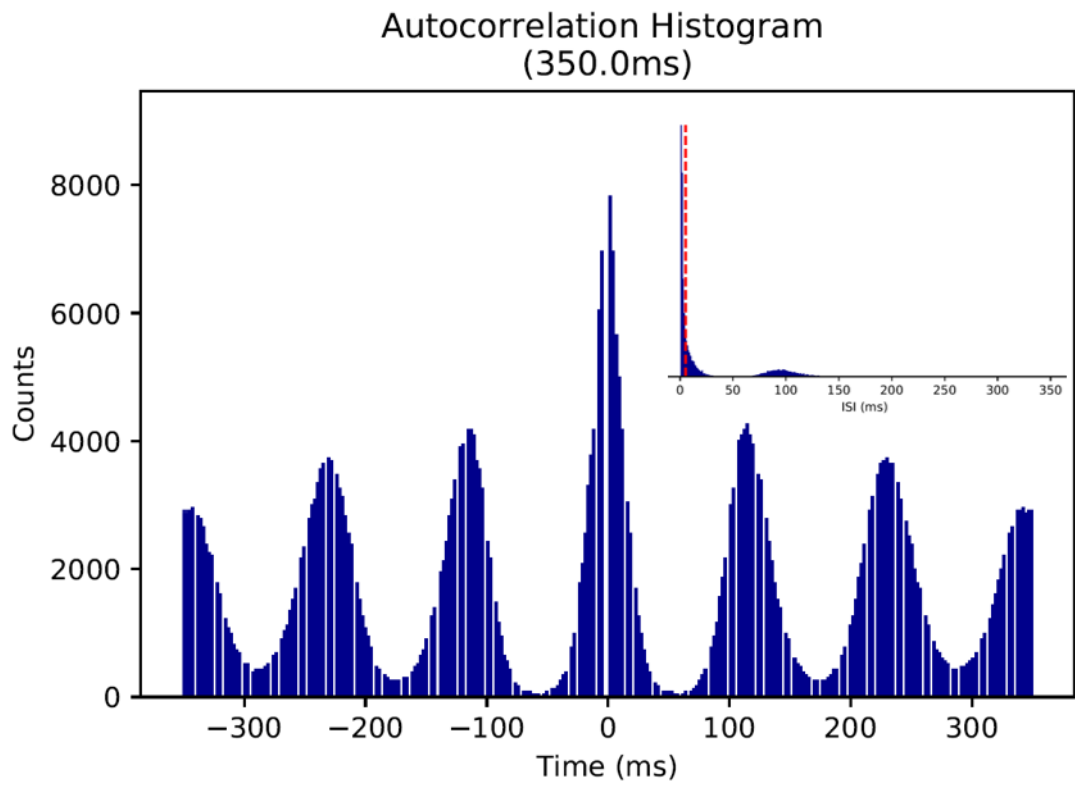


Figure 3.13 The autocorrelation histogram of the ISI of a unit. The peaks-trough rhythmic pattern is more apparent than the ISI histogram (inset) of the unit.

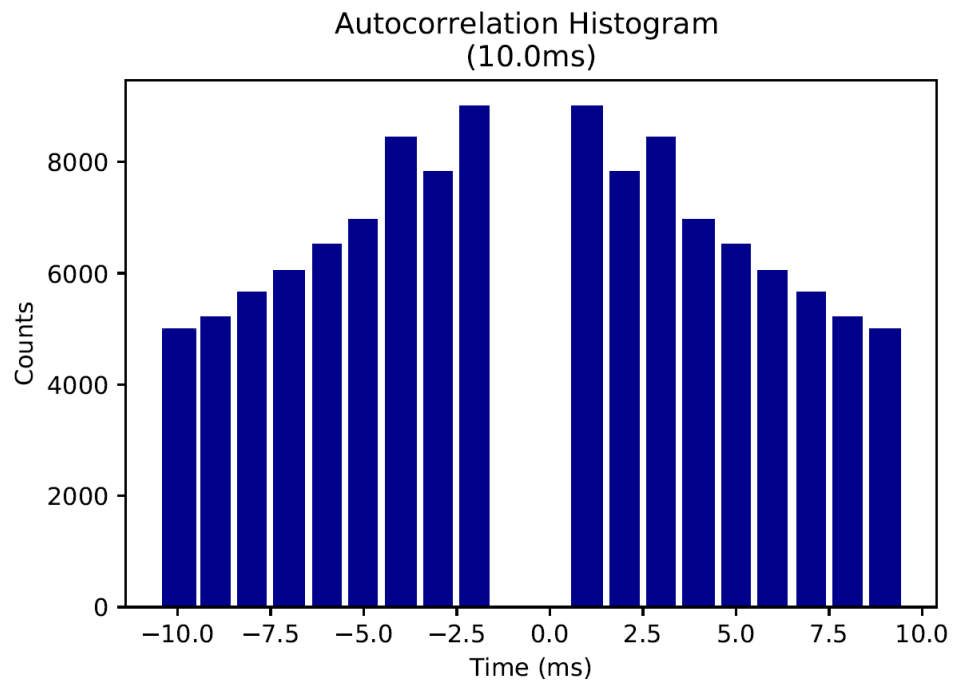


Figure 3.14 The ISI autocorrelation histogram for short lags. This particular example reveals that there is no count of spikes within 2ms interval, which is an indication that, if we consider a 2ms refractory period for the unit, spikes from other units are not incorporated into this unit. This is an alternative assessment of quality of spike sorting.

Analysis of theta-modulated cell

Sometimes the temporal discharge pattern in the neuronal spikes follow a rhythmicity that is revealed as a repeated count of the inter-spike interval. Such frequency of rhythmicity normally lies within the theta-band frequency (6-12Hz) (Tsanov et al., 2011a) and is clearly revealed in the autocorrelogram of the ISI (**Figure 3.13**). This type of units is called theta-modulated cells and are characterized by the strength of theta-modulation which is expressed in terms of ‘Theta-Index’. The non-normalized autocorrelogram is fitted to the **Equation 3.1**,

$$y(x) = a \cdot \cos(\omega x) \cdot \exp\left(-\frac{|x|}{\tau_1}\right) + b + c \cdot \exp\left(-\frac{x^2}{\tau_2^2}\right) \quad (3.1)$$

where y is the autocorrelation signal, x is the lag, a is the amplitude of the oscillating terms with frequency ω and modulated by exponential with decay constant τ_1 assuming that spikes were generated following a Poisson Process. τ_2 is the decay constants for the exponentials with amplitude c . The Gaussian term $c \cdot \exp(-x^2/\tau_2^2)$ was used to help fit the center peak of the autocorrelogram. The baseline shifts for the cosine terms and all the constant errors are interpreted in constant b . The theta index is defined as the ratio of the fit parameters a/b (Royer et al., 2010, Tsanov et al., 2011b). The fitted curve for the unit in **Figure 3.13** is shown **Figure 3.14**.

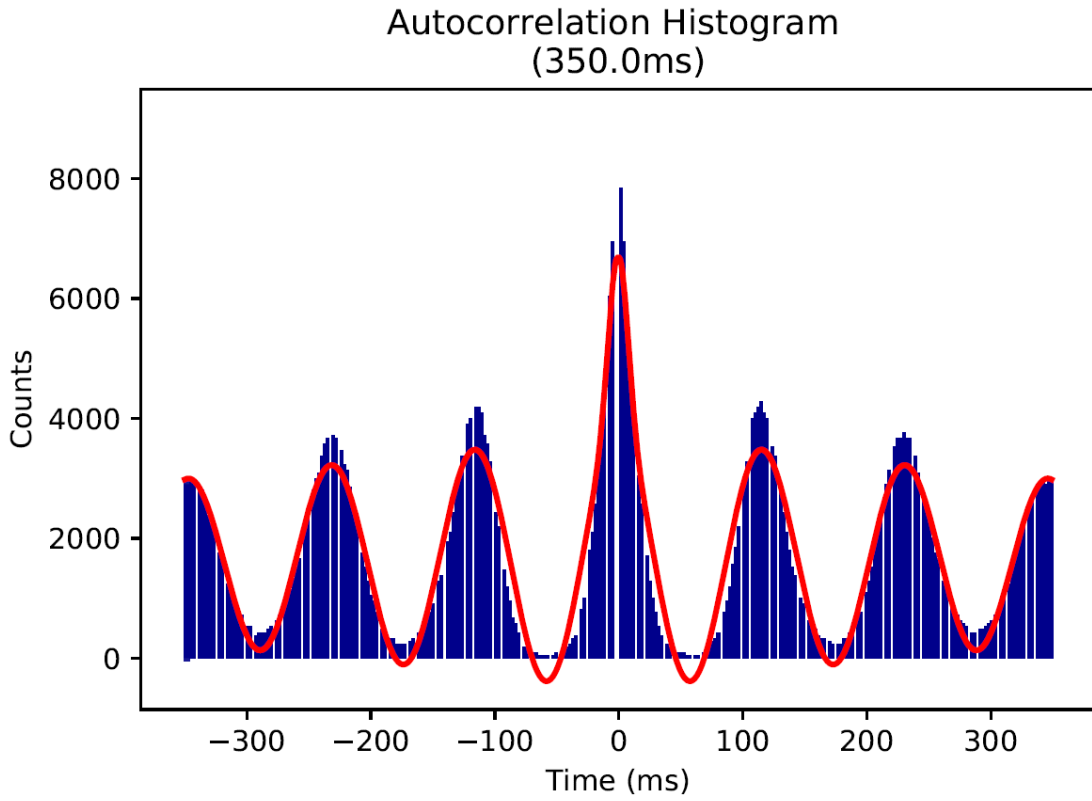


Figure 3.15 ISI autocorrelation histogram of a theta-modulated unit. The red envelope shows the fitted curve from the Equation 3.1. The index for theta modulation is derived from this fitted curve.

Analysis of theta-skipping cell

Some theta-modulated cells show theta-cycling skipping- the first side peak of the autocorrelogram is smaller than the second side peak. This implies that the spikes are occurring at alternating theta cycles (**Figure 3.16**; blue histogram). The non-normalized autocorrelation signal was fitted (**Figure 3.16**-red envelope) to **Equation 3.2** which is an extended form (Jankowski et al., 2014) of the one used in (Brandon et al., 2013).

$$y(x) = [a_1 \cdot \cos(\omega_1 x) + a_2 \cdot \cos(\omega_2 x)] \cdot \exp\left(-\frac{|x|}{\tau_1}\right) + b + c_1 \cdot \exp\left(-\frac{|x|}{\tau_2}\right) - c_2 \cdot \exp\left(-\frac{|x|}{\tau_3}\right) \quad (3.2)$$

here y is the autocorrelation signal, x is the lag, a_1 and a_2 are the amplitude of the oscillating terms with frequency ω_1 and ω_2 and modulated by exponential with decay constant τ_1 . τ_2 and τ_3 are the decay constants for the exponentials with amplitude c_1 and c_2 . The parameters a_1 , a_2 , b , c_1 and c_2 were allowed to vary in the range $[0, N]$,

where N is the peak of the autocorrelation signal. The range of default decay constants are, $\tau_1 = [0, 5000]$, $\tau_2 = [0, 100]$, $\tau_3 = [0, 10]$ in millisecond unit.

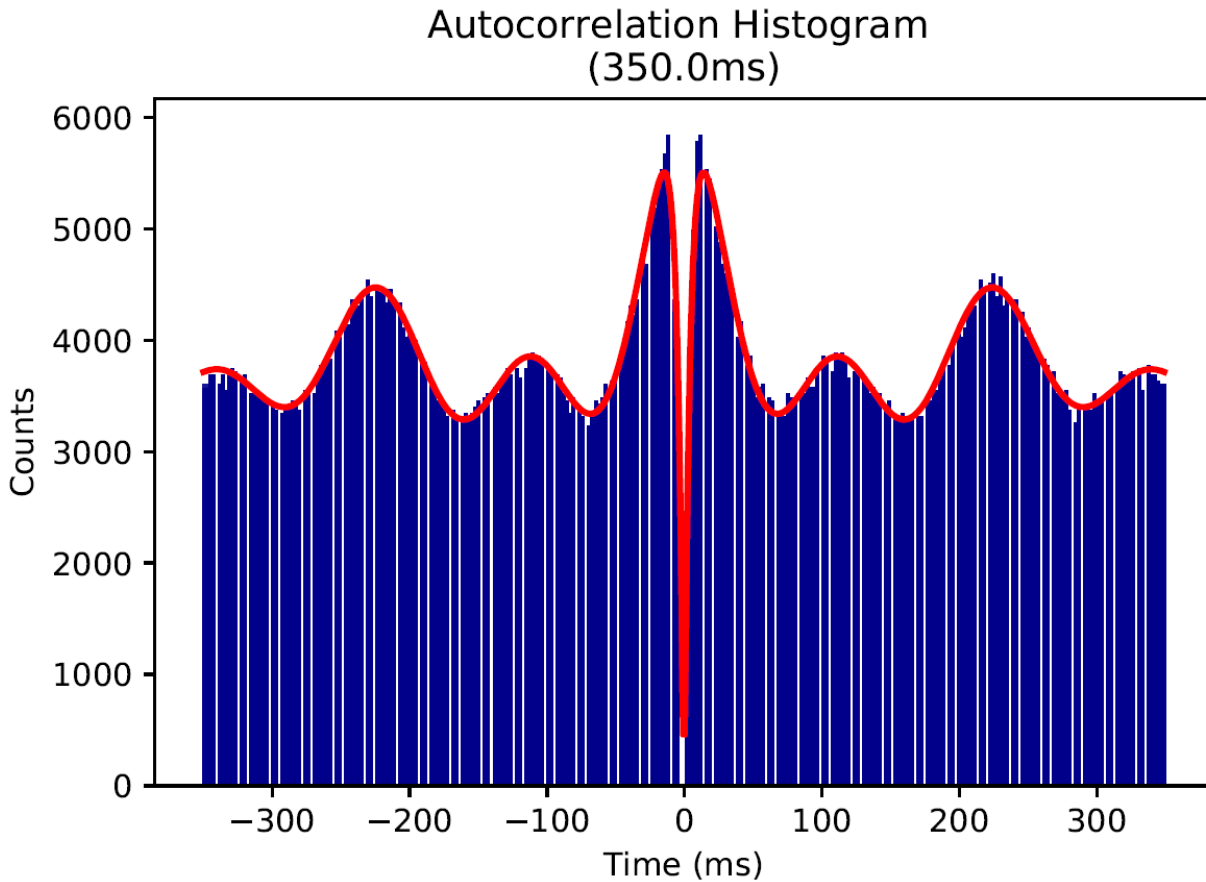


Figure 3.16 ISI autocorrelation histogram of a theta-skipping unit (blue histogram) and the corresponding fitted curve from Equation 3.2 (red envelope).

Similar to theta-modulated analysis, assuming that spikes were generated following a Poisson process, the ISI was considered to follow an exponential distribution which also gives an exponential distribution in its autocorrelation, represented in **Equation 3.2** by the positive exponential component comprised of (c_1, τ_2) . Initial dip in the autocorrelation for the delay in ISI was given by the fast decaying negative exponential (c_2, τ_3) . The alternating low and high peaks are modelled as superposition of two interfering oscillations, given by two slow decaying cosine functions, as if the high peaks are generated when the oscillations are in the same-phase and the low peaks are generated when they are in anti-phase. The baseline shifts for the cosine terms and all the constant errors are interpreted in constant b . **Equation 3.2** does not

hold the prior assumption that one of the periodic functions oscillates exactly at half-frequency of the other as that was assumed in (Brandon et al., 2013).

The curve fitting followed the measurement of ‘jump factor’, defined as the relative contribution of high and low frequency components in the higher peaks given by $a_2/(a_1+a_2)$. Frequency ratio of the cosine functions were also measured to verify the superposition model of **Equation 3.2**. Theta cycle skipping index, TS was measured by **Equation 3.3**

$$TS = \frac{p_2 - p_1}{\max(p_1, p_2)} \quad (3.3)$$

where p_1 and p_2 are the model values at one and two full cycles ($x= 2\pi/\omega$ and $x= 4\pi/\omega$) to the right of the center peak (Brandon et al., 2013). TS range should be $[-1, 1]$. Negative TS is unexpected according to the model in **Equation 3.2** if it is particularly fitted for the theta-skipping cycle cell with alternative low and high peaks, and low peaks appearing first after the center peak. The meaning of positive TS is that the second peak is larger than the first one. The bigger the TS is, the higher the larger peak is jumped from the theta-modulated signal, caused by the interference of the second oscillation.

We measured this effect with an alternative index ‘jump factor’ defined as $a_2/ (a_1+ a_2)$. The advantage of using $a_2/ (a_1+ a_2)$ is that it provides the direct measurement of the relative contribution of the two oscillations. Jump factor >0.5 means that the contribution of slow varying interfering oscillation in the jump is bigger than the theta-range oscillation.

If the Frequency ratio follows a 2:1 ratio (or closer), then the mathematical basis, of using two cosine functions of different frequency ranges to obtain consecutive in-phase or anti-phase superposition to yield lower and higher peaks respectively, is verified.

Analysis of burst property

In NeuroChaT, a burst is defined as a series of spikes in which each ISI was $< T_{th}$, where T_{th} is the threshold for ISI to consider the series of action potentials to constitute a burst-typically $T_{th} \geq T_r$ (T_r = Refractory period of the neuron). Bursts,

therefore consisted of a minimum of two spikes. The analysis first detects the bursts (**Figure 3.17**) based on the V_{th} set by the user. If the unit bursts, number of other parameters are calculated (**Appendix D**). The time intervals between the first spikes of consecutive bursts are defined as inter-burst interval (IBI). Burst duration is measured as the cumulative ISI of the bursting spikes in each burst or, equivalently, the time difference between the first and last spiking event of a burst. The ratio of burst duration to IBI for a detected burst is defined as the duty cycle. A propensity to burst index was calculated as the ratio of the number of burst-forming spikes and total spike count (Anderson and O'Mara, 2003) to measure the likelihood of a unit to fire in bursting mode, with the 0 value indicating the cell never bursts and 1 indicating the unit fires constantly in bursting mode.

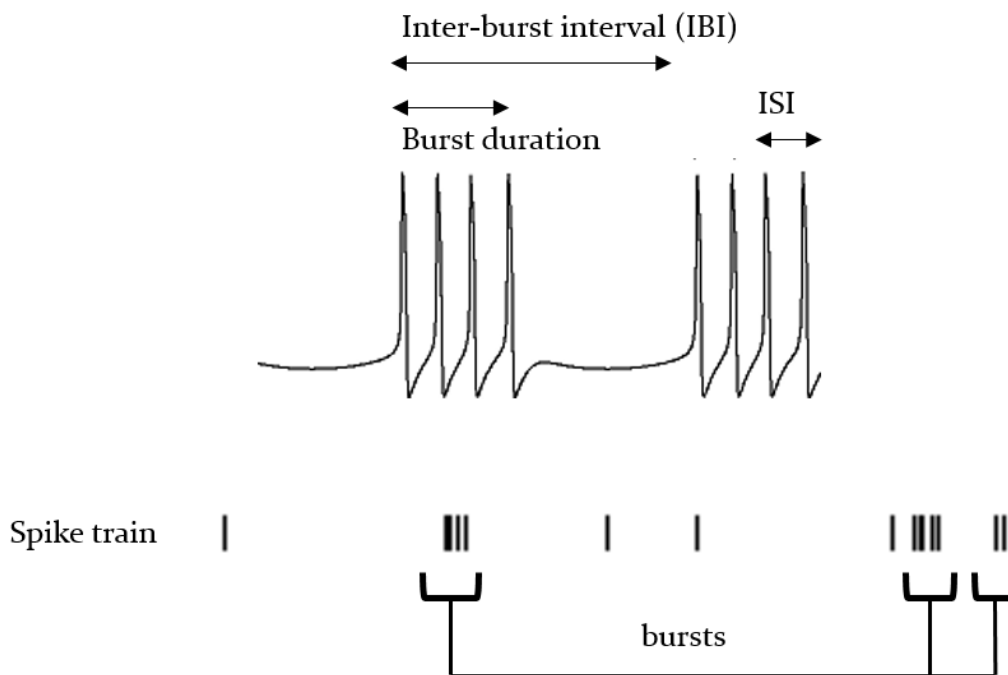


Figure 3.17 Illustration of bursting spike trains and the concept of inter-burst interval (IBI) along with the duration of burst. The ratio of burst duration to the IBI is defined as the duty cycle of bursting (Adapted from <http://www.scholarpedia.org/article/Bursting>).

Bursting units or units with bursts can be visually identified from the ISI histogram from its bimodality (**Figure 3.18**). The short ISIs from the burst spikes form the first mode while the longer ISIs form the second mode. If there exists a trough between the two modes, it may correspond to a) Refractory period of an intrinsic burst, or b) the timescale of the network-induced bursting (Doiron et al., 2003). There are two

approaches for detecting bursts in neural spike trains (Bakkum et al., 2014). The rate-threshold approach detects the spikes whenever the activity rate exceeds a specific value. This approach is more common in detecting network bursts or bursts in multiple spike trains. The ISI-threshold approach detects a burst whenever the ISI between consecutive bursts are smaller than a set threshold. NeuroChaT analyses are performed on a cell by cell basis and, therefore, ISI-threshold is the method of choice for burst-detection. Selecting the threshold can be manual or automated. Automated algorithms are based on finding valleys in the bimodal distribution of the ISIs (Pasquale et al., 2010). There are other approaches where the ISI distribution is modelled with a generative process i.e. Poisson distribution, and the bursts are identified as the ‘surprise’ events (Ko et al., 2012). We will consider implementing the automated detection algorithms in future versions of NeuroChaT. A comprehensive review on different algorithms is presented in (Bakkum et al., 2014) which can be a useful resource for future work in this regard.

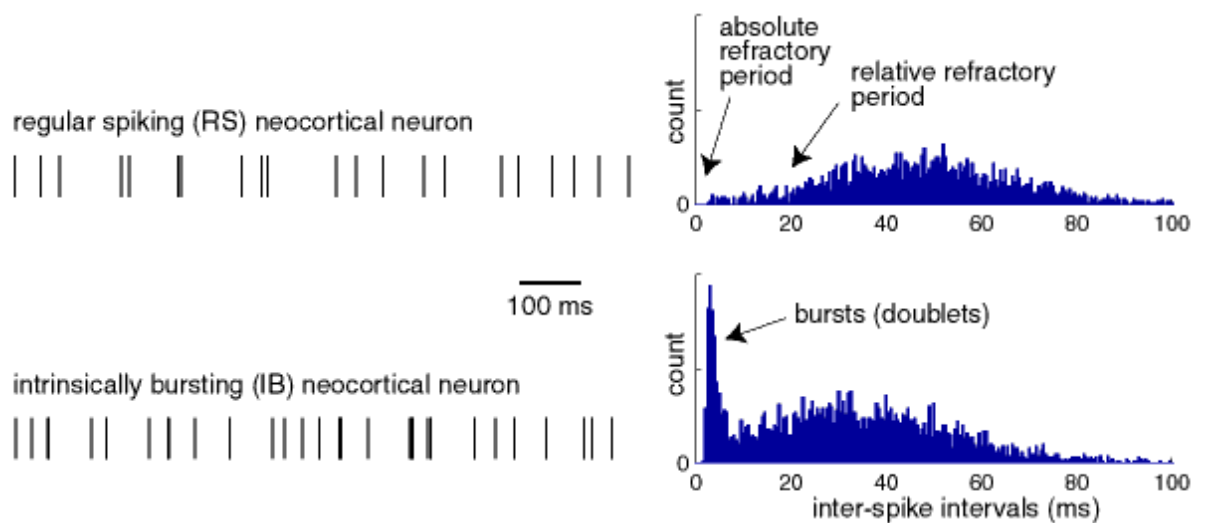


Figure 3.18 Depiction of the use of ISI histogram to identify the bursting units; top: regular spiking neuron, bottom: spike with bursting showing the bimodality of distribution with characteristic high counts of spikes at low intervals (Adapted from <http://www.scholarpedia.org/article/Bursting>).

3.3.2 Analysis pertaining to location of animal

Spike rate as a function of the location of the animal in the environment

This analysis explores the single unit spike activity with respect to the location of the animal in the arena through spatial firing map. The map is produced by obtaining the total number of spikes in each spatial bin using the video data and dividing them by the duration the animal remained in individual bins. The entire process is illustrated in **Figure 3.19**. The rate map is then smoothed using a two-dimensional box-filter of specified length. The unoccupied edges are chopped off and if back-ground firing rate adjustment is enabled in the configuration, then the analysis subtracts the firing map by mean rate of the pixels with 20%-40% of the peak rate if number of such pixels are counts more than 80% of the overall pixels.

Along with the firing map output, the analysis provides some measures of spatial selectivity like- a) Skaggs information content, b) Spatial coherence, c) Spatial sparsity. The Skaggs information content is expressed in bits per spike and calculated according to (Skaggs et al., 1996, Skaggs et al., 1993) expressed in the equation below,

$$Skaggs\ IC = \sum_i P_i \frac{R_i}{R} \log_2 \left(\frac{R_i}{R} \right) \quad (3.4)$$

where P_i is the occupancy probability at i -th pixel, which is measured from the time map in **Figure 3.19** by dividing time at each pixel with the total foraging time. R_i is the firing rate at each pixel, R is the overall firing rate (total number of spikes divided by total foraging time).

Spatial coherence is a measure of orderliness of the local firing pattern, or simply the correlation between firing map before smoothing, and the smoothed firing map (Muller and Kubie, 1989). As the smoothing was done using a box filter of, e.g. length 3, then each pixel is represented by the average of eight surrounding pixels. Thus high positive coherence implies that firing rate at each pixel can be predicted by knowing the rate of neighboring pixels, which in turn means that it is a measure of any consistent, graded, location-related variations in the rate (Cho and Sharp, 2001).

Spatial sparsity is a measure of adaptation that calculates the fraction of the environment in which a unit is active and is measured by the **Equation 3.5** (Treves and Rolls, 1991).

$$Sparsity = \frac{\langle R \rangle}{\langle R^2 \rangle} = \frac{(\sum_i P_i R_i)^2}{\sum_i P_i R_i^2} \quad (3.5)$$

where P_i and R_i , similarly, are the occupancy probability and the firing rate at i -th pixel.

There are other standard measures like coordinates of the centroid of the spatial firing map, firing rates at each pixel being the weights of the averaging.

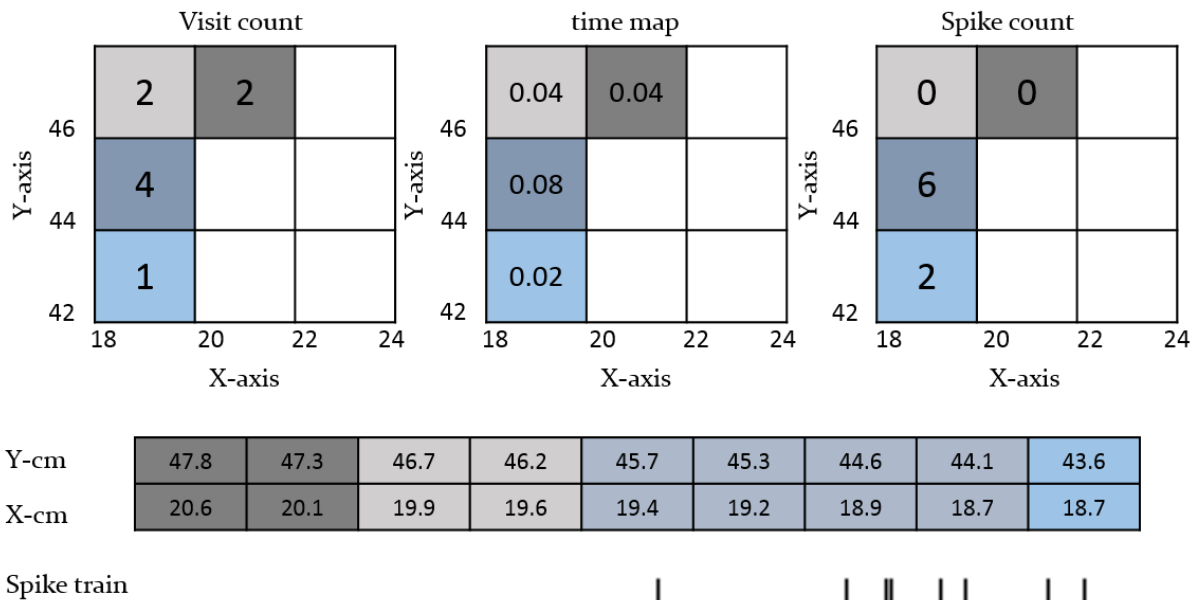


Figure 3.19 Illustration of how locational firing place is created from the location data of the animal and the spike times. A visit count map is created (upper left) from the two-dimensional histogram of the location which shows the number of times animal visited in a particular pixel in a foraging arena. Pixels of 2cm sizes are paired and color coded in the raw locational data in the middle. It is then multiplied by the sampling interval of the location data to obtain the total amount of time the animal has traversed in different pixels of the arena (upper-middle, time map). Another map is created for the spike count at those locations (upper right) by counting the number of spikes in the paired pixels. The map in the right is divided by the time-map to obtain the firing-rate map of a unit in the arena.

A sample firing rate map of a unit with a locational receptive field is provided in **Figure 3.20a**. We also plot the y versus x coordinates to show the traversing path of the animal and along with scatter plot of the location where a spike occurs, i.e. the

coordinates for the first spike in **Figure 3.19** is (19.4, 45.7). Such plots are called spike-plot with respect to the location. A sample plot is shown in **Figure 3.20b**.

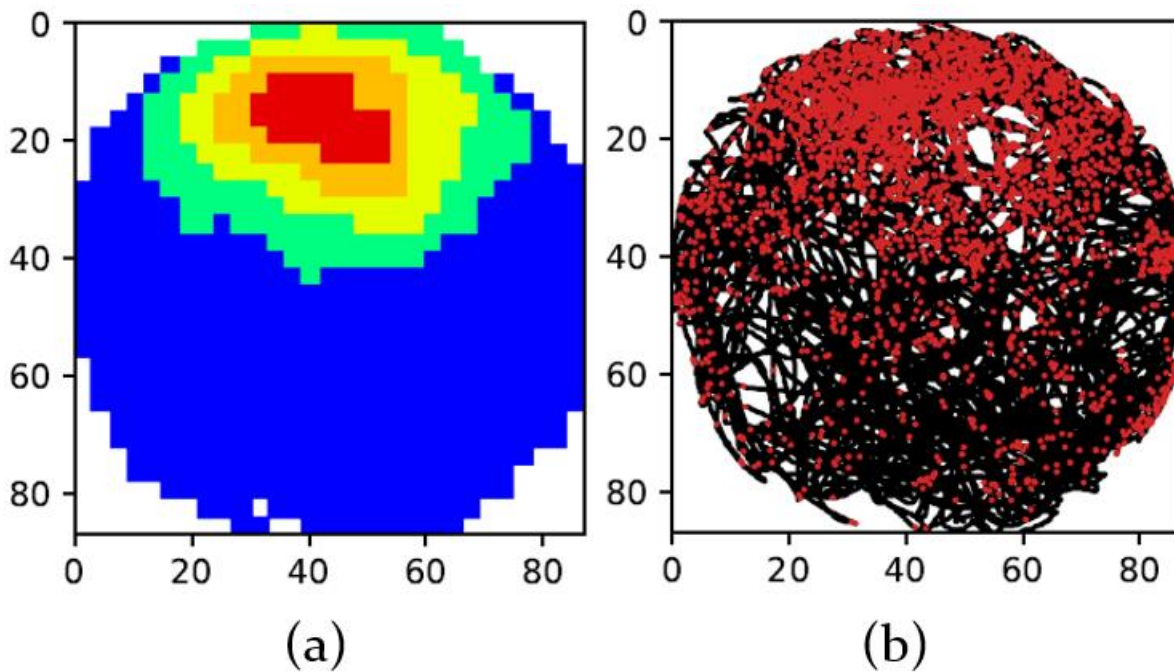


Figure 3.20 Sample outcome of locational firing rate analysis; (a) Firing rate as a function of location of the animal; (b) Black lines show the path of the animal traversed, the red dots represent spiking events.

Locational shuffling analysis

In this analysis, firing maps are created as that of in **Figure 3.20a** for spike-trains created from shuffling the time of the spiking-events within a specified time range from the events or randomizing them, as specified by the user. This procedure is replicated for a specified number of shuffles and spatial specificity measures (Skaggs information content, spatial coherence and spatial sparsity) are calculated for every new spike-train created. Such measures are then binned to create a histogram and observe the distribution. The fundamental idea behind this analysis is that if the spiking activity is independent of location, or the firing rate map that we observe is a random observation, then the specificity measures from the original spike-train will be statistically insignificant when compared to the similar measures observed from randomly shuffled spike-trains. Given that a higher information content and spatial coherence are expected for units with locational receptive field, they are compared to the upper tail of the 95th-percentile of the distribution. The spatial sparsity should be

low for such units, as expected, and would be compared to the lower tail at 5% significance level. The outcome for such analysis is presented in **Figure 3. 21**.

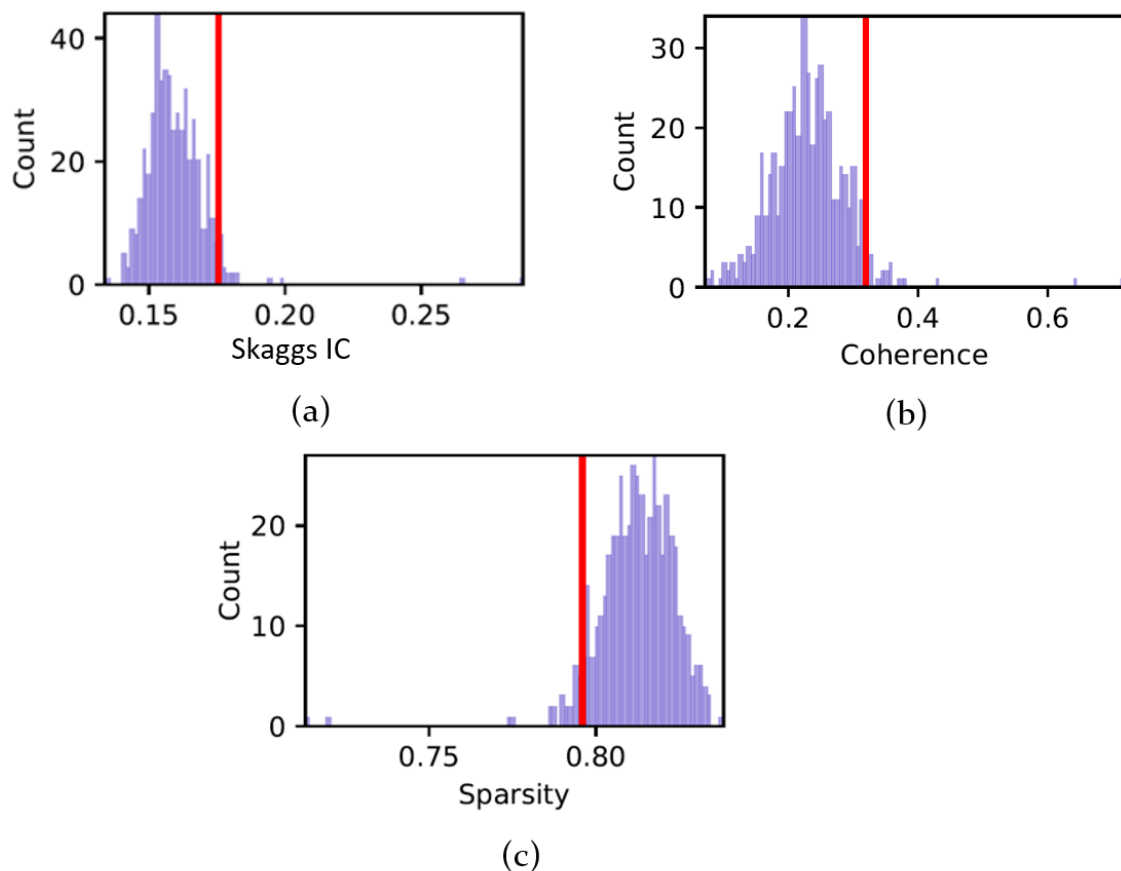


Figure 3.21 Distribution of specificity measures in locational shuffling analysis. Red line shows the 95th-percentile. The Skaggs information content (IC), coherence of the firing map from the original spike-train is compared to the red line to test for significance at 5% level at the upper tail as they are supposed to be higher in original spike-train if the spiking incidents with respect to location are expected not to be random (a, b). As the sparsity is expected to be minimum, the lower tail at 5% significance level is considered for the comparison (c).

Locational time-lapse analysis

Locational time-lapse analysis simply splits the entire recording sessions into smaller segments of creates firing rate maps, as well as the spike-plot, at following intervals:

Cumulative: 0-1min, 0-2min, 0-4min, 0-8min, 0-16min (if recording session is longer than 16min), 0-end of the recording

Progressive: 1-2min, 2-4min, 4-8min, 8-16min (if recording session is longer than 16min), 0-end of the recording.

Sample outcome for such analysis is shown in **Figure 3.22**. This simple analysis is effective in describing a unit whether it is responding to the animal's locational from the very beginning of the session, or first exposure to an environment, or it gradually emerges as the animal forages the environment. In short, it provides a picture of temporal evolution of the firing of a unit in an environment (Jankowski et al., 2014).

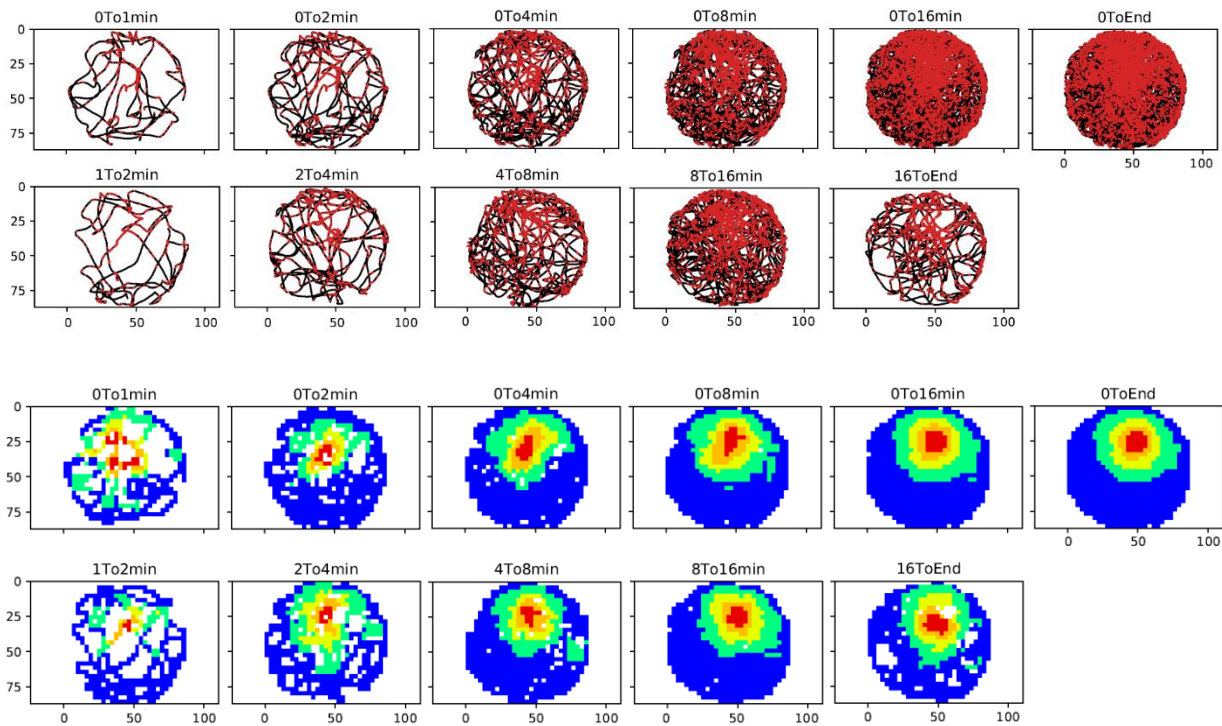


Figure 3.22 The time-lapse spike-plot (top two rows) and firing rate map (bottom two rows). The cumulative time-lapse are 0-1min, 0-2min, 0-4min, 0-8min, 0-16min, 0-end of the recording. The progressive time-lapse are 1-2min, 2-4min, 4-8min, 8-16min, 0-end of the recording.

Locational time shift analysis

A series of time-shifted firing rate maps are created for user-specified number of shifts in time with respect to the video samples. The algorithm remains as same as the firing rate map calculated for the original spike-train of the unit. Only the spike-train has been shifted by a multiple of location timestamps to the left (-ve shift) or to the right (+ve shift) to pair with past (-ve shift), present (no shift) or future (+ve shift) locations of the animal. This is illustrated in **Figure 3.23**.

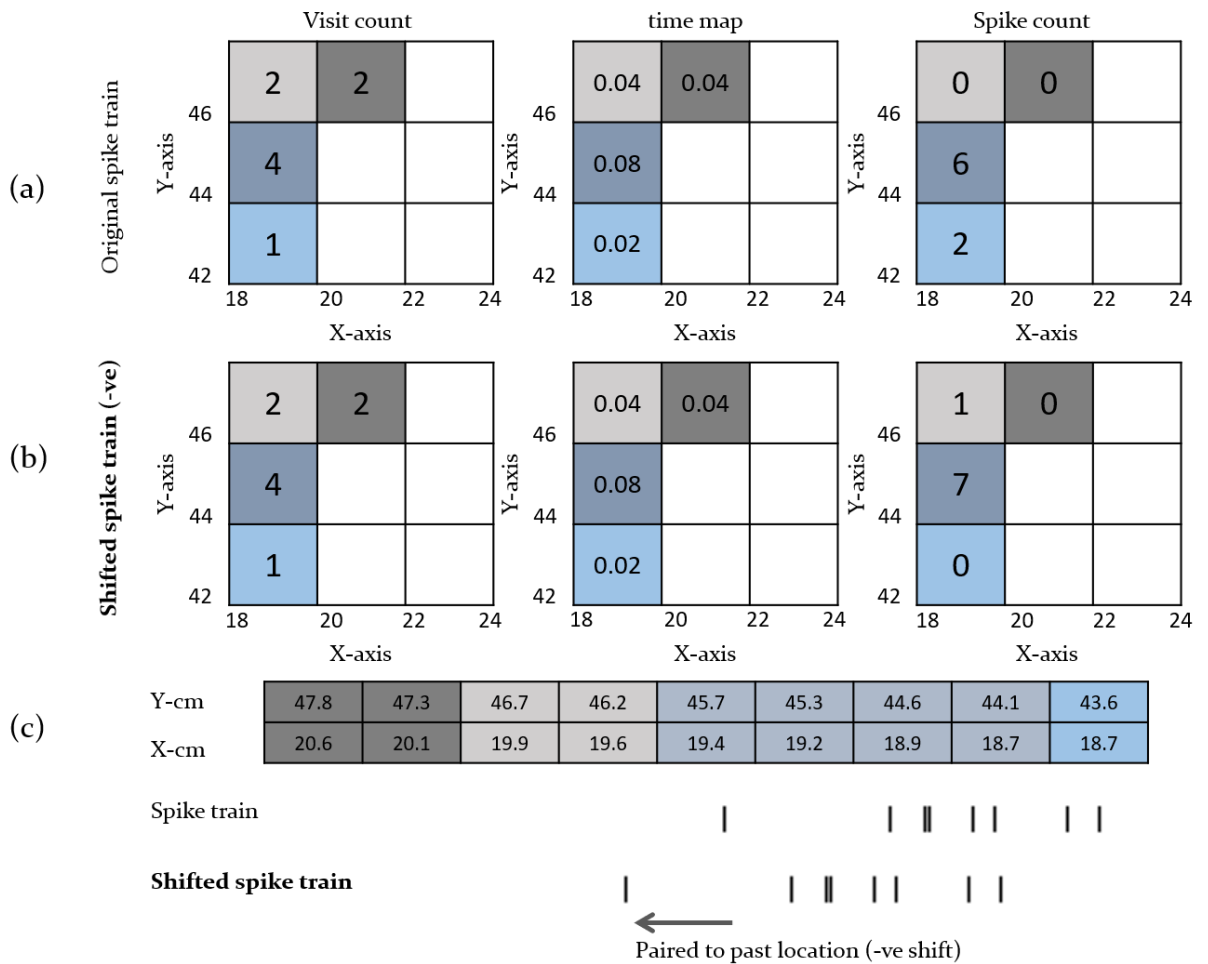


Figure 3.23 Construction of firing rate map from the spike train of the unit and shifted spike train; (a) Visit count, time map, and spike-count map for the unchanged spike-train of the unit; (b) Maps similar to (a), but with a shift in spike train as shown in (c); (c) top row shows the coordinates of the location, middle row depicts the spike train, and the bottom row shows the spike train when it is shifted to match with a past location of the animal. The pixels formed by pairing the x-, and y-coordinates are color-coded. As we can see, the count in spike changes in the shifted spike train, and a new map is formed out of it.

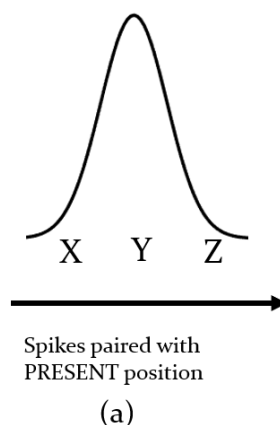
The specificity measures were calculated for each such map. While the firing-rate map and associated measures provide a way to determine whether the unit is responding to location of the animal, this analysis explores the temporal relationship between the unit activity and the location of the animal. The rationale behind this method, as described in (Sharp, 1999), is illustrated in **Figure 3.24**. Consider a one-dimensional trajectory of an animal through a place field (receptive field with respect to location), from left to right as shown by the arrow in **Figure 3.24a**, of the unit under consideration. The unit fires when the animal is at or near the location Y, and over repeated runs through this place field, the average firing rate of the unit forms an approximately Gaussian function, centered on location Y. If we assume that the

timing of the spikes coincides with the animal's arrival at the preferred location, the location signaled by this unit is Y. However, there are two other possible cases,

- The unit is driven to fire by location X, but it takes some time for the information to reach the recording location, so that the unit does not fire maximally until it reaches Y.
- The unit fires in relation to anticipated arrival at Z, and the firing is maximal when the animal is at Y.

Figure 3.24b provides an example of case (a). In this figure (left), when the unit activity is paired with simultaneous (present) position, it appears that the unit fires maximally at Z when traversing in the forward direction, and at X when travelling in the backward direction. But if unit activity is paired with the past location, two directional splits of the firing rate curves merges into one (**Figure 3.24b; right**). This implies that the unit lags and fires after the arrival at the preferred location.

Figure 3.24c provides an example of case (b). In left figure, the unit fires optimally at location X while moving from left to right direction (solid arrow) and at Z when the animal traverses in the opposite direction (dotted arrow). This is the case when the firing rate profiles are created by pairing unit activity with animal's simultaneous or present position. But if the spikes are paired with future position, the actual preferred location of the animal is location Y, as is shown in the right figure. This can be interpreted as the unit signals the arrival at Y ahead of time.



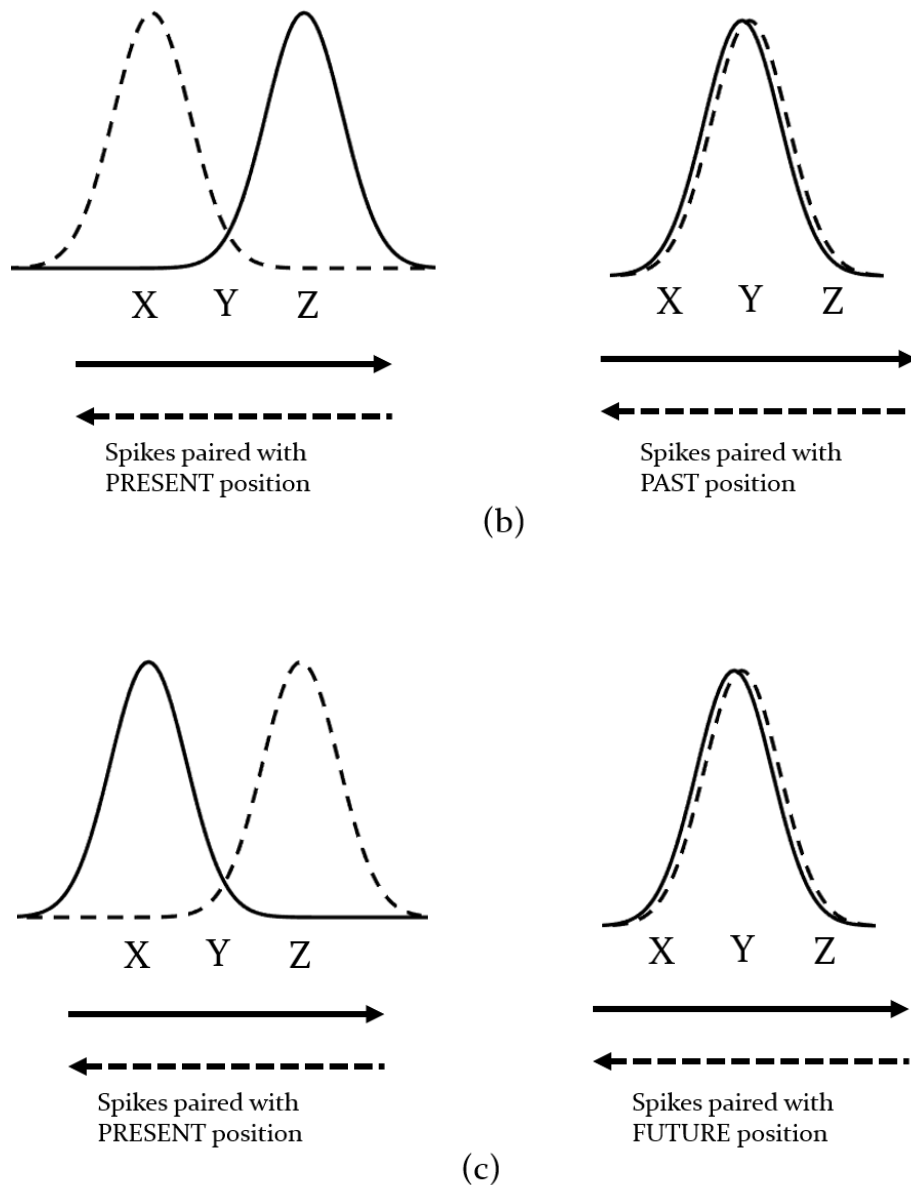


Figure 3.24 Illustration of the rationale behind time-shift analysis. The trajectory is considered one-dimensional for ease of explanation. The unit is assumed to signal the location Y. Solid arrow represents a trajectory from left to right across the place field, the dotted arrows shows the similar but in opposite (right to left) trajectory; (a) Expected firing rate curve if the unit fires simultaneously as the location of the animal changes; (b) An example of case-a where the firing rate is higher at a location (X or Z) after the animal traverses the signaling or preferred location Y (left); The firing rate maps overlaps and the place field becomes narrower when the spike train is paired with a past location (right); (c) An example of case-b where the firing rate is higher at a location (X or Z) before the animal traverses the signaling or preferred location Y (left); The firing rate maps overlaps and the place field becomes narrower when the spike train is paired with a future location (right);

In both the cases, the maximum information or spatial coherence can be expected when the unit activity is paired with a time-shifted position- either with past position when the unit is lagging, or with future position when it is anticipating. When the

specificity measures are plotted against the time-shift, a gradually optimal curve appears in units with place-field, as shown in **Figure 3.25**.

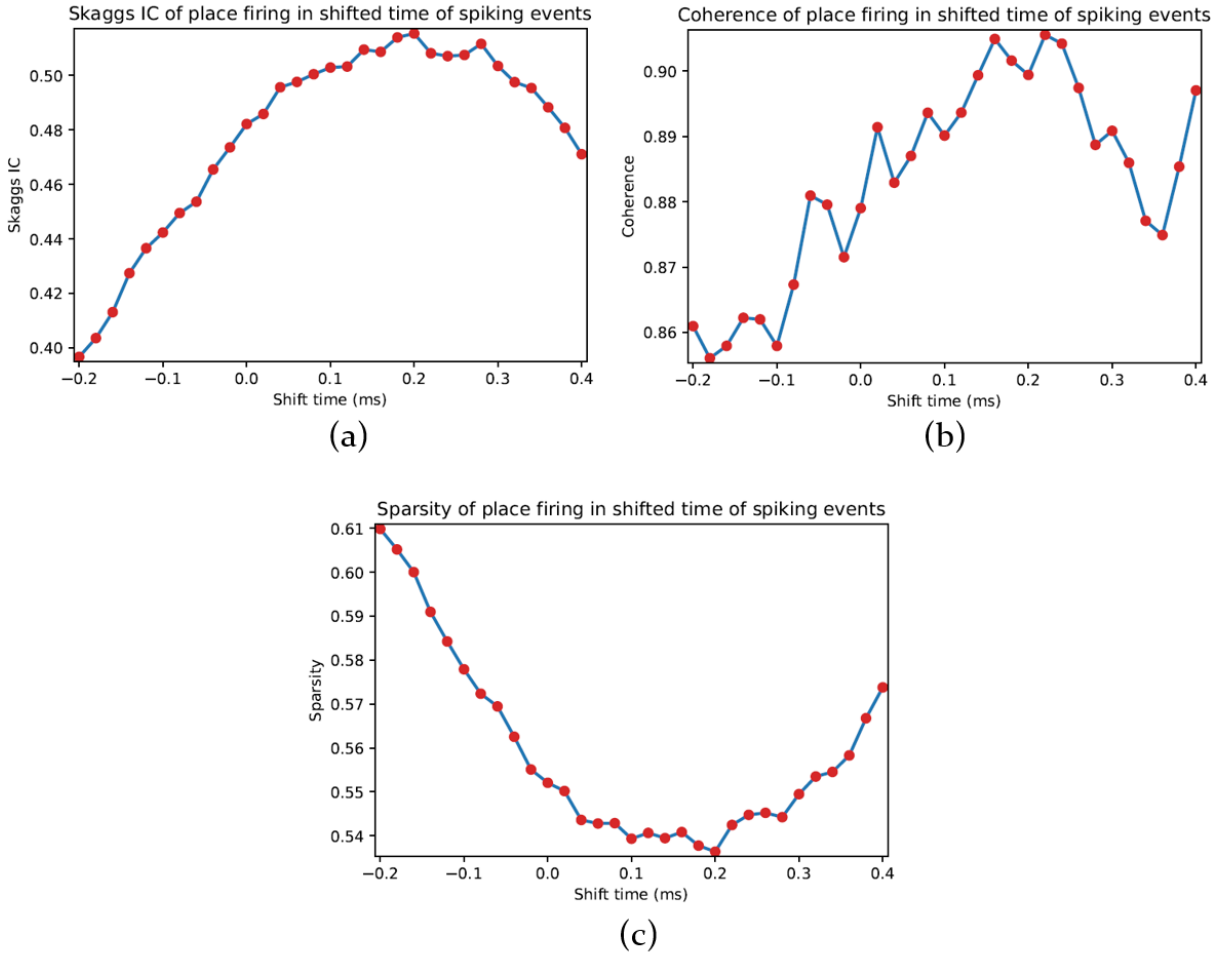


Figure 3.25 Specificity measures for locational firing in a unit with place field; (a) Skaggs information content (IC); (b) Coherence; (c) Sparsity. The Skaggs IC and the coherence are expected to attain the maximum value when the unit is paired with its preferred location where the sparsity will reach to its minimum as place field will be more compact. As they reach the optimum value at a +ve time shift, it can be speculated that the unit is anticipatory.

Spatial and rotational autocorrelation

Spatial correlation is a two-dimensional autocorrelation of the firing rate map. It is estimated using the equation below,

$$r(\tau_x, \tau_y) = \frac{n \sum_i R_i(x, y) R_i(x - \tau_x, y - \tau_y) - \sum_i R_i(x, y) \sum_i R_i(x - \tau_x, y - \tau_y)}{\sqrt{n \sum_i R_i(x, y)^2 - (\sum_i R_i(x, y))^2} \sqrt{n \sum_i R_i(x - \tau_x, y - \tau_y)^2 - (\sum_i R_i(x - \tau_x, y - \tau_y))^2}} \quad (3.6)$$

where R_i denotes the firing rate at i^{th} -pixel, τ_x and τ_y are the spatial lags. The summation is over all overlapping pixels between the firing rate map and its lagged

version. The autocorrelation was not estimated for those lags where the total number of overlapping pixels are less than a user-specified number. The concept of lags in spatial correlation, or 2-dimensional correlation in general is depicted in **Figure 3.26**.

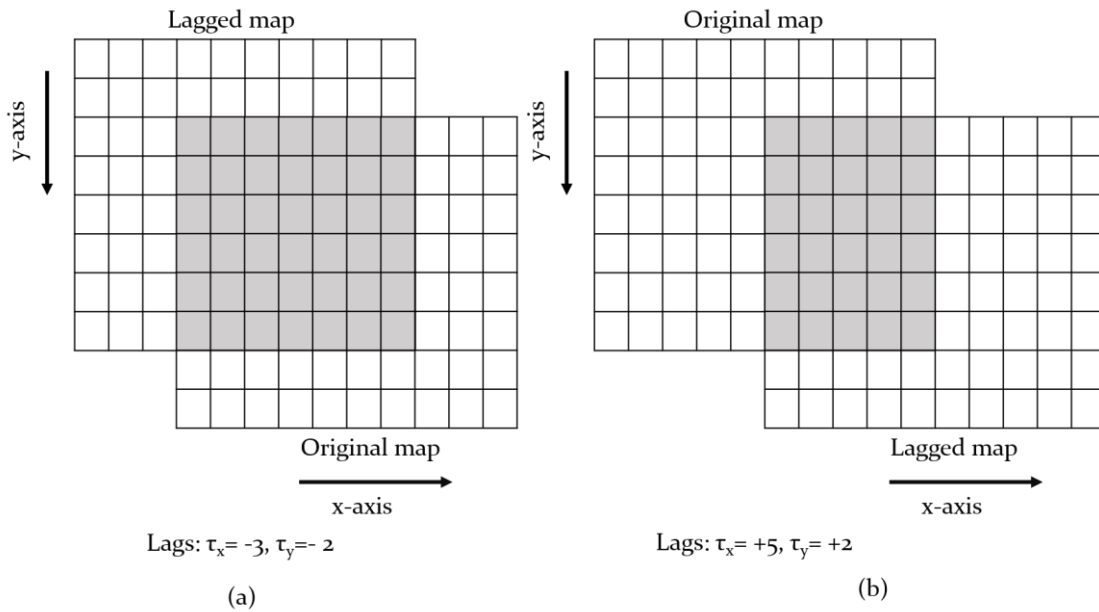


Figure 3.26 Construction of autocorrelation map- concept of lags and how the spatial autocorrelation is calculated. The shaded area shows the overlapping pixels between a lagged map and the original map. (a) overlap at $\tau_x = -3$ and $\tau_y = -2$; (b) overlap at $\tau_x = +5$ and $\tau_y = +2$. Correlation between the shaded areas for a particular lag represents the autocorrelation for that lag.

We further estimated the rotational autocorrelation by rotating the spatial autocorrelation maps in steps of user-specified degree, i.e. 5° , and calculating the 2-dimensional correlation between each rotated map and the original. As we will see in the next section, this analysis helps identifying the periodicity in units firing in grid patterns (Sargolini et al., 2006). The spatial autocorrelation itself also reveals patterns of hexagonal receptive fields in grid cells.

Grid cell analysis

A grid cell reveals spatial organization of subfields forming a grid of regularly tessellating triangles spanning the whole recording surface (Hafting et al., 2005) as shown in **Figure 3.27a**. Once the spatial autocorrelation is calculated, the regular nature of the activity distribution is more apparent, and the central peak of the autocorrelogram is surrounded by six equidistant peaks forming the vertices of a regular hexagon (**Figure 3.27b**). This analysis identifies the geometric structure,

verifies if it forms a grid, and characterizes the structure of the grid. The analysis flow chart is shown in **Figure 3.28** and illustrated in **Figure 3.29**.

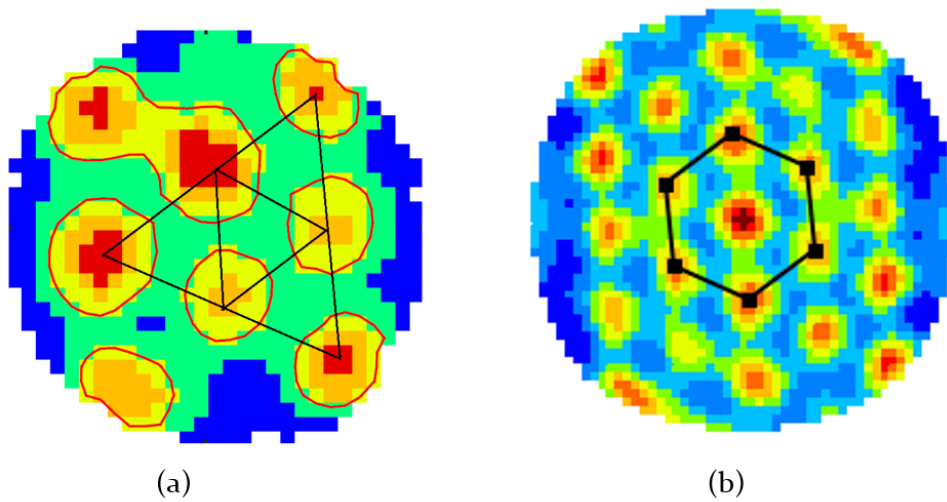


Figure 3.27 Patterned firing fields in a grid cell; (a) pattern of tessellating triangles among the firing fields; (b) Hexagonal pattern around the central peak of the spatial autocorrelation.

The spatial autocorrelation of the firing rate map of the unit is obtained. The maxima in the spatial autocorrelation are identified (**Figure 3.29a**) and sorted by their distance from the central peak. The first 6 peaks are then isolated and their distance from the centers is measured. If they all belong to within $\pm 25\%$ of their mean distance (**Figure 3.29b**), the portion of the autocorrelogram within $\pm 50\%$ of the mean distance is then isolated (**Figure 3.29c**) and its rotational autocorrelation is observed (**Figure 3.29d**). If there exists alternating peaks and troughs at every $60 \pm \text{tolAngle}$, the difference between correlation values at the peaks and troughs is defined as the gridness score of the unit (Sargolini et al 2006). The mean of the central angles (α), interior angles (ψ), orientation of the grid with the positive x-axis measured as the orientation of first peaks are also calculated for further verification and characterization (**Figure 3.29e**).

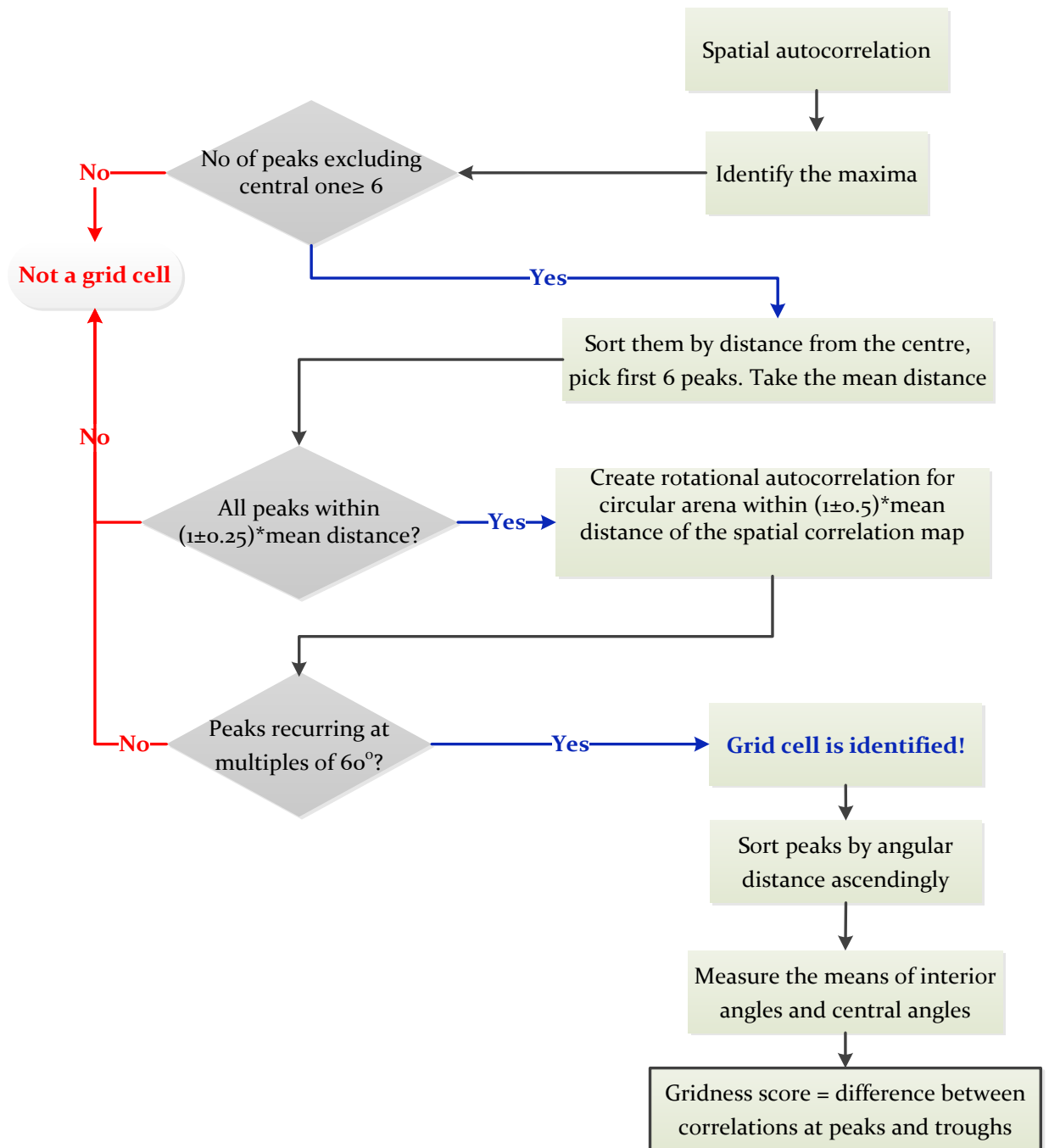


Figure 3.28 Flow chart for the detection of grid cell and measuring the gridness score.

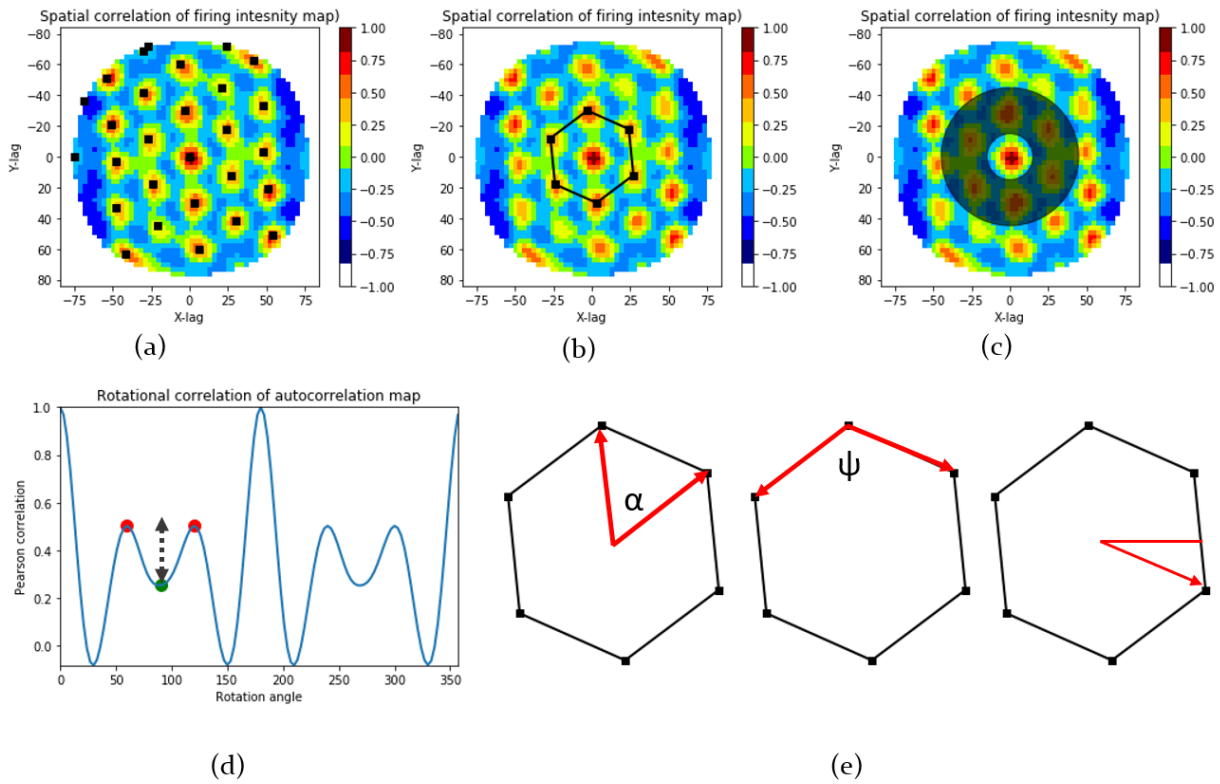


Figure 3.29 Illustration of identifying grid cell and calculating gridness score; (a) Maxima locations in spatial autocorrelation; (b) Peaks forming the central hexagon; (c) Shaded zone shows the $(\pm 0.5) \times$ mean peak distance; (d) Rotational correlation measured with the shaded zone in (c). Red dots show the maxima locations, and the green dot shows the minima in the curve. Gridness score is the difference in between the rotational correlations at peaks and troughs; (e) Definition of central angle (left), interior angle (middle) and orientation of the grid (right).

Border cell analysis

Border cells are units that are activated along the geometric border of the arena an animal traverses (Jankowski and O'Mara, 2015). The activity can be along the entire environmental border or along a segment of the arena (**Figure 3.30**). When the unit has a receptive field in the intersection of the environmental boundary and is located at a specific distance from the animal in a specific allocentric direction, the unit is more specifically called boundary vector cell (BVC) (Lever et al., 2009), although they may also be referred as border cells as in (Solstad et al., 2008).

In NeruoChaT, analysis of such cells starts with identifying the environmental border from the traversing path of the animal as the recording systems usually do not provide this information. The outermost pixel along the path forms the putative environmental border. This is primarily a two-step process (**Figure 3.31**): (i) a binary

map of the traversed arena is produced by placing '0' for the pixels visited, and '1' for the pixels the animal did not visit and (ii) The edging 1s are kept optimized to have a regular pattern and isolated as border.

Once the border is identified, the distance of each pixel is measured as the number of pixels it is away from the border times the pixel size. Pixel where an animal is located at each time-point is identified and corresponding distance is assigned as the distance from border for that spatial data point.

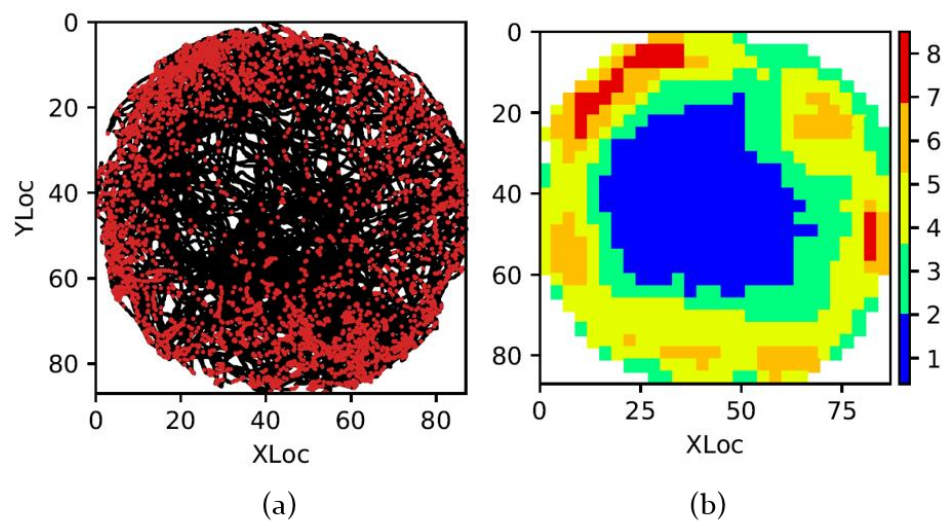


Figure 3.30 Spike plot (a) and firing rate map (b) of a unit with preferential firing along the border of the environment.

The spike firing rate with respect to border than follows the regular rate coding scheme as that followed in calculation of firing rate vs running speed or angular head velocity (see Section 3.3.4). Figure 3.32 and Figure 3.33 show the output graphics of a border cell analysis. Figure 3.32a shows the typical firing rate map, but it reveals that there are zones of higher firing rate near the environmental border, indicating that this is a border cell. The active pixels of this map are defined as the ones with firing rates higher than a user-specified threshold. If the user sets 20% of maximum firing rate as the minimum rate for accepting the activity of pixel significant, the total number of such pixels at every distance ring (Figure 3.31f) are divided by the number of pixels located in that distance within the border. Such normalization is necessary as the total number of pixels away from the environment because of shrinking in the area while the pixel size is remaining same throughout the environment. The

histogram of such count for the example cell is shown in **Figure 3.32b**. The firing rate vs the distance is shown in **Figure 3.32c**. The stair-plot shown in **Figure 3.32d** shows the mean distance of all the pixels within stepwise activity level. For example, the pixels with firing rate within 80% to 100% are identified, and the mean of their distance from the border is obtained. For a border cell, this distance is very low as most of the active pixels will remain near the border (dark grey zone in **Figure 3.32d**). As the distance increases, the firing rate declines, and drops drastically near the center of the arena, resulting in a sharp increase in the mean distance at low firing percentiles (light grey zone in **Figure 3.32d**).

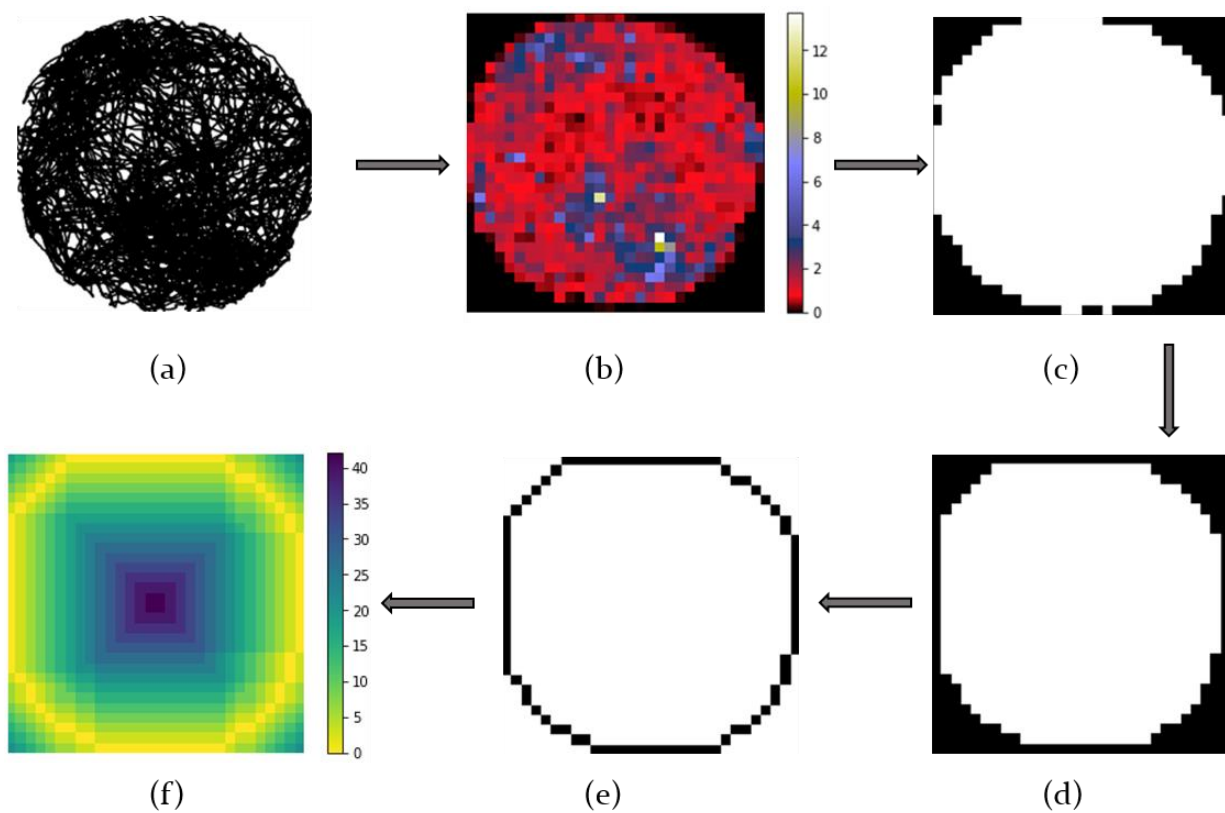


Figure 3.31 Stepwise depiction of border detection procedure; (a) black lines show the path of the animal traversed in the environment; (b) Paths are pixelated to count the amount of time animal visited a pixel; black area shows areas captured by the camera but not visited by the animal; (c) Binary picture of the map in (b). Unvisited area is represented by '1' and visited area by '0' (Colormap is inverted); (d) Binary map is regularized to obtain a smoother boundary; (e) Black pixels are the ones forming the boundary; (f) Map of the distance of each pixel from the boundary pixels.

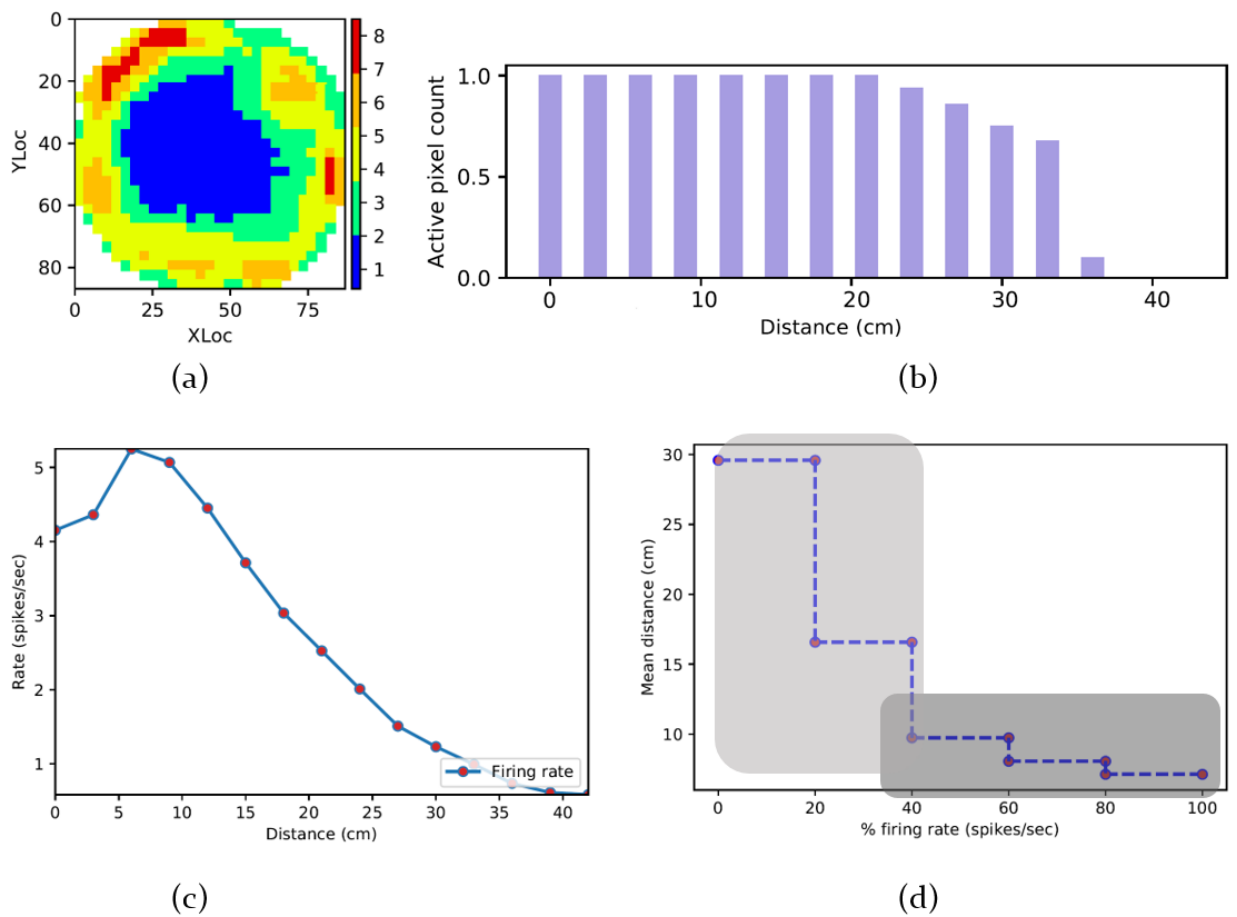


Figure 3.32 Border cell analysis; (a) Spatial firing rate map showing a characteristic circular ring of high firing rate zone near the environmental boundary; (b) Histogram of normalized active pixel count from the border at a particular distance. Active pixels are counted as the total pixels with firing rate higher than the user specified threshold, i.e., 20% normalized by the total pixel at that distance; (c) Firing rate calculated according to the rate coding scheme; (d) Stair plot showing a low mean distance from border for pixels with high firing rates, and sudden increase in distance for pixels with low firing rates, indicating that firing rate is high near the border and almost diminishes near the center of the arena.

Figure 3.33a shows the active pixel count, but instead of the distance from the border, it uses the angular distance from the center of the arena. The pixel count is not normalized. This analysis shows the spread of the activity across the environment. The user can set a threshold that if there is a continuous block of 90degree spread of the activity, covering $1/4^{\text{th}}$ of the environmental border, the cell can be considered as a border cell. **Figure 3.33b** is an extension of **Figure 3.33a**, as it linearizes the entire firing rate map, and counts the number of active pixels at particular distance along a particular directional bin.

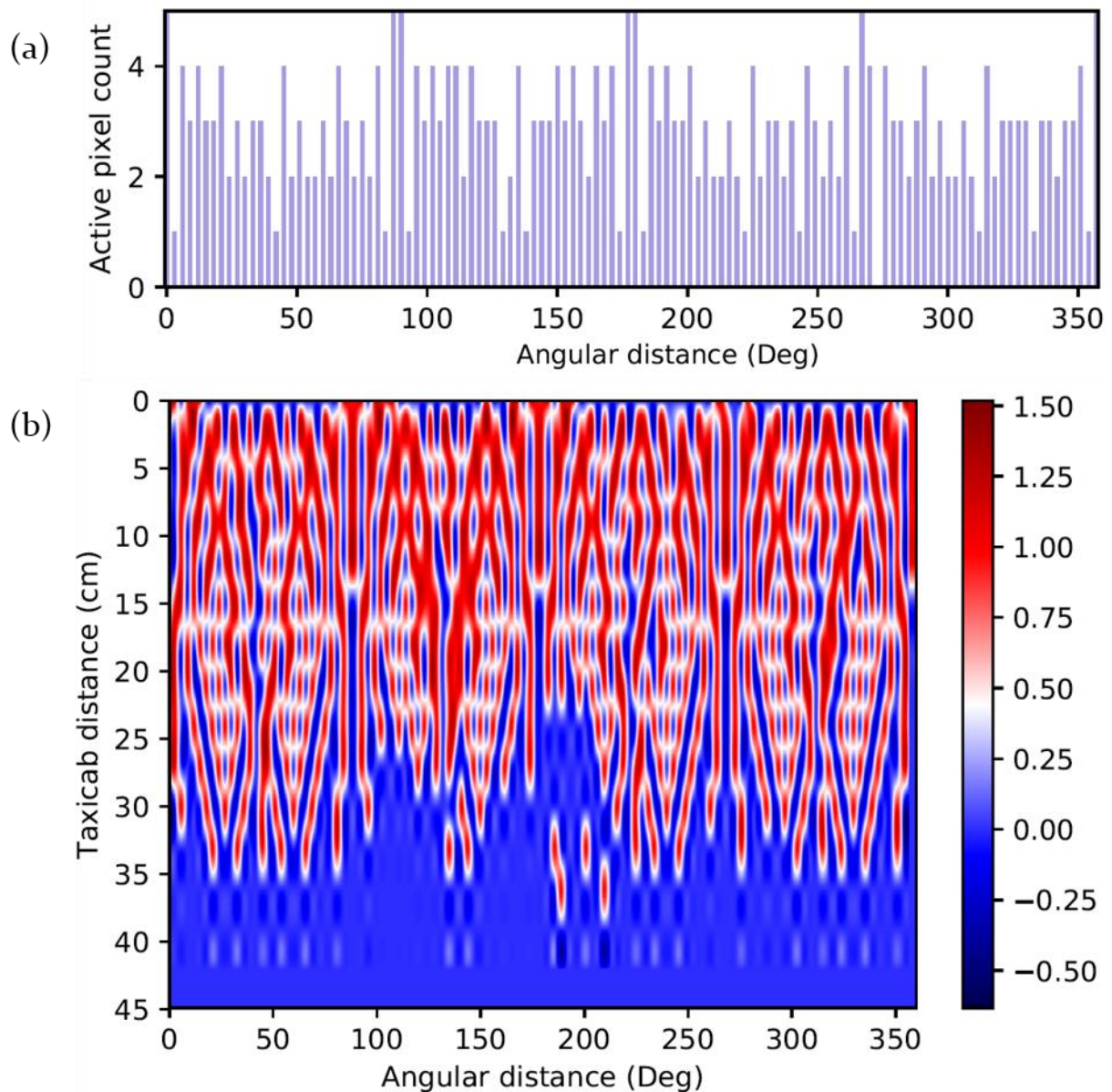


Figure 3.33 Linearization of firing rate map; (a) Number of active pixels (absolute count) at different angular distance of the environment from the center of the arena; (b) Number of active pixel at different distance from the border along particular circular distance.

Gradient cell analysis

Gradient cells have a gradually increasing firing rate from the environmental boundary to the center of the foraging arena. **Figure 3.34a** shows an example of such a unit where the activity of the unit increases gradually as the animal moves from border towards the center, forming bands of activity belonging to a particular firing rate percentage. Each color in the map represents spiking activity of 0-100% in step of 20%. The stair plot (**Figure 3.34b**) is obtained (like the border cells) using a gradual stepping in the mean distance with respect to the firing rate bands, representing a

graded activity pattern in contrast to the border cell where active pixels are located close to the border. The spiking rate vs distance of the animal from the border is calculated like that in border cell and fitted to a Gompertz function (**Equation 3.7**) in **Figure 3.34c**, the goodness of fit is evaluated, and the differential firing rate is calculated by taking the first derivative of the fitted curve with respect to the distance (**Figure 3.34d**).

$$y = a. \exp^{-b \exp^{-cx}} \quad (3.7)$$

where y is the firing rate, x is the distance from the border, a is the Gompertz asymptote setting the saturation of firing rate at the edges, and b and c are positive numbers setting the displacement along the x-axis and the growth rate respectively. The distance at which differential rate maximizes at short distance further from the border within the arena is defined as the deflection point and can be used to characterize the gradient activity. This equals to the $\ln(b)/c$ in the fitted equation.

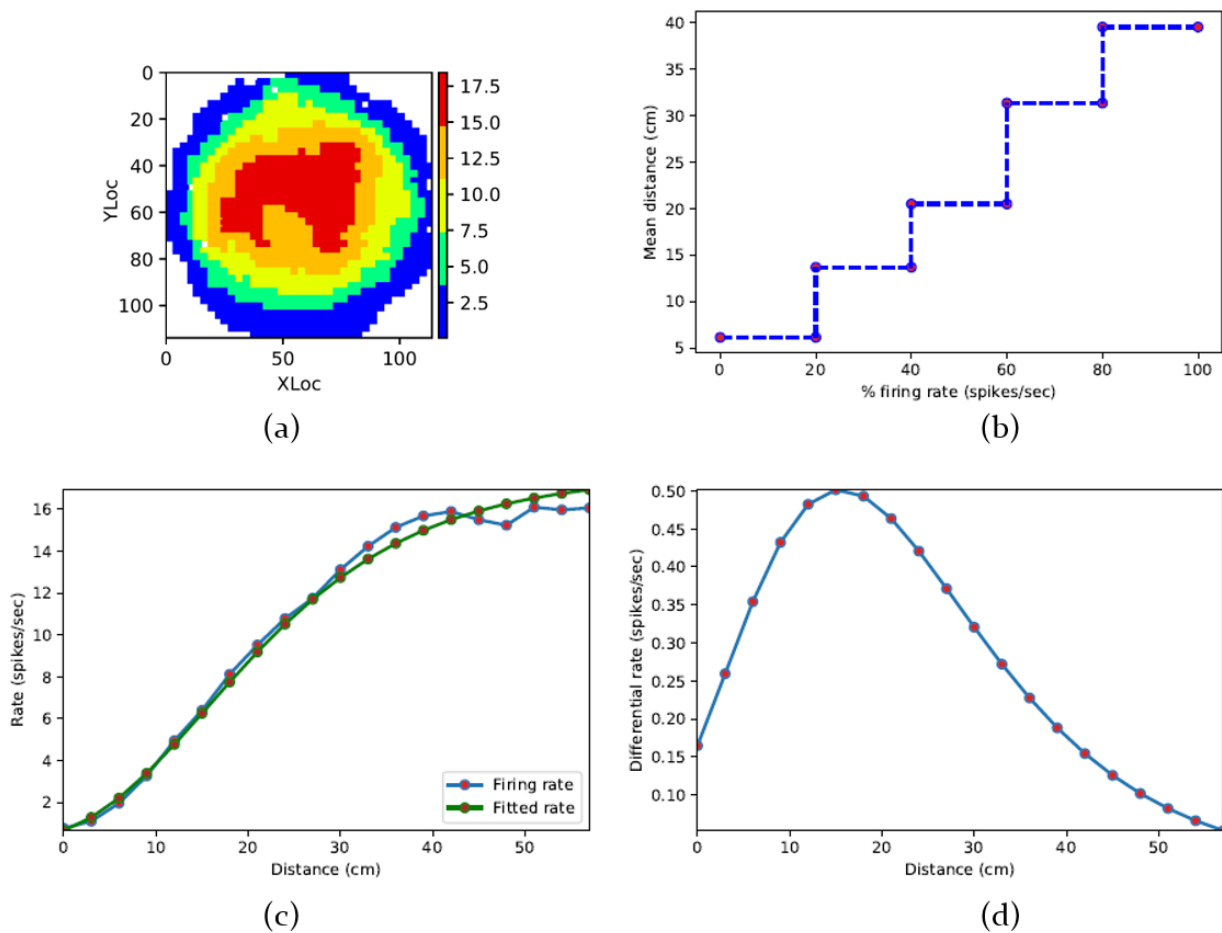


Figure 3.34 Gradient cell analysis; (a) Firing rate map showing that the activity of the unit increases gradually as the animal moves from border towards the center, forming bands of activity belonging to a particular firing rate percentage. Here, 5 colors have been used representing spiking activity of 0-100% in step of 20%; (b) Stair plot like the one calculated in border cell. The stairs are increasing gradually, and there is not abrupt change in the plot, further providing support for the gradient cell; (c) The spiking rate vs distance of the animal form the border is calculated like that in border cell (blue). This curve is fitted with Gompertz function (green); (d) Differential firing rate vs distance showing that the increase in firing rate maximizes at short distance further from the border within the arena. The point where this rate of changes in firing rate maximizes is called deflection point.

3.3.3 Analyses on directional correlations of spikes

Spike rate as a function of head direction of the animal

This function analyses the effect of head-direction (HD) of an animal with respect to the environment in the horizontal plane on spiking rate of a unit. It starts with smoothing the head directional data of the animal by replacing each recorded sample with the circular mean of 5 data samples centered around that sample, or, $n_t = \text{Circular Mean}(n_{t-2}, n_{t-1}, n_t, n_{t+1}, n_{t+2})$, where n_t is the t -th sample of the recording session. The directional heading is divided into bins of user-specified size, i.e. 5° . The spike rate is

calculated by dividing the total number of spikes in a particular head-direction bin by the amount of time the animal was heading that direction bin (Taube et al., 1990). This is depicted in **Figure 3.35**.

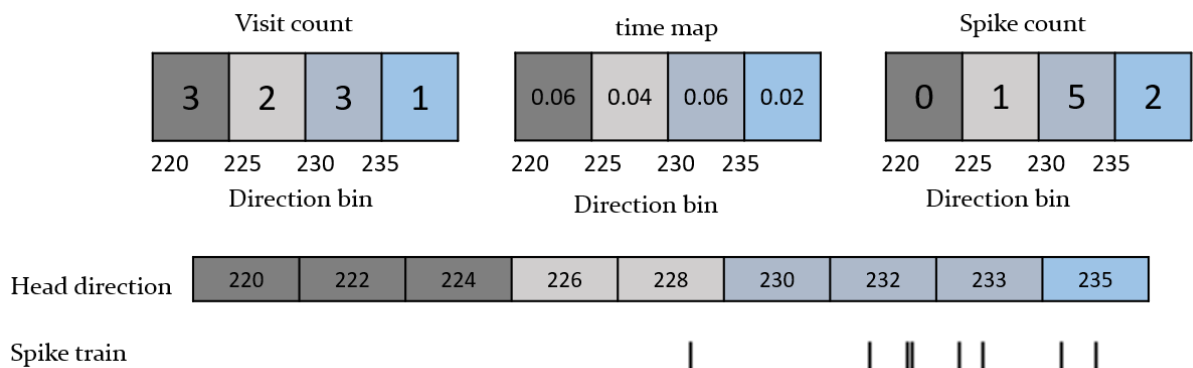


Figure 3.35 Depiction of calculation of the spiking rate with respect to the head direction of the animal. Top row- left: visit count of the animal in different directional bins, middle: amount of time spent in such directions, right: number of spikes paired with such direction; Middle row: head-direction of the animal at different tracker samples.; Bottom row- spike-train of the unit during 9 samples shown. Head directions falling in different bins are color-coded differently.

The head direction vs spike rate curves are called “HD tuning curves”. The idea of such rate coding is similar to that of the place field, but instead of having a 2-dimensional distribution of spiking activity, it is assessed for circular data (directional). Therefore, the statistical analyses are those used for directional data for head-directional information. The circular mean is then calculated which is the preferred direction of firing of a unit. The firing rate in this direction is also reported. Skaggs information content is measured according to **Equation 3.4**, where P_i and R_i represents the occupancy probability and firing rate at i -th directional bin, θ_i , instead of the spatial bins described in that section. Other standard circular statistical tests and measurements- like mean resultant vector, Raleigh’s z-test for uniformity, concentration parameters for von-misses distribution- are calculated according to (Mardia and Jupp, 2008). Corresponding equations are provided in **Appendix F**.

The typical firing rate pattern in a head-direction tuned unit is shown in **Figure 3.36a**. The width of the tuning curve is measured as the angular differences between the points where the firing rates are 50% of the peak firing rate (dotted line in **Figure 3.36b**). It is used to characterize the spread of the directional receptive field.

Drawing the similarity from **Figure 3.24** for one dimensional place field, we can similarly explain that there are possible cases of a unit being anticipating or lagging behind from the preferred direction of firing. To observe such effects, we calculate the angular head velocity (AHV) and split the head directions into clockwise (CW) or counter-clockwise (CCW) head movement depending on the sign of the AHV. A '+ve' AHV indicates that the animal's head moves in CW direction, and '-ve' AHV indicates a CCW movement. In both cases the absolute value of AHV must be greater than a user-specified threshold to avoid flickers in head-movement of the animal during exploration. The two sets of head-directions are then paired with synced spiking-activity to obtain two curves (**Figure 3.36c**; red for CW and blue for CCW) and their mean values are calculated. If the peak firing rates for these two curves occur at θ_{CW} and θ_{CCW} respectively for the head directions at CW and CCW movements, the separation angle is measured as $\delta = \theta_{CW} - \theta_{CCW}$.

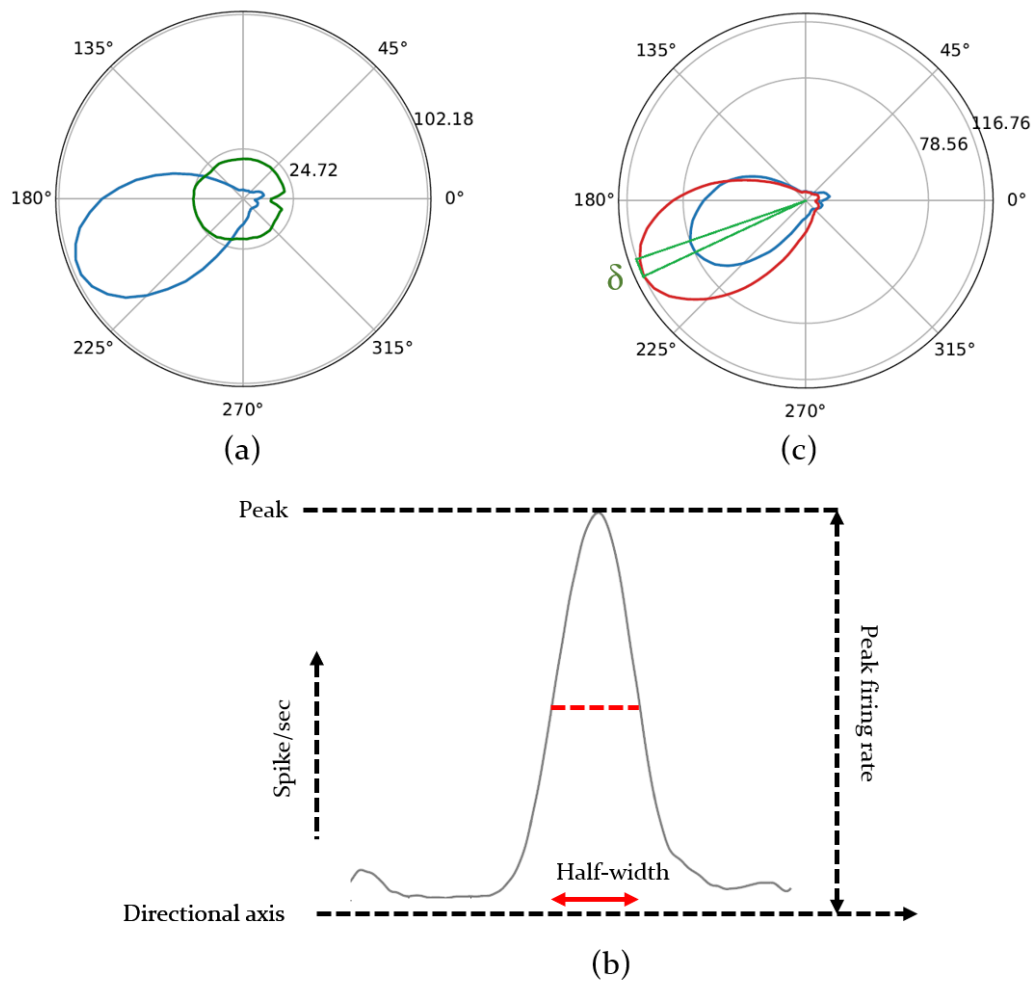


Figure 3.36 Tuning curves from a recorded head-directional unit; (a) Firing rate with respect to directional heading of the animal (blue) and the predicted firing rate (green) to observe the homogeneity in sampling of the directional bins; (b) depiction of the measure of half-width, a characteristic of the tuning curve to describe how wide the directional receptive field is; (c) Measurement of separation angle, the difference between the directions where peak firing occurs during CW (red) and CCW (blue) head movement.

The green line in the head directional tuning curve (**Figure 3.36a**) is the predicted firing rate calculated using the methods described in (Cacucci et al., 2004) based on ‘distributive hypothesis’ (Muller et al., 1994). The approach consists of calculating the directional firing that one would predict under the null hypothesis that place cell firing is only modulated by location and that the only influence of head direction arises from the sampling bias. Sampling bias arises because of inhomogeneous sampling of locations and head directions and may cause a place cell to show an apparent directional modulation of its firing rates and vice versa. For example, if an animal is foraging in a cylindrical arena, a head-direction unit with a preferred

northeast direction may show a place field against the northeast portion of the wall. This is attributable to the fact that whenever the animal is in the northeast region of the cylinder, it will often face northeast and cannot face southeast due to the physical constraint. It may also be the case that a spurious head-direction response arises if the animal samples some head directions more frequently while spending time within a place field, i.e. when the place field is near a wall.

The predicted rate as a function of direction is derived from the locational firing map using the equation,

$$R_{pred}(\theta) = \frac{\sum (R_p T_p(\theta))}{\sum T_p(\theta)} \quad (3.8)$$

where R_p is the firing rate at p -th pixel, $T_p(\theta)$ is total time spent facing head direction θ while at pixel p . The numerator denotes the average number of spikes at direction θ , and the denominator is the amount of time spent across all pixels only when the animal was facing in the direction θ . This split of time by head-direction is illustrated in **Figure 3.37**.

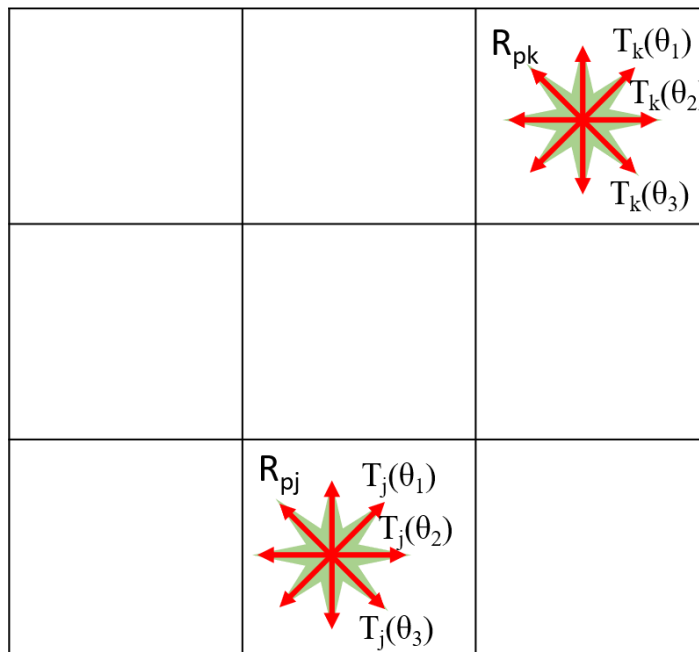


Figure 3.37 Illustration of the directional split at a pixel. The total time at direction θ_1 is the sum of time spent in all pixels j, k, \dots or $T_j(\theta_1) + T_k(\theta_1) + \dots$ where total spikes (average or predicted) at direction θ_1 is the sum of the rates at each pixels multiplied by the time at that pixel or $R_{pj}T_j(\theta_1) + R_{pk}T_k(\theta_1) + \dots$. The predicted rate is calculated by dividing the predicted spikes by total time.

The predicted firing rate (green) can be compared to the observed firing rate (blue) using the distributive ratio (DR) given by the following equation,

$$DR = \sum \left| \ln \left(\frac{1 + R_{obs}(\theta)}{1 + R_{pred}(\theta)} \right) \right| / N \quad (3.9)$$

$R_{obs}(\theta)$ is the firing rate of the unit with respect to head direction measured using the algorithm explained in **Figure 3.35**. N is the number of directional bins. DR is zero for a perfect prediction as the $R_{obs}(\theta)$ and $R_{pred}(\theta)$ are equal. The null hypothesis is true in such cases and the directional modulation is only attributable to the inhomogeneity of sampling, where the place-field firing is only modulated by locations. A higher value of DR indicates a poor prediction and, therefore, if there exists directional modulation, that accounts for some of the variability in the firing of the unit. **Figure 3.38** shows an example where the predicted firing rate resembles the observed or calculated firing rate.

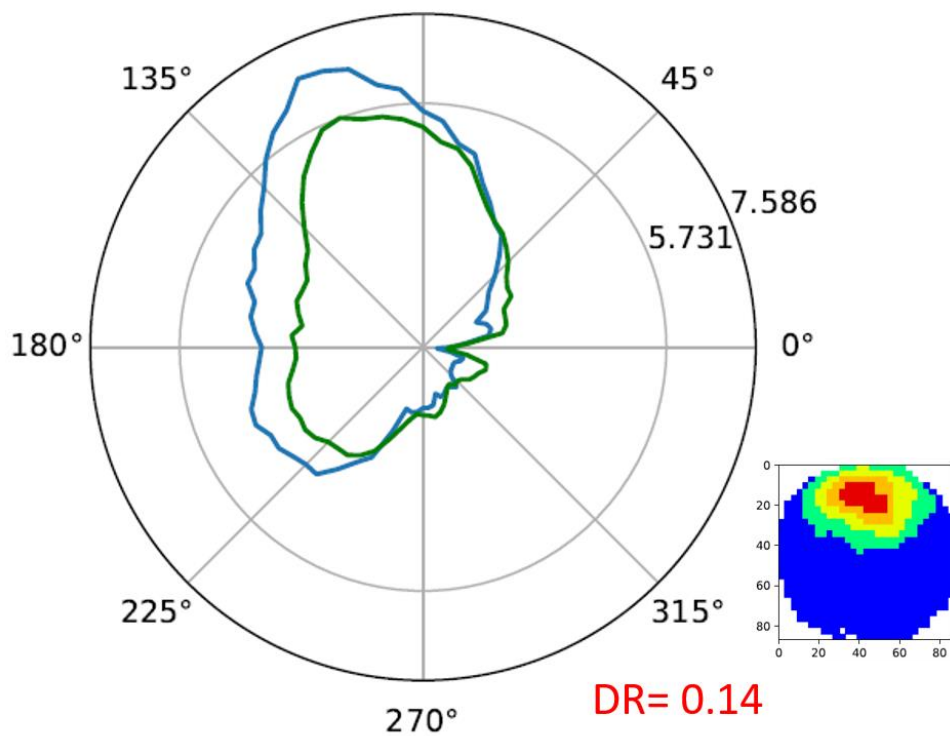


Figure 3.38 Sample place unit with apparent directional tuning recorded in a pellet chasing task in an open arena. The blue line shows that the unit preferentially fires along the north-northwest direction. Green line, the predicted rate, almost resembles the observed firing rate and, therefore, the DR is very small (0.14). Onset shows the behavior of the unit with respect to location. The place field is in the north-northeast wall of the cylindrical arena and may cause the unit apparently fires more along that direction. We can also conclude that there is an inhomogeneity in sampling of the head direction.

The analysis also provides a polar-scatter plot for the head-directions of each spike. The time of each spike is observed and paired to the corresponding head-direction. Such head-directions are then binned at 2° intervals, and each spike is represented by the coordinate (r, θ) where θ is the bin, and r is the incremental radius for each spike count in this bin. Examples plots are shown in **Figure 3.39**.

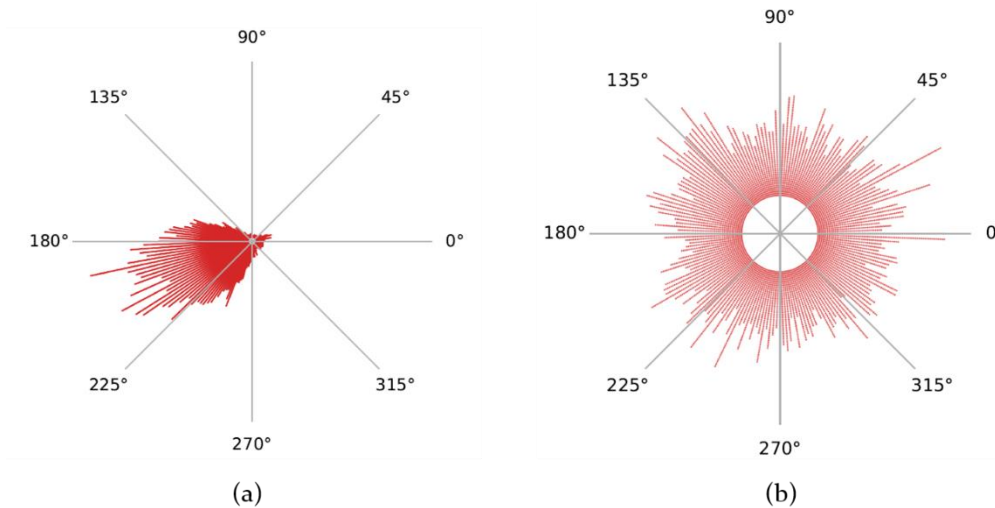


Figure 3.39 Scatter plot of the head direction corresponding to each spike of (a) the head directional unit with the firing rate shown in Figure 3.36 and (b) a non-head directional unit

Head directional shuffling analysis

Like the location shuffling analysis, spike-trains are shuffled, and a new tuning curve is created for each of the new spike-trains. The Rayleigh Z-score and the concentration parameter κ for the von Mises distribution are calculated for each tuning curve. This procedure is replicated for a specified number of shuffles and then binned to create a histogram. This is, as was argued in the shuffling analysis of locational firing map, based on the idea that if the spiking activity is independent of directional heading of the animal and the tuning curve implying a directional receptive field is a random observation, the specificity measures for the tuning curves, namely Rayleigh Z and von Mises parameter κ will be statistically insignificant. These measures are compared to the upper tail of the distribution with a 5% significance level. Sample output graphs are shown in **Figure 3.40**.

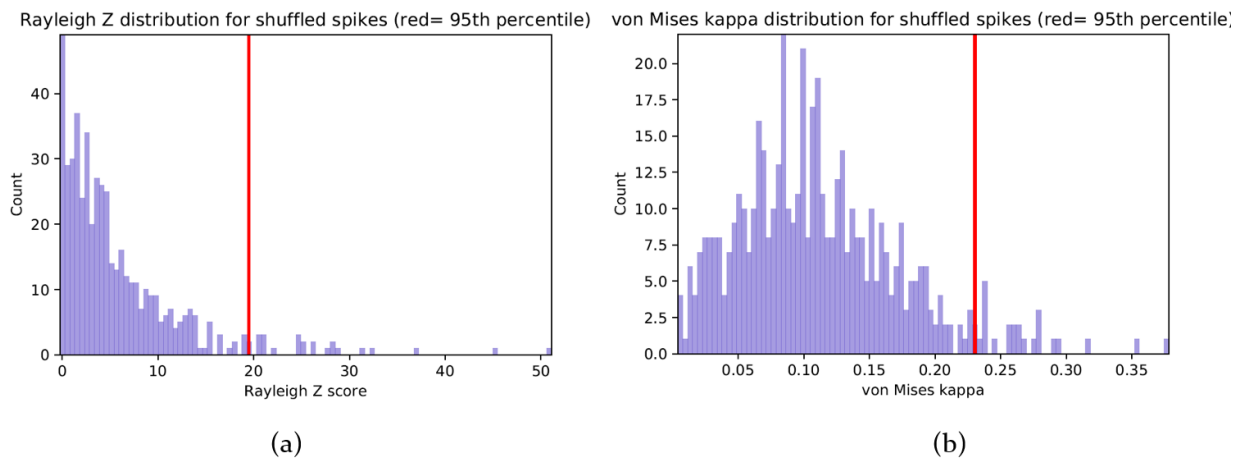


Figure 3.40 Sample output of head-directional shuffling analysis; (a) distribution of Rayleigh Z scores (b) distribution of the von Mises κ . The red line shows the 95th-percentile in the distribution which is then compared to the Rayleigh $Z=930$ and $\kappa=2.44$ to test for significance at 5% level.

Head directional time-lapse analysis

Like the locational time-lapse analysis, this analysis simply splits the entire recording sessions into smaller segments of creates tuning curves and directional spike-plot at following intervals:

Cumulative: 0-1min, 0-2min, 0-4min, 0-8min, 0-16min (if recording session is longer than 16min), 0-end of the recording

Progressive: 1-2min, 2-4min, 4-8min, 8-16min (if recording session is longer than 16min), 0-end of the recording.

A sample outcome for such an analysis is shown in **Figure 3.41**. This analysis is effective in describing a unit whether it is responding to the animal's head direction from the very beginning of the session, or first exposure to an environment, or it gradually emerges as the animal forages the environment. In short, it provides a picture of temporal evolution of the firing of a unit in an environment (Jankowski et al., 2014).

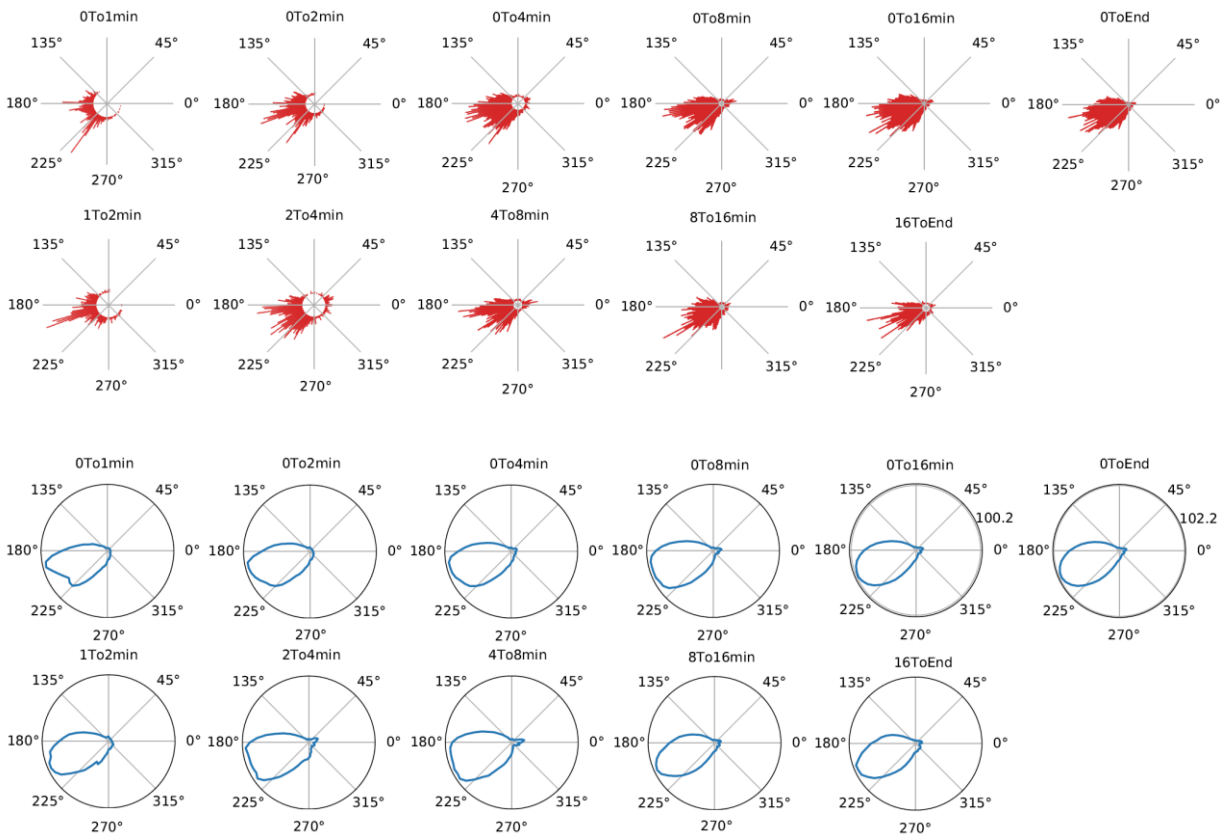


Figure 3.41 The time-lapse directional spike-plot (top two rows) and tuning curves (bottom two rows). The cumulative time-lapse are 0-1min, 0-2min, 0-4min, 0-8min, 0-16min, 0-end of the recording. The progressive time-lapse are 1-2min, 2-4min, 4-8min, 8-16min, 0-end of the recording.

Head directional time shift analysis

A series of time-shifted tuning curves are created for user-specified number of shifts in time with respect to the video samples. The algorithm remains as same as the tuning curve calculated for the original spike-train of the unit. Only the spike-train has been shifted by a multiple of location timestamps to the left (-ve shift) or to the right (+ve shift) to pair with past (-ve shift), present (no shift) or future (+ve shift) locations of the animal. This is illustrated in **Figure 3.42**. The Skaggs information content, peak firing rate and separation angle δ are calculated for each tuning curve and plotted against the amount of time spike-train are shifted (**Figure 3.43**). The δ -values are fitted with a straight line and the shifted time where $\delta=0$ is defined as the anticipatory time interval (Blair and Sharp, 1995). This is the time required for the CW vs CCW tuning curves to overlap with each other (like the ID place field explained in **Figure 3.24**). The information content and the peak firing rate is also expected to be maximum around this time interval.

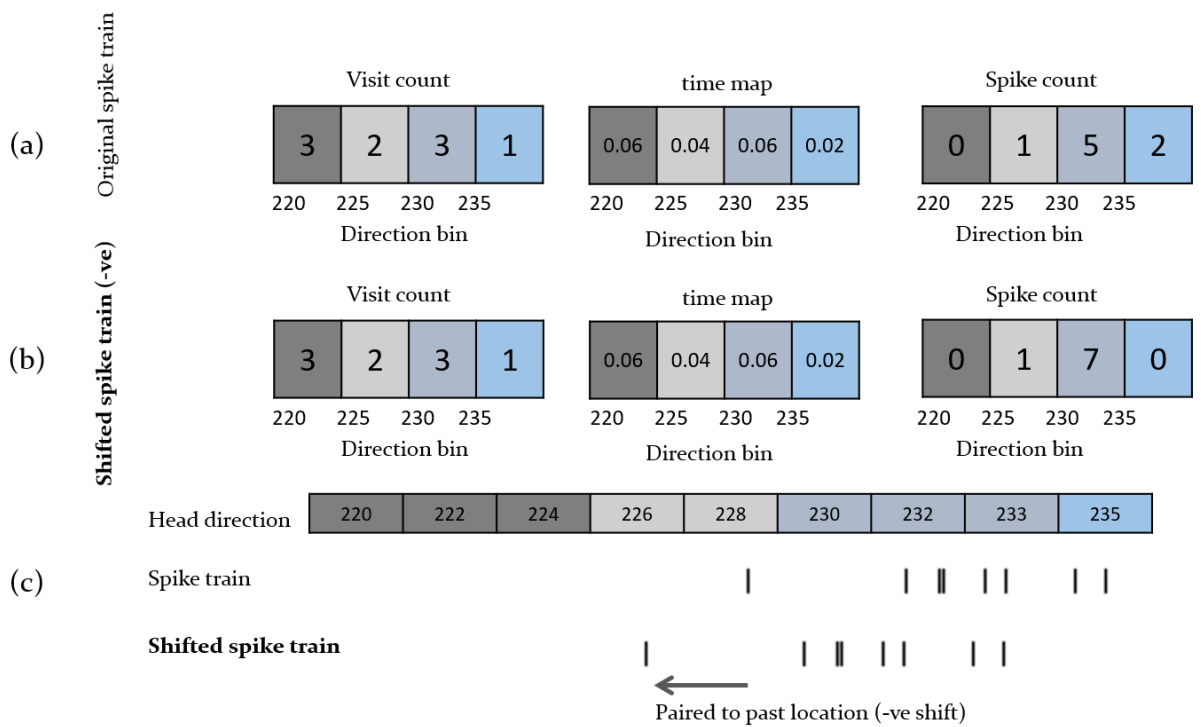


Figure 3.42 Construction of a tuning curve from the spike train of a unit and its shifted spike train; (a) Visit count, time map, and spike-count map for the unchanged spike-train of the unit; (b) Maps similar to (a), but with a shift in spike train as shown in (c); (c) top row shows the head direction, middle row depicts the spike train, and the bottom row shows the spike train when it is shifted to match with a past location of the animal. Directions falling into different bins are color coded. As we can see, the count in spike changes in the shifted spike train, and a new rate is formed out of it.

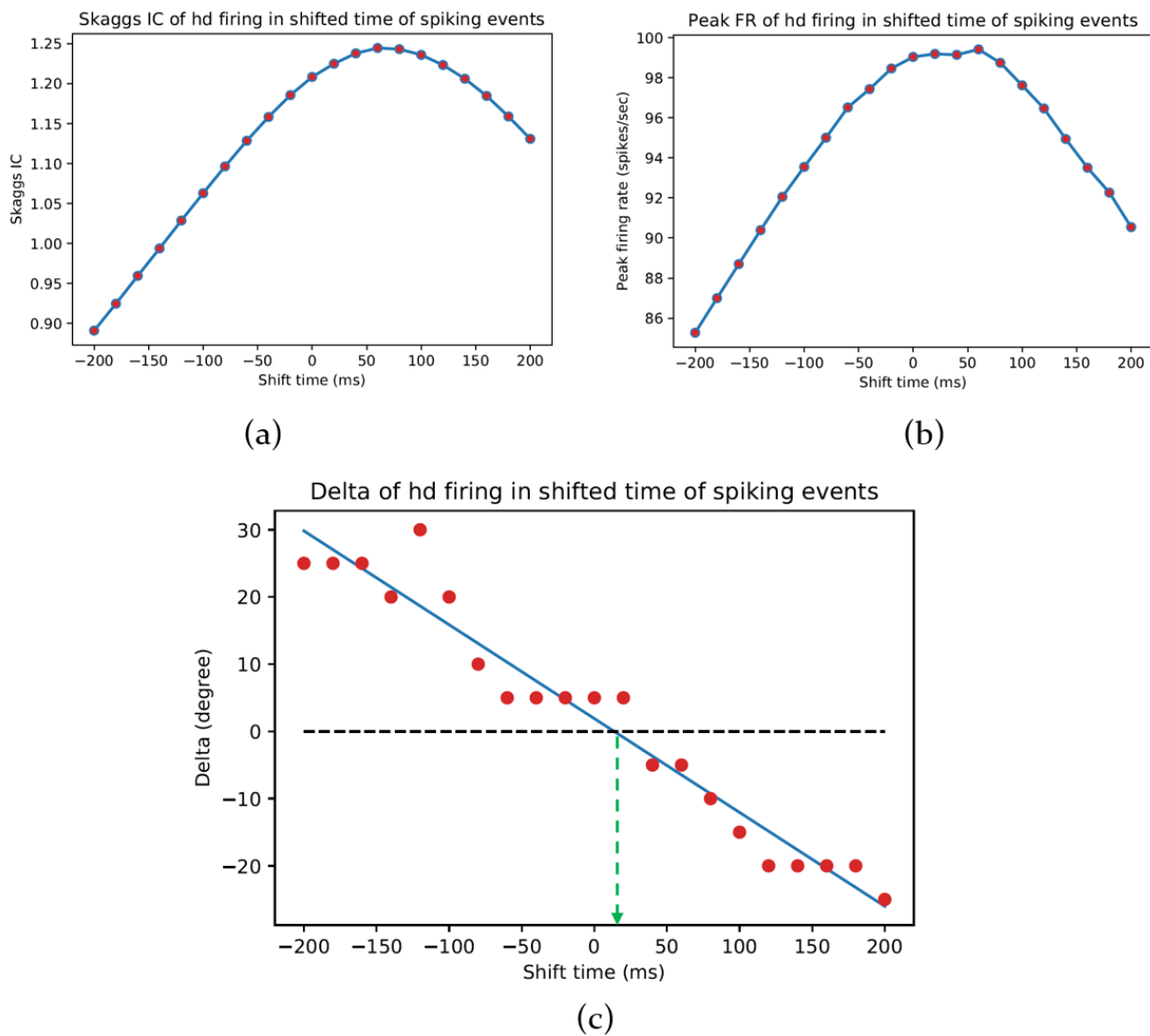


Figure 3.43 Sample results of the head-direction time-shift analysis for a unit that has a directional firing field; (a) The Skaggs information content gradually maximizes towards the preferred direction and declines afterwards as expected for a unit with directional firing field. It maximizes at a +ve time shift, which implies that the unit is anticipating a preferred direction. (b) Similar results can be obtained by using the peak firing rate of the tuning curves at each shifted time. (c) Plot of separation angle δ vs shift of time. Separation of the CW vs CCW curves are expected to be zero at the preferred direction. Therefore, the anticipatory time interval is measured as the shift of time where the fitted δ line intersects x-axis (intersection of the blue and black lines).

3.3.4 Non-spatial navigational information

Spike rate as a function of running speed of the animal

The raw speed data, as provided by the recording system, is smoothed using a moving average filter with rectangular window of user-specified samples, i.e. sample n_t is being replaced by $(n_{t-2} + n_{t-1} + n_t + n_{t+1} + n_{t+2})/5$ for a 5-sample averaging. The histogram counts of the speeds, and hence the number of times the rat moved with certain

speed, are binned up to the maximum speed the user specifies for the analysis. The scheme for this analysis is exactly as same as the calculation for head directional tuning curves. The number of spikes corresponding to the speed for each position data sample is counted. The spike rate is finally calculated by dividing the total number of spikes belonging to each running speed bin by the total time the animal moved at that speed. The Skaggs information content were calculated using the travelling time & spike rate at speed bin and mean firing rate of the cell using the **Equation 3.4** specified in **Section 3.3.2**. Speed bins at which animals did not travel for at least 1sec were then excluded. The remaining raw data are then fitted to a linear regression equation. The quality of the linear fit is assessed using the Pearson's correlation coefficient between the raw spike rate vs the fitted rate. A sample outcome is shown in **Figure 3.44** for a unit that has a strong linear correlation to the speed with its spiking rate.

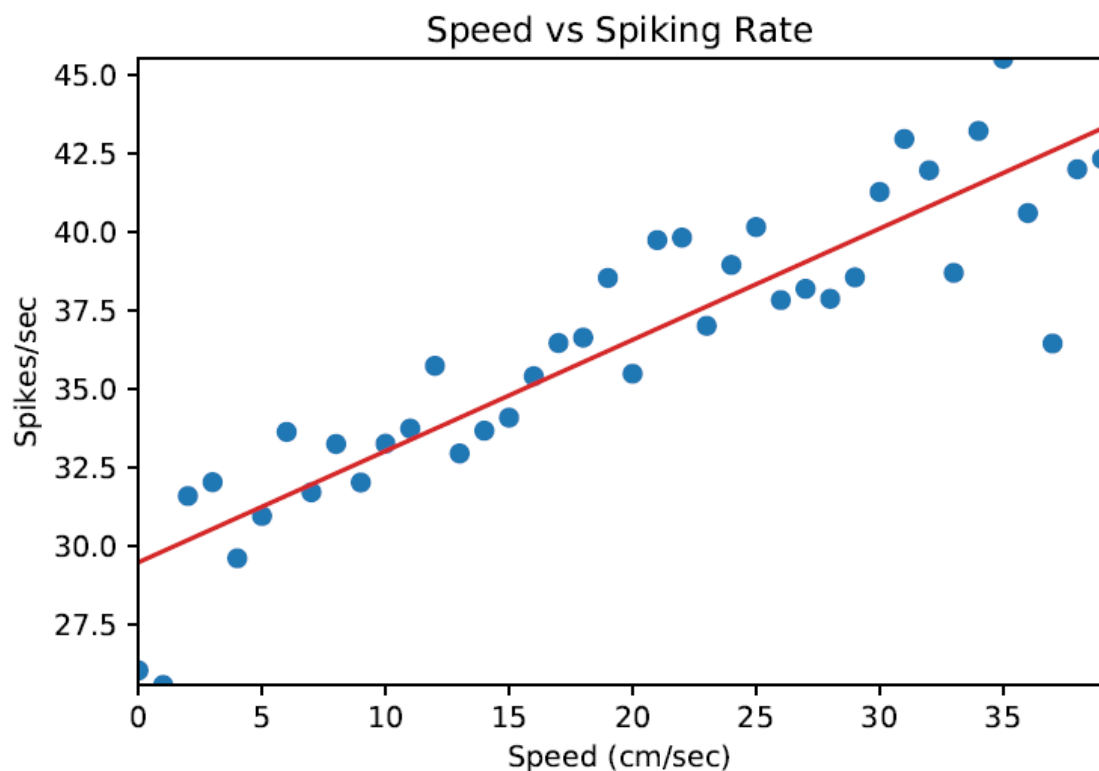


Figure 3.44 Graphical output of a unit with a high correlation its firing rate to the speed of the animal (Pearson's $r= 0.9$, $p< 0.001$)

Spike rate as a function of the angular head velocity of the animal

To calculate the angular head velocity, head directions are first smoothed using a moving circular mean filter over a rectangular window with specified size. The angular head velocity (AHV) at time t_i was then calculated as the slope of the linear regression line passing through head direction at t_{i-2} to t_{i+2} time samples. The -ve and +ve angular head velocities were taken as the clockwise (CW) & counter-clockwise (CCW) angular velocities respectively, as the clockwise movement decrease the head-direction in the reference plane (**Figure 3.45a**). For each of these groups, the spike rate was calculated using a similar approach to that of the running speed. The Skaggs information were calculated taking CW & CCW rates together (the rate and time spent at each bin spanning both the -ve and +ve bins). High AHVs are always preceded by passing through the range of lower AHVs creating an inherent sampling bias in lower AHVs and resulting in lower samples and higher variance at high AHVs (Bassett and Taube, 2001). To minimize this bias, bins with trailing zeros at higher angular velocities and those not visited for at least one second were removed. The lower AHVs are also excluded below a cut-off amplitude, e.g. $10^\circ/\text{sec}$. Both the CW & CCW spike rates were separately fitted to a linear equation, and the quality of the fit were assessed using the Pearson's correlation coefficient between the fitted spike rate and the raw rate. A sample output is shown in **Figure 3.45b**.

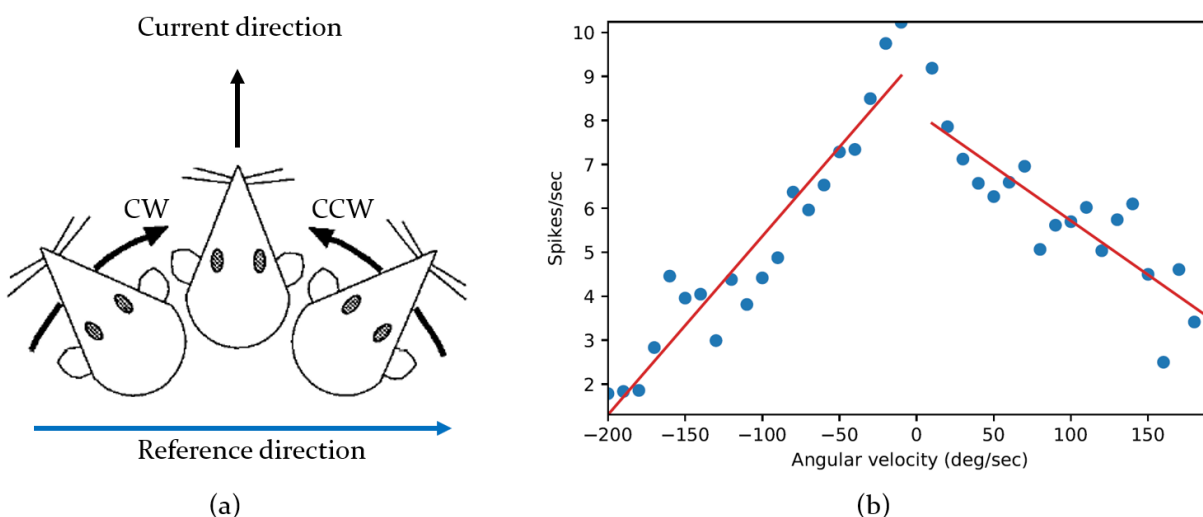


Figure 3.45 Angular head velocity analysis; (a) Concept of clockwise (CW) and counter-clockwise (CCW) head directional movement (Adapted from Blair and Sharp, 1995); (b) Sample firing rate of a head direction cell that fires at a decreasing rate as the magnitude of AHV increases, but does not depend on the directionality of the head movement.

3.3.5 Multivariable analyses

Multiple regression analysis

Multiple regression or MR (linear least square regression) R^2 values measure how well regression line approximates the real data points. In NeuroChaT, MR represents the instantaneous spike-rate as a linear combination of environmental variables. The variables under consideration are location, running speed, angular head velocity, head direction of the animal and its distance from the border in the arena. As the spike rate of the unit is non-linear in nature for head-direction, place and border cells, we use the firing rate of the animal at corresponding binned elements to be representative of the value in the MR. The entire recording is subsampled into a specified interval, e.g. 0.1ms, by taking the intermediate values between samples, and a specified chunk of samples is taken randomly with replacement (bootstrapping) for a, for example, 120sample episode which, along with 10Hz new sampling rate, turns into 120 data samples under consideration. This procedure is replicated for many times, i.e. 1000 replications, and multiple regression is estimated using ordinary least square for each such randomly sampled chunk. The semipartial correlation coefficient for each independent variable is calculated for each multiple-regression estimate. The mean and standard deviation of all semipartial correlations are presented in the graph as shown in **Figure 3.46**,

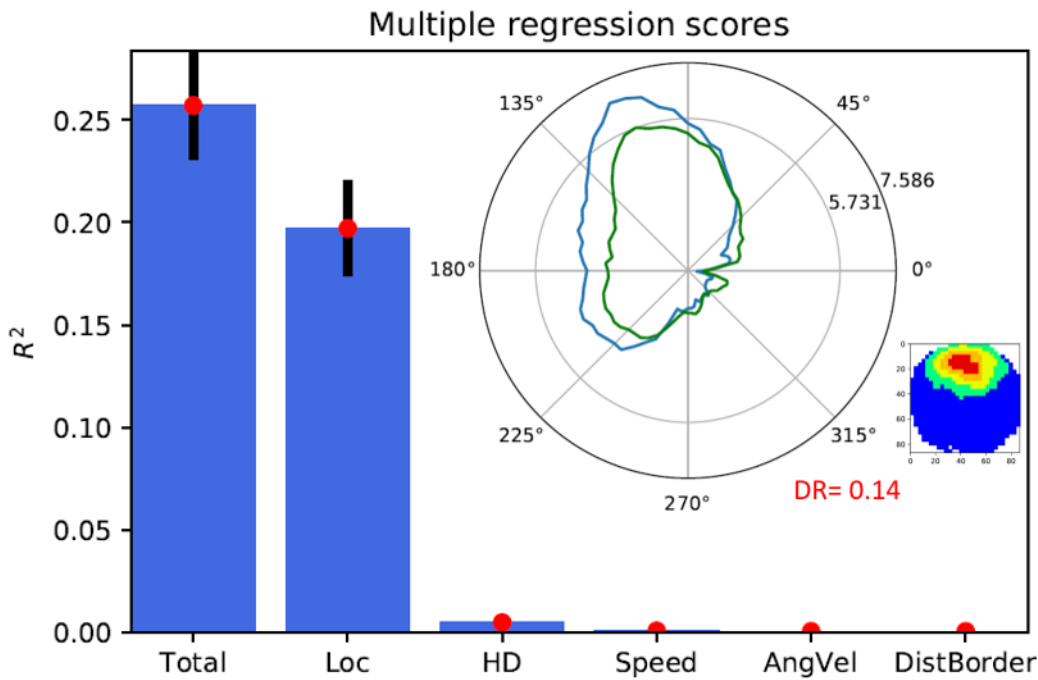


Figure 3.46 Multiple regression correlation parameters (mean \pm std) for different independent variables. The inset shows the apparent directional preference (blue line) for a unit with locational firing preference. As expected, the semipartial correlation dominantly explains the variability in spiking activity, as the directional preference is an artefact explained by the green line- the predicted firing rate- which is very similar to the observed rate (DR= 0.14)

The original version of the MR analysis assumed that the locational and directional firing patterns tend to be idiosyncratic, so that each unit that codes direction has its own unique preferred direction and, each unit that signals location has its own preferred location (Sharp, 1996). Therefore, the point of the analysis was to ask if the unit showed a reliable locally graded influence of either of these variables so that information about the unit's own average firing in a given range of directional headings or spatial locations would assist in prediction of the unit's momentary rate. We extended the concept to border cell as well. But for two other variables, namely speed and AHV, we are assuming a linear relationship between the firing rate and the variable values.

Interdependence analysis

The distributive hypothesis is explained in **Section 3.3.3** for location and head direction of the animal as two potential contributing sources of firing rates. And the predicted firing rate is compared to observed firing rate to obtain the DR value, which explains further whether the unit is preferentially firing signaling animal's location

where the effect of head directional modulation is an artefact due to physical constraints, and vice versa (Cacucci et al., 2004). In this analysis, we extended the concept for other independent variables, namely speed, AHV and distance of the animal from the border, as they can also be constrained and may represent the locational firing as the response to these variables, i.e. place field in the border etc. The split of each of the variables into respective bins and the calculation of predicted firing rates followed the same equations as that of directional information. The analysis provides parametric output showing the DR for each variable.

3.3.6 Local field potential

LFP frequency spectrum

The raw LFP signal is passed through a bidirectional zero-phase shift Butterworth filter. The user can specify the cut-off frequencies, filter order etc. for the filter. This signal is then considered for two different spectral analyses-

- A standard power spectral density obtained by the Welch's method- dividing the data into overlapping segments of specified size, called window size (e.g. 1sec), with specified amount of overlap (e.g., 0.5sec) and computing modified periodogram for each segment and averaging the periodogram. The resulting output provides a mean for quick screening of the power distribution in the entire signal (**Figure 3.47a**).
- A time resolved spectrogram, where spectrogram was calculated for each moving window of, for example, 1 sec, with consecutive windows having overlap of specified length (e.g., 0.5sec). This is also called short-term Fourier Transform or STFT (Oppenheim and Schaffer, 2009).

In both the case, the output graphic is band limited to a specified frequency, say 40Hz, and the STFT is converted into decibel scale (**Figure 3.47b**)

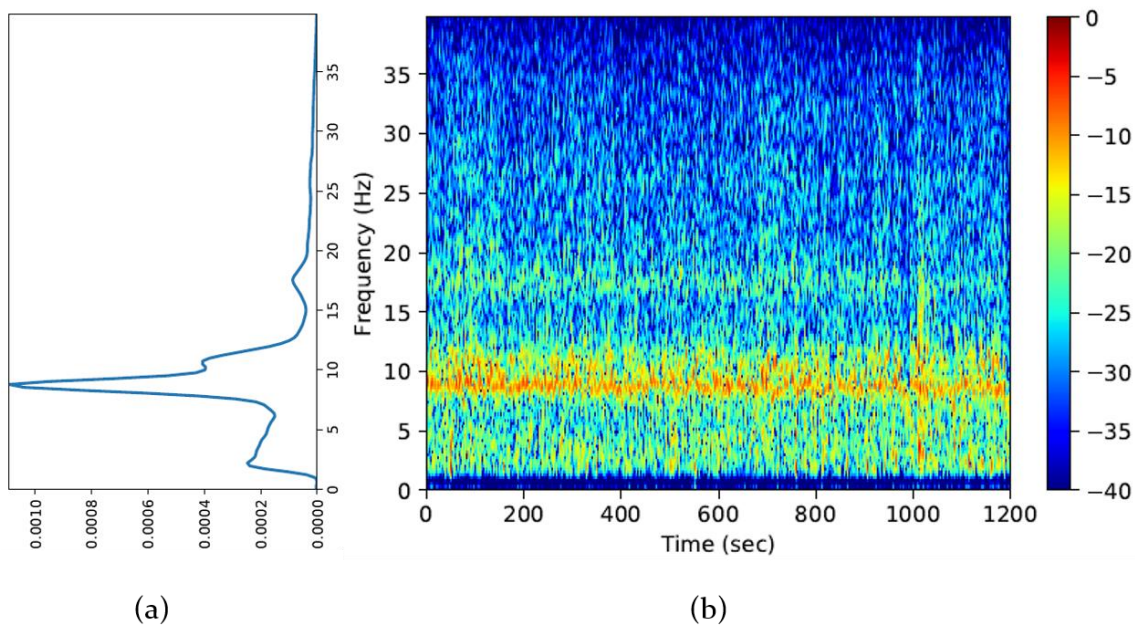


Figure 3.47 Result of the LFP analysis; (a) The PSD ($\mu\text{V}^2/\text{Hz}$) calculated using the Welch's method; (b) The power spectrogram of the LFP band-limited to 40Hz and converted into decibel in this example. It is apparent that there are stronger theta components as it peaks at around 8Hz and 10Hz.

3.3.7 Single units to local field potential phase synchrony

Unit to LFP phase distribution

The algorithm for determining the LFP phases within a band $[f_{min}, f_{max}]$ at spiking-events is shown in **Figure 3.48**. After the pre-filtering to remove the effects of non-stationarity due to trends and drifts, the raw LFP signal (L) is bandpass filtered with a fifth order bidirectional zero-phase Butterworth filter within $[f_{min}, f_{max}]$ to obtain the band signal (B) (**Figure 3.49a**). The amplitude and phase of the B signal are obtained using its Hilbert transform. We first divided the entire L and B signals into segments to explore the portion of the filtered signal with significant strength. The segments were taken between the points where the amplitude of the B crossed the 50% of the mean amplitude taken from the signal envelope (**Figure 3.49b**). A segment is merged with the next segment until the newly formed segment is longer than the temporal length corresponding to the lower cut off frequency of the filter, or $1/f_{min}$ Hz. The power spectrum of the corresponding L segment is obtained through Fast Fourier Transform (**Figure 3.49c**). If the power within $[f_{min}, f_{max}]$ Hz is at least *pratio* (specified by the user, i.e. 15%) times of the total LFP power of the segment, and the peak-to-

peak LFP amplitude within that segment is at least *aratio* times (user-specified, i.e., 25%) of that of the entire LFP, then this segment was regarded to contain significantly strong filtered band signal. The phase of all spikes lying within such acceptable segments of the filtered signal were taken by interpolating the phase calculated using Hilbert transform at the timestamp of each spiking event (**Figure 3.49d**). The phases are then binned, and the raster and histogram of phases are calculated (**Figure 3.49e and f**). To determine if the spikes were phase-locked, NeuroChaT further assesses circular statistics of the spike-phase distribution using Rayleigh's Z-test (**Appendix F**). The mean phase and von Mises distribution parameter, κ , are also determined (Fisher, 1993). The description of the input and output parameters are explained in **Appendix C** and **Appendix D** respectively.

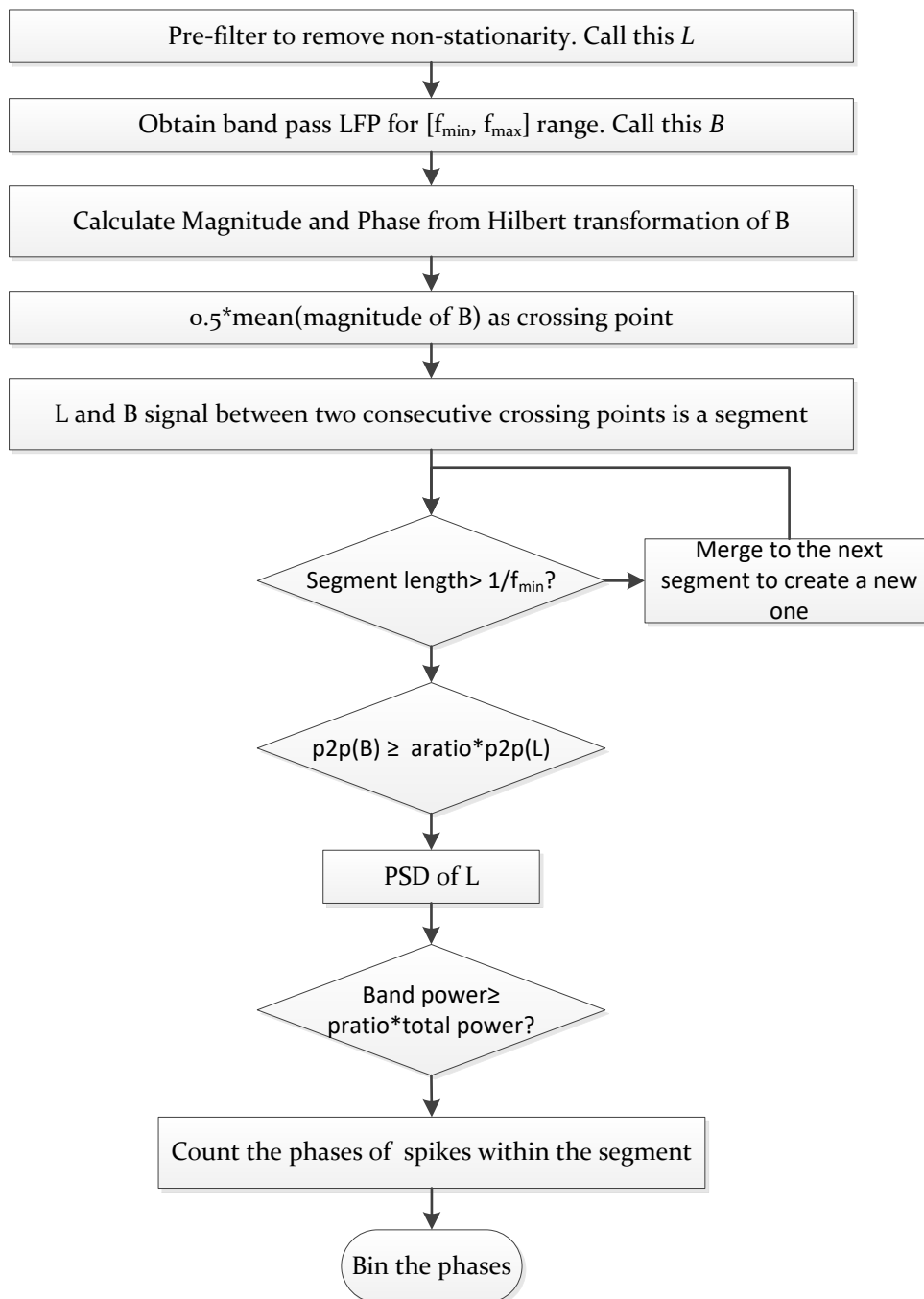


Figure 3.48 Algorithm to obtain the phases of a unit locked to the LFP signal.

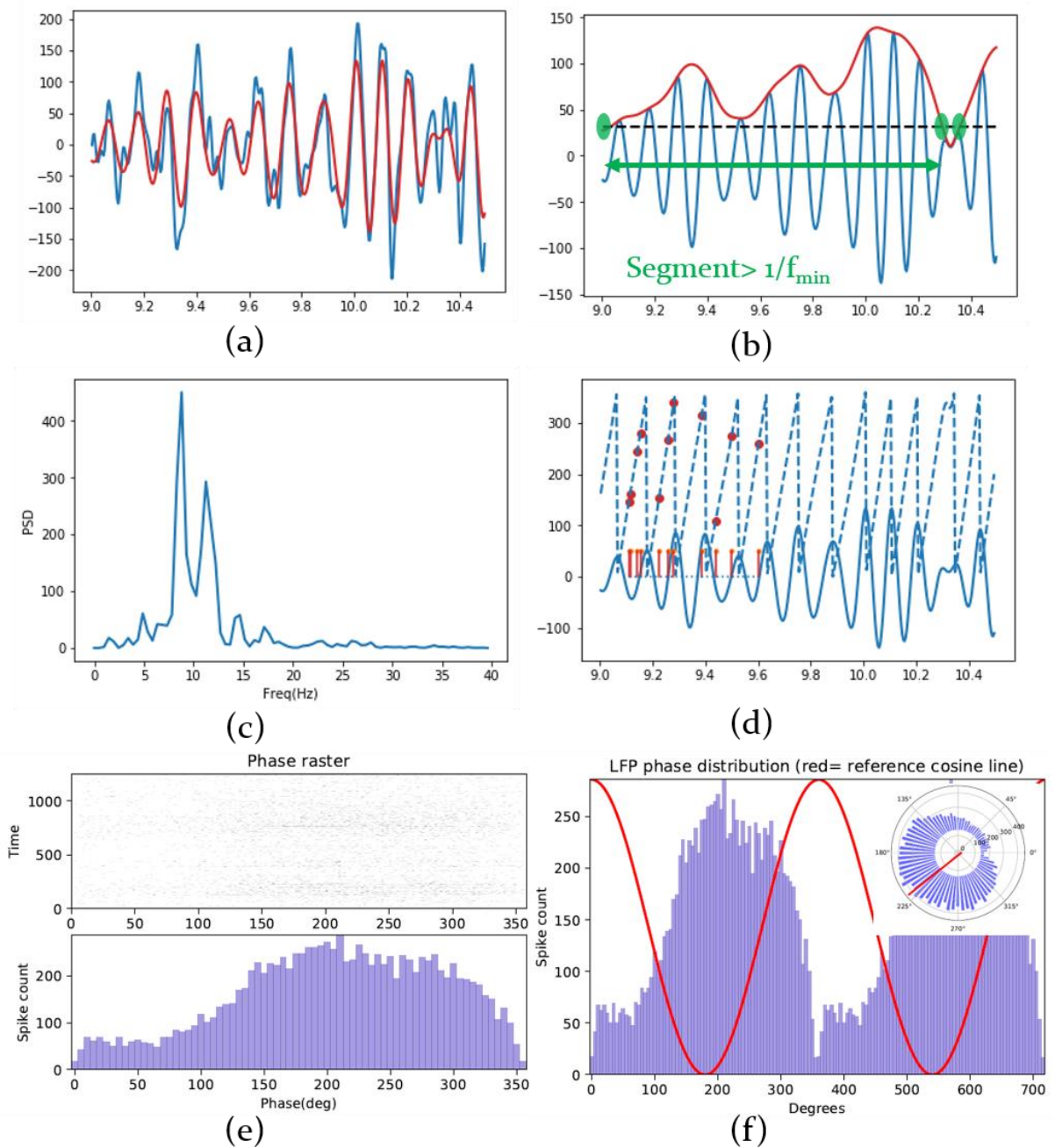


Figure 3.49 Stepwise illustration of calculating the spike to LFP phase locking within a particular band of LFP signal in the frequency range $[f_{min}, f_{max}]$; (a) The blue line shows the high-pass filtered LFP signal L . Red line shows the band-pass filtered signal B ; (b) Blue line is the B signal. Red line shows the amplitude of the signal derived from Hilbert transform. Black dotted line is the crossing-line to determine segments for further analysis. Green dots are crossing points, where green arrow shows a segment that is longer than $1/f_{min}$ or the minimum temporal length to represent the slowest component of the B signal; (c) Power spectral density measurement for the L signal. The power within the B -band is sum of the power between $[f_{min}, f_{max}]$ and must be greater than a set percentage (pratio) of the total power; (d) Once the segment is verified to carry significant band-power and having peak-to-peak (p2p) amplitude comparable to the overall p2p amplitude of the L signal, phases of the spikes within that segment (red stems) are calculated from the phases (blue dotted lines) of the B signal using Hilbert transform (red dots); (e) The raster (top row) shows the phases of the spikes at each segment (y-axis), and the bottom row shows the counts at different phase-angles; (f) The bottom row in (e) is shown along with a cosine curve to distinguish the portion of the distribution with phase-preference. The inset in the figure shows a circular plot of the same distribution along with the mean phase (red line).

Unit to LFP phase locking

NeuroChaT includes another approach to determine the phase-locking of a unit to the LFP signals using the phase-locking value (*PLV*) (van Wingerden et al., 2010) and spike-field coherence (*SFC*) measures (Rutishauser et al., 2010). The algorithm for such measurements is shown in **Figure 3.50** and illustrated in **Figure 3.51**. A segment of the LFP is cut out, called spike-triggered LFP, for a window of, for example, [-0.5, 0.5] sec centered around the time of the spiking events of the unit (**Figure 3.51a**). Individual traces are averaged to obtain the spike-triggered average, or *STA* (**Figure 3.51c**). Each of the traces is multiplied by a hanning window and the Fourier transformations are obtained (**Figure 3.51b**). The average of the power spectrums provides the spike-triggered power spectrum, or *STP* (**Figure 3.51d**). The power spectrum of the *STA*, *fSTA*, is also calculated. The spike-field coherence is obtained by,

$$SFC(f) = \left[\frac{fSTA(f)}{STP(f)} \right] \times 100\% \quad (3.10)$$

where *PLV* is obtained by normalizing each frequency spectrum and taking the resultant vector length which is given by the following equation,

$$PLV(f) = \left| \frac{1}{N} \sum_{i=1}^N \frac{X_i(f)}{|X_i(f)|} \right| \quad (3.11)$$

where $X_i(f)$ represents the frequency spectrum of the LFP trace corresponding to the i -th spike. N is the total number of spikes in the unit. Sample analysis outcome are shown in **Figure 3.52**.

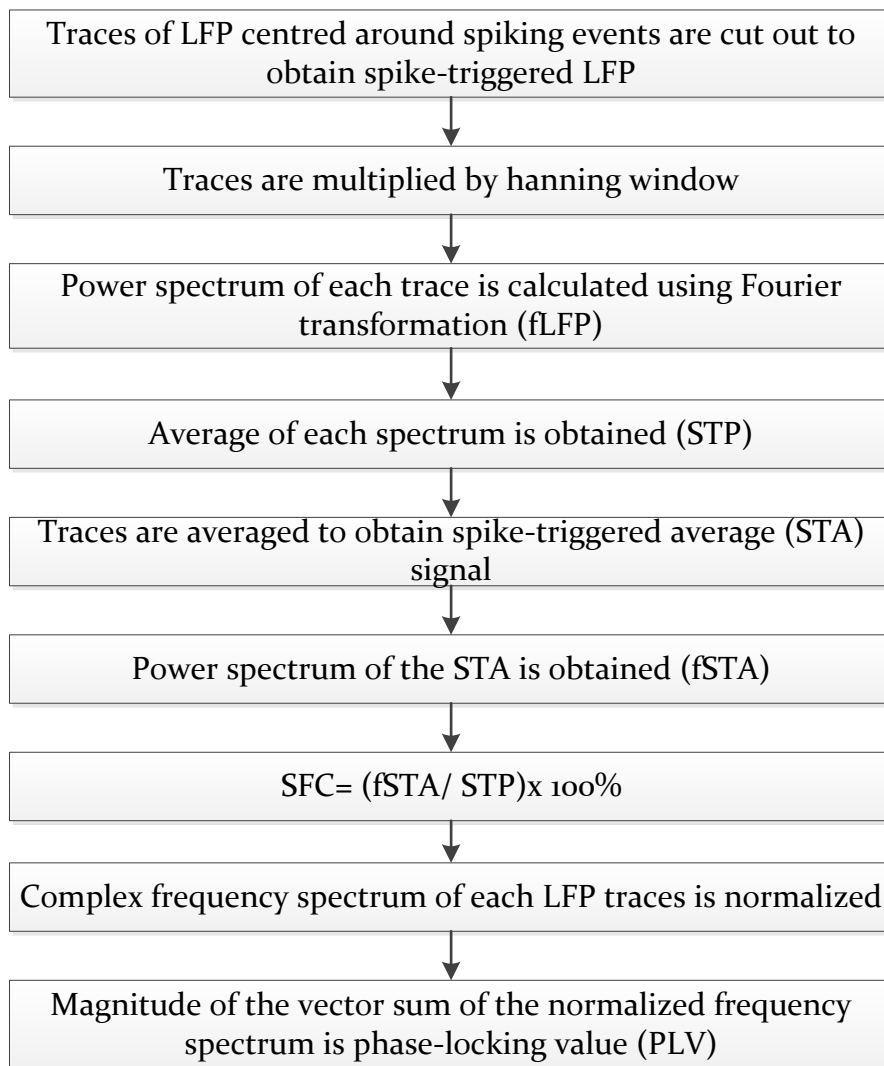


Figure 3.50 Flow chart for calculation of spike-field coherence (SFC) and phase-locking value (PLV)

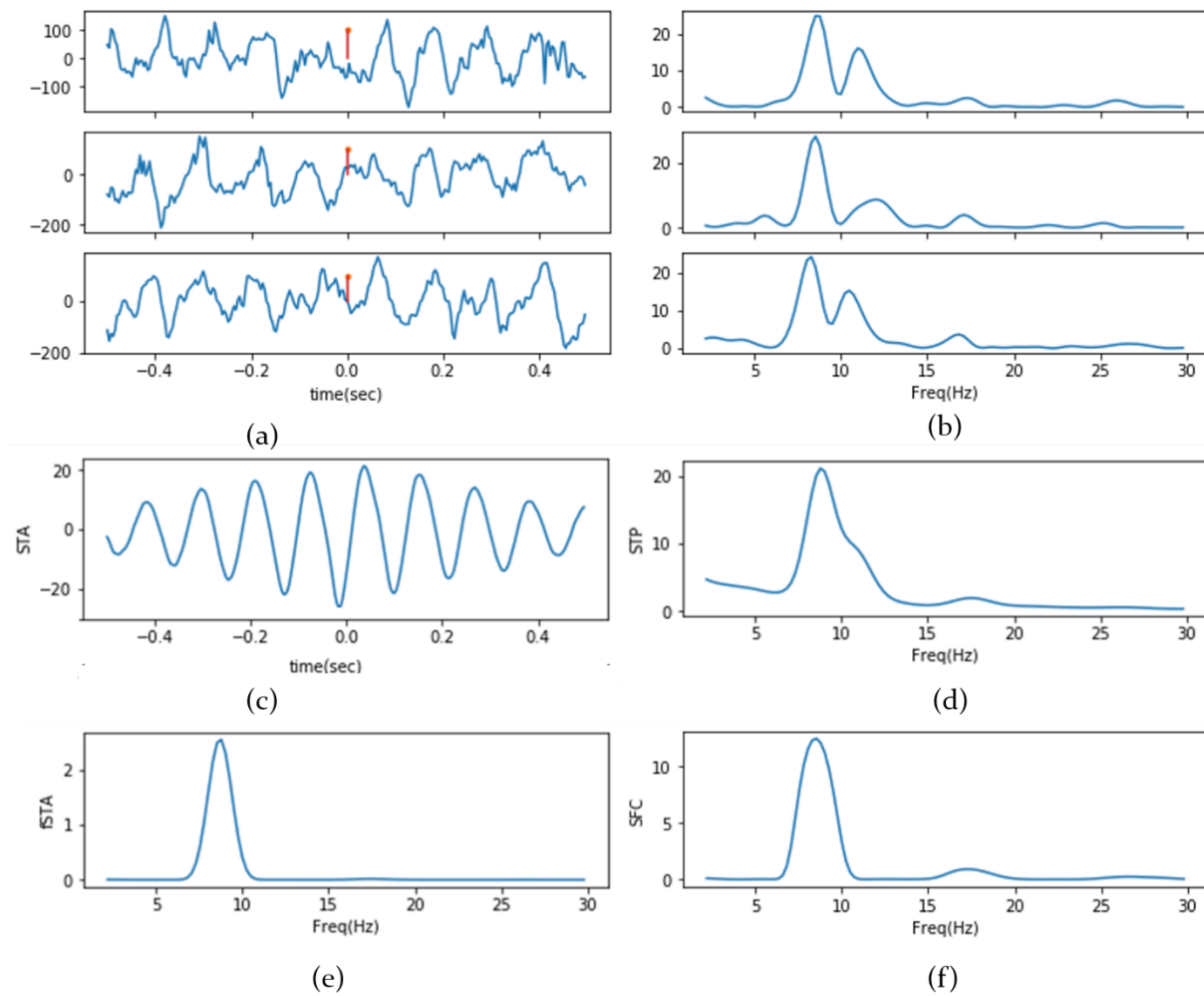


Figure 3.51 Stepwise illustration of measuring the Phase-locking metrics; (a) A segment of the LFP is cut out, called spike-triggered LFP, for a window of, for example, $[-0.5, 0.5]$ sec centered around the spiking time. The red stems show the spikes and the blue traces are the LFP segments; (b) Each segment is multiplied by a hanning window and the Fourier transformations are obtained. These graphs show the power spectral density (PSD) obtained from the Fourier transform; (c) The LFP segments are averaged to obtain spike-triggered average (STA); (d) Power spectrum of the segments are averaged to obtain the average spectrum (STP); (e) The Power spectrum of the STA; (f) The spike-field coherence (SFC) measured from fSTA and STP.

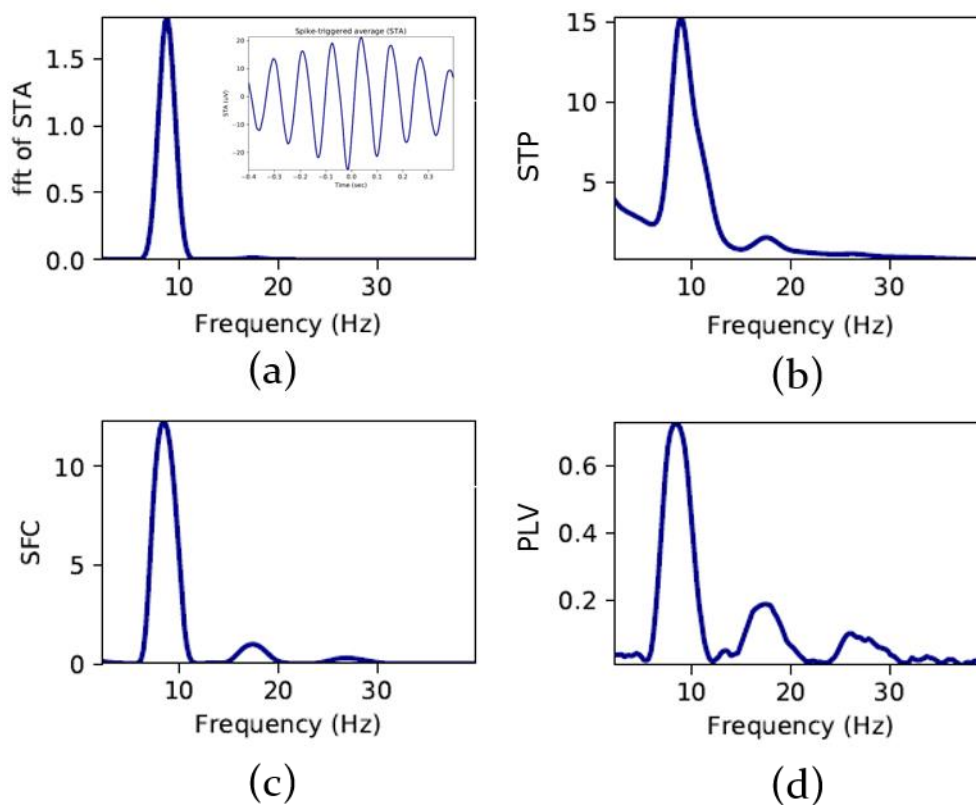


Figure 3.52 Results of different phase-lock metrics; (a) Power spectrum of the spike-triggered average. Inset shows the STA; (b) Average of the power spectrum of individual LFP traces; (c) Spike-field coherence; (d) Phase-locked value. All these graphs show that there is a strong coupling of the spiking activity to the underlying LFP signal at 9Hz.

We further extended the analysis for measuring the time-resolved phase-lock metrics. The window of the LFP traces are offset by a series of time-shifts in a step of 25ms. *SFC* and *PLV* are measured for each shifted LFP traces according to the algorithm explained above. Sample outcomes for such analysis are shown in **Figure 3.53**. Such analyses not only provide the information about the lock to the phases of different frequency components, but also indicates the temporal dynamics between the spikes and LFP from the shifts for maximum locking. Therefore, they can be a good indicator of causal relationships between the spikes and the LFP signals.

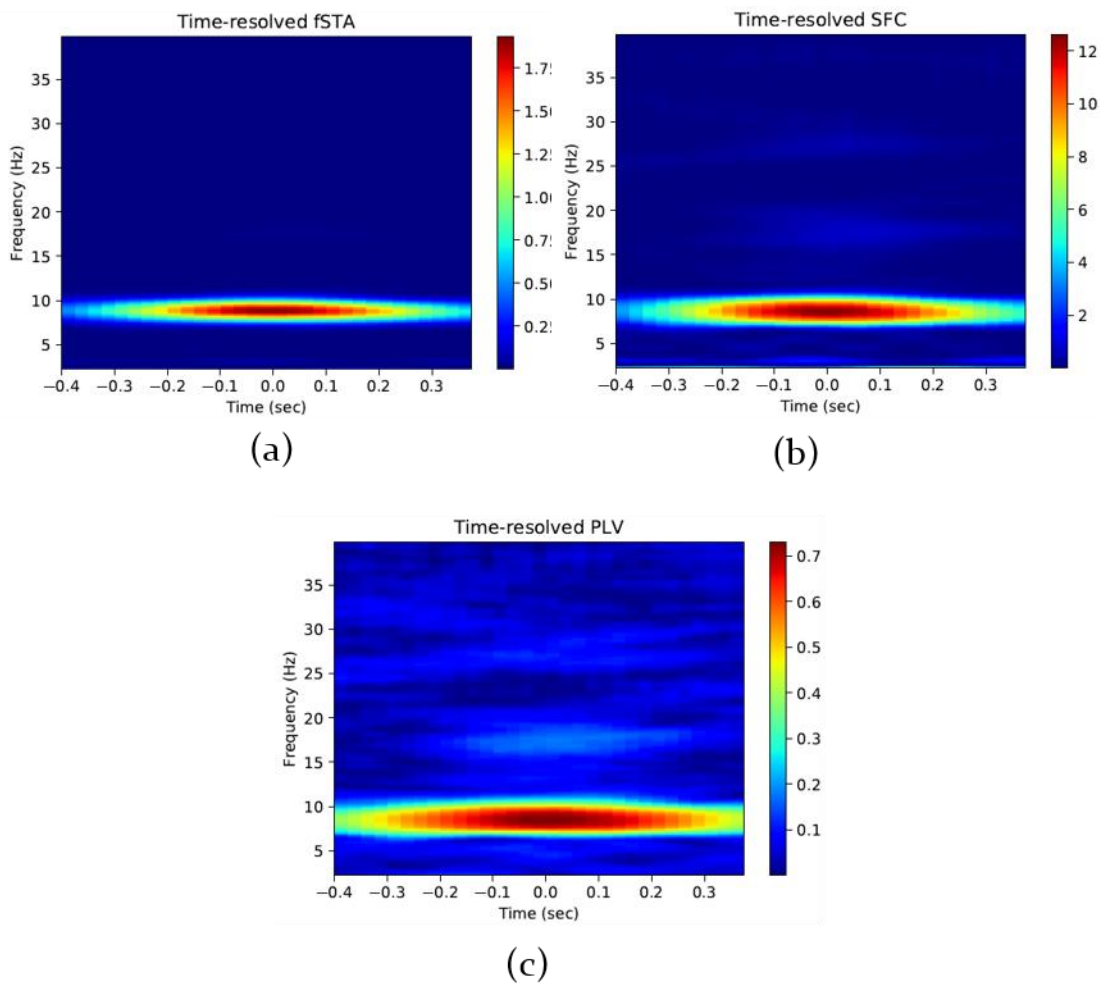


Figure 3.53 Time-resolved phase-lock metrics; (a) Power spectrum of the spike-triggered average; (b) The spike-field coherence; (c) The phase-locking value;

Given that the more spikes we use to obtain our statistical estimate of the *PLV* using the resultant length equation, the more reliable it becomes (van Wingerden et al., 2010). Therefore, the *PLV* itself is a biased measure with respect to the number of spikes for the unit under consideration. To compare across different experimental conditions or to reduce the statistical variances between units, NeuroChaT also estimates the phase-locking values using the bootstrapping procedure. A pool of a fixed, i.e. 100, number of spiking-events is drawn, and the phase-locking metrics are estimated for this group of spikes. The procedure is replicated for a user-specified number of time and the mean and standard error of mean for each of the metrics are calculated. A sample outcome is shown in **Figure 3.54**.

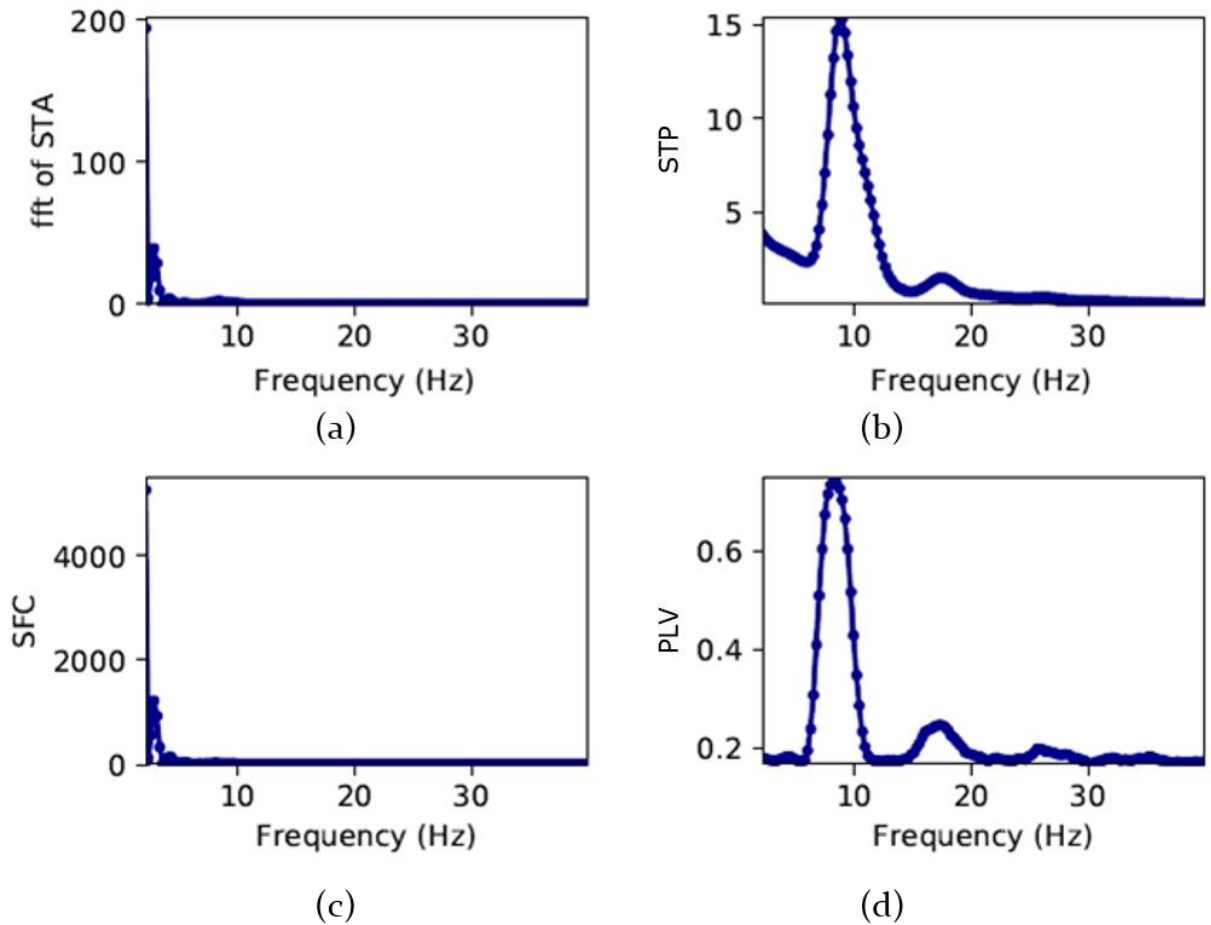


Figure 3.54 Phase-lock metrics using bootstrap technique; The mean (dotted)±sem of the (a) power spectrum of the spike-triggered average; (b) average of the power spectrum of individual traces; (c) spike-field coherence; (d) phase-locking value; In this particular example, the sem values are very small, and their shaded plots are not recognizable.

3.4 Data management made easy

NeuroChaT provides for better data management by standardizing the nomenclature in its output data. It also reduces efforts and attention by facilitating batch mode processing and activity logs and verifying the specification of the data before NeuroChaT starts analyzing them. It facilitates the management of output graphics in an easy to use approach. It also facilitates the data integrity, accessibility, and acceptability by converting the proprietary format data into a standardized data format.

3.4.1 Nomenclature

NeuroChaT creates a unique name for each unit of a recording session using the following format:

`unit_id = record_id+ 'TT'+ tet_no+ '_SS_'+ unit_no + '_' + eeg_file_ext`

`record_id`= Unique file or folder identifier for each recording session that are used to store the individual data and identify them properly.

`tet_no`= Tetrode number where the unit is identified.

`unit_no`= Tag of the unit or the cluster number in spike-sorting.

`eeg_file_ext`= Filename or the extension used for naming an LFP data file.

Name conventions in the supported data formats of NeuroChaT are shown in **Table 3.3**. Each HDF5 file represents one recording session, and the data for units, LFP, and spatial behavior are organized in folder-like paths. The HDF5 data format used in NeuroChaT is further explained in **Section 3.4.5**.

Table 3.3 Naming conventions of data formats supported in NeuroChaT.

Recording system	Spike File	LFP File	Spatial File
Axona	(record_id).(tet_no), e.g., 120412_1.1. Here, 120412_1 is the recording id, and 1 is the tetrode number where the unit is recorded	(record_id).(eeg_file_ext), e.g., 120412_1.eeg8 representing LFP data recorded in channel 8 of the recording system	(record_id)_x.txt, e.g., 120412_1_1.txt, where x can represent any of the tetrode number in that session.
Neuralynx	~/record_id/TT(tet_no).ntt, e.g., data in tetrode 8 for the	~/record_id/CSC(tet_no).ncs, e.g., LFP data recorded in channel 1 in	~/record_d/VT(x).nvt, e.g., where x can be any user specified entity, but normally it

	recording session 120412_1 are saved in ~/120412_1/TT8.ntt	recording session 120412_1 are saved in ~/120412_1/CSC1.ncs	is set to 1. So, the spatial data resides in ~/120412_1/VT1.nvt
NWB/ HDF5	(record_id).hdf5 + path within file= /processing/Shank/tet _no	(record_id).hdf5 + path within file= /processing/Neural Continuous/LFP/lfp _id	(record_id).hdf5 + path within file= /processing/Behavior al/Position

The filenames are resolved to extract individual components of the unique unit name. The output graphics from the analysis of one unit are stored in unit_id.pdf (or .ps), and the relevant analysis parameters, graphical output data and parametric output are stored in (record_id).hdf5 file in path /analysis/unit_id/. The parametric outputs are also listed in a table where each unit is indexed (row number) with the unit_id.

The existing analysis in NeuroChaT produces more than 50 graphical outputs. Storing them in one file creates the initial layer of output data management. These output files are stored in the respective data folder, so that they can be easily traced. The unique nomenclature of the unit information is essential when working with many such units from the same study, otherwise keeping track of the output graphics would be overwhelming in terms of number of graphic files and the amount of disk space they would require. We also rasterize the graphics and store them at 400dpi to preserve the image quality for publication. The Excel table of the parametric output are also easy to use as the user can perform inferential statistics on them using any standard analysis package including Microsoft Excel itself. The list can also be used to easily identify the units of interest based on certain property of the output parameter, i.e. Skaggs information content for spatial units. Thus, the nomenclature used in NeuroChaT brings in efficiency in managing and scrutinizing the analyses outcome.

3.4.2 Batch mode processing

NeuroChaT facilitates batch mode processing by providing the unit and spatial information in an Excel list. It is a very usual practice for researchers to keep track of the identified single units or the units of interest using an Excel file. We harnessed this opportunity to facilitate the analysis using this list. **Appendix A** shows how to provide the input to NeuroChaT using such lists. Once input and other configurations are set, and the user starts the execution, it fetches individual unit information one at a time, performs the analyses specified, and finally gives the results in a table. While it takes time for the analyses to complete, the user can do other tasks, and make efficient use of the time. Moreover, as the output are all stored accordingly in the respective data folder, the user does not need to worry about saving individual files. Once the analysis is finished, the parametric output table is displayed, and it waits for the user to save in his/her desired location. Meanwhile, the errors or warnings keeps up showing in the log box if used with the GUI or in the Python command interface. The user can traceback which units showed such notifications and at what point of the analysis. Therefore, there is no need for the neuroscientist to sit in front of the computer till the analyses are finished. On top of that, units with speculated similar properties, i.e., head-directional, can be listed in one file for convenience of post-hoc inferential analysis of population data.

3.4.3 Verification utility

We observed from our experience that a frequent reason for having error while analyzing a unit comes from the wrong specification of data. If a user has already a list prepared, he or she can verify the information using this utility, i.e., whether the specified path or files exist, or whether the cluster unit of interest belongs to the recording or is mistyped. The utility works by verifying the existence of each file, loading each spike file, and exploring whether it contains the specific unit of interest. This assures that the user is not wasting time for finding issues after running the analyses and knows well ahead about the problematic specifications. As many of the

NeuroChaT analysis are very time consuming, this is a convenient way of eliminating a common human error and reduce the waste of time.

3.4.4 PDF management utility

Given that a large number of units are recorded over time, the number of .pdf or .ps output files grows linearly. The PDF management utility in NeuroChaT facilitates the user to merge the output file of interested units into one file or to relocate them in a specific folder to group them together. The utility can be used either by providing a list of such units, or by manually choosing the files using an interactive window (**Appendix A**). At the end of each execution, NeuroChaT provides a list of pdf files where the graphical output for each analyzed unit are stored. User can export this list from the GUI *Utility* menu and can use the same list for merging or accumulating them into one folder. Thus, NeuroChaT also bridges the gap of tracing, by using unique nomenclature, and managing hundreds of graphical outputs in a logical and non-contradictory approach.

3.4.5 Data accessibility through NWB/ HDF5

The proprietary format data are converted into and are accessible through the HDF5 file viewers (www.hdfgroup.org/) once they go through NeuroChaT. The name came from Hierarchical Data File format, and as it suggests, data within the file are stored in hierarchical groups and datasets, making it more like organizing the data in folder in an operating system. Every time NeuroChaT analyses a unit, it stores the relevant data in the HDF5 file that has been named following the convention in NeuroChaT as described in **Section 3.4.1**. HDF5 is behind many popular open electrophysiology file formats (**Figure 3.55**) such as NIX (Stoewer et al., 2015), LBNL BRAINformat (Rübel et al., 2015), Orca, KWIK (Rossant et al., 2015), and most recently the Neurodata Without Border or NWB (Teeters et al., 2015). A comprehensive effort was particularly taken for establishing NWB file format by considering a number of evaluation criteria such as:

- Ability to store electrophysiology data
- Making the data open-source
- Extensibility to accommodate new data or methods
- Available and robust tool ecosystem
- Support for popular electrophysiological data analysis platforms like Python or MATLAB®
- Viewing the data without coding
- User's ability to describe the data after reviewing its contents
- Ease of extracting and plotting data by naïve users etc.

Therefore, we also considered NWB as the standard output and input data file format for NeuroChaT. One of the biggest hurdles in using NWB was its poor API design which does not allow extending on an existing file to include new data. It always requires creating a new file. But this extensibility is essential for NeuroChaT because of the cell-by-cell approach it uses. We convert the data for each unit a time at the end of analysis and store it in the file. There is always one HDF5 file for one session of recording as mentioned earlier so that we can manage the data compactly. We also wanted to enable the user to try for different sets of analyses. As the analysis data are also stored in the HDF5 file, it is very inefficient to recreate a file every time a new set of analysis is performed. Given that HDF5 itself has all the essential criteria, and the API for reading or writing is available in all major programming languages, we decided to store our data in the native HDF5 file format, instead of one of the formats that are built on it.

Other key advantages that HDF5 file format provides are:

- Ability to store heterogeneous data or data of different types i.e., numbers or texts, or complex data objects
- There is no vendor lock-in
- Self-describing which implies all data and the meta-data, or the description of the data can be passed along in one file
- The reading and writing process are very fast and reliable

- There is virtually no limit on the file size and the number of data objects in can store, therefore it is very flexible for big data and potentially very useful for storage of data from high-density electrodes, i.e., Neuropixels (Jun et al., 2017).

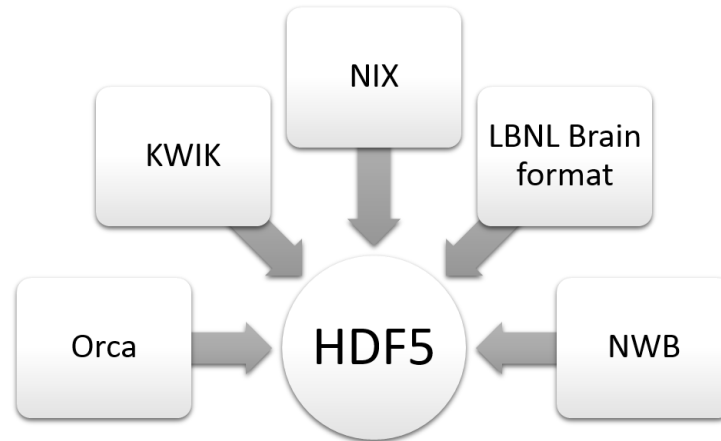


Figure 3.55 Popular file formats using the HDF5 format.

NeuroChaT stores the data in the HDF5 file following a very similar naming convention as are used in NWB format so that if NWB brings changes in its read/write API, the HDF5 file can be compatibly used by it. The hierarchy in the storage for NeuroChaT-created HDF5 file is shown in **Figure 3.56**. The acquisition, epochs, general, and stimulus folders are currently unused. The video tracker in each recording system processes the raw video files to provide the spatial behavior of the animal, and therefore, is placed in the *processing* folder (or, path or group as called in HDF5). This is also the case for LFP and single-unit data which are stored in the *Neural Continuous* and *Shank* paths respectively within *processing* to remain consistent with the NWB format. A path is divided into modules, each module into interfaces, and each interface as series in NWB, although in terms of data organization within HDF5 file, they are paths at descending hierarchy. The data paths and comparable items in NWB are listed in **Table 3.4**.

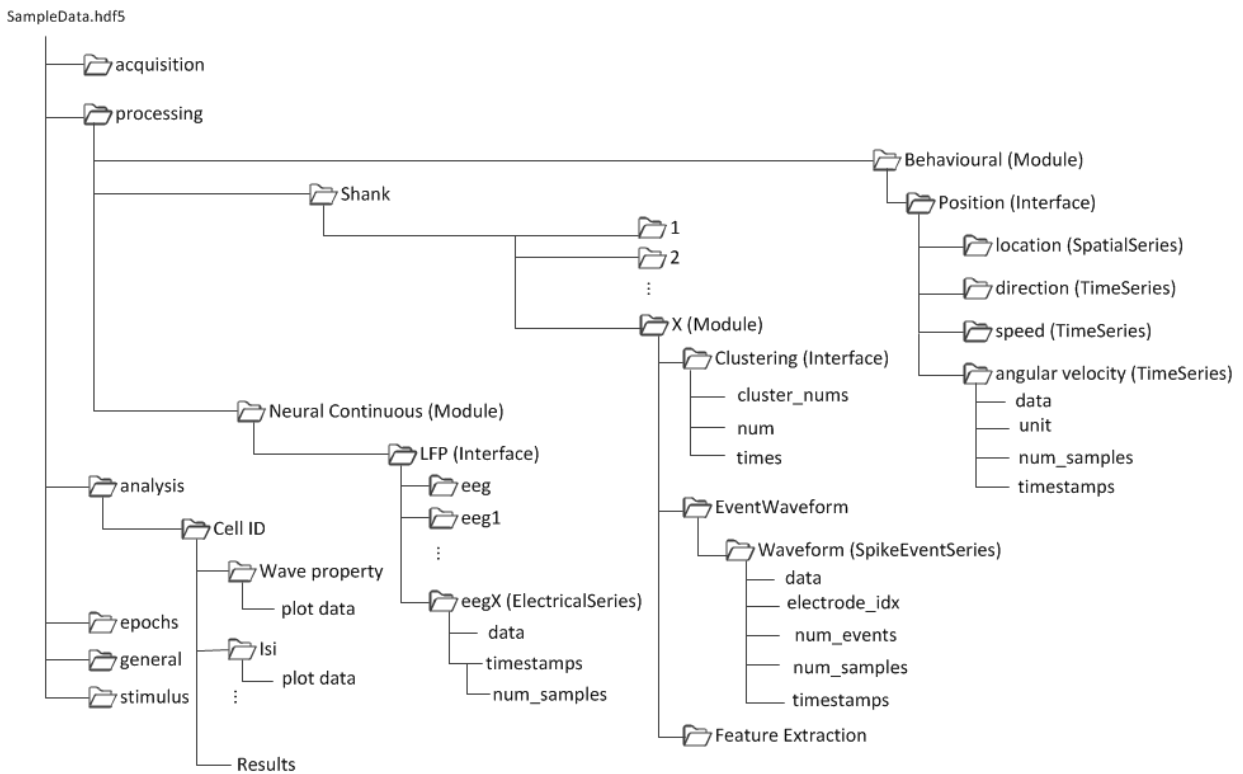


Figure 3.56 The hierarchy used for data storage in the HDF5 file format

Table 3.4 NeuroChaT-generated HDF5 file data paths and items comparable to the NWB formats.

Data	Path	NWB Module	NWB Interface	NWB Series	NWB datasets
Spatial	/Behavioral/Position/location etc.	Behavioral	Position	Spatial Series: location, Time Series: direction, speed etc.	data, unit, num_samples, timestamps
Spike	Shank/X/Clustering or Shank/X/EventWaveform/ Waveform	X (tetrode number)	Clustering, Event-Waveform	Spike Event Series: Waveform	Clustering: cluster_nums, num, times Waveform: data,

					electrode_idx , num_events, num_samples , timestamps
LFP	/Neural Continuous/ LFP/eegX	Neural Conti- nuous	LFP	Electrical Series: eegX (LFP file extension)	data, num_samples , timestamps

The toolbox also has a utility that converts the units specified in a list of using an Excel file like the one used for the batch-mode processing (**Appendix A**). This is another step towards the integrity in data throughout the entire NeuroChaT workflow and data processing.

3.5 References

- ANDERSON, M. I. & O'MARA, S. M. 2003. Analysis of recordings of single-unit firing and population activity in the dorsal subiculum of unrestrained, freely moving rats. *Journal of neurophysiology*, 90, 655-665.
- BAKKUM, D., RADIVOJEVIC, M., FREY, U., FRANKE, F., HIERLEMANN, A. & TAKAHASHI, H. 2014. Parameters for burst detection. *Frontiers in Computational Neuroscience*, 7.
- BASSETT, J. P. & TAUBE, J. S. 2001. Neural correlates for angular head velocity in the rat dorsal tegmental nucleus.
- BLAIR, H. T. & SHARP, P. E. 1995. Anticipatory head direction signals in anterior thalamus: evidence for a thalamocortical circuit that integrates angular head motion to compute head direction.
- BRANDON, M. P., BOGAARD, A. R., SCHULTHEISS, N. W. & HASSELMO, M. E. 2013. Segregation of cortical head direction cell assemblies on alternating theta cycles. *Nat Neurosci*, 16, 739-48.
- CACUCCI, F., LEVER, C., WILLS, T. J., BURGESS, N. & O'KEEFE, J. 2004. Theta-modulated place-by-direction cells in the hippocampal formation in the rat. *J Neurosci*, 24, 8265-77.
- CHO, J. & SHARP, P. E. 2001. Head direction, place, and movement correlates for cells in the rat retrosplenial cortex.
- DOIRON, B., CHACRON MJ FAU - MALER, L., MALER L FAU - LONGTIN, A., LONGTIN A FAU - BASTIAN, J. & BASTIAN, J. 2003. Inhibitory feedback required for network oscillatory responses to communication but not prey stimuli.
- FISHER, N. I. 1993. *Statistical Analysis of Circular Data*, Cambridge, Cambridge University Press.
- GAMMA, E., HELM, R., JOHNSON, R. & VLISSIDES, J. 1995. *Design patterns: elements of reusable object-oriented software*, Addison-Wesley Longman Publishing Co., Inc.
- HAFTING, T., FYHN, M., MOLDEN, S., MOSER, M. B. & MOSER, E. I. 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436, 801-6.
- JANKOWSKI, M. M., ISLAM, M. N., WRIGHT, N. F., VANN, S. D., ERICHSEN, J. T., AGGLETON, J. P. & O'MARA, S. M. 2014. Nucleus reuniens of the thalamus contains head direction cells. *Elife*, 3.
- JANKOWSKI, M. M. & O'MARA, S. M. 2015. Dynamics of place, boundary and object encoding in rat anterior claustrum. *Front Behav Neurosci*, 9, 250.
- JUN, J. J., STEINMETZ, N. A., SIEGLE, J. H., DENMAN, D. J., BAUZA, M., BARBARITS, B., LEE, A. K., ANASTASSIOU, C. A., ANDREI, A., AYDIN, C., BARBIC, M., BLANCHE, T. J., BONIN, V., COUTO, J., DUTTA, B., GRATIY, S. L., GUTNISKY, D. A., HAUSSER, M., KARSH, B., LEDOCHOWITSCH, P., LOPEZ, C. M., MITELUT, C., MUSA, S., OKUN, M., PACHITARIU, M., PUTZEYS, J., RICH, P. D., ROSSANT, C., SUN, W. L., SVOBODA, K., CARANDINI, M., HARRIS, K. D., KOCH, C., O'KEEFE, J. & HARRIS, T. D. 2017. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551, 232-236.
- KO, D., WILSON, C. J., LOBB, C. J. & PALADINI, C. A. 2012. Detection of bursts and pauses in spike trains. *Journal of Neuroscience Methods*, 211, 145-158.
- LEVER, C., BURTON, S., JEEWAJEE, A., O'KEEFE, J. & BURGESS, N. 2009. Boundary vector cells in the subiculum of the hippocampal formation. *J Neurosci*, 29, 9771-7.
- MARDIA, V. K. & JUPP, P. E. 2008. *Directional Statistics*, John Wiley and Sons, Inc.
- MULLER, R. U., BOSTOCK E FAU - TAUBE, J. S., TAUBE JS FAU - KUBIE, J. L. & KUBIE, J. L. 1994. On the directional firing properties of hippocampal place cells.
- MULLER, R. U. & KUBIE, J. L. 1989. The firing of hippocampal place cells predicts the future position of freely moving rats. *The Journal of Neuroscience*, 9, 4101.
- OPPENHEIM, A. V. & SCHAFFER, R. W. 2009. *Discrete-Time Signal Processing*, Prentice Hall Press.
- PASQUALE, V., MARTINOIA, S. & CHIAPPALONE, M. 2010. A self-adapting approach for the detection of bursts and network bursts in neuronal cultures. *J Comput Neurosci*, 29, 213-229.

- ROSSANT, C., KADIR, S. N., GOODMAN, D. F. M., SCHULMAN, J., BELLUSCIO, M., BUZSAKI, G. & HARRIS, K. D. 2015. Spike sorting for large, dense electrode arrays. *bioRxiv*.
- ROYER, S., SIROTA, A., PATEL, J. & BUZSAKI, G. 2010. Distinct representations and theta dynamics in dorsal and ventral hippocampus. *J Neurosci*, 30, 1777-87.
- RÜBEL, O., PRABHAT, M., DENES, P., CONANT, D., CHANG, E. & BOUCHARD, K. 2015. BRAINformat: A Data Standardization Framework for Neuroscience Data. *bioRxiv*.
- RUTISHAUSER, U., ROSS, I. B., MAMELAK, A. N. & SCHUMAN, E. M. 2010. Human memory strength is predicted by theta-frequency phase-locking of single neurons. *Nature*, 464, 903-7.
- SARGOLINI, F., FYHN, M., HAFTING, T., MCNAUGHTON, B. L., WITTER, M. P., MOSER, M. B. & MOSER, E. I. 2006. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312, 758-62.
- SHARP, P. E. 1996. Multiple spatial/behavioral correlates for cells in the rat postsubiculum: Multiple regression analysis and comparison to other hippocampal areas. *Cerebral Cortex*, 6, 238-259.
- SHARP, P. E. 1999. Comparison of the timing of hippocampal and subicular spatial signals: implications for path integration.
- SHINOMOTO, S. 2010. Estimating the Firing Rate. *Analysis of Parallel Spike Trains*.
- SKAGGS, W. E., MCNAUGHTON, B. L. & GOTHARD, K. M. 1993. An Information-Theoretic Approach to Deciphering the Hippocampal Code. *Advances in Neural Information Processing Systems 5, [NIPS Conference]*. Morgan Kaufmann Publishers Inc.
- SKAGGS, W. E., MCNAUGHTON, B. L., WILSON, M. A. & BARNES, C. A. 1996. Theta phase precession in hippocampal neuronal populations and the compression of temporal sequences. *Hippocampus*, 6, 149-172.
- SOLSTAD, T., BOCCARA CN FAU - KROPFF, E., KROPFF E FAU - MOSER, M.-B., MOSER MB FAU - MOSER, E. I. & MOSER, E. I. 2008. Representation of geometric borders in the entorhinal cortex.
- STOEWER, A., KELLNER, C. J., SOBOLEV, A., SONNTAG, M., BENDA, J., WACHTLER, T. & GREWE, J. 2015. Integrating data storage and annotation in the data workflow using the NIX format and libraries. *Frontiers in Neuroscience*.
- TAUBE, J. S., MULLER RU FAU - RANCK, J. B., JR. & RANCK, J. B., JR. 1990. Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis.
- TEETERS, J. L., GODFREY, K., YOUNG, R., DANG, C., FRIEDSAM, C., WARK, B., ASARI, H., PERON, S., LI, N., PEYRACHE, A., DENISOV, G., SIEGLE, J. H., OLSEN, S. R., MARTIN, C., CHUN, M., TRIPATHY, S., BLANCHE, T. J., HARRIS, K., BUZSAKI, G., KOCH, C., MEISTER, M., SVOBODA, K. & SOMMER, F. T. 2015. Neurodata Without Borders: Creating a Common Data Format for Neurophysiology. *Neuron*, 88, 629-34.
- TREVES, A. & ROLLS, E. T. 1991. What determines the capacity of autoassociative memories in the brain? *Network: Computation in Neural Systems*, 2, 371-397.
- TSANOV, M., CHAH, E., VANN, S. D., REILLY, R. B., ERICHSEN, J. T., AGGLETON, J. P. & O'MARA, S. M. 2011a. Theta-modulated head direction cells in the rat anterior thalamus. *J Neurosci*, 31, 9489-502.
- TSANOV, M., CHAH, E., WRIGHT, N., VANN, S. D., REILLY, R., ERICHSEN, J. T., AGGLETON, J. P. & O'MARA, S. M. 2011b. Oscillatory entrainment of thalamic neurons by theta rhythm in freely moving rats. *J Neurophysiol*, 105, 4-17.
- VAN WINGERDEN, M., VINCK, M., LANKELMA, J. & PENNARTZ, C. M. 2010. Theta-band phase locking of orbitofrontal neurons during reward expectancy. *J Neurosci*, 30, 7078-87.

(This page is intentionally kept blank)

4 Discussion

4.1 Attaining NeuroChaT objectives

4.1.1 Interactive tool

One of the objectives of this project was to create an interactive tool for neuroscientists. NeuroChaT achieved this by producing a graphical user interface that helps the user interact with the underlying large volume of codes by simple clicks and enjoying analyses in different modes, i.e. one cell at a time or batch processing of list of cells, along with setting up the analyses parameters using the interactive panels. Other utilities like verification of the unit list before batch mode processing, selecting the analyses individually as per the requirements or by cell types, output graphical file management utility can also be done using just clicks. Current setting of analyses types and parameters can be stored and retrieved from the menu- further reducing the amount of efforts for executing similar analyses on different sets of data using similar input to the analysis techniques. Selecting the unit, LFP channel, browsing the data, comparing units for similarity, assessing the quality of clustering, converting the proprietary format data to the standard HDF5 file format data- all these features define the UI as very useful and interactive tool.

The development framework PyQt5 is a cross-platform framework and, therefore, the software UI can be rendered in most operating systems. We have not tested the interface on touch enabled devices or in mobile operating system platforms. Given that the computing power of the mobile devices are increasing very rapidly, it might be a good opportunity to design the UI using a framework, e.g., Kivy (<https://kivy.org>), that is more suitable for such platforms, but considering the maturity, versatility, stability, existence of large community of users, and the ease of implementation and extension compared to other platforms, we have decided to build our entire graphical interface in PyQt5. Qt is a cross-platform C++ application framework and widget library and is free to use for open source project under the GPL v3, and LGPL v3 licenses (<https://www.qt.io>). PyQt5 is a set of Python bindings for Qt v5 which enables Python as an alternative language for application development and is released under the GPL v3 license (<https://pypi.org/project/PyQt5/>). As NeuroChaT source is made open for non-commercial use, there is no restriction on usage of either Qt v5 or PyQt5.

4.1.2 How can it work as a framework?

The general concept of software framework is that it provides generic functionalities which can be used along with the user-written codes to develop new application. The concept of framework in NeuroChaT is that it provides the user with diversified functionalities, which are both class specific and generic, for managing the attributes and input/output control from an object. The user can utilize the objects in many fashions. For example, user can define a separate spike object and a spatial object and can pass the information selectively to the spatial object, e.g., the spike event timestamps, for a rate coding analyses while the spike object remains unchanged and can be used independently for analysis built within itself. The entire spike object can be passed as a member of the spatial object which will harness the information carried by the spike object within itself. Similar principle applies for spike-LFP data coupling. Alternatively, they can be the members of NData object which can manage the transfer of information among these objects- thus providing for encapsulation of all three recorded data types. As each object is readily extendable with the helper functions to facilitate the transfer of information, a variety of combinations of these objects or data within them can be used to build a set of analyses according to the user requirement. This use of collection of NeuroChaT classes as an application programming interface (API) or skeletal support by bringing together different components to develop a customized project can be compared to what is traditionally defined as software framework, although the code base is not big enough to fit into the classic definition.

4.1.3 Unification of analyses methods

We achieved this objective by implementing a big set of analyses techniques that are already widely in use in relevant researches. There are altogether 25 different analyses developed, tested, and being used- resulting in their unification in one platform. We implemented well established methods for spatial analyses, analyses of spike trains, LFP and its relationship to the spiking activities. Most of the toolboxes currently available put emphasis on a particular datatype or analyses as explained in **Section**

1.3.5. They are either made for spike-sorting, working with spike-trains and event related analyses, or analyzing LFP signals. But description of the spatial behavior in terms of neuronal firing can also be modulated by underlying LFP signals, and the units may have traits in their spiking-trains to describe the dynamics of local population ensembles (**Section 1.1.2**). Therefore, collecting all these analyses in one platform made it easier for researchers to examine their data in different ways without switching from one toolbox to another and without additional burden of managing data format compatibility among different analyses platforms.

4.1.4 Developing problem specific analysis

The architecture of NeuroChaT easily facilitates integration of new analyses methods. The object-oriented design enables ease of extension of the classes by including new methods into the class. This can be achieved in two ways: (a) classes can be subclassed either by composition or inheritance, and (b) Python allows inserting a method to an instance of an object dynamically which, therefore, does not require sub-classing the original class. There are large sets of methods in each class for managing input and output of data and information in and out of the class instances. These methods provide a means for data management within each class instance without interfering to the variables or data that are in use exclusively within each class. Please refer to the NeuroChaT documentation (**Appendix A**) for references to these methods.

In API use guide (**Appendix C**), we provide an example of how problem specific analysis technique can be developed using the data classes in NeuroChaT by writing a few lines of code. The example depicts how easily an analysis can be implemented and integrated in NeuroChaT with the help of a large number of methods included in its classes. These are not part of the analysis techniques but designed for bringing efficiency and eloquence in programming new analysis algorithm.

4.1.5 Bringing efficiency in research yield

The batch analyses mode in NeuroChaT facilitates serializing the analyses of hundreds of units using a list in an Excel sheet. This implies the users can set all the analyses and associated parameters for the analyses they want to perform, can upload the Excel file with the list of units and start the analyses. NeuroChaT will take care of the analyses and will provide feedback in terms of progress, warning and error messages. Therefore, the user does not need to sit in front of the computer for the analyses to complete. They can plan other studies, do experiments, or review the output of previous run for inference while NeuroChaT is running. The user can efficiently use their time this way. The specification of batch mode data input is a simple approach and align with how the researchers keep track of their recording information- usually using an Excel sheet with all the information. The tabular parametric output and the table of graphical file location that can be used for merging or accumulating them into one file or folder, respectively, can then be used for further assessment of the spatial or rhythmic nature of the units. Logs displayed during the process can be analyzed for errors and can be addressed easily. The overall scheme of working in NeuroChaT is thus not only user friendly, but also very efficient in terms of working hours. Since NeuroChaT prototype had been developed, there is an increased pace in the analyses of data in the lab, yielding number of publications and few others under preparation or review- proving how NeuroChaT is contributing to the overall yield to our experimental works.

4.2 Towards a culture of openness and reproducibility

4.2.1 Open Data

Why Open Data initiative?

Data in Neuroscience exist at a range of scales in space and time as summarized in **Figure 6.1** (Sejnowski et al., 2014). Decades of developments enabled obtaining data using a wide variety of techniques, using patch clamp to record intracellular action

potentials to fMRI for system level organization of neural functionalities. Each of these techniques has their own vocabularies, assumptions and experimental practices or norms. Integrating information from different recording techniques is one of the fundamental problems in Neuroscience study. This is further complicated by the fact that Neuroscience research is conducted on a range of problems, i.e. learning and memory, decision making, perception, motor activities etc. and across different species (Keifer and Summers, 2016)- requiring a horizontal integration of problem specific findings along with the vertical integration on the same problem or species using different levels of organization of neuronal populations. This can be achieved through the mega-analyses or pooled analyses of raw data and meta-analysis of published data as already been exemplified by genome projects (Costafreda, 2009). Data sharing also comes into play when large scale efforts like Brain Research through Advancing Innovative Technologies (BRAIN) (Insel et al., 2013) and Human Brain Project (HBP)(Markram et al., 2011) needs to combine the findings from different laboratories which would otherwise be nearly impossible to accomplish from one lab. Research involving human subjects have inherent ethical demands to minimize the number of participants and maximize the benefits of research (Brakewood and Poldrack, 2013, Choudhury et al., 2014) unless doing so would increase the risk to the subjects. Sharing data also has economic advantage as it reduces the cost of doing science by providing a bigger sample set for the same research questions and increasing the reproducibility in terms of analyses. The increased concern of reliability and reproducibility (Anonymous, 2013) in scientific research are often embedded in the low statistical power in many published studies (Button et al., 2013) - posing an increased pressure to deploy larger samples. But this can be hindered by the regulation on the use of animals or subjects and financial limitations (Poldrack and Gorgolewski, 2014) which can be overcome by amassing large number of samples within the similar species and experimental paradigms by sharing data among laboratories. Almost all the neuroscientific techniques involve multistage processing streams resulting in a large number of variations in analysis parameters (Carp, 2012). This poses another concern on reproducibility to obtain similar results using different analysis pipelines. As the computation power and the new generation of computational neuroscientists evolve, there is a continuous development of analytic techniques using complex algorithms and machine learning methods for more

comprehensive and translational research (Lloret-Villas et al., 2016). Sharing data may provide the test bed as well as the benchmark for the new techniques. For example, data from OpenfMRI have been used to examine the effects of different processing pipelines on analyses outcomes (Poldrack and Gorgolewski, 2014, Carp, 2012), and German Neuroinformatics Node (spike.g-node.org) provides the platform for benchmark data and evaluation of spike-sorting algorithms. Developing open-source toolboxes is a cultural practice in neuroscience which benefitted researchers to a great extent. Sharing data in an open and widely used format can also foster the development of open-source tools and standardizing them along with increasing the code reuse between different toolboxes. The use of in-house or proprietary formats and source codes prevent the researchers from modifying and unifying them across different tools and languages (Siegle et al., 2015). **Figure 6.2** summarizes the necessity of open-data initiative and that of sharing the data.

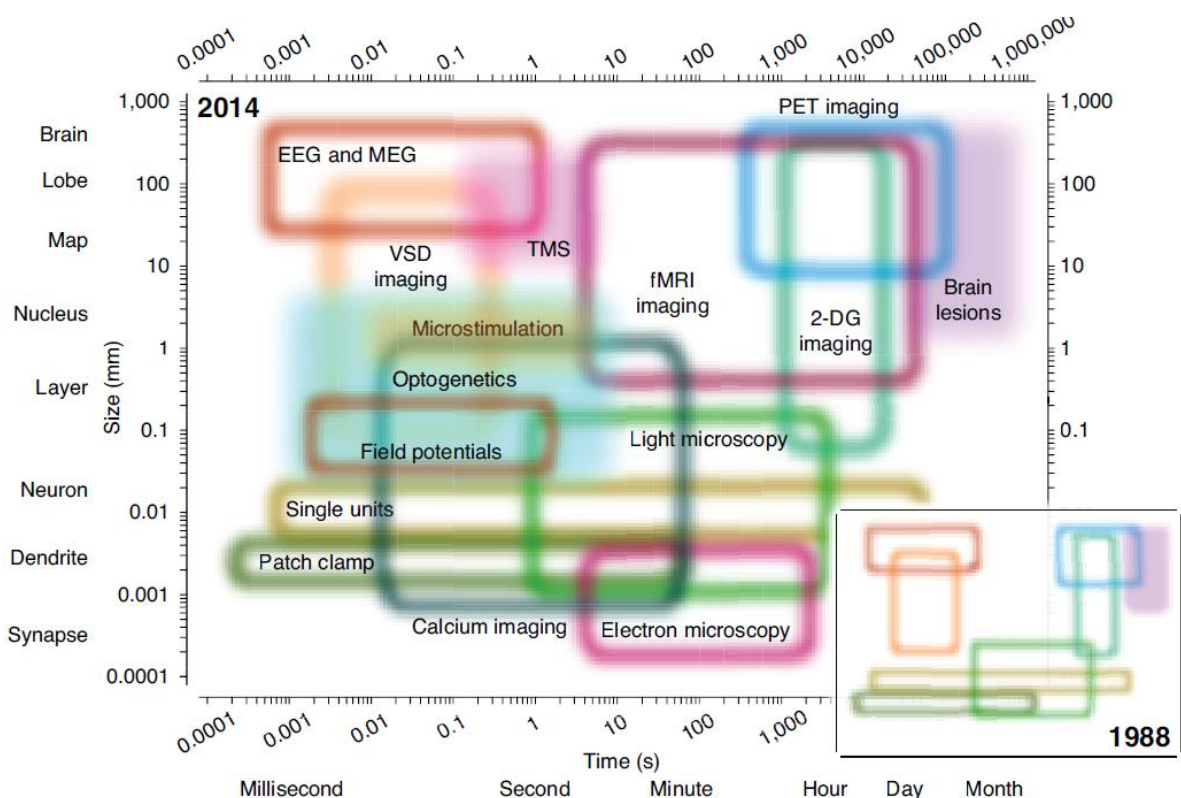


Figure 4.1 The spatiotemporal domain of neuroscience and of the main methods available for the study of nervous system in 2014. Each colored region shows the extent of spatial and temporal resolution for a method. Filled boxes show perturbation techniques for alternating the functions of neuronal circuitry and the unfilled boxes show the measuring techniques. Inset shows a cartoon rendering of the methods available in 1988. (Adapted from Sejnowski and Churchland., 2014). EEG= electroencephalography; MEG= magnetoencephalography; PET= positron emission tomography; VS= voltage-sensitive dye; TMS= transcranial magnetic stimulation; 2-DG, 2-deoxyglucose. (Adapted from Sejnowski et al., 2014)

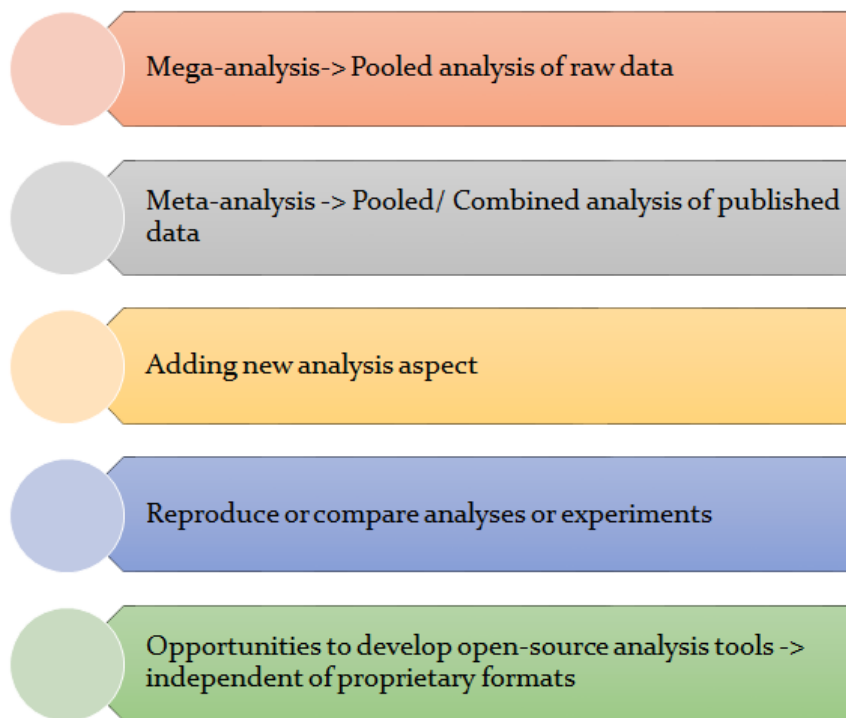


Figure 4.2 Summary of why open data initiative is necessary

Open Data Requirements

Sharing data for open science has some preliminary requirements. Collaborative projects like genome projects standardize the file format and store them in massive databases so that both the raw and the published data can be pooled. The toolboxes for such datasets should also be such that researchers using various computer systems and analysis machinery can access them irrespective of the platforms they are using. Often reading the data and information about the data, also known as metadata, provide an additional confidence about the data structure, types, size, and usability. These requires a format that is human readable, or at least widely provided with appropriate software to make it so. Neuroscience data format needs to support the management and organization of complex datasets from various sources, e.g. neuronal recording, video and audio recording, eye-tracking, external stimuli, neural stimulation, task-related events, analytic results etc. Supporting the metadata can make interpretation and analysis much easier. Storing similar data elements in a common, compatible way with common descriptions and terminologies can further

enable interpreting the data without any need for an API or any other translation mechanism between one lab and another (Zehl et al., 2016). Apart from these technical requirements, as summarized in **Figure 6.3**, there are other aspects to consider for creating the ecosystem of data sharing (**Figure 6.4**). Given the tremendous amount of data growth in Neuroscience, the file formats should also need to be extensible while still maintaining portability. They must be widely supported by programming languages and analysis tools for fostering the growth of open-source toolboxes and for unifying custom-developed codes. The sharing of the data implies that they must be usable and easily accessible by other users. Hence, the principle of data sharing dictates that the data must reside in a repository which is discoverable to everyone with appropriate open-storage location along with the traceability of the data by the types of technologies used and the use of experimental paradigms. The quality of the data must be assessable by engaging the community to adopt a metric which will also incentivize individual researchers (Wiener et al., 2016). Moreover, the repository or the storage mechanism must be provided with incentives for sustainability of the data over the years. This implies that the data must be curated and governed, their life-cycle must be managed in a marketplace where data-exchange can be rewarded as part of the research product instead of calling them supplementary materials (Ferguson et al., 2014, Wiener et al., 2016). One such approach can be sharing data in the form of publication named as data-articles (Poline, 2018) and developing a citation system equivalent to the literature citation. Agencies like Wellcome Trust mandate the release of data, and there is an increase number of journals, e.g. Nature Scientific Data, GigaScience, F1000Research, MNI Open Research, eNeuro, eLife, Wellcome Open Research, that are accepting data from neuroscience-focused experiments. Such credit attribution through stable digital identifier can also help tracking of data and improving most of the ecosystem requirements of open-data sharing.

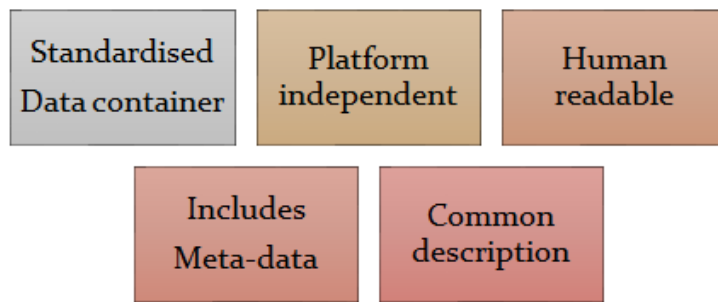


Figure 4.3 List of open data technical requirements



Figure 4.4 List of open data ecosystem requirements

Open Data Challenges

Open data challenges mostly surround the requirements of data sharing. Identifying the challenges as a multi-factor construct, (Wiener et al., 2016) mentioned a number of challenges to initiate the data sharing ecosystem and enable its survival so that new collaborative projects can thrive. The problem-space of neuroscience data sharing can be presented by **Figure 3.5** non-exhaustively. The most significant challenge in data-sharing is sociological and ethical, rather than technical (Choudhury et al., 2014). Researchers lack motivation to share their data unless an incentive to promote data-sharing is introduced (Poline et al., 2012). There is a classic competition to be recognized for novel findings. It is also expensive to generate data, and, when obtaining funding for research is extremely competitive, anxiety to be undercut by other researchers in terms of data collection is a very real challenge unless there is a strong motivation to release hard-earned data. There are also concerns of data quality and the fear of being viewed as incompetent if the researchers lack confidence about their research methods not being of the highest quality. The willingness to share data may further be constrained if the study involves human subjects where there are

concerns of privacy and confidentiality. Data used in Neuroscience studies suffers from diverse formats, methods and models. Sharing data in open repositories need to address the standardization of the data and make them discoverable which will otherwise be used by only a fraction of closely-operated researchers. The large volume of datasets also poses an upper bound on the capacity and throughput of the data-sharing infrastructures. Neuroscience datasets are highly specialized and very domain specific- requiring the training of a new workforce of researchers to ask what they should do with these data and to employ modern techniques, i.e., machine learning, for obtaining insights on the large volume of recorded data (Vogelstein et al., 2014). Projects with massive funding, e.g., US BRAIN and European Human Brain Project, produce large-scale datasets, but neuroscientists across the research spectrum produce lots of small datasets which creates a granularity in the data-sharing. These datasets are also called long-tail data (Ferguson et al., 2014). Sharing these richly diverse and heterogeneous small datasets are also in question for data-sharing as it requires continuous integration and adaptation of new data types. Sustainability of the stored data depends on the discoverability, usability, governance of the data and establishing a market-place to trade the data in research ethos. If there is a barrier to enter the database and the researchers cannot readily find the data they are interested in, the enrichment process for the ecosystem will diminish. This also implies that there should be policies and governance of data sharing allowing for federating disparate datasets. Finally, there should be financial safeguards for the continued existence of the ecosystem and its scalability to accommodate new generations of complex data recorded from sophisticated technologies which is also a demand against the traditional fixed-period funding strategies.

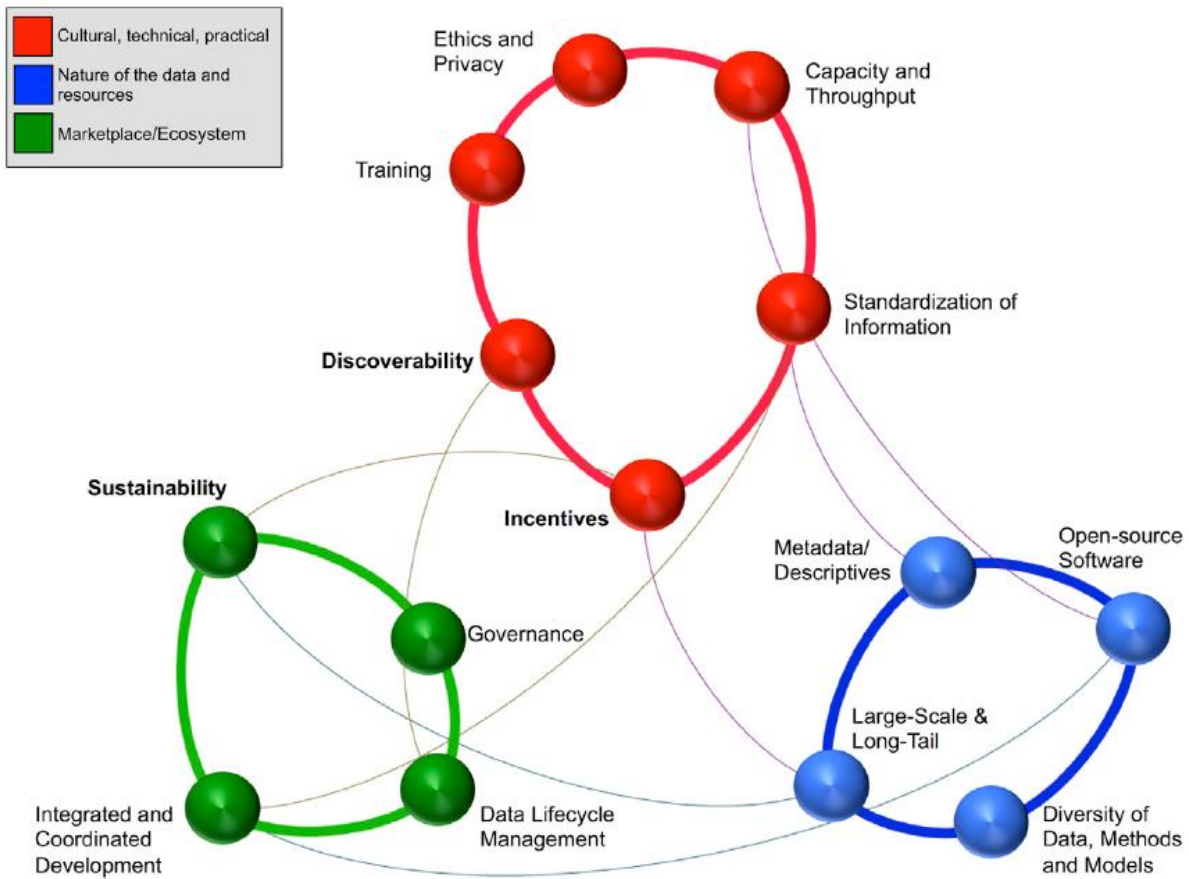


Figure 4.5 Landscape of data sharing in Neuroscience and the issues to address. These issues can be grouped into three clusters- red, blue and green- each for three different categories: 1. cultural, technical and practical, 2. Nature of the data and resources, and 3. Marketplace or ecosystem. (Adapted from Wiener et al., 2016)

4.2.2 Open Source

It is a very common practice to make the analytic and software source-codes publicly available across all fields of Neuroscience. In neurophysiology alone, there are lots of open source toolboxes as we discussed in **Chapter 2**. The open source tools help neuroscientists to extract, translate, analyze, and visualize the data. Neuroscientists also use software like Statistical Parametric Mapping (SPM) and Neuron for analyzing, modelling and simulating their data. But there are also a growing number of publications where custom-built software and scripts are being used, either stand-by or utilizing the codes from other open-source projects. When the results are set for publication, it undergoes a complex analysis pipeline involving many processing steps from the pre-processing of the data for extracting the exact information that the scientists are interested in to the statistical analyses and visualization of the

information. All these steps cannot be described in full-details in the accompanying publications- making it difficult to reproduce the results and cross-validate the findings of the original research. This issue brings a compromise in the integrity and reliability of research as addressed in (Open Science Collaboration, 2015). Although there are other experimental steps like use of transgenic animals and reagents that cannot be exactly reproduced, this limitation does not apply when the collected raw data enters the computation. Analyses techniques are becoming more sophisticated, implying that there is a wide gap between the collected raw data from the experiment to what is published as results. Availability of codes make it more transparent and enables other researchers to reuse and potentially improve it. The publication itself works as an advertising tool for the usefulness of the code or the software (Gleeson et al., 2017), and the feedback received by other users, contributions to the codes, tests and sharing of one another's software or codes can create a network of tools or neuroscientific developers. This can enable a multi-fold improvement in the quality control and efficiency in neuroscience research- particularly those engaging data analyses and model development. But it also necessitates developing a platform for open source repository and shares similar challenges as that of open data initiative. Interestingly, many good code-sharing solutions exist like GitHub and BitBucket. GitHub alone has 15 million users and its popularity has been increasing exponentially among scientists, many of them from Neuroscience, to share, maintain, and update their code (Perkel, 2016). Dedicated scientific sites like Zenodo (<http://zenodo.org>) and Figshare (<http://figshare.com>) provide citable unique identifiers (DOI) and also allow GitHub users to upload snapshots of their repositories (Eglen et al., 2017).

There is also an observable reluctance among researchers to share codes who believe that their codes are not written properly following the standardization or are poorly documented (Barnes, 2010) - implying that the codes are not 'good enough'. But it can be argued that if the code is good enough to do the job, it can be considered good enough to release. This will eventually provide the peers with at least a tentative idea of what data processing steps were undertaken and what set of parameters or assumptions were used to obtain the results of a published model or figure. The codes can still be useful contrary to popular expectation, although well-documented codes are obviously easy to reuse or test.

Releasing the codes used for analyses of data in a publication had not been a prerequisite until recently. A growing number of journals are now mandating that the authors ensure the public availability of all the codes, computational models and analysis tools (Eglen et al., 2017). F1000Research stated that “software papers describing non-open software, code and/or web tools will be rejected”. BioMed Central introduced a checklist for minimum standards of reporting for BMC Neuroscience, BMC Biology, Genome Biology and GigaScience which require the submission to include a statement about code availability and, when applicable, citing the code using a DOI or similar unique identifier (Kenall et al., 2015). Nature journals now require the authors to include a statement declaring the availability of the codes underlying the central results. Given that the altruistic motives for sharing codes are clear, funding agencies such as Wellcome Trust in the United Kingdom and the NIH in the United States already have developed policies to obligate the researchers and change their attitudes towards code-sharing.

4.2.3 Open Storage

Apart from the social challenges in making data open, the next level of difficulty arises when it comes to storage of the vast amount of data. Taking as an example, brain imaging studies such as the Alzheimer’s Disease Neuroimaging Initiative (ADNI) (Weiner et al., 2012) which produces multiple petabytes of genetics data alone with its initially available samples of 832 subjects (Van Horn and Toga, 2014). The size of the data increases significantly when this is combined with the multi-modal neuroimaging data types. The increase in the capability to obtain data and the need for open-data for sustainable reproducibility generated interest among the neuroscientists to create and share large scale databases. Neuroimaging community is particularly well ahead in manifesting the data-sharing culture. There are several major initiatives that provide publicly available datasets such as the Human Connectome Project, OpenfMRI, XNAT Central, 1000 Functional Connectomes/International Neuroimaging Datasharing Initiative, OASIS, ADNI, NIH MRI Study of Normal Brain Development (NDAR), the FBIRN consortium etc. Further briefings on these projects are available (Poline et al., 2012). Numbers of

organizations are also fostering the neuroimaging data sharing, e.g., the Neuroimaging Data Access Group (NIDAG), INCF and its Neuroimaging Data Sharing initiative, and the Biomedical Informatics Research Network (BIRN; Helmer et al., 2011).

But such large projects are multi-site collaboratives in nature which require an extensive amount of funding and are specifically financed to do so. For example, ADNI itself received a grant of \$60 million USD at the first stage. Although projects like ADNI, BIRN and XNAT produced infrastructure and tools to help groups share their data, small laboratories often struggle in capturing, preparing, and releasing their data (Poline et al., 2012). Sharing and curating “long-tail” data (Ferguson et al., 2014) and facilitating small laboratories to share their data within reasonable budgets are not yet well-practiced, and there is a necessity for more generalized and sustainable storage platform so that the data can be available for re-analysis in a distributed and lightweight manner.

So far, neurophysiologists do not have many community repositories like their colleagues in neuroimaging except the German Neuroinformatics Node (<https://www.g-node.org>). All the open data repositories can be browsed by subject ‘Neuroscience’ through the Registry of Research Data Repositories (<https://www.re3data.org>). There is a proliferation of ‘generalist’ data repositories that host data from different scientific fields with an objective to provide a systematic data stewardship practices and enable datasets collection, curation, preservation, long-term availability, dissemination and access (Assante et al., 2016). The following five repositories have been identified as being recommended by data journals for the deposition and publication of research data (Candela et al., 2015):

- **4TU.ResearchData:** Previously known as 3TU.Datacentrum (<https://researchdata.4tu.nl>) is the result of cooperation between four technical universities in the Netherlands.
- **CSIRO Data Access Portal:** This repository (<https://data.csiro.au>) is established by the Commonwealth Scientific and Industrial Research Organization (CSIRO) and is a part of the Australian National Data Service (ANDS).

- **Dryad:** This repository is an initiative of number of journals and scientific societies to adopt a joint data archiving policy (JDAP) (<http://datadryad.org>). This is governed by a membership organization where the membership is assigned to the scientific stakeholders such as journals, publishers, research institutions etc. One of the unique features of Dryad is that data files are always associated with a published article. Usually, journals facilitate the data submission by sending notices of new manuscripts to Dryad. Authors submit data either during the review of the associated article or after it has been published.
- **Figshare:** This repository provides free unlimited space for public data and is one of the portfolio business of Digital Science, the global technology division of Macmillan Science and Education (<https://figshare.com>).
- **Zenodo:** This repository is a by-product of the EU FP7 project OpenAIREplus (Manghi et al., 2012) in collaboration with CERN to establish a European-wide data-sharing infrastructure (<https://zenodo.org>). It has an integration with GitHub for creating citation of the hosted codes.

All five of these repositories provide DOI, an essential requirement for increasing the visibility and access to data-articles. (Assante et al., 2016) provides a detailed analysis of their features such as formatting, documentation, licensing, publication costs, validation process, availability of the data, discoverability and access, and provision for citation. Although Dataverse (Crosas, 2011) is often cited as a repository, it is indeed a technology that enables creating repositories.

Google's G Suite for Education provides free unlimited Google Drive spaces. For example, all students of Trinity College Dublin avail of this unlimited cloud-space through Trinity's MyZone service. Amazon Web Services (AWS) has launched 'AWS Public Dataset Program' to cover the cost of storage for publicly-available high-value cloud-optimized datasets to democratize access to data when it is available for analysis using the cloud-computing facilities of AWS. Other objectives of the program are to develop techniques, formats and tools that are native to cloud-computing and to encourage the development of communities who may possibly benefit from the shared data. Data analysis in neuroscience will soon require adoption to 'big-data' philosophy and will involve use of rich open-source ecosystems for distributed

computing and storage (Freeman, 2015). The storage facilitation by two technology giants for the greater scientific and academic community will hopefully ease the cultural shift in data-handling of neuroscientific studies.

4.2.4 NeuroChaT and open science

Open Data and NeuroChaT

NeuroChaT meets most of the technical requirements of open-data concept by adopting a widely used platform independent format called Hierarchical Data File (HDF5) format (www.hdfgroup.com) as its default output. This format provides a view of the data without decoding using freely available software called HDFView available in the link above. Popular electrophysiological data analysis platforms like Python or MATLAB® have API for reading and writing data in the HDF5 format which eases the extraction and plotting of data by naïve users. There is no vendor lock-in for this file-type. It is also extensible, and there is virtually no limit on the file size and the number of data objects it can store. This makes the format very flexible for big data and potentially very useful for storage of data from high-density electrodes, i.e., Neuropixels (Jun et al., 2017b). Different NeuroChaT data types are stored hierarchically using the name specifications of NWB format. Data from each recording session are stored in one file, reducing the management of multiple files generated in single recording in most of the commercial recording systems. The analyses outcome, both graphical and parametric, are stored in different folders with individual subfolders for each analysis. Use of this format and the arrangement of information within the file (**Figure 3.56**) conforms to most of the technical requirements of open data. We do not have specific mechanism to store meta-data apart from those automatically retrievable from the original data files. But NeuroChaT I/O module for HDF5 enables writing data and attributes to any of its paths or data without rewriting the entire file which was a major limitation in NWB API. Storing meta-data in multiple formats or extracting them into one common file using a structured file like odML (open metadata Markup Language) provides ways for keeping the information organized, easily accessible and machine-readable which are crucial for enhancing reproducibility in neuroscience research (Zehl et al., 2016).

Future extension of codes on system specific meta-data extraction or to insert user-input as meta-data to the HDF5 files can be easily deployed as a plug-in and without further requirement of new file formats for storing them separately aside the data files. The ecosystem requirements of open data are not addressed yet apart from the fact that we deposited analyzed data and PDFs of graphical output for the publication of (Jankowski et al., 2014) in *eLife*, a prominent journal funded by the Wellcome Trust, the Howard Hughes Medical Institute, and the Max Planck Society. But we briefly mentioned some of the options for the open storage to meet the requirements in **Section 6.2.3**.

Open Source and NeuroChaT

We used Open Science Framework (OSF) (<https://osf.io>) to make NeuroChaT source codes publicly available. OSF is an open-source software project of Center for Open Science, a non-profit organization in Charlottesville, Virginia, that facilitates management and open collaboration in scientific research. All files, data, codes and protocols can be stored in one centralized location which is managed in a secure cloud. Access to these files can be controlled by the user to make part of the project public or to keep it private. The user can view all the projects in one dashboard, add collaborators and view the changes in file version and their contributors. Each project may contain a Wiki for the description of the project. The framework currently also provides unlimited storage along with supporting third-party add-ons for storage like Google Drive, Dropbox, OneDrive, Amazon S3, Bitbucket, Box and Figshare. Citations manager like Mendeley and Zotero can be integrated as well. GitHub and GitLab can also be used from the project dashboard with full control to the git repository. Thus, OSF brings in a new opportunity to store-keep source-codes, data, and documentation along with other project-management features-making it a suitable candidate for hosting NeuroChaT with respect to all open-initiative aspects.

The entire NeuroChaT project can be found in the citable link, <https://osf.io/kqz8b>. NeuroChaT provides access to the source codes by hosting them in GitHub repository (<https://github.com/mnislamraju/NeuroChaT>). The same codes are also hosted in the OSF project where they are essentially linked and synced to the GitHub repository. Therefore, the users can avail the version control and collaboration in the same way as in git. Common git actions like creating repository, branching,

committing, sending pull requests and downloading the entire repository can be performed similarly. The project page also contains a Wiki describing essential aspects of the NeuroChaT. Few example datasets in HDF5 formats are also provided using the free storage that OSF provides. These datasets are also publicly available so that the users can use or test NeuroChaT by using these files.

4.3 Data integrity and open science in NeuroChaT

Scientific misconduct and fraud span creating data, manipulating results to make them more statistically significant or more compelling, appropriating ideas and approaches of other researchers and plagiarizing (Lisberger, 2013). While most other issues are addressed through enforcing and nurturing ethical practices, the ones involving data can be handled through direct speculation. When it comes to neuroscience, particularly to neurophysiology, there are challenges pertaining to the artefact created by the analysis pipeline, lack of peer-reviewed code and lack of replication of analysis. Novel developments are facilitating increased neuroscience data volume and posing challenges on computational errors apart from the intentional misconducts. This is further exaggerated by the veracity of neuroscience data types and formats, and lack of analysis tools supporting data standards apart from the copyright formats. There are also issues associated with data storage, maintenance, exploring across data across historical framework, referencing, indexing and data sustainability. So far, responsibility for the truth and accuracy of data is on the onus of the researcher (National Academy of Sciences et al., 2009), but it is also essential to address the challenges in compounding results from multiple scales of resolution (gene expression datasets, cellular imaging and physiology, functional imaging and clinical data) (Akil et al., 2011, Gordon, 2003, Deco et al., 2015, Bouchard et al., 2016). Transparency in data manipulation, interpretation, statistical measurements etc. can be assured through the practice of open science in the community and cultivating data integrity within individual laboratories. The US National Academy of Science makes the following recommendation for ensuring integrity of research data (Editorial, 2009):

- Provision of appropriate tools for management of research data and for efficient utilization of computational and human resources
- Documentation of experimental protocols and their relevance to data acquisition and manipulation
- Making data and experimental details accessible and archiving to allow for replication and consequent studies

The data integration process by using NeuroChaT starts with standardizing the processing pipe-line as shown in **Figure 6.6**. Data is handled by the experimenter from the design of the studies to publication. Once the behavioral and neural data are recorded, they are stored in the recording computer with a backup to an external storage medium or a secure cloud storage of experimenter's choice. Experimental details including details of the animal under use are written on spreadsheet as a source of metadata for subsequent studies. The unit identification and isolation follow the analysis of behavioral correlates using the NeuroChaT software. Once units are all analyzed, they are characterized based on the analysis graphics and output parameters. Units with similar characteristics, i.e., tuned to head-directional information, are grouped for statistical inference. The expert developer creates routine for appropriate use of the NeuroChaT or assesses the credibility of built-in algorithms. Thus, an intra-lab peer-reviewing is ensured. The computational expert also provides feedback on the interpretation of the inferential statistics and an intra-lab data-integrity is established.

The source code of NeuroChaT is open. Once the storage of the data is appropriately handled, it ensures an overall integrity of the research data to the wider community for reproducibility and accountability. The open-methodology principle has been complied by using peer-reviewed analyses techniques and explaining the entire data-processing pipeline in the published literatures. Apart from the open source and open data, the trinity of open science (Kraker et al., 2011) is accomplished by archiving them in platforms with open peer-review and open access such as *bioRxiv* (Matulewicz et al., 2018). The working philosophy in NeuroChaT therefore created an ecosystem for aligning our research to open science principles by addressing associated challenges.

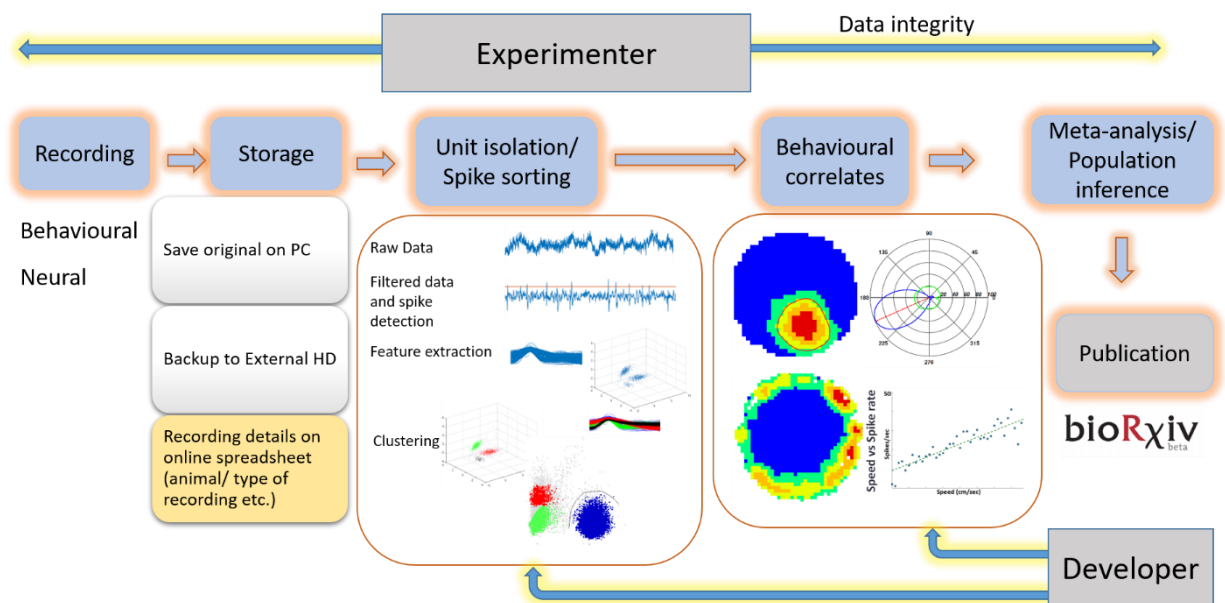


Figure 4.6 Standard pipeline for the processing of behavioral neurophysiology data from recording to publication.

4.4 Future works

Owing to the rise of big-data in neurophysiology and envisioning the use of cloud computing (Bouchard et al., 2016), future development of NeuroChaT can target a cloud-native version to support distributed computing and work with algorithms to support such technologies. Neo (Garcia et al., 2014) has a support for conversion of electrophysiology data from several copyright formats i.e., Axona, Balckrock, Plexon, NeuroExplorer etc. to HDF5 format. NeuroChaT currently supports Axona and Neuralynx formats. Integrating other data formats will be useful to provide for the analytic need of scientists using recording systems from a wide range of vendors. Currently, NeuroChaT supports analyses that pertain to assessing the dynamics of neural responses in response to the spatial information. Analysis of stimulus-response dynamics are also widely studied in neurophysiology. Extensive development of event-related analysis using both the LFP and single-unit data will potentially open the door for wide-spread reception among the neurophysiologists. We currently do not provide tools for visualizing the raw behavioral or neural data and the spike-sorted data. Therefore, we may further consider integration of tools necessary for such visualizations.

4.5 References

- AKIL, H., MARTONE, M. E. & VAN ESSEN, D. C. 2011. Challenges and Opportunities in Mining Neuroscience Data. *Science*, 331, 708-712.
- ANONYMOUS 2013. Announcement: Reducing our irreproducibility. *Nature*, 496.
- ASSANTE, M., CANDELA, L., CASTELLI, D. & TANI, A. 2016. Are Scientific Data Repositories Coping with Research Data Publishing? *Data Science Journal*, 15.
- BARNES, N. 2010. Publish your computer code: it is good enough. *Nature*, 467, 753.
- BOUCHARD, KRISTOFER E., AIMONE, JAMES B., CHUN, M., DEAN, T., DENKER, M., DIEMANN, M., DONOFRIO, DAVID D., FRANK, LOREN M., KASTHURI, N., KOCH, C., RUEBEL, O., SIMON, HORST D., SOMMER, FRIEDRICH T. & PRABHAT 2016. High-Performance Computing in Neuroscience for Data-Driven Discovery, Integration, and Dissemination. *Neuron*, 92, 628-631.
- BRAKEWOOD, B. & POLDRACK, R. A. 2013. The ethics of secondary data analysis: Considering the application of Belmont principles to the sharing of neuroimaging data. *NeuroImage*, 82, 671-676.
- BUTTON, K. S., IOANNIDIS, J. P. A., MOKRYSZ, C., NOSEK, B. A., FLINT, J., ROBINSON, E. S. J. & MUNAFÒ, M. R. 2013. Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, 14, 365.
- CANDELA, L., CASTELLI, D., MANGHI, P. & TANI, A. 2015. Data journals: A survey. *Journal of the Association for Information Science and Technology*, 66, 1747-1762.
- CARP, J. 2012. On the Plurality of (Methodological) Worlds: Estimating the Analytic Flexibility of fMRI Experiments. *Frontiers in Neuroscience*, 6, 149.
- CHOUDHURY, S., FISHMAN, J. R., MCGOWAN, M. L. & JUENGST, E. T. 2014. Big data, open science and the brain: lessons learned from genomics. *Front Hum Neurosci*, 8, 239.
- COSTAFREDA, S. 2009. Pooling fMRI data: meta-analysis, mega-analysis and multi-center studies. *Frontiers in Neuroinformatics*, 3, 33.
- CROSAS, M. 2011. The Dataverse Network: An Open-source Application for Sharing, Discovering and Preserving Data. *D-Lib Magazine*, Volume 17.
- DECO, G., TONONI, G., BOLY, M. & KRINGELBACH, M. L. 2015. Rethinking segregation and integration: contributions of whole-brain modelling. *Nature Reviews Neuroscience*, 16, 430.
- EDITORIAL 2009. Ensuring data integrity. *Nature Neuroscience*, 12, 1205.
- EGLER, S. J., MARWICK, B., HALCHENKO, Y. O., HANKE, M., SUFI, S., GLEESON, P., SILVER, R. A., DAVISON, A. P., LANYON, L., ABRAMS, M., WACHTLER, T., WILLSHAW, D. J., POUZAT, C. & POLINE, J.-B. 2017. Toward standard practices for sharing computer code and programs in neuroscience. *Nature Neuroscience*, 20, 770.
- FERGUSON, A. R., NIELSON, J. L., CRAGIN, M. H., BANDROWSKI, A. E. & MARTONE, M. E. 2014. Big data from small data: data-sharing in the 'long tail' of neuroscience. *Nat Neurosci*, 17, 1442-7.
- FREEMAN, J. 2015. Open source tools for large-scale neuroscience. *Curr Opin Neurobiol*, 32, 156-63.
- GARCIA, S., GUARINO, D., JAILLET, F., JENNINGS, T., PROPPER, R., RAUTENBERG, P. L., RODGERS, C. C., SOBOLEV, A., WACHTLER, T., YGER, P. & DAVISON, A. P. 2014. Neo: an object model for handling electrophysiology data in multiple formats. *Front Neuroinform*, 8, 10.
- GLEESON, P., DAVISON, A. P., SILVER, R. A. & ASCOLI, G. A. 2017. A Commitment to Open Source in Neuroscience. *Neuron*, 96, 964-965.
- GORDON, E. 2003. Integrative Neuroscience. *Neuropsychopharmacology*, 28, S2.
- HELMER, K. G., AMBITE, J. L., AMES, J., ANANTHAKRISHNAN, R., BURNS, G., CHERVENAK, A. L., FOSTER, I., LIMING, L., KEATOR, D., MACCIARDI, F., MADDURI, R., NAVARRO, J. P., POTKIN, S., ROSEN, B., RUFFINS, S., SCHULER, R., TURNER, J. A., TOGA, A., WILLIAMS, C. & KESSELMAN, C. 2011. Enabling collaborative research using the Biomedical Informatics Research Network (BIRN). *J Am Med Inform Assoc*, 18, 416-22.
- INSEL, T. R., LANDIS, S. C. & COLLINS, F. S. 2013. The NIH BRAIN Initiative. *Science*, 340, 687.

- JANKOWSKI, M. M., ISLAM, M. N., WRIGHT, N. F., VANN, S. D., ERICHSEN, J. T., AGGLETON, J. P. & O'MARA, S. M. 2014. Nucleus reuniens of the thalamus contains head direction cells. *Elife*, 3.
- KEIFER, J. & SUMMERS, C. H. 2016. Putting the “Biology” Back into “Neurobiology”: The Strength of Diversity in Animal Model Systems for Neuroscience Research. *Frontiers in Systems Neuroscience*, 10.
- KENALL, A., EDMUNDS, S., GOODMAN, L., BAL, L., FLINTOFT, L., SHANAHAN, D. R. & SHIPLEY, T. 2015. Better reporting for better research: a checklist for reproducibility. *Genome Biol*, 16, 141.
- KRAKER, P., LEONY, D., REINHARDT, W., NBSP, #252 & BEHAM, N. 2011. The case for an open science in technology enhanced learning. *Int. J. Technol. Enhanc. Learn.*, 3, 643-654.
- LISBERGER, S. G. 2013. Sound the Alarm: Fraud in Neuroscience. *Cerebrum: the Dana Forum on Brain Science*, 2013, 6.
- LLORET-VILLAS, A., DAUDIN, R. & LE NOVÈRE, N. 2016. “Big Data” in neuroscience: open door to a more comprehensive and translational research. *Big Data Analytics*, 1.
- MANGHI, P., BOLIKOWSKI, L., MANOLA, N., SCHIRRWAGEN, J. & SMITH, T. 2012. OpenAIREplus: the European Scholarly Communication Data Infrastructure. *D-Lib Magazine*, 18.
- MARKRAM, H., MEIER, K., LIPPERT, T., GRILLNER, S., FRACKOWIAK, R., DEHAENE, S., KNOLL, A., SOMPOLINSKY, H., VERSTREKEN, K., DEFELIPE, J., GRANT, S., CHANGEUX, J.-P. & SARIA, A. 2011. Introducing the Human Brain Project. *Procedia Computer Science*, 7, 39-42.
- MATULEWICZ, P., ULRICH, K., MATHIASSEN, M., AGGLETON, J. & O'MARA, S. 2018. Proximal perimeter encoding in the rat rostral thalamus. *bioRxiv*.
- NATIONAL ACADEMY OF SCIENCES, N. A. O. E., INSTITUTE OF MEDICINE COMMITTEE ON ENSURING THE U. & INTEGRITY OF RESEARCH DATA IN A DIGITAL, A. 2009. *Ensuring the Integrity, Accessibility, and Stewardship of Research Data in the Digital Age*. Washington (DC): National Academies Press (US)
- Copyright 2009 by the National Academy of Sciences. All rights reserved.
- OPEN SCIENCE COLLABORATION 2015. Estimating the reproducibility of psychological science. *Science*, 349.
- PERKEL, J. 2016. Democratic databases: science on GitHub. *Nature*, 538, 127-128.
- POLDRACK, R. A. & GORGOLEWSKI, K. J. 2014. Making big data open: data sharing in neuroimaging. *Nat Neurosci*, 17, 1510-7.
- POLINE, J.-B., BREEZE, J., GHOSH, S., GORGOLEWSKI, K., HALCHENKO, Y., HANKE, M., HELMER, K., MARCUS, D., POLDRACK, R., SCHWARTZ, Y., ASHBURNER, J. & KENNEDY, D. 2012. Data sharing in neuroimaging research. *Frontiers in Neuroinformatics*, 6.
- POLINE, J. 2018. From data sharing to data publishing [version 1; referees: 1 approved, 1 approved with reservations]. *MNI Open Research*, 2.
- SEJNOWSKI, T. J., CHURCHLAND, P. S. & MOVSHON, J. A. 2014. Putting big data to good use in neuroscience. *Nat Neurosci*, 17, 1440-1.
- SIEGLE, J. H., HALE, G. J., NEWMAN, J. P. & VOIGTS, J. 2015. Neural ensemble communities: open-source approaches to hardware for large-scale electrophysiology. *Curr Opin Neurobiol*, 32, 53-9.
- VAN HORN, J. D. & TOGA, A. W. 2014. Human Neuroimaging as a “Big Data” Science. *Brain imaging and behavior*, 8, 323-331.
- VOGELSTEIN, J. T., PARK, Y., OHYAMA, T., KERR, R. A., TRUMAN, J. W., PRIEBE, C. E. & ZLATIC, M. 2014. Discovery of brainwide neural-behavioral maps via multiscale unsupervised structure learning. *Science*, 344, 386-92.
- WEINER, M. W., VEITCH, D. P., AISEN, P. S., BECKETT, L. A., CAIRNS, N. J., GREEN, R. C., HARVEY, D., JACK, C. R., JAGUST, W., LIU, E., MORRIS, J. C., PETERSEN, R. C., SAYKIN, A. J., SCHMIDT, M. E., SHAW, L., SIUCIAK, J. A., SOARES, H., TOGA, A. W. & TROJANOWSKI, J. Q. 2012. The Alzheimer's Disease Neuroimaging Initiative: a review of papers published since its inception. *Alzheimers Dement*, 8, S1-68.

- WIENER, M., SOMMER, F. T., IVES, Z. G., POLDRACK, R. A. & LITT, B. 2016. Enabling an Open Data Ecosystem for the Neurosciences. *Neuron*, 92, 617-621.
- ZEHL, L., JAILLET, F., STOEWER, A., GREWE, J., SOBOLEV, A., WACHTLER, T., BROCHIER, T. G., RIEHLE, A., DENKER, M. & GRUN, S. 2016. Handling Metadata in a Neurophysiology Laboratory. *Front Neuroinform*, 10, 26.

Appendix

A. 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:

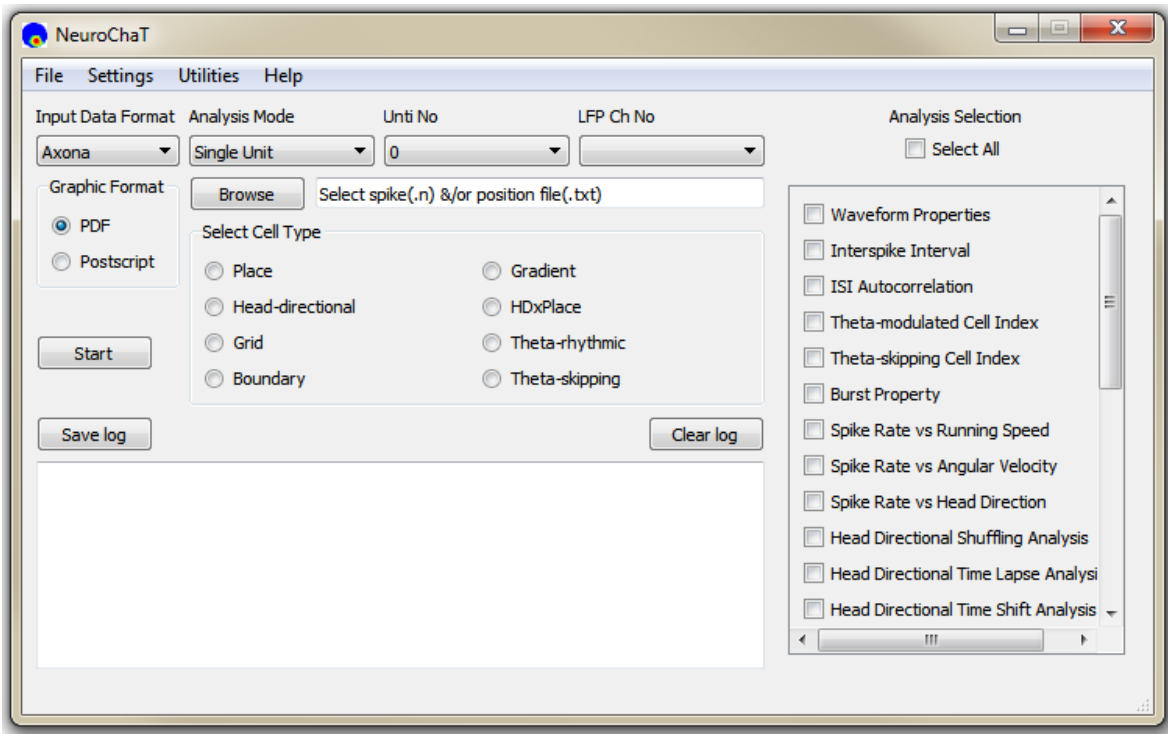


Figure 0.1: NeuroChaT graphical user interface

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.

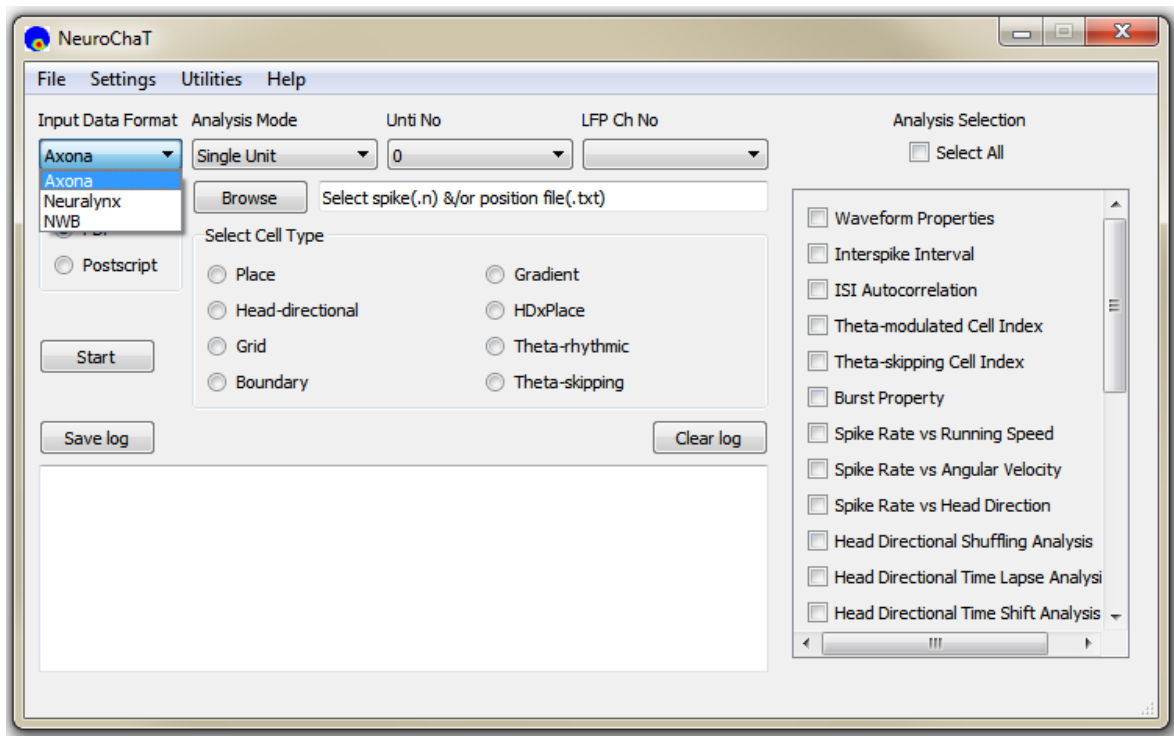


Figure 0.2: Select the input file format from one of the three options in the 'Input Data Format' dropdown menu.

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 analyze 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)

The figure shows two Excel spreadsheets. The top spreadsheet is for HDF5 data with columns: dir, nwb, cell id, lfp. The bottom spreadsheet is for Axona and Neuralynx data with columns: Directory, position file, Spike file, cell id, lfp chan.

	A	B	C	D
1	dir	nwb	cell id	lfp
2	C:\Users\██████████	120213_26	6	eeg

	A	B	C	D	E
1	Directory	position file	Spike file	cell id	lfp chan
2	C:\Users\██████████	120213_26_1	120213_26.6	4	eeg

Figure 0.3: 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 the information from there as .txt file.

Unit No

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

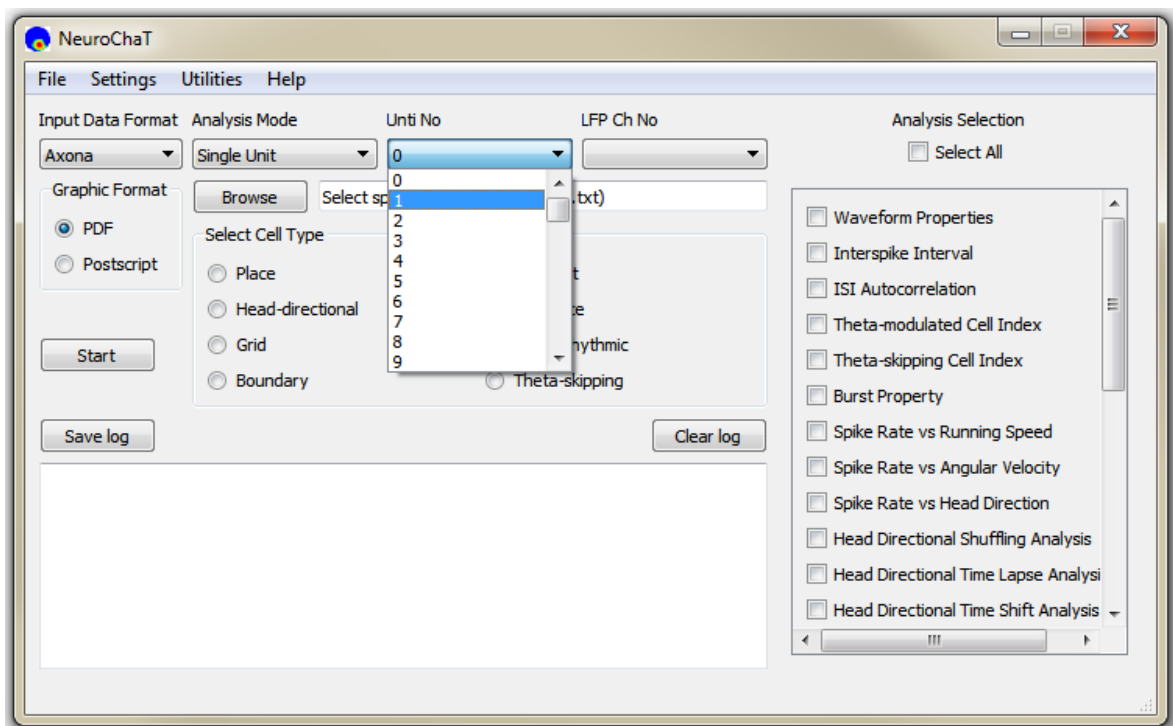


Figure 0.4 Dropdown list for the selection of cell number.

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.

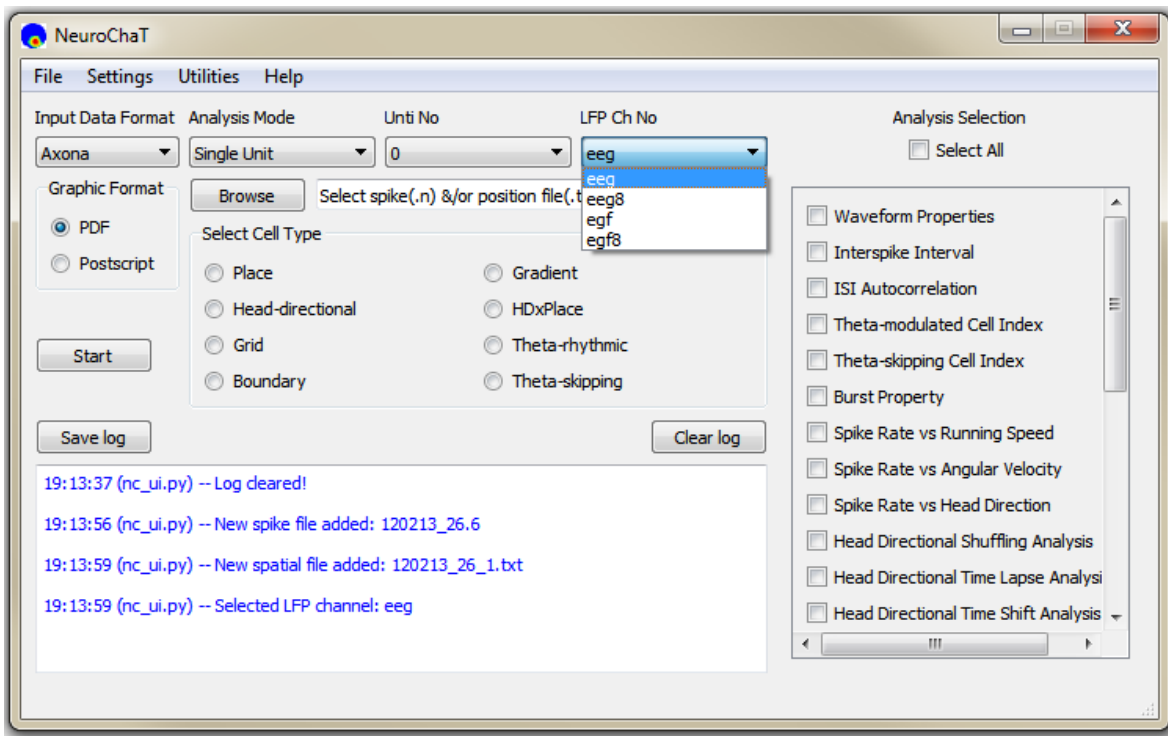


Figure 0.5: Dropdown list to select the LFP channel

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 characterizes 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 color 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 6**. Clicking on the 'Export results' prompts the user to save the tabular data in Excel file (**Figure 6, Bottom**).

	Mean Spiking Freq	Mean width	Mean amplitude	Mean height	Std height	Std width
TT6_SS_4_eeg	9.73022481266	241.153944891	203.199721643	204.495651026	21.4653091143	64.52079776
TT3_SS_1_eeg	21.0133222315	126.49912166	241.516174434	241.710288474	18.5259256623	13.3141324968
TT7_SS_2_eeg	0.706666666667	358.195754717	172.367887677	182.170939895	27.0331326065	48.0785555413
TT5_SS_1_eeg	35.682764363	189.698790495	357.959282513	743.145034856	25.8458682168	15.9444303437

	A	B	C	D	E
		Mean Spiking Freq	Mean width	Mean amplitude	Mean height
1					
2	TT6_SS_4_eeg	9.730224813	241.1539449	203.1997216	204.495651
3	TT3_SS_1_eeg	21.01332223	126.4991217	241.5161744	241.7102885
4	TT7_SS_2_eeg	0.706666667	358.1957547	172.3678877	182.1709399
5	TT5_SS_1_eeg	35.68276436	189.6987905	357.9592825	743.1450349
6	TT4_SS_1_eeg	54.81681932	144.4640642	148.4015041	323.181652

Figure 0.6 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 option 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.

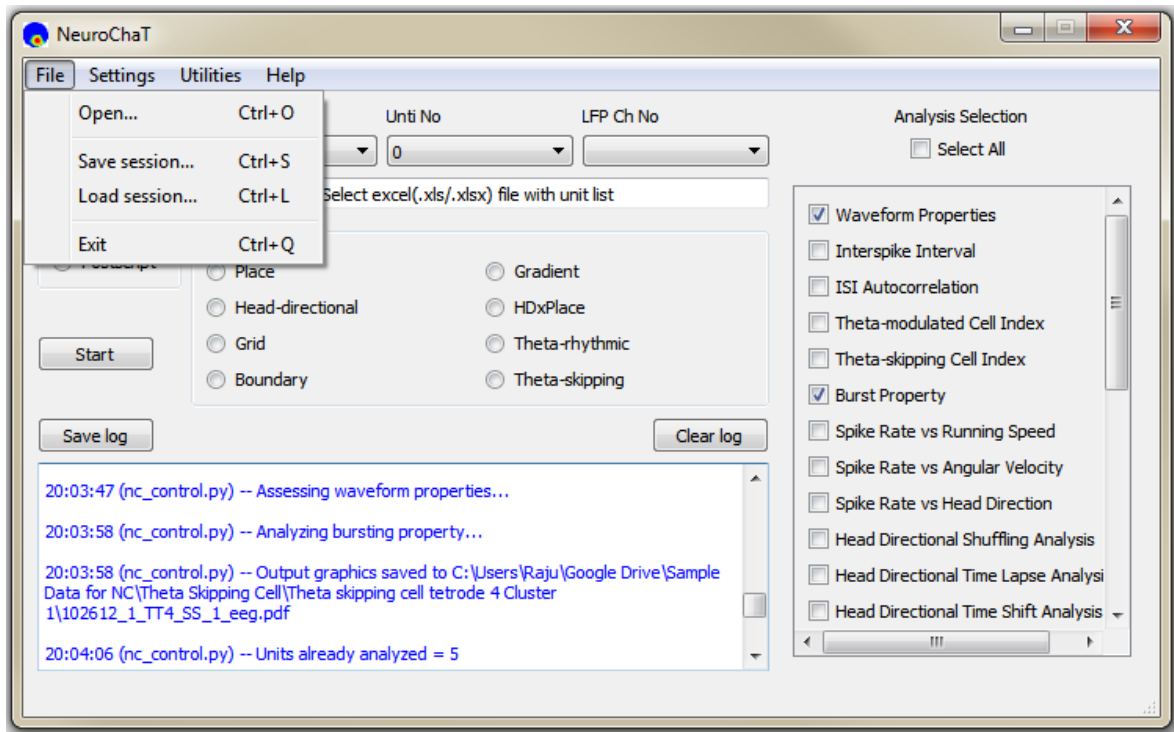


Figure 0.7 'File' menu items. Shortcuts for invoking each of these actions are also shown.

Settings

Settings menu has one item: *Parameters*. Clicking this item initiates the parameter selection box as shown in **Figure 8**. Analysis specific parameters and their accepted values are listed in **Appendix C**. Clicking on an item on the left panel will display the parameter setting panel for that analysis on the right panel.

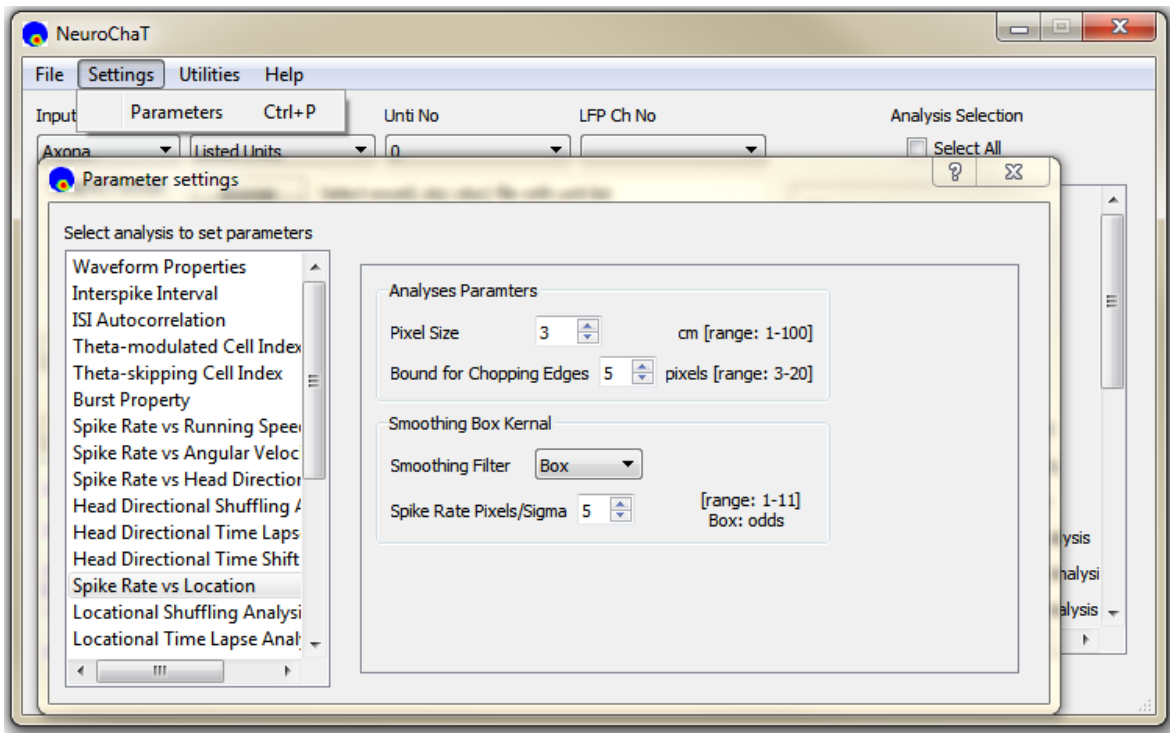


Figure 0.8: Window for analysis specific input parameter settings.

Utilities

This menu comprises of useful NC utilities described below (**Figure 9**):

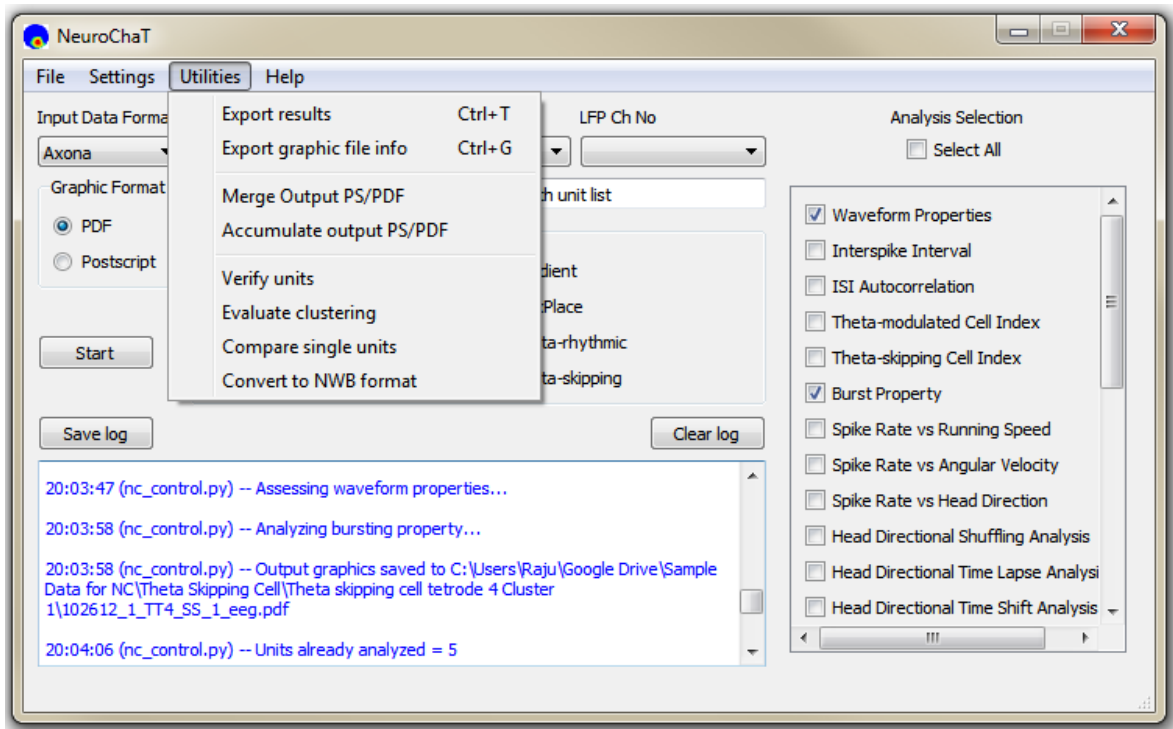


Figure 0.9 Items in the 'Utilities' menu.

- 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:

	A	B	C
1		Graphics Files	NWB Files
2	TT6_SS_4_eeg	\\120213_26_TT6_SS_4_eeg.pdf	\\120213_26.hdf5
3	TT3_SS_1_eeg	\\120412_1_TT3_SS_1_eeg.pdf	\\120412_1.hdf5
4	TT7_SS_2_eeg	\\062612_1_TT7_SS_2_eeg.pdf	\\062612_1.hdf5
5	TT5_SS_1_eeg	\\112512_1_TT5_SS_1_eeg.pdf	\\112512_1.hdf5
6	TT4_SS_1_eeg	\\102612_1_TT4_SS_1_eeg.pdf	\\102612_1.hdf5

Figure 0.10 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 II**: 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.

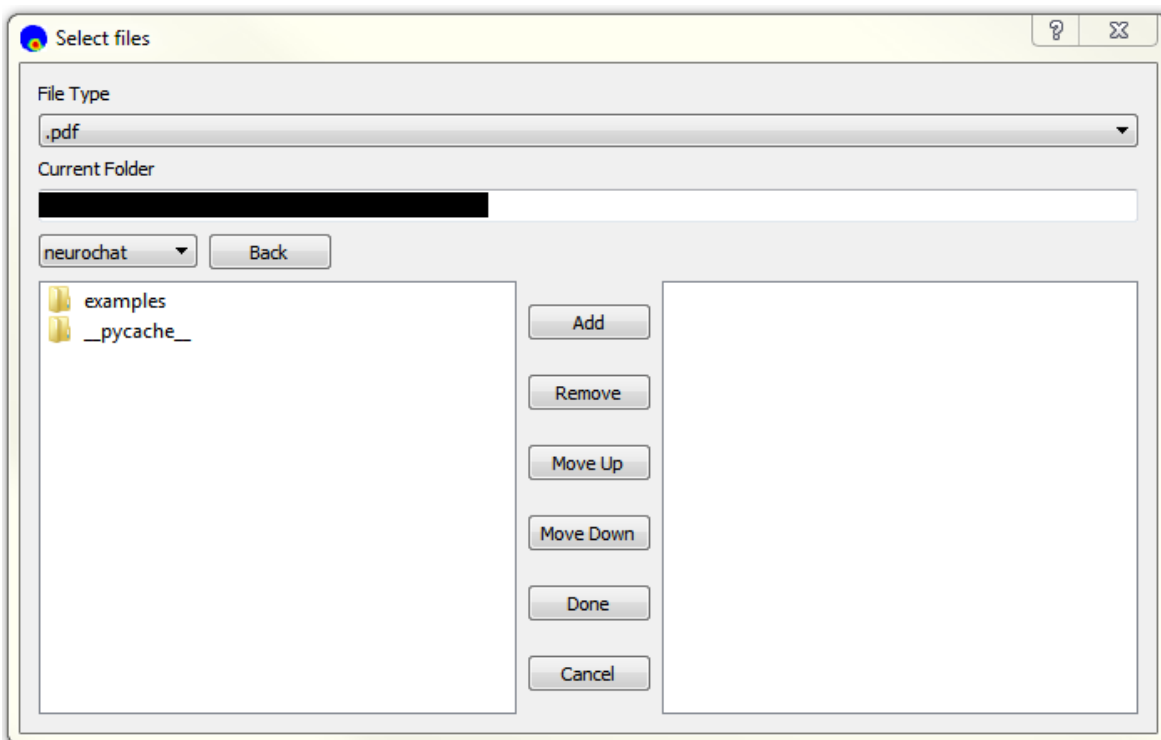
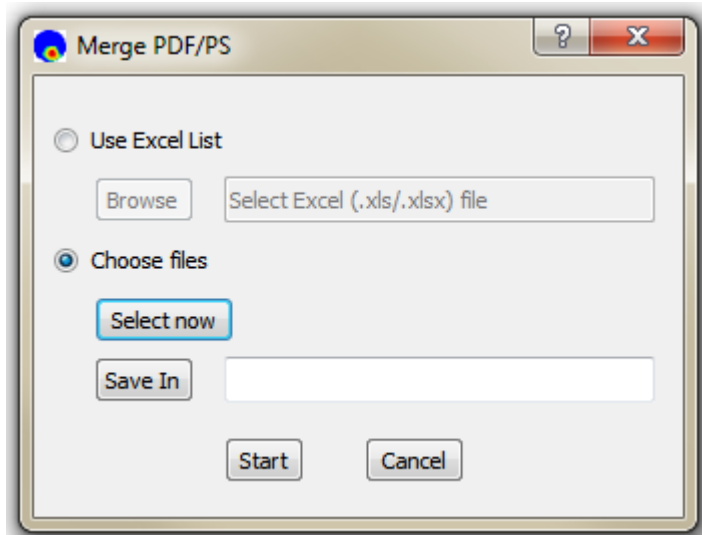


Figure 0.II: Upper row: Merge file utility showing the options of 'Use Excel List' or 'Choose Files'. Bottom Row: The file-picker for selecting files to merge

d. Accumulate output PDF/PS: This utility help accumulating the analyzed 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 11, 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 12**. Output of this analysis adds the last two columns that shows if the file and the unit in that file exists.

	A	B	C	D	E	F
1		Directory	Spike file	cell id	fileExists	unitExists
2	0		120213_26.6	4	TRUE	TRUE
3	1		120412_1.3	1	TRUE	TRUE
4	2		062612_1.7	2	TRUE	TRUE
5	3		112512_1.5	1	TRUE	TRUE
6	4		102612_1.4	1	TRUE	TRUE

Figure 0.12: 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 13**.

	A	B	C	D	E	F
1		Directory	Spike file	cell id	BC	Dh
2	0		120412_1.3	1	1.16E-12	1
3	1		062612_1.7	2	5.04E-08	1
4	2		112512_1.5	1	8.72E-19	1

Figure 0.13: Sample Excel file input for cluster evaluation. The analysis outputs 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 χ^2 -similarity) or their statistical distance (Hellinger distance). Sample input style and results are shown in **Figure 14**.

	A	B	C	D	E	F	G	H	I
1	Directory 1	Spike File 1	Cell 1	Directory 2	Spike File 2	Cell 2	BhattCoeff	Hellinger	Chi Sq Sim
2		061516_ELSSW1.4	1		061716_ELSNW2.4	1	0.393037264	0.779078	0.31384119
3		061516_ELSSW1.4	1		061816_ELSW3.4	1	0.430171872	0.75487	0.32798797
4		061516_ELSSW1.4	1		061516_ELSSW1.4	1	1	0	1

Figure 0.14: Sample Excel file input for cluster similarity measurement. Two sets of specifications are required for comparison. The analysis outputs 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 15**, 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.

	A	B	C	D
1	Directory	Behavioural file	Spike file	lfp chan or file
2		120213_26_1	120213_26.6	eeg
3		120412_1_1	120412_1.3	eeg
4		062612_1_1	062612_1.7	eeg

Figure 0.15 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.

B. 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](#) or [GitHub](#)

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 CO and SO 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/Behavioral/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/Behavioral/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

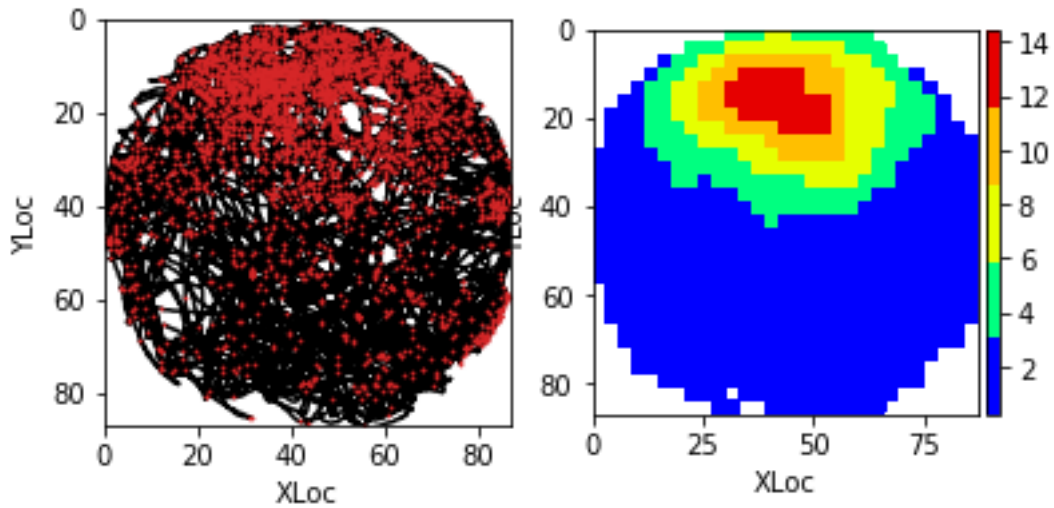
```
placeData= spat.place(spike.get_unit_stamp(), pixel = 3, filter = ['b', 5])
```

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)
```

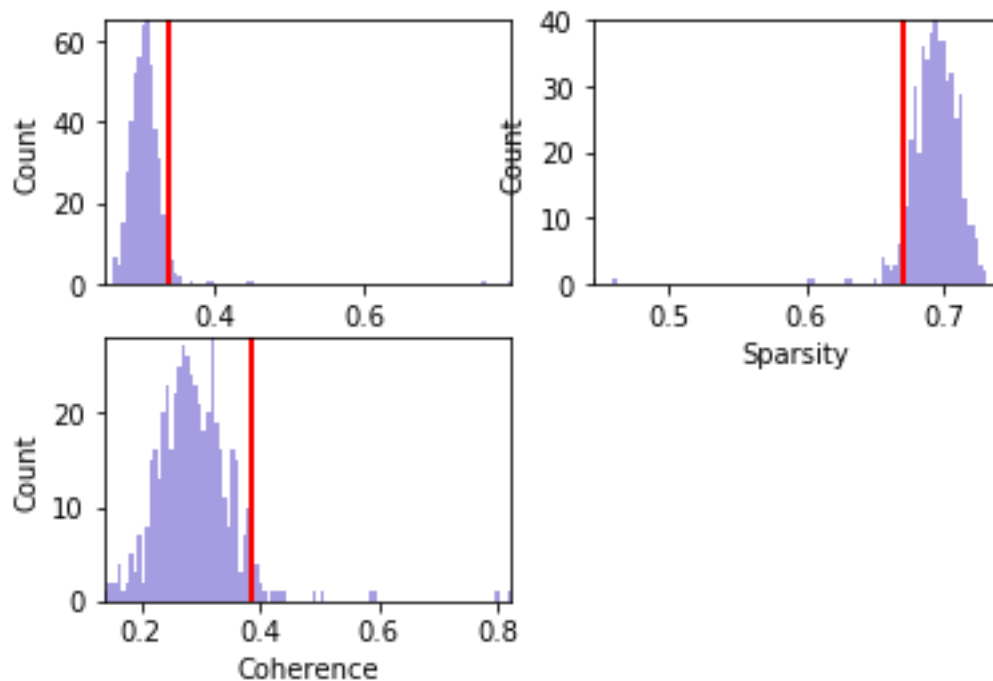


Analysis and plotting of locational shuffling analysis using default parameters

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)
```

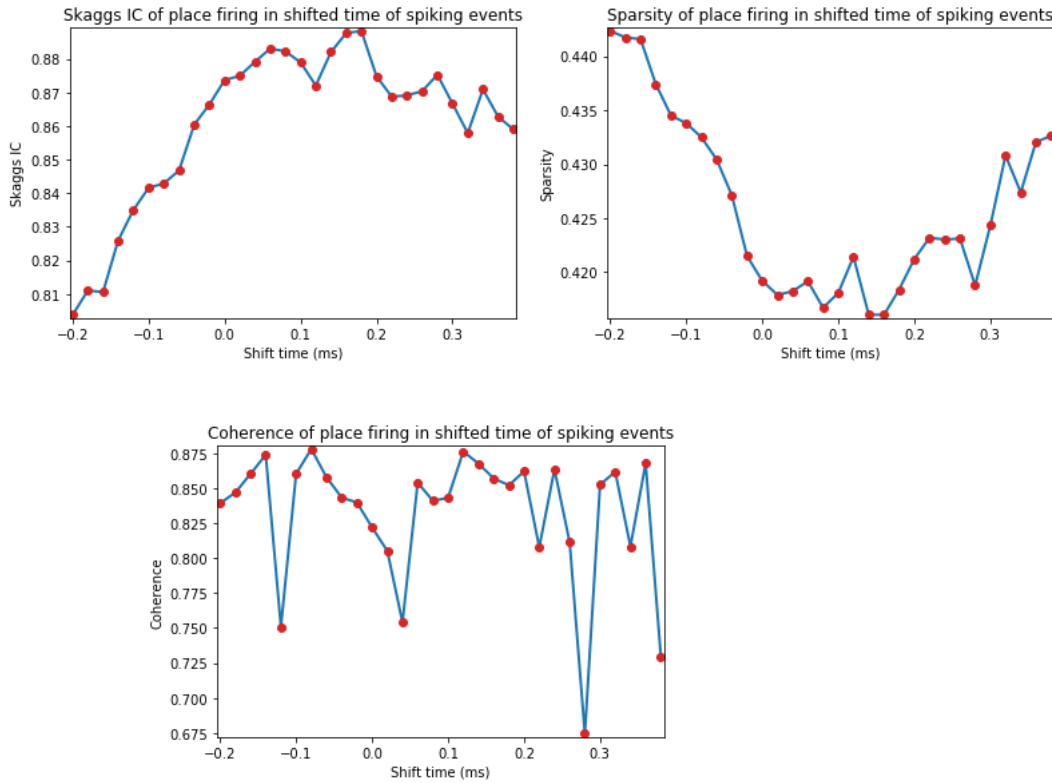
Distribution of locational firing specificity indices



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

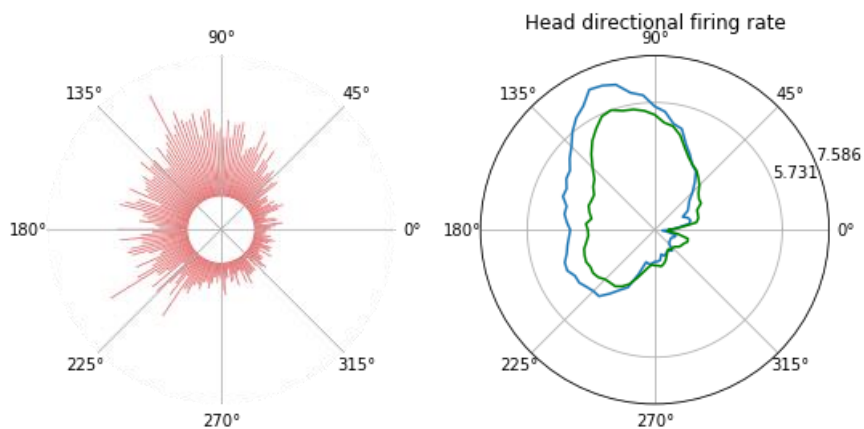
Spike timestamps are gradually shifted from -10 to +20 units of spatial time-resolution. If the video for tracking animal behaviour is sampled at 50Hz, this means the spike-train is shifted from -200ms to +400ms

```
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)
```



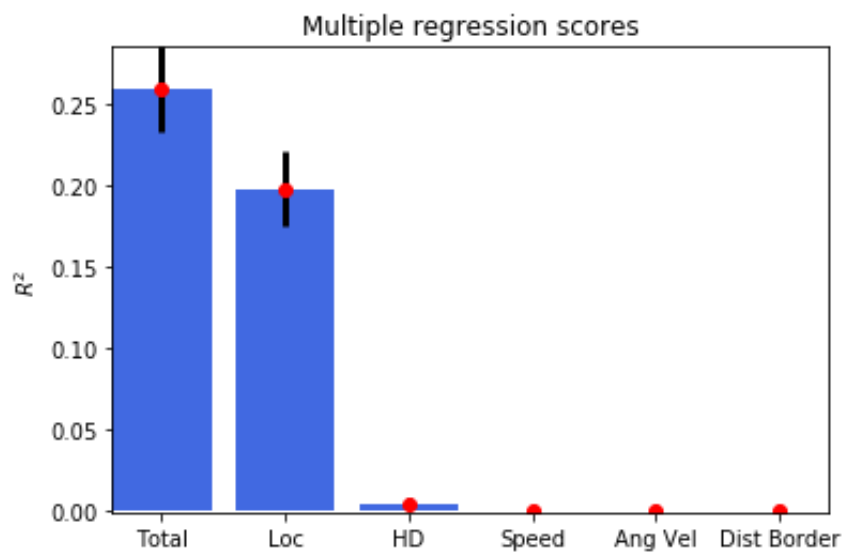
Head directional analysis of this unit

```
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/Behavioral/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.

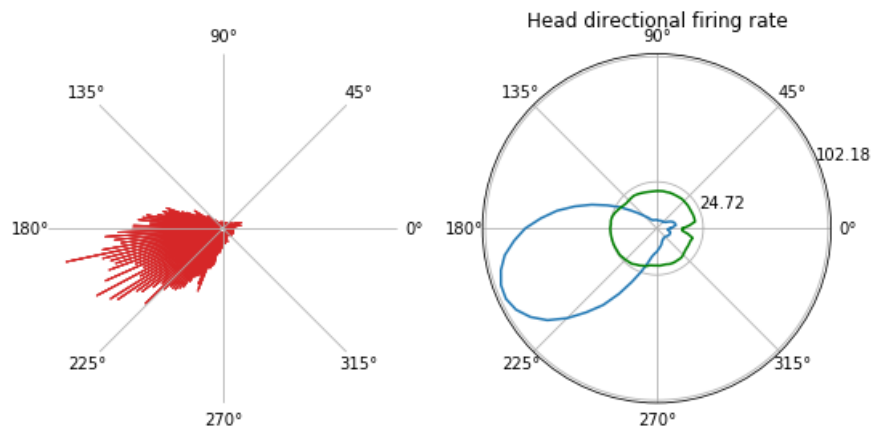
```
ndata.reset_results()
```

Or, results can be reset using individual data objects

```
spat.reset_results()  
spike.reset_results()
```

Head-directional firing rate analysis and plot

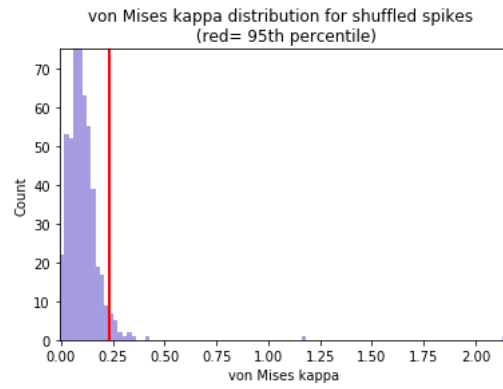
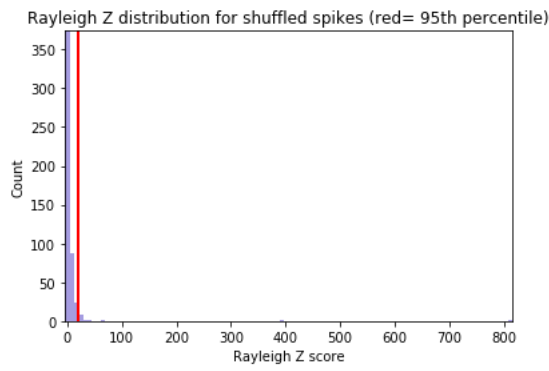
```
hdData= nndata.hd_rate()  
fig= nc_plot.hd_firing(hdData)
```



Head directional shuffling analysis and plot

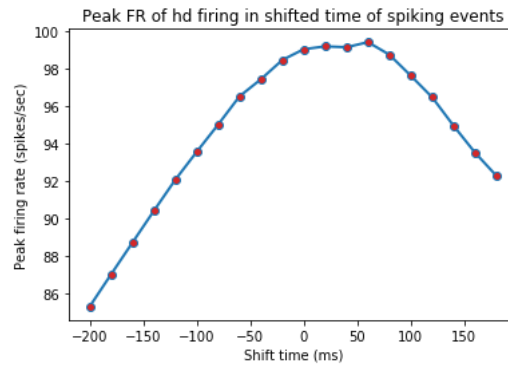
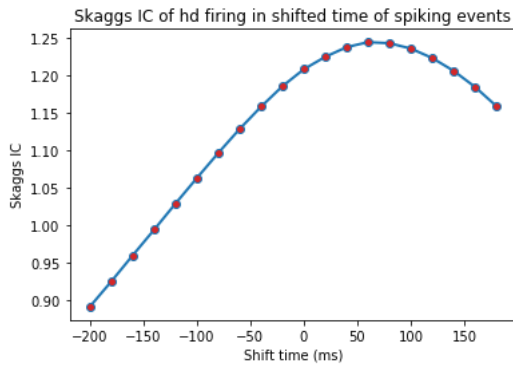
Number of bins for the histogram of the shuffled data is set to 100

```
hshuffleData= nndata.hd_shuffle(nshuff = 500, limit=0, bins= 100)  
fig= nc_plot.hd_shuffle(hshuffleData)
```



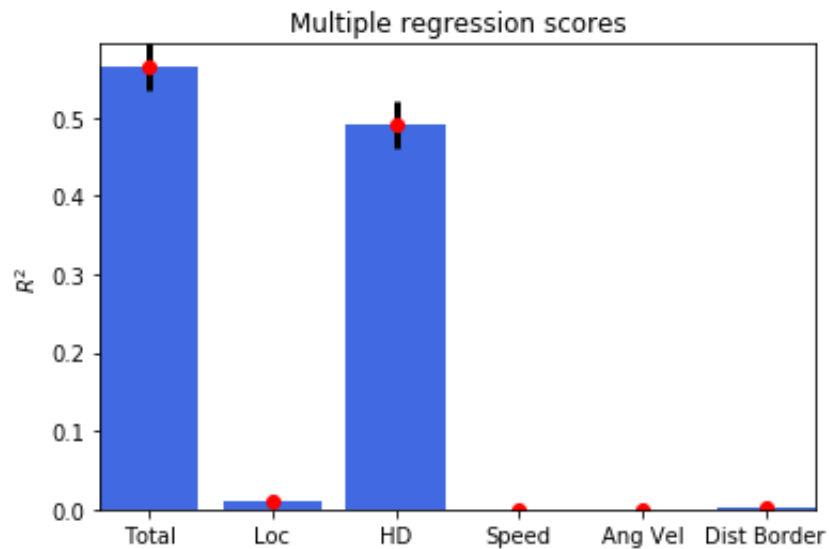
Head directional time-shift analysis

```
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/Behavioral/Position')  
spike.set_filename(data_dir + '112512_1.hdf5+/processing/Shank/5')
```

```
spat.load()  
spike.load()
```

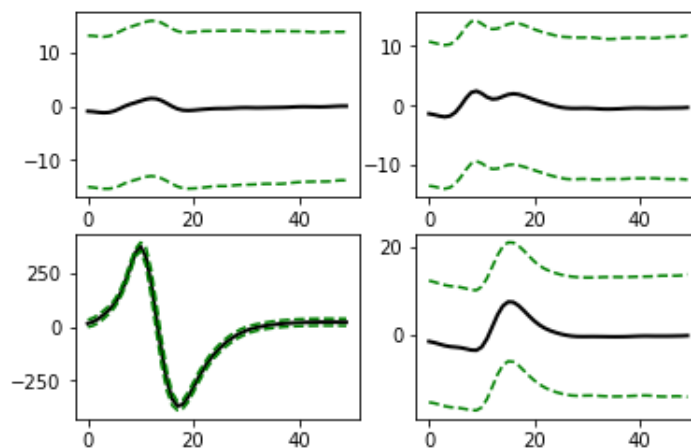
```
spike.set_unit_no(1)
```

Reset results to omit parametric output of previously analysed unit

```
ndata.reset_results()
```

Waveform properties of the unit

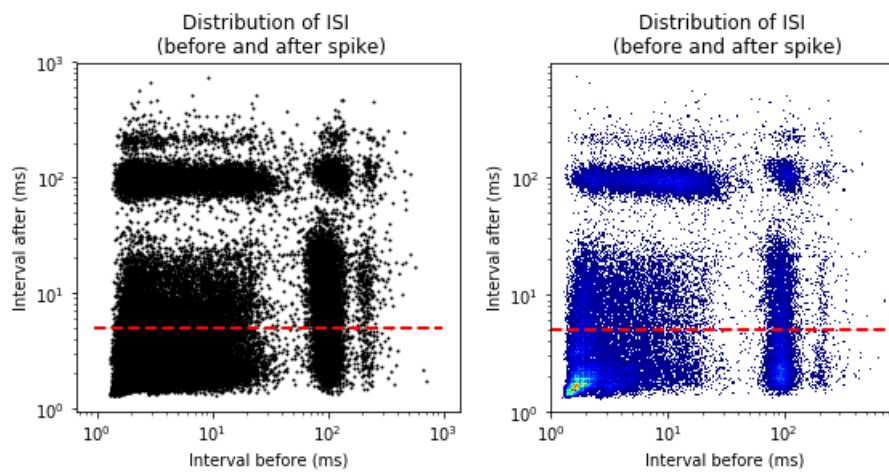
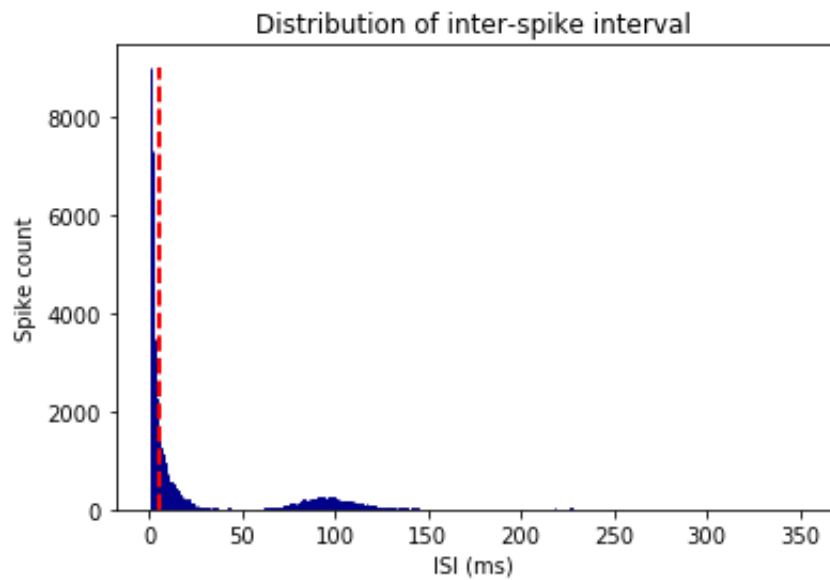
```
graphData= ndata.wave_property()  
fig= nc_plot.wave_property(graphData, [int (spike.get_total_channels()/2), 2])
```



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 repeatedly from now on for reusing the memory

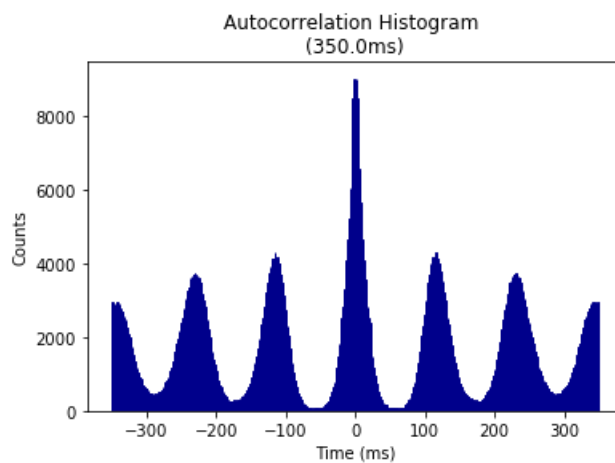
```
graphData= ndata.isi(bins = 350, bound = [0, 350])  
fig= nc_plot.isi(graphData)
```



ISI autocorrelation histogram for longer length

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

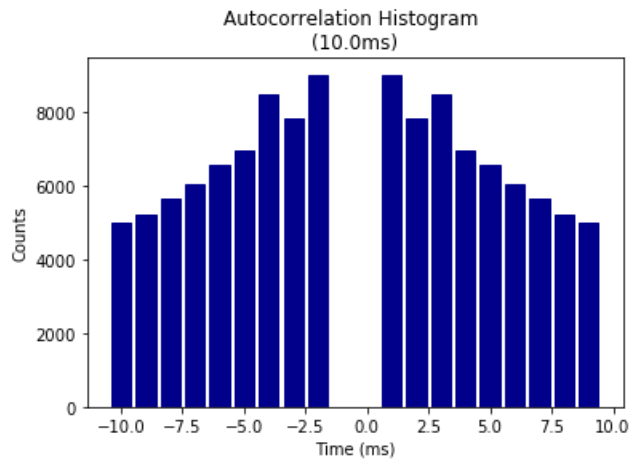
```
graphData= ndata.isi_corr(bins = 1, bound = [-350, 350])
fig= nc_plot.isi_corr(graphData)
```



ISI autocorrelation histogram for shorter length

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

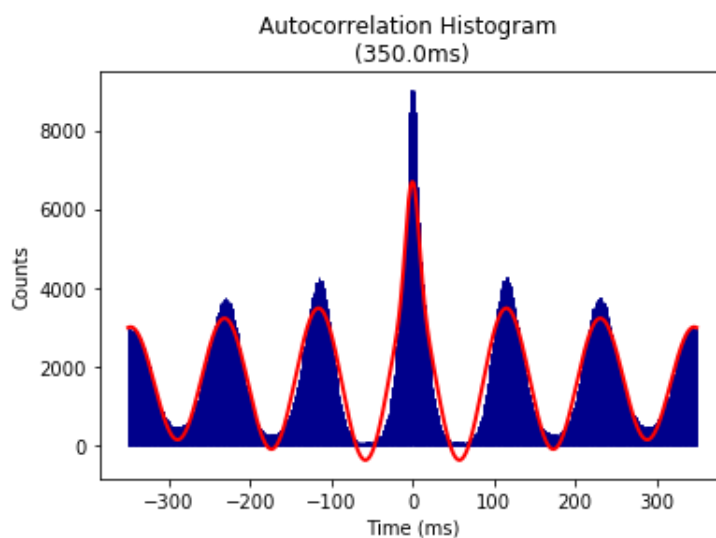
```
graphData= ndata.isi_corr(bins = 1, bound = [-10, 10])  
fig= nc_plot.isi_corr(graphData)
```



Theta modulation Index analysis

Input parameters are for [Frequency, tau1, tau2] and provides the starting value, lower, and upper bound for the fitted sinusoidal equation. Binsize and temporal 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
```

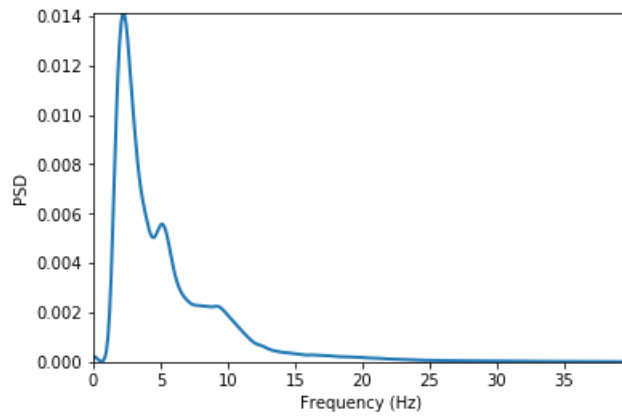
Instantiate LFP data object, set the filename, load data, and add to ndata

```
lfp= NLfp(system= 'NWB')
lfp.set_filename(data_dir+ '\\112512_1.hdf5+/processing/Neural Continuous/LFP/eg')
lfp.load()
ndata.lfp= lfp
```

LFP frequency spectrum analysis

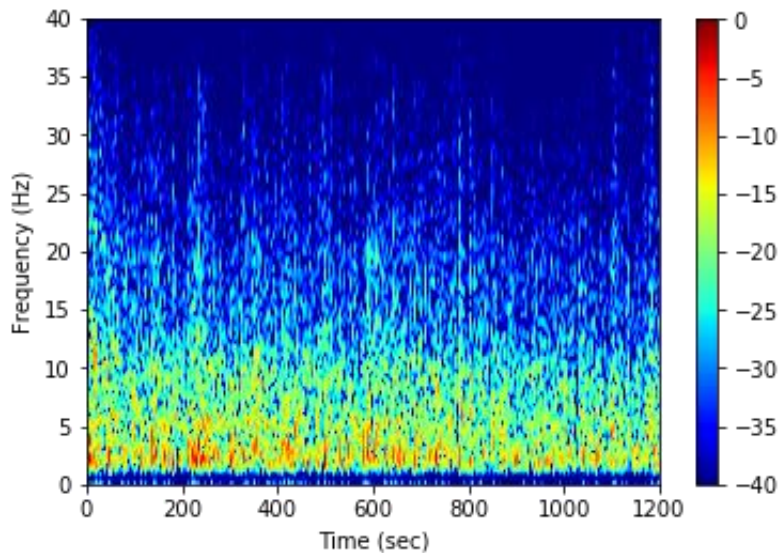
Hanning window of 2sec with 1sec overlap and number of FFT components= 2048. ptype is 'psd' which means power-spectral density. Other option can be 'power'. prefilt set 'True' for pre-filtering the LFP signal with a bandpass filter as set by filtset. filtset= [filter order, lower cutoff frequency, higher cutoff frequency, type of filtering]. fmax defines the maximum frequency to analyse. db set to 'True' will convert the spectrogram in decibel unit. tr set to 'True' creates a time-resolved spectrogram with 'window'-resolution and 'overlap' amount of signal overlap. tr set to 'False' calculates the spectrogram 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`

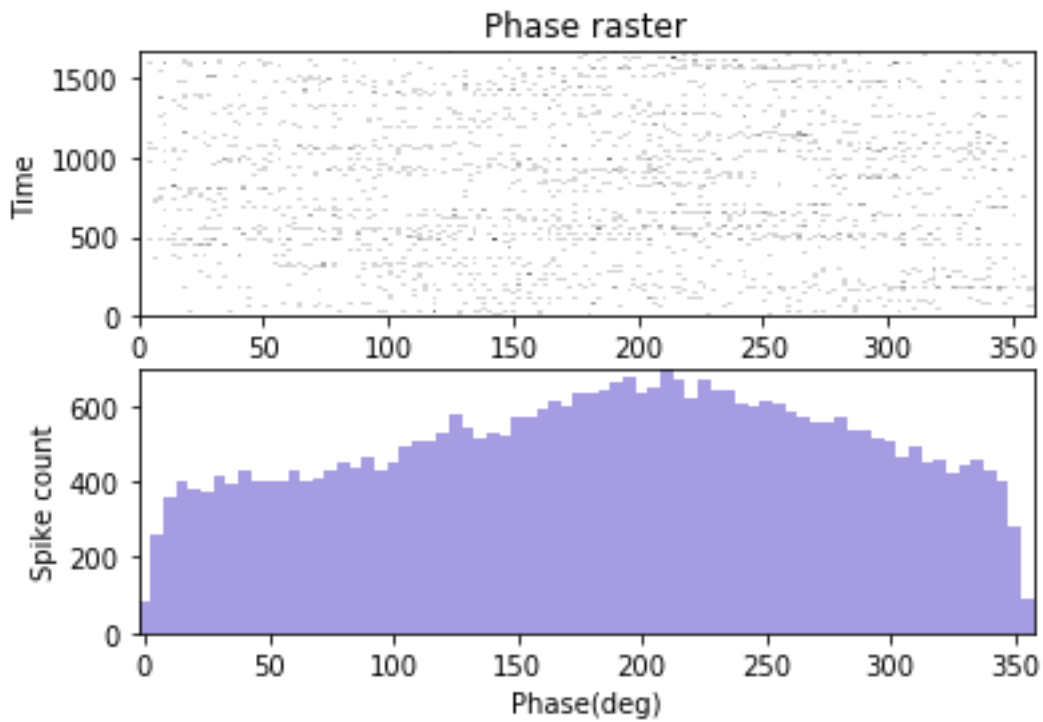
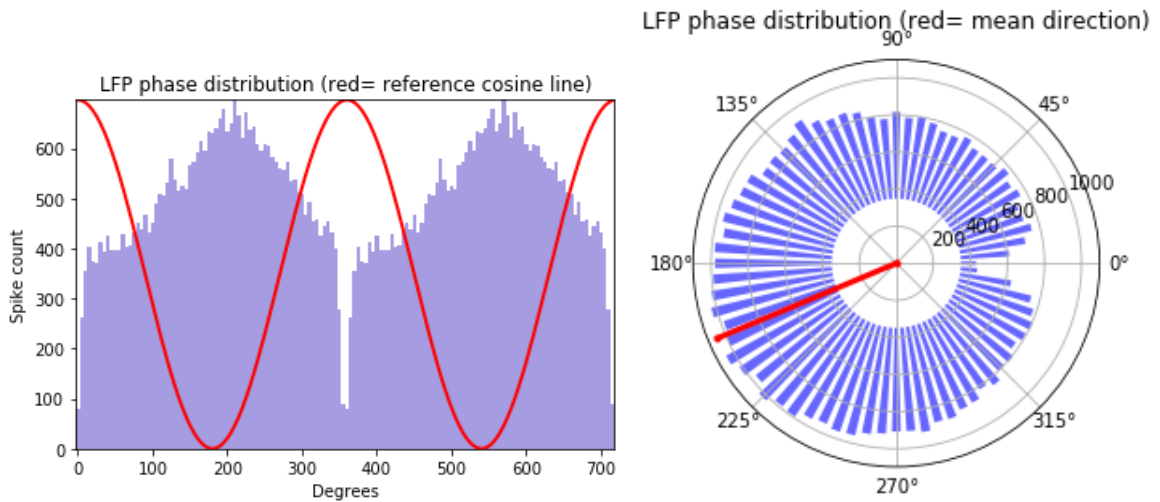
```
graphData= lfp.spectrum(window = 2, noverlap = 1, nfft = 2048, ptype = 'psd', \
    prefilt = True, filtset = [10, 1.5, 40, 'bandpass'], \
    fmax = 40, db = True, tr = True)
fig= nc_plot.lfp_spectrum_tr(graphData)
```



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, fw
in = [6, 12],\
    pratio = 0.1, aratio = 0.15, filtset = [10, 1.5, 40, 'bandpass'])
fig= nc_plot.spike_phase(graphData)
```

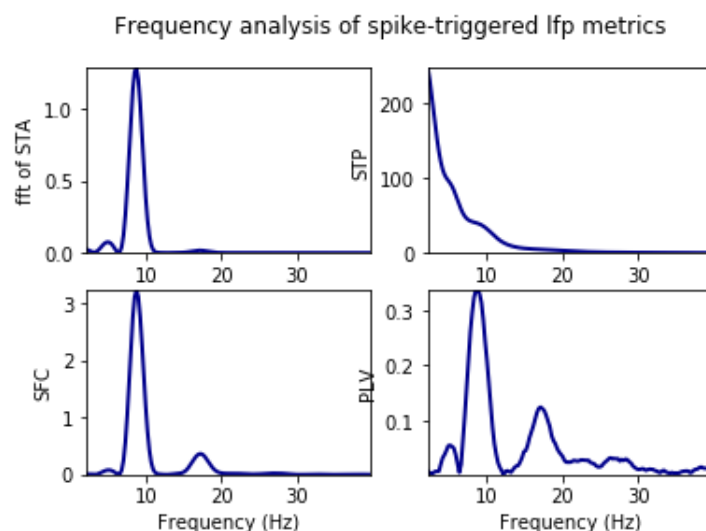
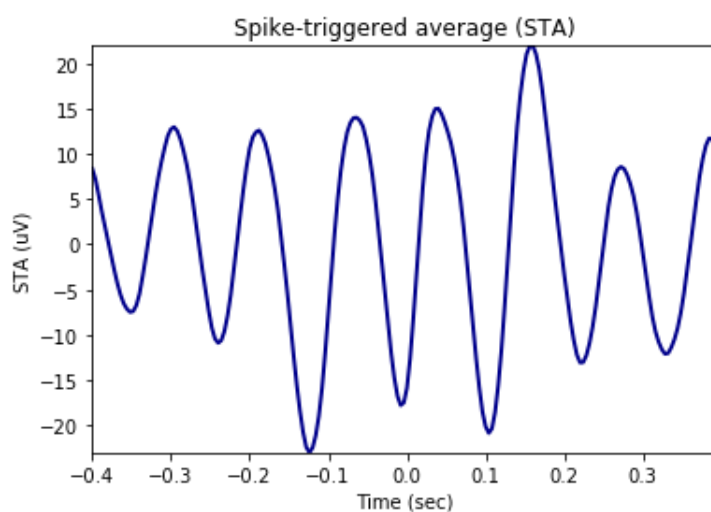
Using the spike object:

```
graphData= spike.phase_dist(lfp = lfp , binsize = 5, rbinsize = 2, fwin = [6, 1
2],\
    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-triggered average (STA)

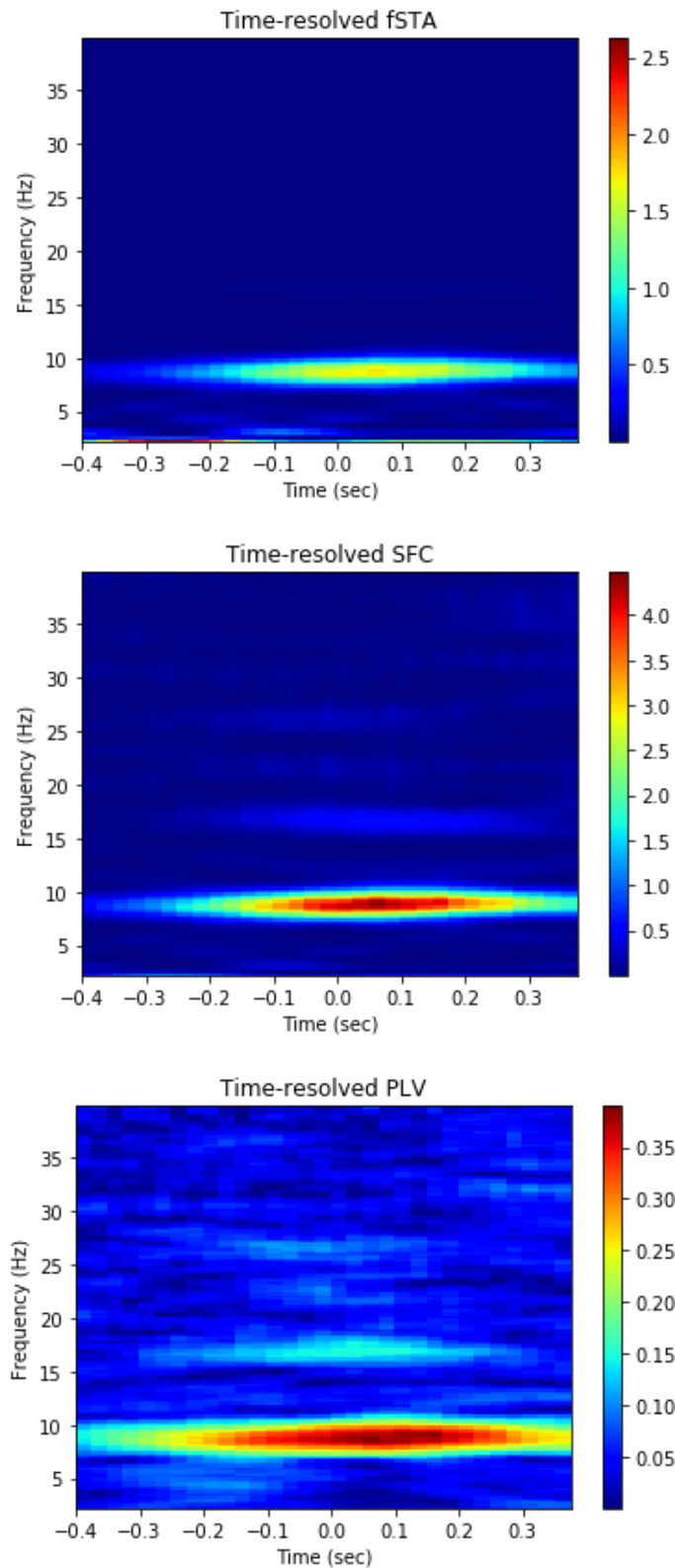
Window of the LFP chunks in reference to the spike timestamps is set to -400ms to +400ms Frequency of interest for the analysis is set as 2Hz to 30Hz

```
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() or 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 individual 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 utilises 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 in 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) # NeuroChaT 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 = dat
e_to_store, create_group = True)
hdf.save_attributes(path= '/path/to/group/or/dataset/', attr = dict_of_attribut
es)

```

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 default 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 skip cell turned off
```

Analyses can be performed in different modes, namely: 1. 'Single Unit'-one cell at a 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 = 'SingleUnit') # 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 analyses. 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 used 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 analyses 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, parametric 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:\Users\Raju\Google Drive\Sample Data for NC...
NWB Files
TT6_SS_4_eeg C:\Users\Raju\Google Drive\Sample 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 w
riting 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

Although we are initialising 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)
```

Evaluating 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
```

C. 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 τ_1 decay constant in curve fitting
thetaCellTau1Start	0.1	0-15	Sec	Starting value of the decay constant τ_1 decay constant in curve fitting
thetaCellTau2Max	0.05	0-0.1	Sec	Upper limit of the decay constant τ_2 decay constant in curve fitting
thetaCellTau2Start	0.05	0-0.1	Sec	Starting value of the decay constant τ_2 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 analyze
speedMax	40	10-200	cm/sec	Maximum limit on the speed to analyze
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 Gompertz function, e.g., ± 0.25
--------------	------	------------	--	--

gradDisplaceLim	0.25	0.1:0.05:1		Range of displacement parameter 'b' on Gompertz function, e.g., ± 0.25
gradGrowthRateLim	0.5	0.1:0.05:1		Range of growth rate parameter 'c' on Gompertz 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

D. 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 taul 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= 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 neighboring 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 center 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 measured using (Sargolini et al., 2006)

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 R^2 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 Gompertz 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

E. Equations for circular statistics

Note: Interested readers are referred to (Mardia and Jupp, 2008) or (Zar, 1999) for detailed explanation.

Following equations apply for circular data, with magnitude R_i at i -th angle θ_i :

$$\text{mean magnitude, } R_n = R_{mean} = \sqrt{\left(\sum_i R_i \cos \theta_i\right)^2 + \left(\sum_i R_i \sin \theta_i\right)^2}$$

$$\text{mean angle, } \theta_{mean} = \arctan\left(\sum_i R_i \sin \theta_i / \sum_i R_i \cos \theta_i\right)$$

$$N = \sum_i R_i$$

$$\text{resultant length, } R_l = R_{mean}/N$$

The Rayleigh's Z-test in circular statistics is performed to test for circular uniformity. The null hypothesis H_0 is that, θ_i are uniformly distributed around a circle, or $R_1 = R_2 = \dots = R_i = \dots = R_m$, where m is the number of angular bins under consideration. Raleigh test assumes that the sample is generated from von Mises distribution. The Raleigh's Z-score and the approximate P-value under H_0 is computed as (Zar, 1999),

$$\text{Rayleigh's } Z = R_n^2/N$$

$$\text{Rayleigh's } P = \exp\left[\sqrt{1 + 4N + 4(N^2 - R_n^2)} - (1 + 2N)\right]$$

The von Mises concentration parameter κ is calculated using the following set of equations (Fisher, 1993),

$$\kappa = \begin{cases} 2R_l + R_l^3 + \frac{5R_l^5}{6}; & R_l < 0.53 \\ -0.4 + 1.39R_l + \frac{0.43}{1 - R_l}; & 0.53 \leq R_l < 0.85 \\ \frac{1}{3R_l - 4R_l^2 + R_l^3}; & R_l \geq 0.85 \end{cases}$$

The smaller κ is, the distribution is closer to uniformity. If κ is zero, the distribution is uniform. A large κ indicates that the distribution is very much centered on the mean angle.

F. Reference

- FISHER, N. I. 1993. *Statistical Analysis of Circular Data*, Cambridge, Cambridge University Press.
- MARDIA, V. K. & JUPP, P. E. 2008. *Directional Statistics*, John Wiley and Sons, Inc.
- SARGOLINI, F., FAYN, M., HAFTING, T., MCNAUGHTON, B. L., WITTER, M. P., MOSER, M. B. & MOSER, E. I. 2006. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312, 758-62.
- ZAR, J. H. 1999. *Biostatistical Analysis*, Upper Saddle River, Prentice Hall.