Statistical Analysis and Modeling of Brain cells' Activity

Thesis submitted for the degree "Doctor of Philosophy"

Itay Gat

Submitted to the Senate of the Hebrew University in Jerusalem (1999)

This thesis is dedicated to my wife Tamar and my children Naama and Nadav.

This work was carried out under the supervision of

Prof. Moshe Abeles and Prof. Naftali Tishby.

Acknowledgments

Special thanks are due to:

- Prof. Naftali Tishby, my supervisor, for his ideas, continuing interest and enthusiasm.
- Prof. Moshe Abeles, who shared his knowledge unstintingly with me, for his important comments on the work in progress.
- Dr. Hagai Bergman and Prof. Eilon Vaadia for sharing their data with me, and for the numerous stimulating and encouraging discussions of this work.
- The Ph.D. students of machine learning group: Shai Fine, Elad Schneidman, Noam Slonim, Yoram Gdalyahu, Dr. Golan Yona, Dr. Shlomo Dubnov, Dr. Ran El-Yaniv and Dr. Lidror Troyansky for their friendship, help and support during this study.
- Aeyal Raz, Yifat Prut, Iris Haalman and Hamutal Slovin for their valuable assistance in the laboratory.
- My wife, Tamar, who now knows more about behaving monkeys than she ever thought possible, for her encouragement, support, and invaluable editing.

This research was supported in part by a grant from the Unites States Israeli Binational Science Foundation (BSF).

Contents

Abstract

One of the intriguing questions in brain research concerns the neural code, i.e. the code used by neurons in the brain to represent and process information. In order to gain more information about the neural code, the association paradigm can be used. In this paradigm researchers attempt to associate the spike trains taken from extra-cellular recordings with the stimuli the animal was presented with at the same time. The next step is to characterize this association under different conditions. In principle, this association can be performed in two ways:

- Compute the *average* response of cells near a known stimulus, and compare that value for different stimuli.
- Search for the locations of a given neural response, and learn about the association of the findings with the external stimuli.

The first method is more commonly used due to the high variability of the spike trains produced by a cell as a response to repeating presentations of the same stimulus. Temporal averaging over many repetitions is used to accurately estimate the response properties of a single cell as well as the strength of its functional connectivity with other cells.

The averaging method, however, suffers from some severe limitations: the alignment problem and the non-continuous search, to name only two of them. The alignment problem is based on the assumption that every time the animal is presented with the same stimulus, it goes through the same mental process – obviously, a problematic assertion. We are not able to construct exactly the same process in the brain because there are many other variables which we can not control directly. The non-continuous problem means that we "look only under the lamp", i.e. we are not searching in all the continuous recordings taken from the animals' brain – but rather only a short time before and after something we consider important has happened. It is possible that the response we are looking for appeared in a non-stimuli-locked fashion. It might appear in unexpected places – which today are not searched.

In this work, when we had to choose a method for searching the neural code, we took the one less traveled by. The method of searching for a single trial response in a continuum of data. We make use of the fact that modern technology enables us to get recordings from several cells recorded simultaneously, and therefore to reduce the variance of response in

such a manner. The view taken here is that it might suffice to accumulate the information across several cells in order to be able to gain information about the activity taken by the animal.

In this work I present a list of statistical tools used to processes such activity:

1. The detection (or classification) of a single trial in its correct behavioral mode.

2. The synergy value computed from the detection value of cells.

3. The spotting of a neural activity template in continuous data.

4. The search for excessively repeating patterns.

The detection of a single trial is the process of predicting the behavioral mode of the animal during the time the single trail was produced. In this part of the work, several methods spanning the current views regarding spike trains were examined. Among these we list the non-homogeneous Poisson process, hidden Markov models, correlation function, metric distance and empirical classification.

The most impressive result of this work is that a single segment of simultaneously recorded spike trains can predict, with good accuracy, the behavioral task of the animal. This is a non-trivial result taking into consideration the low firing rate of the cells, and the high variability from trial to trial. Building the statistical models describing the data, required the use of aligned data. Nevertheless, this result stands in contrast to many of the common analysis methods which align and average a large number of data segments, in order to get reliable results. The impressive power of detection shown here may lead to a new examination of this paradigm.

The detection methods that were based on parametric distribution function supplied us with the best detection rates. The parametric function used was based on the assumption that the spike trains are distributed according to non-homogeneous Poisson distribution. The methods that used non-parametric estimation of the probability functions had significantly worse results. The combination of these facts provides another support to the non-homogeneous Poisson statistical assumption regarding the activity of neurons.

The hidden Markov model outperformed the simplified non-homogeneous Poisson model. The possible advantages of the HMM over the PSTH are:

1. Dynamical time modeling – the ability to wrap the time scale for every example.

2. Building a high-dimensional mode and thus addressing the fact that several cells were recorded simultaneously.

The fact that the HMM was proven superior to the PSTH suggest by using these features, the description of neural data is more accurate.

The detection mechanism allows us to weigh the contribution of each cell separately, and compare it to the combination of all cells. The results show that in all the cortical delayed response paradigm recording sessions examined, the detection was a cooperative achievement and was not only based on one single cell.

This finding supports the use of simultaneous recording as a superior method for understanding the neural computation. It highlights it's importance and suggests that more cells would allow better detection, and more reliable prediction of the animal's actions. This result can be seen from yet another point of view. According to this view the HMM achieved better results because the exploitation of the multi-unit activity is an integral part of such a model.

Combining the tool for detecting the neural response with information theory notions such as mutual information, the idea of synergy was presented. Synergy is defined as the information gathered by examining two cells that was not found when examining each of these cells separately. The synergy value of pairs of cells is used to quantify cooperativity among neurons.

In this work the value of synergy was computed both for cortical cells and TANs' recorded in the basal ganglia. The results show that in almost all of the cortical recording sessions a positive synergy value was found, whereas no such value was found in basal ganglia recordings. The prevailing difference between the synergy measurements in the cortex and in the TANs' of the basal ganglia is also strengthened by the different mechanisms underlying those cells. The TANs' are assumed to be *global* modulators of neuronal interactions in the striatum, whereas the information processed in the frontal cortex in this task is believed to be much more collective and complicated. Here we suggest a first handle for quantitative detection of such different neuronal activities.

The next issue under examination was the question of spotting of a specific pattern of multi-neural spike trains within the ongoing neural activity of these neural cells. Given a neural pattern, a fundamental question is: How unique or characteristic this pattern is to this particular stimulus? In other words, how specific this "code word" is in the ongoing activity of these cells? To answer this question one should try to detect, or *spot*, this specific pattern within a *single* recording independently of any stimulus, a rather difficult pattern recognition task, considering the low firing rate of these cells.

The decision theoretic problem of spotting a pattern of sequenced data in a "noisy background" is classically considered a binary statistical hypothesis test, where the two possible sources are a statistical model of the pattern and a background statistical model. Given a complete description of the two sources there is a fully understood statistical approach, the Likelihood-Ratio-Test (LRT). Nevertheless, when the background model is not given (as is the case of our data) the LRT is not applicable.

The main novelty of this work lies in presenting the adversary background model. According to this model one could say that we are trying to spot the neural pattern - but the background is being produced by a malicious "rival" who is trying to confuse us. Fortunately this "rival" has only limited information about our pattern. In general such a competition is far from straightforward – but we have shown in our case how this can be done. We assume to what extent does our rival "knows" our pattern and devise a statistical method that enabled us to spot the pattern even against such a background model.

Our method was implemented on data from the TAN's, and we tried to spot the neural pattern usually related to the outcome of an action. As a result we found that the neural pattern under examination was not necessarily time-locked to the given action. To strengthen our results we also verified that we could not find the pattern in surrogate data. We found an interesting correlation between the spotted pattern and correct trial vs. incorrect trials. We showed that the spotting of the pattern was a collaboration of several cells. Finally, we observed the disappearance of the pattern in animals in which Parkinsonian symptoms were induced. We believe that these results could not have been obtained without such a method of spotting the reward pattern on the basis of unaligned continuous recordings.

The last work presented here consists of devising statistical methods for the search after excessively repeating precise spikes patterns. In this work we have shown several important statistical features of neural data. The outcome of this work is a complete verified method for exact pattern detection.

The main contribution of all the works presented here, lies in suggesting statistical views for the single trial multi-unit activity. Such tools allow us to investigate more elaborate questions regarding the neural code. All of these methods are motivated by the current ability to record from a small network of cells. This ability allows as to gain a glimpse into the complicated computations that are carried on by the brain. These computations are believed to rely on large number of cells, and the large number of connections between the cells.

We believe that as the number of cells recorded simultaneously increases, there will be more need for such statistical methods. By and by such recording will also allow us to ask questions that are inaccessible today due to the small number of cells recorded simultaneously. It will be fascinating to find out how much information can be gained from one subset of cells on the other.

Our hope is that the kind of statistical methods discussed here will increase the global knowledge about the neural code and how the brain functions.

Part I

Theoretical Background

1 introduction

1.1 Prologue: The aim of the work

The aim of this work is to create and test new tools for statistical description of a single trial multi-neuronal activity. Such tools are based on combining the information gathered from simultaneous recordings from behaving animals with ideas postulated in the field of learning theory. Such a combination may shed light on issues that were unfathomable before. An example is the spotting of neural template in a continuous neuronal.

1.2 In search of the neural code

One of the intriguing questions in brain research concerns the neural code, i.e. the code used by neurons in the brain to represent and process information. The following sections will present a few physiological and mathematical models for this problem. These models are ordered according to the evolution from single cell models to population models, ending with population models which include interacting elements. These models are not mutually exclusive, and can overlap to some extent.

1.3 Biologically oriented models

1.3.1 The single cell as a basis for neural code

The idea that the firing rate of the single cell is the basis of the neural code was suggested by many (e.g. Hubel and Weisel [52, 53], and Barlow [22]). According to this view, neurons integrate their input, and emit action potentials, as a function of some parameter of the stimulus. This means that the firing rate is the code the system uses in a "sparse" coding scheme [20]. The rich connectivity serves to funnel information to these computational elements [22, 21]. Some evidence for the existence of such cells has been found (*e.g.* [38, 66, 83]). Further support for this coding hypothesis was reported by Newsome et al. [69], who found a high degree of agreement between psychophysical performances near threshold, and the prediction from the firing rate of a single neuron.

From the computational point of view, this idea is attractive in its simplicity: firing rate is a convenient parameter for formulating a theoretical model of brain function as it can easily be approximated by a continuous function.

The model does, however, suffer from several drawbacks:

1. The model itself is rigid, and requires large numbers of cells for reasonable representation, while it fails to explain simple abilities [88].
2. Evidence for the existence of cardinal ("grandmother") cells is very limited [2].
3. It is difficult to apply the idea of firing rate as the neural code in associative areas, since its average level drops in these areas, and its temporal modulation becomes sluggish [7].

1.3.2 Temporal modulation of the firing rate

Richmond and Optican[80, 65] suggested the temporal modulations of the rate function as the code used when processing information. It was shown that the amount of information carried by this signal is higher than the one carried by the constant rate. The difference between these two entities increases as one moves towards more associative areas. This theory looks attractive because temporal modulations in firing are found when examining responses of cortical neurons. However, the way in which the systems reads this temporal code is not clear. More recent work re-estimates the information gain from applying the temporal code at only 20% [59], a lower value then previous estimations. This highlights the issue of the justification for using this complex parameter for neural coding.

1.3.3 Population model: (weighted) summed activity across groups of neurons

Another approach which focuses on the summed activity of neurons, based on a different contribution of the single cell to this summing, was suggested by Georgopoulos [44]. Here, the coding of information (in this study, the direction of arm movement) is determined by the *weighted* summed activity over a group of neurons. Each cell contributes a vector directed towards its preferred direction, at a length proportional to its activity level. The model is based on the fact that in the cortex, many cells code any given property, with relatively broad tuning. Sharpening of the tuning curve as well as noise reduction were achieved in the model by averaging over the population. Although this model exploits a population of neurons, it does not call for any specific interaction between the units. This could be an outcome of the fact that, at least initially, the experimental results were obtained from a single unit recording. Thus information about population activity was obtained in a sequential manner rather then simultaneously. In this way, no data was collected regarding the functional connectivity between the units. An effort to introduce the single cell interactions into this scheme was reported in [45, 99].

1.4 Mathematical oriented models

Two such models are described here: the Syn-Fire Chain [5, 4, 7], and the Attractor Neural Network [51, 13]. These models assume that "learning" is based on selective modification of the connectivity among neurons, and "computations" are carried out by selective activation of interconnected neurons. These models do not contradict each other, and can be viewed as different aspects of the same physical system.

1.4.1 Syn-Fire Chains

The Syn-Fire Chain model is based on synchronous temporal structures of spike trains. According to this model, the activity in the cortex is propagated in a network made of a chain of diverging-converging links, where each of these links is made of several neurons. The information in this model is propagated in the chain by secure transmission of a synchronous volley from link to link. This model, like the cell-assembly, takes not only an anatomical form, but also a functional one, i.e. the neurons participating in the Syn-Fire Chain may be scattered and intermingled with each other and with many other neurons. Furthermore,the same neuron may participate in several syn-fire chains. This model predicts the existence of temporal structures in the spike trains of cells, and analyzes them [5, 6]. The information is coded therefore in the currently activated syn-fire chain and the location in the chain.

1.4.2 Attractor Neural Network models

In the last decade there has been impressive development in the field of physical models of neuron-like networks, many of which can be analyzed by statistical mechanics, via the analog with spin-glass systems. One of the notable models is the Attractor Neural Network (ANN).

This model presents a network consisting of a large number of simple look-alike units, that imitate some of the basic functions of real neural networks. Each of these units is connected via weighted links to some, or all, of the other units, and alters its state according to these links and to its current state. These units are, therefore, referred to as *neurons*, and the whole system is referred to as a *neural network*.

This description defines a *dynamical system*, which for any given initial set of neuron states, goes on wandering among the 2^N possible configurations of the network, where N is the number of neurons. The basic concept in this framework is the *attractor*. An attractor is defined as a state (or a cycle of states) of the network that is reached through the dynamics of the system, from different initial configurations. Reaching an attractor in the network can be interpreted as recalling a pattern that is stored in memory. The attractors reached can be static ones (an example of this type can be seen in the Hopefield model [51]). In a more general setup, cyclic attractors may appear.

This description suggests that an ANN is capable of performing memory recall by using the attractors of the network [56]. This ability may be seen as the first step towards a computing network that dynamically goes through a sequence of alternating attractors. This sequence of attractors is in this case the neural code used by the network.

1.5 Differentiating between models

In order to gain more information about the neural code and to be able to choose between the theories described above, the association paradigm can be used. In this paradigm researchers attempt to associate the spike trains taken from extra-cellular recordings with the stimuli the animal was presented with at the same time. The next step is to characterize this association under different conditions. In principle, this association can be performed in two ways:

- Compute the *average* response of cells near a known stimulus, and compare that value for different stimuli.
- Search for the locations of a given neural response, and learn about the association of the findings with the external stimuli.

The spike trains produced by a cell as a response to sequential presentations of the same stimulus are variable [41]. The common method of overcoming this variability is to temporally average the rate signal in order to accurately estimate the response properties of a single cell as well as the strength of its functional connectivity with other cells [2]. This averaging is conducted over a relatively long period of recording time, longer than the time needed for perception, cognition and/or motor execution [27]. Thus, apparently, accuracy is not compatible with brain activity.

However, the averaging method suffers from some severe limitations: the alignment problem and the non-continuous search, to name only two of them. The alignment problem has to do with the assumption that each time the animal is presented with the same stimulus, it goes through the same mental process – obviously a highly questionable assertion. We are not able to construct exactly the same process in the brain because there are many other variables which we can not control directly. The non-continuous problem means that we "look only under the lamp", i.e. we are not searching in all the continuous recordings taken from the animals' brain – but rather only a short time before and after something we consider important has happened. It is possible that the response we are looking for appeared in a non-stimuli-locked fashion. It might appear in unexpected places – which today are not searched.

The main problem that prevented researchers from performing the single trial search lies in the nature of the spike trains. The spike trains (especially in areas involved with complicated tasks) are known to be of low rate (mean of 5 spikes per second), and of high variability [1]. The combination of these facts creates a signal that has a very high variance, and therefore most of the research was involved in the process of reducing this variance by aligning several repetitions of the same task and calculating an average response.

1.6 Previous work on statistical modeling

The work presented here was partially based on ideas which emerged from the author's thesis [42], concerning the way that information processing is carried out in the cortex. The previous work was based on the assumption that this processing may best be described by a Markovian state machine, whose output is the spike trains of cells. The model used was a Hidden Markov Model, whose parameters were the firing rate vectors of several cells recorded simultaneously. Computing such a model yields a temporal segmentation and labeling of the spike trains of the cells.

The main contributions of the previous work were:

- The presentation and implementation of the statistical modeling technique for neural data.
- Showing that the temporal segmentation of the data inspired by the model revealed different functional connectivities.

The second issue implies that the recorded cells may participate in different assemblies and may be activated under different conditions at different times, i.e. in different cognitive states of the small neural network recorded. Changes in functional connectivities between different behavioral events have been found in previous research [10, 9]. The novelty in the work done for the Masters thesis was that these changes in functional connectivities were discovered by the model **directly** from the data, **without** making use of the events of the experiment.

The check carried out showed clearly that in almost all cases in which functional connectivity existed, it changed in different states of the model, i.e. almost any pair of cells that showed a non-flat correlation function, also showed a change in functional connectivity between states of the model. It should be noted that the number of pairs of cells showing correlative behavior was far from negligible: 25% of the pairs of cells showed a state dependent non-flat correlation function.

1.7 Current work

In this work, when we had to choose a method for searching the neural code, we took the one less traveled by[39]: the method of searching for a single trial response in a continuum of data. We make use of the fact that modern technology enables us to get recordings from several cells recorded simultaneously, and therefore to reduce the variance of response in such a manner. The view taken here is that it might suffice to accumulate the information across several cells in order to be able to gain information about the activity carried on by the animal.

Some of the advantages of this view are:

- The ability to access questions not asked before concerning the differences between correct and incorrect trials.

- The ability to search the whole recording session, including time intervals which had previous been discarded.
- Specifically addressing the contribution of each cell to the neural code versus the contribution of all cells simultaneously.

Thesis organization

In this work we present a list of statistical tools used to processes such activity. In the following section (sections 2) the data used for this work will be presented. The first issue examined is the problem of detecting the origin of a single trial (sections 3 to 4). This questions deals with correct classification of a single example with the behavioral mode of the animal. In some sense it can be seen as an attempt to predict the animal's thoughts from its brain recordings. An important tool whose practical implementation draws from the detection ideas is the amount of cooperation between cells – or the synergy level (see sections 5 to 7). The next part shows the tools that can be used in order to spot a specific neural pattern in a continuous data stream (section 8 to 11). An implementation of the ideas of spotting is given in sections 12 to 14. The Next part is devoted to the search for firing patters (sections 15 to 19). Finally a discussion of the methods and results obtained is given in section 20.

2 Origin of data

The data used in this report are simultaneously recorded spike trains, which were recorded at the Hadassah higher brain function laboratory. The recording were made by Y. Prut, H. Slovin, I. Halmaan, A. Feingold, and A. Raz as part of their PhD work. The work was done under the supervision of M. Abeles, H. Bergman and E. Vaadia. A detailed description of the data collection and classification methods can be found in [74, 89].

2.1 Areas under investigation

The data used in this report were taken from 24 recording sessions taken from 4 different vervet (Cercopithecus Aethiops Aethiops) monkeys. Throughout this work each monkey will be referred to by a single letter (B,C,H and I). The recording sites were the frontal cortex (FC), which is located in the anterior third of the brain, and the basal ganglia.

2.1.1 Frontal cortex

The frontal cortex is supposed to be a "high" cortical area, i.e., an area which is not directly connected to sensory or motor systems, but is reciprocally connected to many other cortical areas. The FC has been traditionally divided into three broad regions: the motor cortex (MC), the premotor cortex (PMC) and the prefrontal cortex (PFC). Two of these areas were investigated in this work, the PMC and PFC [46, 71]. Fuster claims that *"the PFC is critical for temporal organization of behavior. It mediates cross-temporal sensorimotor contingencies, integrating motor action (including speech in humans) with recent sensory information"* [40]. This area is also involved in the composition and execution of plans [31]. The PMC is essential for the preparation of movement and the ability to develop an appropriate strategy for movement [46]. These areas are characterized by multiple connections with many other cortical areas, most of which are associative areas. A schematic figure of a monkeys' brain with the frontal cortex marked is given in figure 1.

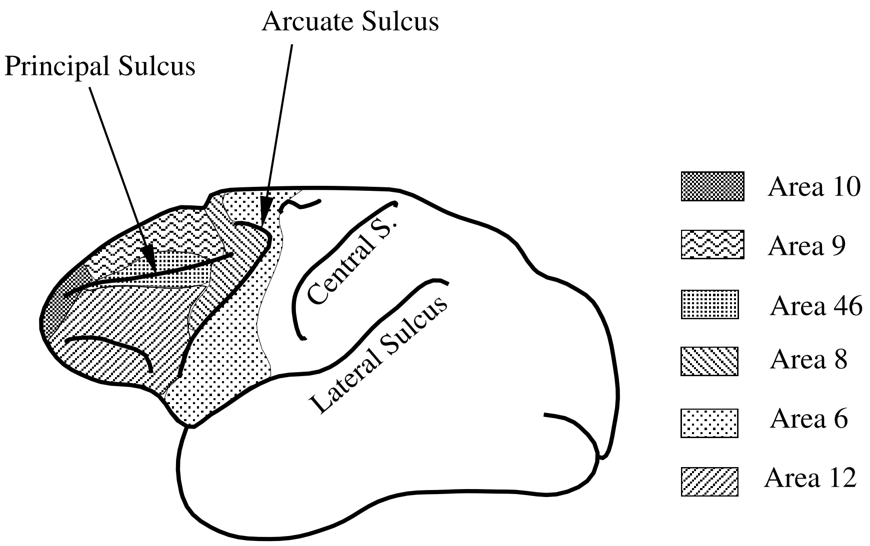

Figure 1: The areas of the frontal cortex shown on a schematic figure of a monkeys' brain.

2.1.2 Basal ganglia

The term "basal ganglia" refers to a loosely grouped collection of large sub-cortical nuclear (gray) masses derived from the telencephalon and located deep within each cerebral hemisphere. The TANs' (tonically active neurons) are the cholinergic inter-neurons of the basal ganglia. TANs have a spontaneous firing rate of 3-15 Hz, and after training, show strong and robust responses to cues predicting future rewards. This response is characterized by a reduction in firing rate (pause), often flanked by brief elevation of the firing rate ([18]). Lately, these type of cells have drawn much attention due to the fact that although the TANs represent only 1-5% of the total population of striatal neurons, they have a major role in the functions of the basal ganglia ([28]). Another interesting feature of these cells is the fact that all the TANs show the same response simultaneously. A schematic figure of the basal ganglia parts is given in figure 2.

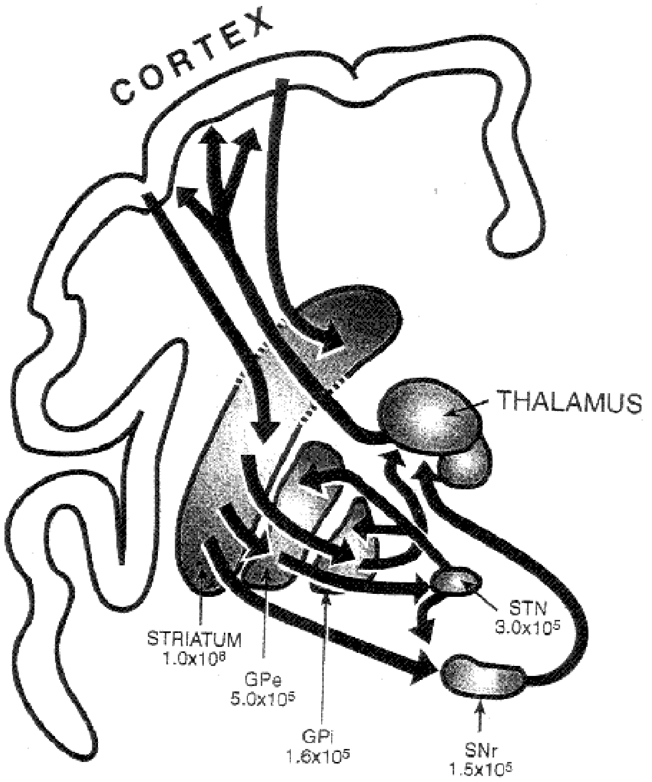

Figure 2: A schematic information flow in and out of the basal ganglia. The components of the basal ganglia are: the striatum, the globus palidus (both the external (GPe) and internal (GPi) segment), and the substantia nigra (SNr).

2.2 Behavioral modes of the monkey

The work reported here was recorded under two behavioral paradigms. The vast majority of the results was produced under the *delayed response* paradigm, and a comparison is made also to the *puzzle box* paradigm.

2.2.1 Delay response paradigm

In this paradigm the monkeys were trained to localize a source of light blink and then, after a delay, to touch the target from which the light blink was presented. The monkeys started a trial by touching a *ready key*, then the central *ready light* was turned on. Three to six seconds later, a *visual cue* was given in the form of a 200 ms light blink coming from the left or the right. Then after a delay of 1-32 seconds the color of the ready light changed from red to orange (the *GO* signal) and the monkeys had to release the ready key and touch the target from which the cue was given. Correct responses were reinforced by a drop of juice. This paradigm was called the *GO* mode. In this mode, the monkey almost always fixed its gaze on the ready light in the first four seconds. As a control, the monkeys were also trained to perform a different paradigm in which all the events were identical to those of the GO mode, except that after the *GO* signal the monkeys had to refrain from responding. This paradigm was referred to as the *NO-GO* mode. A block diagram of this paradigm is shown in figure 3.

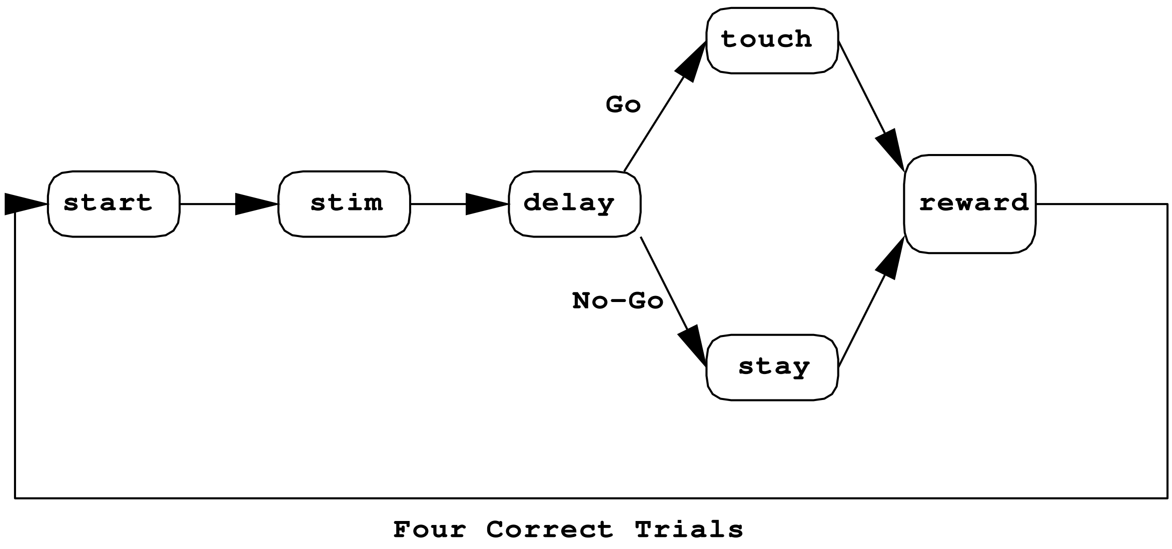

Figure 3: A block diagram of the behavioral modes of the monkey.

The monkeys were trained to switch between modes when a set of lights was turned on for 3-4 seconds. Modes were switched after 4 correct trials. In this way one could study the relation between brain activities and stimuli in two different "sets of mind".

Trials in which the monkey responded with the required response within the time constraint were defined as correct trials. The trial was aborted immediately following any error and the central light was turned off. There was no active punishment and, following an inter trial interval (ITI), the monkey was able to initiate a new trial.

One of the monkeys (monkey I) was trained for a similar but simpler task. The monkey was trained exclusively in the "GO' mode with a 0.25s delay between the offset of the visual cue and the GO signal, and 0.1-0.3s variable delay between the end of a correct trial and the reward. All other parameters (e.g. pre-cue interval, and error definitions) were the same as for the other monkeys.

2.2.2 Box opening puzzle paradigm

In this paradigm the monkey had to select one solution from a range of options. Our implementation of this type of paradigm was the following: Nine boxes, identical in shape, had different opening mechanisms (an example of one option may be seen in figure 4).

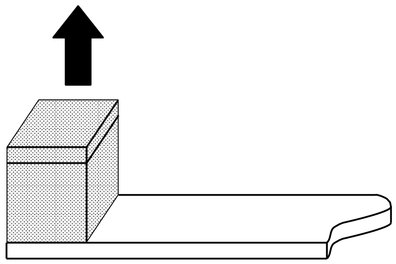

Figure 4: **A Puzzle Box**. An example of a single puzzle box. This box is opened by lifting the upper cover upwards. The external surface of the boxes is identical in texture.

The external faces of the boxes had grooves arranged in a grid manner, to avoid recognition of a specific opening strategy by tactile manipulation. Each box was presented to the monkey after a piece of apple was put inside it. The monkey tried to open the box to get the reward. The boxes were given to the monkey in random order. The temporal sequence of a trial was as follows (schematic description is shown in figure 5):

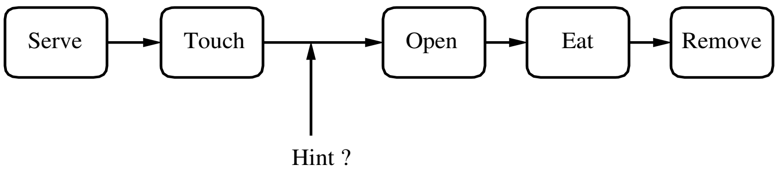

Figure 5: **Behavioral paradigm : Puzzle Box Opening**. Schematic illustration of the event sequences composing the puzzle box opening paradigm. The box was presented to the monkey (Serve), who touches it in an effort to open it. After opening it, the monkey eats the reward inside the box, and then the box is removed. If the monkey had difficulties in opening the box, a Hint was given to help him. As this is a "free trial" no time limits were set.

The opening mechanisms of the boxes are as follows:

- Box 1 - lifting top cover up.
- Box 2 - sliding in x direction.
- Box 3 - sliding in y direction.

- Box 4 - left rotation of the cover.
- Box 5 - right rotation of the cover.
- Box 6 - lifting up the whole box.
- Box 7 - front drawer.
- Box 8 - side drawer.

This type of trial was composed of two main intervals:

- Pre-trial period. From the red light signal until box presentation.
- Trial period. Box presentation until box removal.

2.3 Surgery

When the monkeys were fully trained, they were prepared for recording of single unit activity. Recording chambers were attached to the skull to allow access to the brain tissue under examination. Surgery was performed under aseptic conditions and under deep anesthesia. Upon recovery and retraining, activity was recorded daily while the monkey performed the task. All procedures were conducted according to Hebrew University guidelines for animal care and in accordance with the NIH Guide for the Care and Use of Laboratory Animals (revised 1985).

During the recording sessions, the monkey's head was immobilized and four to eight glass-coated tungsten micro-electrodes confined within a cylindrical guide (2.2-mm outer diameter) were advanced to the recording sites. Neuronal activity from each electrode was amplified, filtered (300-6000Hz), sorted and classified as belonging to a single neuron by a template-matching algorithm. Only spike-trains emitted by well-isolated single neurons (as judged by the stable spike waveforms, stationary firing rates and constant responses to behavioral events) were included in this study.

We stress that this report does not relate to the process of learning anything new, but to the sequence of states through which the neural networks (in which the recorded neurons might be embedded) pass, while the monkey performs a task with which it is familiar.

2.3.1 MPTP treatments

When recording was carried out in the basal ganglia, the monkeys were subjected to MPTP treatment. After recording in the normal state the monkeys were rendered Parkinsonian by systemic MPTP treatments. The MPTP treatment (MPTP HCl, Aldrich, Milwaukee, WI, U.S.A.) 0.4-0.5 mg/kg/day I.M was administrated for a total of four days to monkey H and I respectively. Two to three days after the completion of the MPTP course the monkeys developed severe Parkinsonian symptoms that were stable until the end of the MPTP recording period.

2.4 Periods of interest

We sliced the neuronal recording into several periods of interest.

2.4.1 Differentiating between four delay types

The segments of data used consisted of the delay periods of the trials, whose length was equal to, or longer than, 4 seconds. Each of these segments was further categorized into one of the following different refined behavioral modes:

Mode 1: "Go" mode to the left.

Mode 2: "Go" mode to the right.

Mode 3: "No-Go" mode to the left.

Mode 4: "No-Go" mode to the right.

Table 1 summarizes the characteristics of the data-base used in terms of the number of cells used, the total number of segments used and the distribution of segments among the different behavioral modes.

Table 1 shows that on most of the recording sessions used in this work, the distribution of the different modes is rather uniform.

2.4.2 The reward response

When recordings were taken from the TANs', we concentrated on the period following the reward. Specifically we defined two segments of interest:

Table 1: The list of recording sessions used in the delay response recording in the frontal cortex. For each session the following information is given: the number of cells used, the number of delay period segments used. The percentage of data segments in each of the different modes is also given.

Session	# Cells	# Segments	Mode1	Mode2	Mode3	Mode4
b1	3	107	24	31	24	21
b2	4	235	28	23	24	26
b3	8	256	25	26	24	25
b4	6	198	26	25	26	24
b5	8	243	25	24	26	26
b6	4	198	26	25	26	23
b7	8	197	24	25	26	24
b8	5	245	23	26	26	25
b9	6	348	24	25	25	25
c1	5	207	24	28	26	23
c2	5	242	24	26	26	29
c3	5	409	23	27	25	25
c4	5	476	9	41	10	40
c5	4	426	42	10	40	8
c6	5	327	44	11	37	9
c7	6	722	48	12	33	7
c8	7	249	27	27	22	23
c9	5	182	26	25	22	27
c10	4	180	24	27	26	24

- The reward period which directly follows the reward cue.
- The period just after the beginning of the trial.

We defined different events of the task performed by monkey H and I as the reward cue, i.e., the event that was used by the monkey as the cue predicting the reward. The reward cue was defined as the buzz signaling the actual reward in Monkey H who performed the more complicated GO/No-GO task, with variable delays between the touch of the target and the reward. In the studied sessions, the average success rate for Monkey H was 58%, and it is therefore possible that the monkey used only the buzz tone that was linked with the physical reward as a valid prediction of reward. On the other hand, Monkey I was performing an easier task (only GO mode and a fixed delay between the spatial cue and the GO signal). The average success rate of Monkey I in the studied sessions was 85%. The spatial cue was therefore defined as the reward cue for monkey I. The definition of the reward cue was validated by examining the responses of all TANs recorded from the same monkey, and by the finding that the reward cue was able to elicit the typical (on-off-on) response of TANs to reward predicting events [14, 17].

Table 2: The list of recording sessions used in the delay response recording in the basal ganglia. For each session the following information is given: the number of cells used, the number of delay period segments used. The percentage of data segments in each of the different modes is also given.

Session	# Cells	# Segments	Start of trial	Reward
h1	4	266	133	133
h2	3	186	93	93
h3	3	260	130	130
i1	4	476	238	238
i2	2	180	90	90

Table 3: The list of recording sessions used in the box puzzle experiment. The information supplied as in table 1 and 2.

Session	# Cells	# Segments	Mode1	Mode2
b3	7	107	24	31
b7	8	256	25	26

The following table summarize the data used for this detection experiment:

2.4.3 The puzzle box

In the puzzle box paradigm we concentrated on the first 1000 ms of the trial period. We tried to differentiate between possible groups of opening schemes. In table 3 the list of recording sessions and segments used are given.

Part II

Detection of neuronal response

3 Detection definition

The goal of this work was to examine the ability to classify a single trial response according to its correct behavioral mode. We compared different classification (detection) algorithms which shared a general framework:

- All the data segments belonging to a specific behavioral mode were listed together. A typical length of such a list (number of examples per mode) was 60.
- The data included in each of the different lists were used to train four different models.
- All the data segments were evaluated sequentially according to the previously computed models. Each data segment was assigned a probability of its being the outcome of any of the models. This probability is usually referred to in the literature as the *likelihood* of the data segment given the model. After computing the likelihood of the data segment for all the possible models, the data segment was associated with the most likely model. If the data segment under investigation was actually recorded under the same behavioral mode to which it was associated, it was declared a *correct detection*. Otherwise, it was declared a *wrong detection*.
- After evaluating all data segments, the overall detection was computed. The overall detection was the percentage of correct detection segments among all the segments. More refined detection ratios including the detection within each mode, were also computed. These later detection results were computed using only the segments of a specific mode.

Due to the small number of examples used to build each model, we did not split the data into training and testing sets, rather we used the *Jacks' Knife* procedure. In this procedure when trying to detect a response o, we excluded it from the train set and re-computed the model. The likelihood of o was then computed for each of the newly computed models.

The detection procedure is carried out therefore, via the following algorithm:

1. Go over all the behavioral modes.

2. For each such mode, go over all the data segment examples making it.

3. For each such example compute the probability of it's being the outcome of **all** the models except for the correct one.

4. When the probability of an example is being computed with the **correct** model, this model is altered in the following way: the model is computed using all the examples **without** the current example.

5. For each example, the highest probability is found.

6. The algorithm counts the ratio of correct detections for each behavioral mode.

An example of the detection needed is given in the following two figures. In figure 6A the spike trains of 5 cells in the 4 different behavioral modes are presented, where each spike is represented as a small dot. In figure 6B one segment of data is shown. The goal of the detection problem is to tell which behavioral mode that segment of data was taken from.

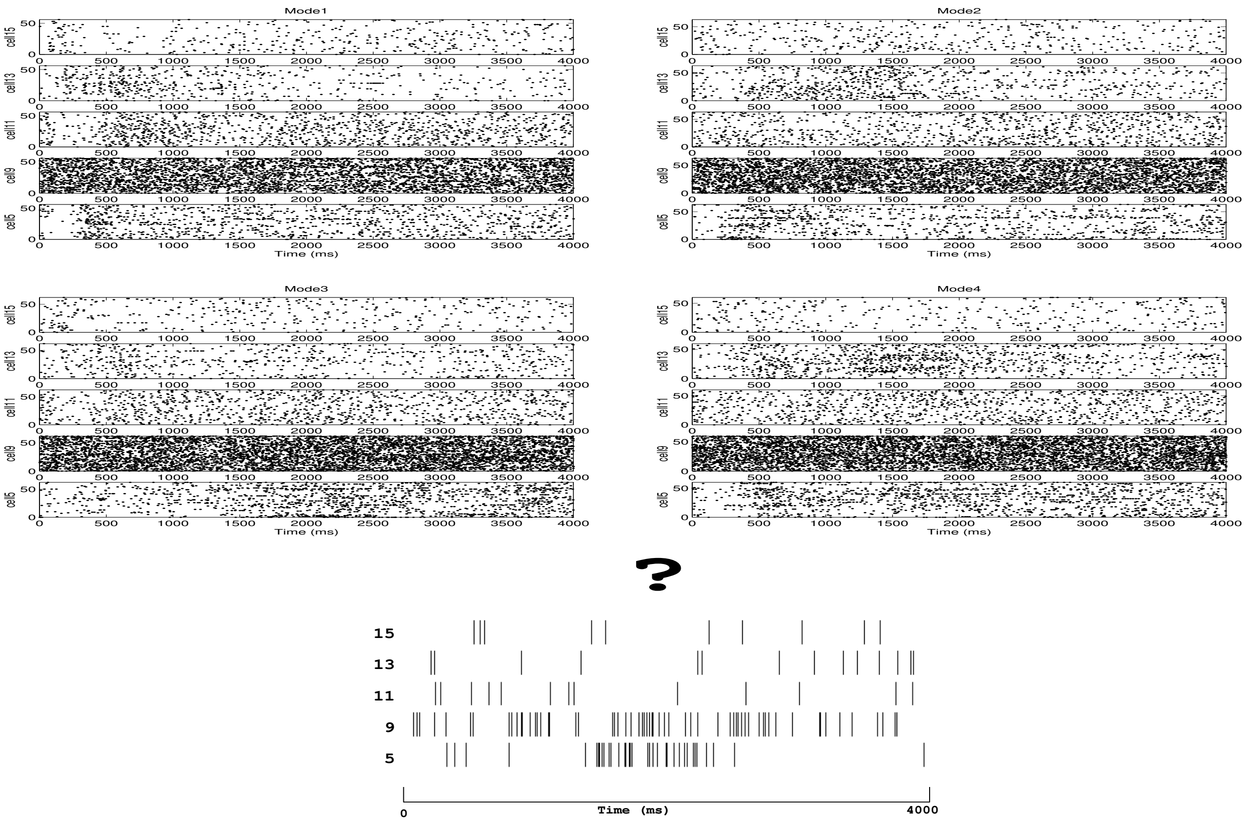

Figure 6: The spike trains of all the segments in one recording session (b8). In this session there were 5 cells recorded, and the segments are parceled into 4 different behavioral modes. In the lower part we see a single trial taken from the same recording session, and we must decide which behavioral mode it is drawn from.

The segment presented in figure 6B was taken from the first behavioral mode (Go to the Left), and was identified correctly by most of the detectors that are shown in this work. The example given above shows how difficult the detection was. The average firing rate of cells in the recorded area is around 5 spikes-per-second, so on the average 4 second segments including 5 spike trains would hold 100 spikes. The detection scheme should be able to reveal the correct behavioral mode using these 100 spikes.

3.1 Modeling techniques

The different modeling techniques used in this work are:

1. Assuming the spike trains are the outcome of a Poisson process. The process may be either homogeneous (spike-count per response) or non-homogeneous (PSTH).
2. Hidden Markov models (HMM) based on firing rate vectors [43]. We also examined another type of HMM which was based on fine resolution data.
3. Pair wise cross-correlation [72].
4. Metric distance between spike trains [92, 93].
5. Optimal detection scheme based on the Jensen-Shannon information measure [48].

3.2 PSTH analysis

3.2.1 Technique description

The basic modeling technique applied to this data consisted of computing the PSTH - peri-stimulus-time histogram. In this method, all the data segments are aligned, and parceled into bins. The number of spikes found in each bin is counted (for each different cell), and normalized by the number of data segments given. The final result, therefore, is the average firing rate of the cell in terms of spikes-per-second. The resolution of the PSTH is the length of data segments divided by the number of bins used. As the number of spikes used to compute the firing rate in each bin is increased (bin width is increased) the accuracy of each bins value is increased, but the time resolution is decreased. When the number of bins increase, the opposite occurs: better time resolution but less accuracy.

Two major methods were used here in order to compute the firing rate of each bin. According to the first, the bin size was fixed and the number of bins in it was counted. According to the second method, the number of spikes to be counted was fixed, and the bin size was changed in order to include that amount of spikes. The second method is thus an *adaptive* method which depends on the temporary firing rate for its evaluation [86].

If the number of bins used in the PSTH is reduced to only one bin, the PSTH computation turns out to be rate computation. In this case, the average rate of firing in each cell is computed, and when a new data segment is presented the probability assigned to it is based only on the count (number of spikes) found in that section. This computation is carried out for each cell separately.

An example of four PSTH models is shown in figure 7.

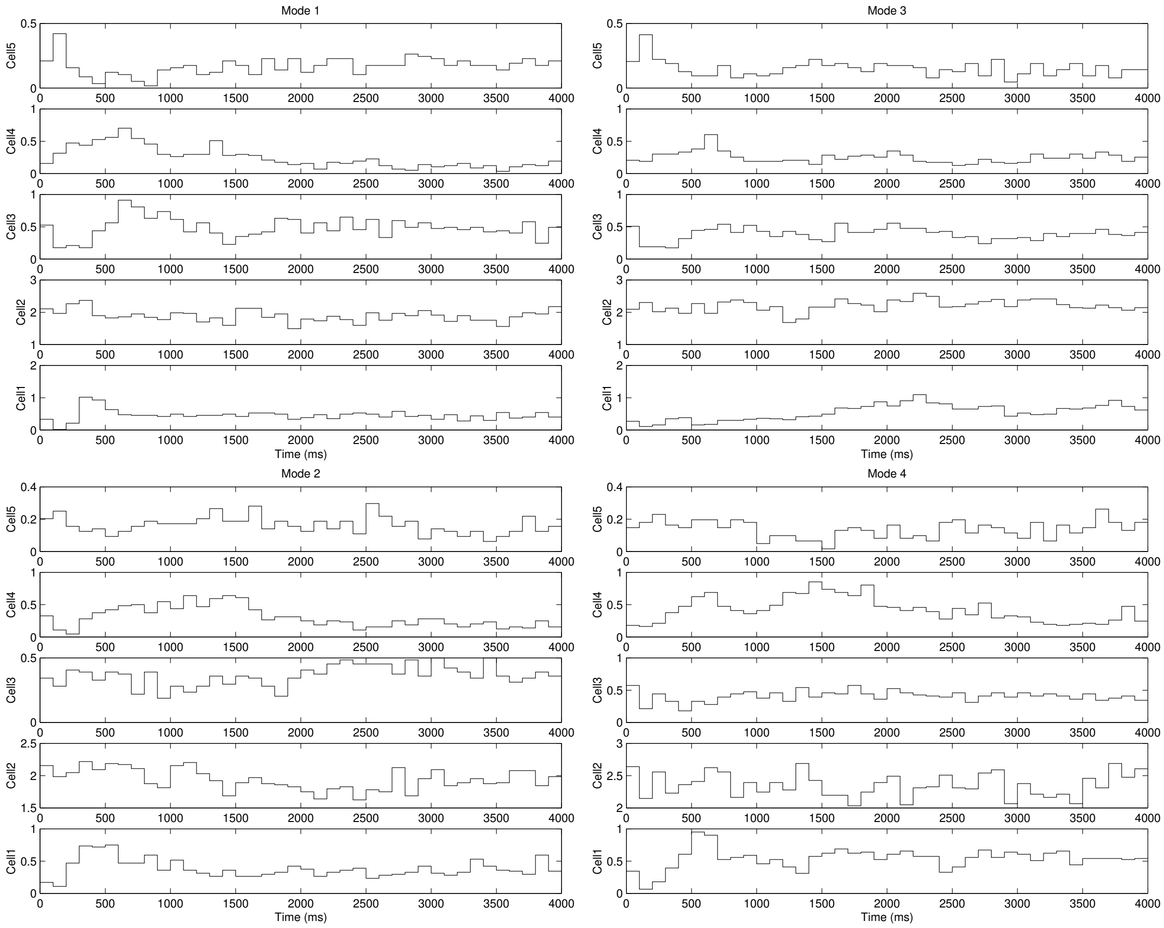

Figure 7: A PSTH model consisting of 5 cells in 4 different modes.

In this example the PSTH of 5 cells in the 4 different modes used are presented. The value in each bin is shown in a bar whose height is proportional to the average number of counts. It can be observed that the PSTHs of different behavioral modes look alike, with some differences. These difference are the ones sought by this type of modeling.

3.2.2 Technique motivation

The simplest interpretation of a PSTH is that it is an estimate of a Poisson process [33]. In the extreme case of one bin, the process is an homogeneous Poisson, and in the case of multiple bins it is a non-homogeneous Poisson process. The assumption made here is that the cells are independent of one another, and therefore there is no correlation between them. Furthermore, the cells' responses in the defined time segments are locked to the stimulus. This lock is not flexible, and it does not allow any warping in time.

3.2.3 Probability computation

The computation of the probability of a given spike train segment to be the outcome of a specific PSTH, is based on the following assumption. The model PSTH is assumed to be a non-homogeneous Poisson process, in which each bin has its own mean. The likelihood of the data assuming a specific PSTH model is therefore:

$$\prod_{i=1}^{N}\prod_{j=1}^{L} P(c_i(j), m_i(j)) = \prod_{i=1}^{N}\prod_{j=1}^{L} \frac{e^{-m_i(j)} m_i(j)^{c_i(j)}}{c_i(j)!}, \tag{1}$$

where:

- N is the number of cells.
- L is the number of bins in the PSTH.
- $c_i(j)$ is the count in the j'th bin of the i'th cell of the current example.
- $m_i(j)$ is the PSTH value in the j'th bin of the i'th cell.

For practical and computation reasons we use the log-likelihood probability measure of the data, i.e.

$$\sum_{i=1}^{N}\sum_{j=1}^{L} -m_i(j) + c_i(j)\log(m_i(j)) - \log(c_i(j)!). \tag{2}$$

The log-likelihood of the data according to all behavioral modes is computed, and the detected mode is the one with highest value. It should be noted that this computation does not depend on the time resolution of the data, but only on the time resolution of the PSTH model.

In order to demonstrate this we will give a proof for a specific case of modeling, and show later how this can be generalized to the multi-cell, many bins case.

Independence of window length: Suppose the PSTH model m of one cell is constant over time T ms, with value λ (where λ is the mean number of action potentials in time T), and a given data segment x holds n spikes. The log-likelihood is:

$$\mathcal{L}_1(x|m) = -\lambda + n\log(\lambda) - \log(n!), \tag{3}$$

if we use bin size of T ms. Using T bins each of length 1 ms would produce the following result:

$$\mathcal{L}_2(x|m) = \sum_{i=1}^{T}\left(-\frac{\lambda}{T} + n_i\log(\frac{\lambda}{T}) - \log(n_i!)\right). \tag{4}$$

Where n_i are the number of spikes in the ith bin. The rate at each ms bin is $\frac{\lambda}{T}$. Assuming that two spikes are at least 1 ms apart, the last equation can be divided into two sums. The first one holds the bins in which the cell has not fired and the second one hold all the other bins.

$$\mathcal{L}_2(x|m) = \sum_{i=1}^{n}\left(-\frac{\lambda}{T} + \log(\frac{\lambda}{T})\right) + \sum_{i=1}^{T-n}\left(-\frac{\lambda}{T}\right) = -\lambda + n\log(\lambda) - n\log(T). \tag{5}$$

It can be seen that the difference between the two log-likelihood measures depends on n and T, but does not depend on λ.

$$\mathcal{L}_1(x|m) - \mathcal{L}_2(x|m) = n\log(T) - \log(n!). \tag{6}$$

Therefore, if we compare the log-likelihood of the same data according to different models (which differ by their λ) all the terms that depend only on n and T are eliminated.

This proof can be generalized to the case of more than one bin PSTH, by simply summing up the log-likelihoods in the different bins, and further summation of the number of cells.

3.2.4 Technique parameters

In the results reported in the delayed response paradigm, the length of the PSTH was 4000 ms, and all the cells were used in the computation. We have varied the bin length (in the fixed bin size type of PSTH), or the number of spikes to be found in a bin (for the variable bin length PSTH). In the latter case, the maximal number of bins was used.

3.3 Stochastic modeling

3.3.1 Technique description

HMM is a stochastic modeling technique for the study of complex time series [73, 32, 76]. It is essentially a stochastic function of a Markov chain, i.e., to each state of a Markov process, a conditional probability distribution on the possible output observations is attached. Hidden Markov models theory enables modeling with many, discrete or continuous, probability distribution functions. The popularity of this technique is largely due to its efficient and robust implementation. The computational costs of both training and evaluation of the model are linear in the length of the data. In addition, the model is sufficiently rich in the possible structure of its output, which has made the Hidden Markov technology attractive for tackling problems in language, speech and signal processing[63, 77].

A detailed description of two HMM types which were applied to neuronal data can be found in [43, 85]. These method uses basically the same HMM mechanism, and differs in the feature space used for the modeling. In the first technique, firing rate vectors are the focus of observation of the HMM. In the second work the actual emission of spikes are considered the observations. The first method allows more accurate short time firing rate, whereas the second method provides us with fine resolution data. The first method was used so far in unsupervised modeling of the data, and the second method was used in supervised modeling. In this work both of these methods are used in supervised modeling.

Using the HMM enables us to find the probability of an observation sequence given a model. This ability will be discussed in detail in the next section. The other algorithm related to the HMM is the ability to build (train) a model given a list of observations. For this algorithm, there are no known optimal methods which can be performed in reasonable time (i.e. polynomial with the number of observations). An optimal method would be one in which the likelihood of the data, given the model is the global maximum over all the existing models. Gradient ascent methods do exist, however, which bring the model into a local maxima of the likelihood function. These methods are sometimes highly governed by the initial conditions posed to the algorithm. The initial conditions in the HMM are comprised of the transition probabilities between states, and the probability of seeing a specific observation in each state. A schematic example of an HMM classification is shown in figure 8.

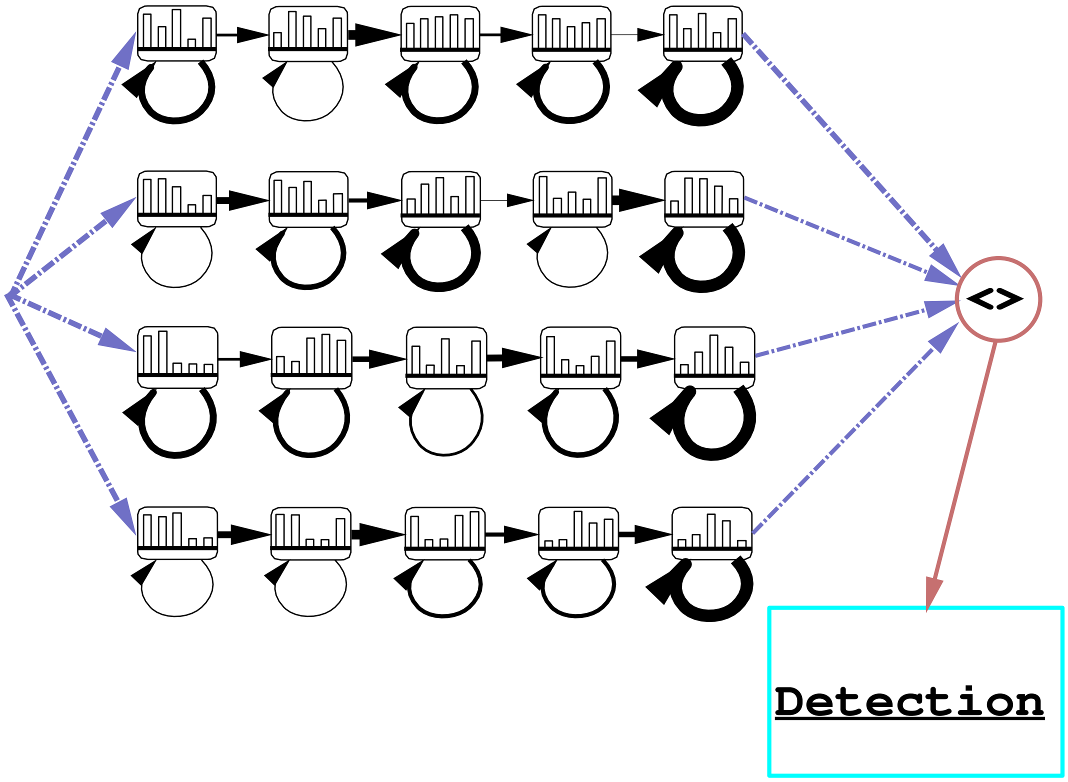

Figure 8: Classification using HMM. The example is based upon 5-state HMMs' and on recording of 5 cells simultaneously. In this classification 4 left-to-right models were created separately. Each such model is drawn as a horizontal sequence of states where each state is characterized by the firing rates of the cells and the transition matrix. The firing rate are drawn as bars whose height is proportional to the cells' firing rate. The transitions to the next state are drawn as arrows whose thickness is proportional to the their probability. The four models were combined together into a larger HMM and the classification is performed by computing the sequence of states with the highest probability for each data example. For more details see [76]

3.3.2 Technique motivation

The HMM assumes that the spikes are the outcome of a hidden markov process, whose states cannot be observed directly but rather through some manifestation (spikes, in this case). It may be assumed that these observations are the spikes themselves (fine-resolution HMM), or the firing rate vectors. The method could be advantageous if the data is a truly vectorial process – if the combination of spike trains reveals more information than just one cell, or any subgroup of cells. The other advantage of the method could result if it transpires that the time locking of the data is not a very strict one, i.e. dynamic time changes appear.

3.3.3 Probability computation

All the stochastic algorithms described in this section rely heavily on the optimality criterion. *"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision"* [24]. Straightforward implementation of the HMM may seem at first exponential by the number of frames being modeled. However, a more sophisticated implementation may be carried out using the dynamic programming algorithm which is based on the optimality criterion [23]. The run time of this algorithm is linear with the number of frames.

In order to compute the probability of an observation sequence given the model (the likelihood of the data), we need to average the probability of a given observation sequence over all possible underlying state sequences. The number of possible state sequences is N^T, where N is the number of states and T is the number of data frames. An algorithm, known as the forward-backward algorithm [76], performs the same calculation in time linear with T. The popularity of the HMM is, to large extent due, to these efficient algorithms.

3.3.4 Technique parameters

The models' parameters are:

1. The type of transition matrix. It can allow any transition, in which case the matrix is said to be ergodic. A special case of transition probabilities is the one referred to as Left-to-Right type. This type allows a transition from state i to state j only if $i = j$ or $j = i + 1$. In the work presented here we used a Left-to-Right type transition matrix. Such a transition matrix exploits the serial nature of response.

2. The length of the estimation window and the transition between successive windows. The length of the window used was 500 ms with transition of 100 ms between frames. These values were derived from optimizations performed in a previous work [42].

3. The number of states used by the model to describe the data. The numbers used here

ranged from 5 to 20. These values were derived from optimizations performed in a previous work [42].

3.4 Pair-wise cross-correlation

3.4.1 Technique description

According to this technique, all the pair-wise cross-correlations of the cells are computed. The values of correlations are then normalized to a correlation function such that the sum of values in each function is the same. This correlation functions serve as a model (in a similar manner to the PSTH models). The normalization performed here eliminates the information about the firing rate of the cells involved.

3.4.2 Technique motivation

In this technique all the information about the firing rate of the cells is removed, and the only information used is the correlation between the cells. We therefore expect this type of modeling to reveal a different type of alternation between the modes. The information found in the correlation functions would, however, be much more noisy.

A typical model for one of the recording sessions is shown in figure 9.

This figure shows the pair-wise cross-correlation of 5 cells, yielding 15 correlation functions. The correlation functions were computed with a bin size of 13 ms and the correlation length was 1000 ms (for the positive and negative lags).

3.4.3 Probability computation

The probability computation is achieved by multiplying all the correlation vectors in scalar multiplication. In the PSTH case there were N histograms (where N is the number of cells), and in this case there are $\frac{N(N-1)}{2}$ correlations. The probability of an example being the outcome of a specific model is:

$$P = \prod_{i=1}^{\frac{N(N-1)}{2}} \prod_{j=1}^{L} c_i(j) m_i(j). \tag{7}$$

Where:

- N is the number of cells.

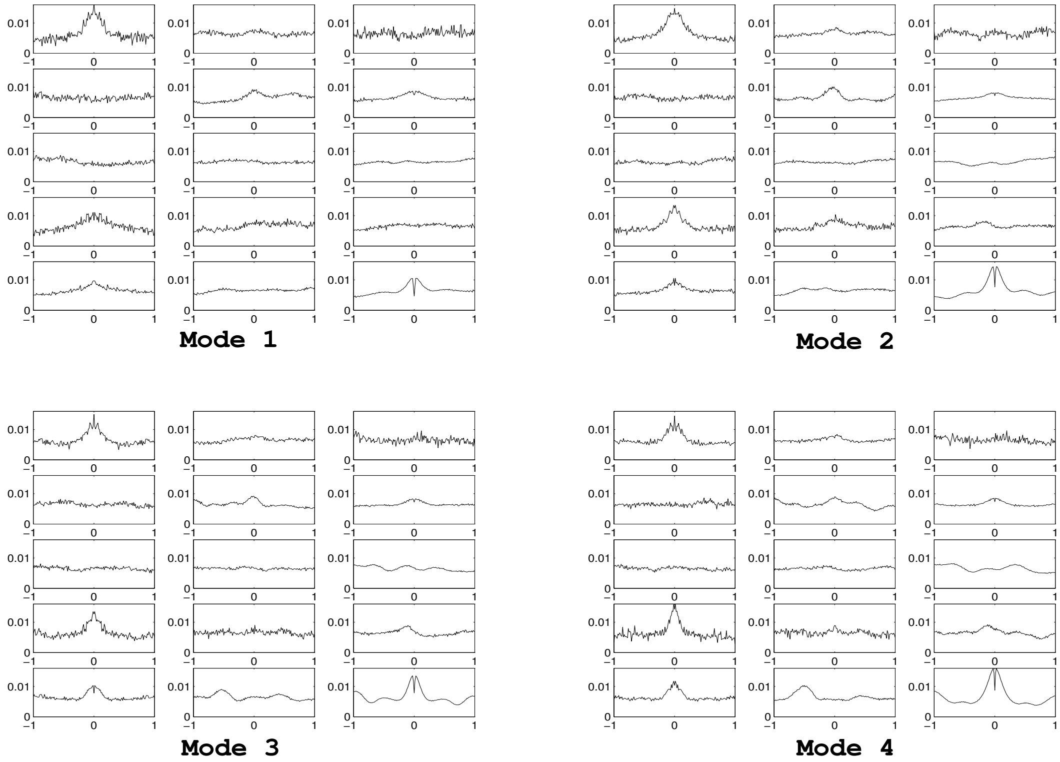

Figure 9: Four models of pair-wise cross-correlation. In this example 5 cells were used, and the number of cross-correlations is therefore 15. The correlation functions are ordered from left to right.

- L is the number of bins in the correlation function.
- $c_i(j)$ is the count in the j'th bin of the i'th cell of the current example.
- $m_i(j)$ is the normalized count in the correlation model of the j'th bin of the i'th cell.

3.4.4 Technique parameters

The parameters used were:

1. The bin length being used, typically ranging from several ms to several tens of ms.
2. The correlation length, ranging from 100 ms to few seconds. The correlation length is computed symmetrically for both sides of the correlation. The output data is, therefore, twice the length of the correlation, i.e. in a 1 second correlation the data spreads over 2 seconds.

3.5 Metric distance method

3.5.1 Technique description

An interesting detection method tries to embed the spike trains in a metric space, and then uses a stimulus-dependent clustering [93]. This can be achieved by providing a metric distance between each two spike trains. Each metric defines the distance between two spike trains as the minimal "cost" required to transform one spike train into the other. The transformation is done via a sequence of accepted elementary steps, such as inserting or deleting a spike, shifting a spike in time, or changing an inter-spike interval length.

3.5.2 Technique motivation

There are several advantages to mapping spike trains into a metric space: Each metric, in essence, represents a candidate temporal code in which similar stimuli produce responses which are close and dissimilar stimuli produce responses which are more distant.

3.5.3 Probability computation

Spike trains are considered to be points in an abstract topological space. A spike train metric is a rule which assigns a non-negative number $D(S_a, S_b)$ to pairs of spike trains S_a and S_b, which expresses how dissimilar they are.

A metric D is essentially an abstract distance. By definition, metrics have the following properties:

- $D(S_a, S_a) = 0$
- Symmetry: $D(S_a, S_b) = D(S_b, S_a)$
- Triangle inequality: $D(S_a, S_c) <= D(S_a, S_b) + D(S_b, S_c)$
- Non-negativity: $D(S_a, S_b) > 0$ unless $S_a = S_b$.

Cost-based metrics fulfill the above general definition of a metric, and are constructed with the following ingredients:

- A list of accepted elementary steps (accepted transformations of spike trains).

- An assignment of non-negative numerical costs to each elementary step.

For any such set of choices, one can define a metric $D(S_a, S_b)$ as the least total cost of any accepted transformation from S_a to S_b, via any sequence of spike trains $S_a, S_1, S_2, ..., S_n, S_b$.

This is a family of cost-based metrics, parametrized by a "cost per unit time" q (units of 1/sec). The elementary steps and associated costs are:

- Insert a spike: cost $= 1$
- Delete a spike: cost $= 1$
- Shift a spike by an amount of time t: cost $= qt$ (the normal distance).

For a matching example see figure 10.

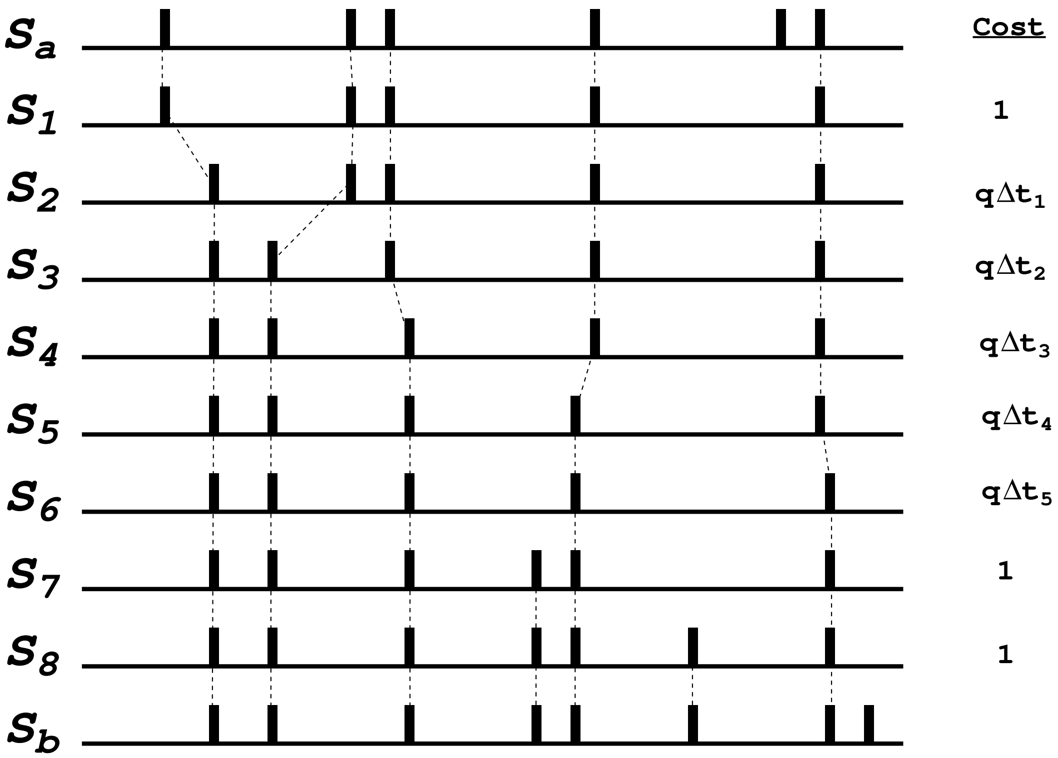

Figure 10: Example of matching spike trains S_a to S_b via insertion, deletion and substitutions. Each line represent an intermediate stage in the matching process. On the left, the stages are marked, and on the right, the cost associated with each stage is given.

If q is very small, this becomes the spike count distance. If q is very large, all spike trains are far apart from each other, unless they are nearly identical. For intermediate values of q, the distance between two spike trains is small if they have a similar number of spikes, occurring at similar times.

The motivation for this construction is that neurons which act like coincidence detectors should care about this metric. The value of q corresponds to the temporal precision of the coincidence detector. This distance can be calculated efficiently by a dynamic programming algorithm.

The full procedure for classification also includes the idea of assigning a spike train to a class of responses if it is closer to the class responses than to any other set of responses. For a classification of spike train S we first exclude it from the set of examples it belongs too (s_α). Now the average distance for any stimulus class s_γ is

$$d(S, s_\gamma) = [\langle D[q](S, S')^z_{S'\ elicited\ by\ s_\gamma} \rangle]^{1/z}. \tag{8}$$

where $\langle\rangle$ denotes an average over all spike trains S' elicited by a stimulus in behavioral mode s_γ. The value of z used was -2.

"The manner in which spike trains from multiple neurons combine to transmit information is a matter of great interest, but not one that we address here"[92].

3.5.4 Technique parameters

The most important parameter in the technique is the value of q. The value of q determines the accuracy needed in time – it determines whether an action potential is added, or shifted. It is possible to use different types of cost-based metrics - but for practical purposes it was sufficient to work with the cost as presented above.

The main obstacle was the one quoted from [92], relating to the use of the metric in the case of multiple neurons. In this study we used several methods, such as summing the overall distance, or using some form of multiplication. In the results section we give the maximum value for each recording session, i.e. we actually give results which are optimized to our data set.

3.6 Empirical classification

3.6.1 Technique description

Modern classification theory concerns the problem of how to classify observations into predefined categories. In principle this classification may be achieved using either a known probabilistic description, or just the training data (empirical classification). Recently, more attention has been given to the empirical problem, and the method presented in this section is one of the optimal implementations of such a technique.

The technique described here was first shown by Gutman[48] and is also referred to as the Jensen-Shannon (*JS*) divergence [64]. Using this technique, a statistic is computed. It has been shown [48] that this statistic is not only optimal in a certain sense, but will also, *given enough training and testing data*, produce error probabilities with the *same* exponential rate as the likelihood ratio classifier that clairvoyantly knows the underlying statistical model for the training data.

3.6.2 Technique motivation

The advantages of using empirical classification are due to the amount of knowledge we have a priori regarding the statistical description of our data. Suppose that our knowledge is very limited, or that we do not have full access to the correct statistical model. In such a case, assuming the wrong statistical model may insert distortion into the results instead of aiding in classification. It is therefore preferable to use the empirical classification.

3.6.3 Probability computation

The scenario in this method is as follows: we are given a training set $\mathbf{T}_j$, where the index j stands for different categories of the data. The data is made up of observations, and each observation can only assume a finite set of values: $R_l \in r_1, ..., r_K$ (K which is the alpha-bet size, is finite). When given a test set $\mathbf{R} = R_1, ...R_{L_R}$, the statistics are:

$$G_j = \frac{L_T}{L_R}\mathcal{D}(\hat{P}_{\mathbf{T}_j}||\bar{P}_{\mathbf{T}_j,\mathbf{R}}) + \mathcal{D}(\hat{P}_{\mathbf{R}}||\bar{P}_{\mathbf{T}_j,\mathbf{R}}) \tag{9}$$

$$\bar{P}_{\mathbf{T}_j,\mathbf{R}} = \frac{L_T\hat{P}_{\mathbf{T}_j} + L_R\hat{P}_{\mathbf{R}}}{L_T + L_R} \tag{10}$$

where $\hat{P}$ is the histogram estimate of the probability mass function:

$$\hat{P}_{\mathbf{T}_j}(r_k) = \frac{\#\ times\ r_k\ occurs\ in\ \mathbf{T}_j}{L_T}\quad \hat{P}_{\mathbf{R}}(r_k) = \frac{\#\ times\ r_k\ occurs\ in\ \mathbf{R}}{L_R}, \tag{11}$$

and $D(P||Q)$ is the KL-divergence [60]. For a detailed description of this function see [32, 42].

3.6.4 Technique parameters

The method can be implemented directly by building the histogram for all response values – for more details see [54]. Nevertheless, the number of entries in the histogram would usually prevent us from such an implementation. Having N neurons whose responses are measured across B bins, the number of possible response values (the value of K defined above) is 2^{NB}. In our case $N = 4$ and $B = 600$ which results in the inability to estimate the probability for each of these values.

In order to overcome the huge set of values we need to define stationary segments of data and to quantize the possible values of response.We are forced to give up the promise of optimality, but the computation becomes feasible. Figure 11 shows an example of such a procedure. In the example it is shown that the data is sliced into stationary bins, and within each bin the possible values are computed over bins which define the size K of the possible alpha-bet (which is the number of possible counts in each bin to the power of the number of cells). In the example given here we allow each such bin to have at most one spike, and given 3 cells we have K equal 8. The distribution of symbols in the alpha-bet is shown in the histogram computed for one of the stationary bins.

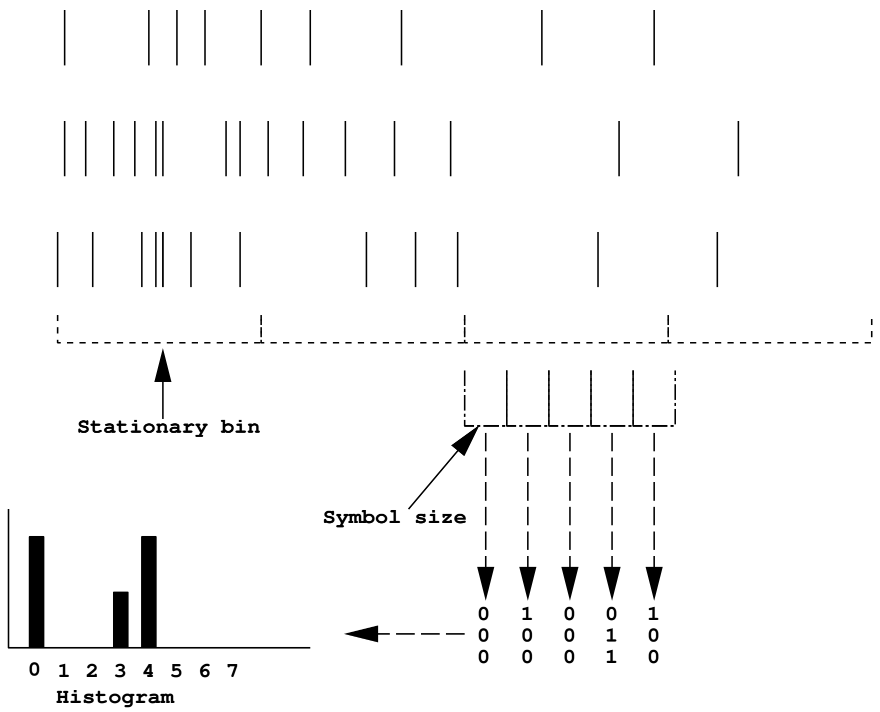

Figure 11: Schematic description of slicing the data into stationary segments and defining the alpha-bet size via further slicing the stationary bins into symbol-size bins.

The computation of the Gutman statistics from that point on is straightforward.

4 Detection results

4.1 How to interpret and compare the results

For each of the detection methods we computed the detection rate per recording session. The first element of comparison is to examine the mean detection rate. In order to evaluate the significance of the change between detection methods, it is not enough just to compute the variance of the results. The existence of intrinsic changes between recording session must also be taken into account. An example of this phenomenon is shown in figure 12. The results in this figure show that although the mean detection rate of the solid bars is within one tenth of standard deviation of the same rate in the dashed bars, the results are significantly different. In most of the recoding sessions the detection rate difference is in the same direction.

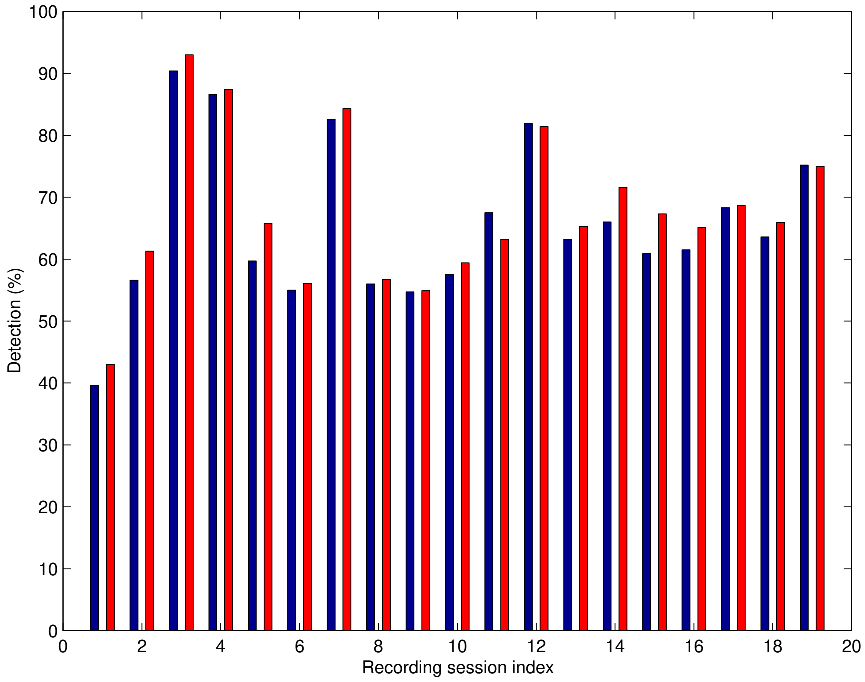

Figure 12: Comparison between two detection methods. Although the mean value of detection of the first method in each point is within one standard deviation, the difference between detections is significant. This fact can be clearly seen when examining the difference between detections point-wise.

A better way of estimating the significance changes between detection methods is by measuring the average and the variance of the difference between the methods along the recording sessions.

Another issue which arises when examining these results is the detection significance issue. We need a method which will be able to compute if the detection results are significantly above chance level. The chance level in the delay behavioral mode detection was 25% (four different states). Given the number of examples presented to the detector we can easily compute if the results are truly significance or could be produced by chance.

4.2 Comparison between methods

The basic results of detection are summarized in table 4.

Table 4: Summary of the detection per recoding session over all the detection techniques. The codes of the techniques are: PSTH1 – the homogeneous Poisson process, PSTH2 – non-homogeneous Poisson process with fixed bin width. PSTH3 – non-homogeneous Poisson process with variable bin width. HMM1 – firing rate hidden markov model. HMM2 – fine resolution hidden Markov model. CORR – pair-wise cross correlation method. METRIC – Metric distance method. EMPIR – empirical classification.

Sess	PSTH1	PSTH2	PSTH3	HMM1	HMM2	CORR	METRIC	EMPIR
b1	29.9	43.0	39.6	44.7	35.4	24.5	31.1	28.6
b2	37.0	61.3	56.6	65.6	52.7	24.9	55.3	46.5
b3	71.9	93.0	90.4	92.9	55.0	37.2	64.1	86.8
b4	64.1	87.4	86.6	81.6	69.2	41.9	62.1	73.6
b5	43.6	65.8	59.7	72.8	45.9	27.4	49.4	52.4
b6	39.4	56.1	55.0	64.0	47.3	35.0	45.0	44.6
b7	66.0	84.3	82.6	85.4	68.5	59.0	69.0	79.9
b8	44.1	56.7	56.0	65.5	44.0	29.1	48.6	49.6
b9	39.7	54.9	54.7	68.8	50.2	29.7	50.3	48.3
c1	41.5	59.4	57.5	68.8	47.3	31.7	46.9	45.8
c2	49.2	63.2	67.5	69.0	51.3	25.4	47.1	56.9
c3	56.2	81.4	81.9	79.5	58.7	54.9	69.9	79.3
c4	39.5	65.3	63.2	86.7	49.5	33.8	46.8	44.7
c5	46.2	71.6	66.0	66.0	59.1	35.2	60.3	43.0
c6	39.1	67.3	60.9	63.5	51.1	33.3	65.8	34.9
c7	49.9	65.1	61.5	65.5	57.8	27.0	56.0	25.6
c8	55.8	68.7	68.3	69.2	51.4	30.6	61.0	58.0
c9	37.9	65.9	63.6	70.3	49.3	33.2	48.9	51.1
c10	51.6	75.0	75.2	78.2	64.4	36.2	66.1	70.8
Av.	47.5	67.7	65.6	70.5	53.1	34.2	56.2	53.7

4.2.1 PSTH model

The results of detection of the four behavioral modes are given in table 4. From this table, the superiority of the non-homogeneous Poisson process over the homogeneous Poisson process is clearly seen.

The dependency of the non-homogeneous fixed bin size detection method upon the bin size was examined and is displayed for one typical recording session in figure 13. In this figure, the detection percentages of PSTH with bin size varying from 20 ms to 4000 ms are presented for two different recording sessions.

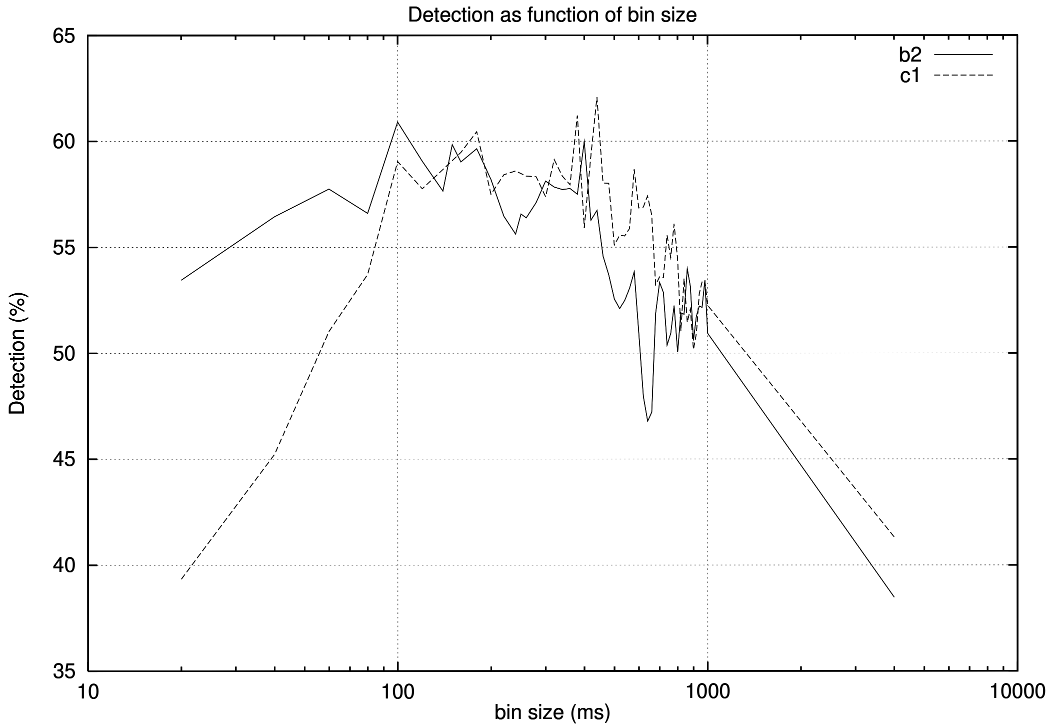

Figure 13: Detection as a function of bin size in two different recording sessions. The detection percentage of PSTH with bin size varying from 20 ms to 4000 ms are presented for two different recording sessions.

Figure 13 shows that the maximal value is reached around the bin size of $100ms$. Thus the detection values shown in table 4 are close to optimal detection. From this point onward the parameters of the PSTH were fixed.

We also examined the different methods for estimating the Poisson rate and comparing fixed bin width with variable bin width. The results show a tendency towards the fixed bin size. In 16 out of the 19 recording sessions examined the detection was better in the fixed bin width algorithm (in both algorithms optimal values were used).

4.2.2 Stochastic models

The firing rate hidden Markov model proved to be the optimal method for the detection of the behavioral modes. The model used was a Left-to-Right model with 8 states. The dependency upon the number of states used can be seen in figure 14. This graph shows that the optimal number of states is between 7 and 15 states.

The fine-resolution method was far from optimal even when compared with other methods

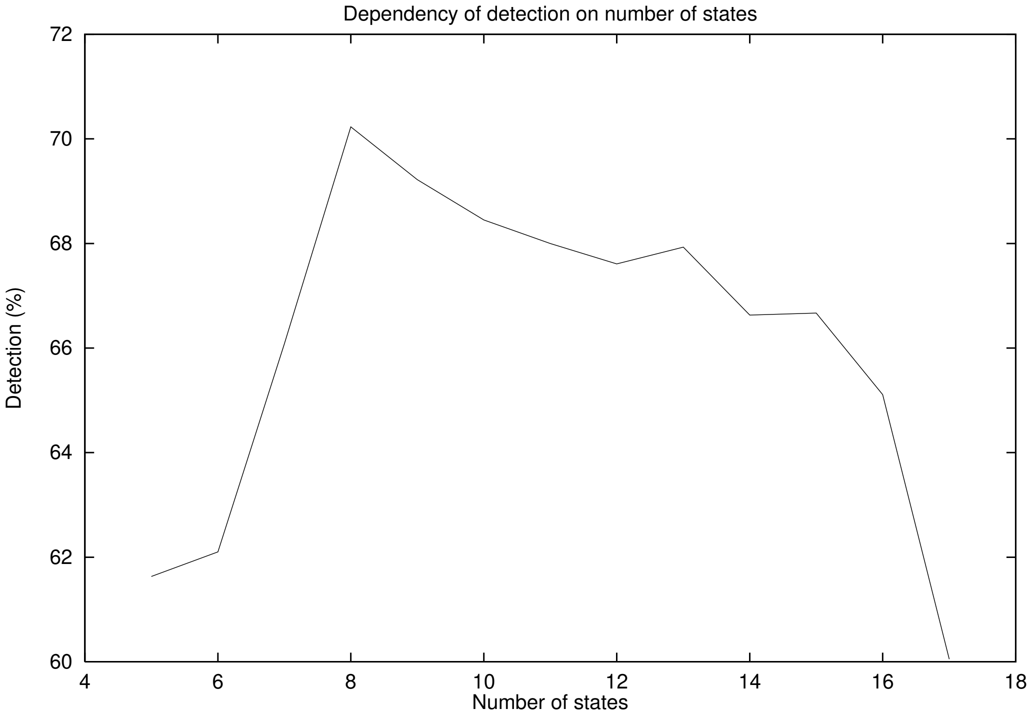

Figure 14: Detection as a function of the number of states in the firing-rate HMM.

(e.g. the different PSTHs'). In this case, if the number of cells is n, there are 2^n different possible observations, ranging from no spikes, to spikes in all the cells. For more details about the method see [86].

The results shown in the *HMM2* column of table 4 were produced by the following initial conditions:

1. Strong diagonal in the transition matrix, with all other elements randomly chosen.
2. Random observations probabilities.
3. 5 states of the model.

4.2.3 Pair-wise cross-correlation

The results shown in table 4 in the cross-correlation column are the outcome of using the basic correlation algorithm described in section 3.4. The bin size used was $13ms$ and the correlation length was 1 second.

Contrary to the results achieved by the other methods, in the cross-correlation method not all the results are significant. Only in 8 sessions out of 19 were significant results found. In the other sessions the same results could have been produced by chance.

Next we examined the following question: is the information in the correlation only partial information of the PSTH, or does it give some novel perspective on the data. In order to answer the question, we conducted the following test: We examined the erroneous trials of the PSTH, and checked the performance of the correlation algorithm on them. These figures were compared with the detection of the correlation over all trails. If the information in the correlation that enables the significant detection was only some partial information from the PSTH, we would not expect significant detection in the erroneous segments. The results of this test are shown in table 5.

Table 5: The detection of the correlation on the whole data, and on erroneous segments of the PSTH. The sessions shown are the ones for which a significant detection was achieved by the correlation method.

Session	Erroneous segments	All segments
b3	38.9	37.6
b4	36.0	43.0
b6	42.5	38.1
b7	48.4	60.0
c3	40.8	52.0
c4	30.3	33.7
c5	23.1	34.0
c10	24.4	40.6
Mean	31.5	42.4

The results shown in this table are not conclusive. On the one hand, the average detection on the erroneous segments is much lower than the over-all detection of the correlation method. On the other hand, this is due to the detection in 3 sessions (c4,c5 and c10). In the other sessions detection remains significant and in some cases (b3 and b6) is even higher than the over-all detection

4.2.4 Metric space method

As already claimed in [92] there is no rigorous method of combining the information from multi unit recordings. In this work we utilized several methods of score combining (e.g. geometrical mean), and used the optimal one for each recording session. This procedure produced an over-fit to the data. The results of using this method are summarized in table 4.

It can be clearly observed from this table that the metric space method is not as strong as the previous ones. The results are significantly less than for the PSTH and HMM methods.

Although these results gained somewhat from the over-fit to the data (as described above).

4.2.5 Empirical classification

We varied the parameters of transforming the spike trains into a symbol list. At first we found the optimal value for the stationary segment of data. The results of this computation are presented in figure 15A. The next step was to use the optimal value for the alpha-bet size bin. The results are presented in figure 15B.

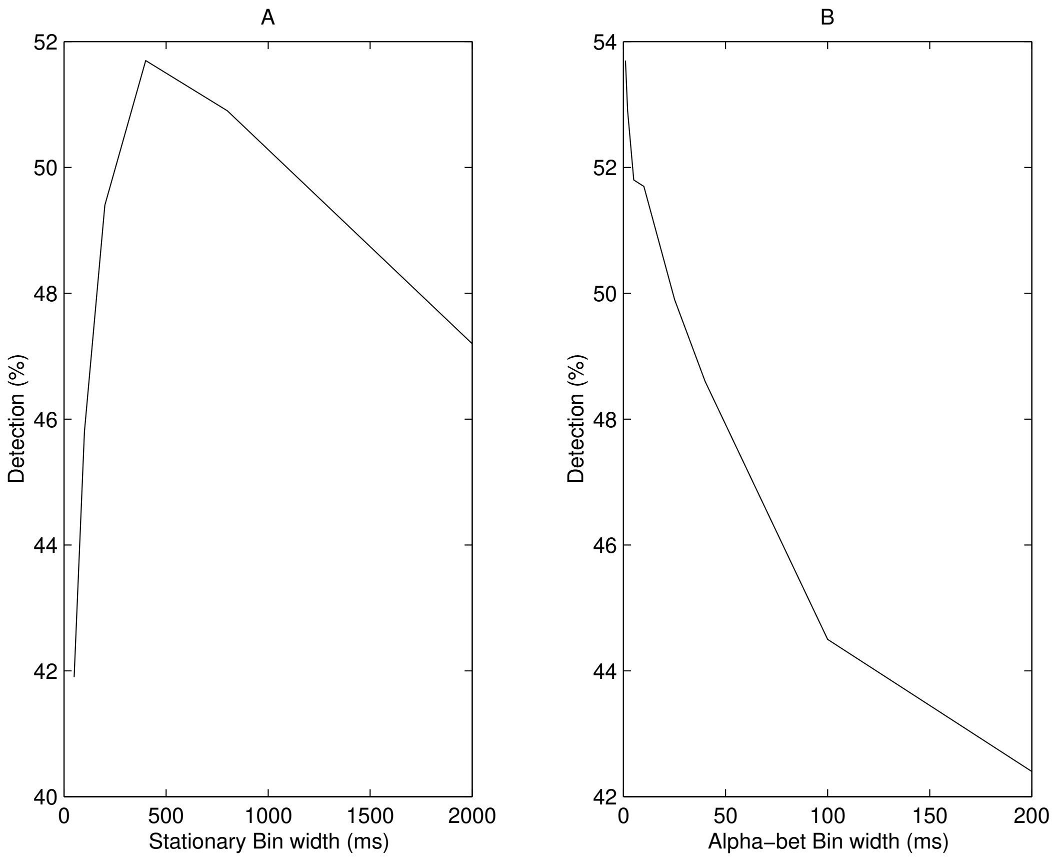

Figure 15: Tuning of the empirical classification bin widths. First we set the optimal stationary bin width, then we set the inside alpha-bet size bin width.

The results shown in table 4 were computed from these optimal values (stationary bin width of 400 ms and alpha-bet size based on 1 ms bins). The results are significantly less than for the PSTH and HMM methods.

4.3 Technique summary

The results reported from this point onward are the ones achieved using the PSTH fixed-bin method, unless otherwise stated. We used this method despite the fact that it did not produce the optimal results as the computation of the HMM was very cumbersome. This is due to the need to re-compute the whole model for each example, so as to avoid an over-fit to the correct data. The penalty for over-fitting the data is presented in section 4.6.

4.4 The amount of cooperation

We examined the contribution of each cell (or pair of cells) to achieving the goal of detection. The question posed was whether the ability to accurately reveal the behavioral mode of the animal is due mainly to a small subset of the cells, or to a collective effort. Two checks were conducted in order to answer this question. In the first, we examined the dependency of detection upon the number of cells used. In the second, we checked the *average* contribution of each cell to the correct result.

4.4.1 Detection based on single cells

The results of the first check are summarized in table 6.

Table 6: The averaged detection of a single cell, using the fixed bin size PSTH. For each recording session the detection based on single cells is shown together with the average detection (two column before last), and the maximum detection value (one column before last). The last line averages the maximum and mean value of detection over all the recording sessions. The last column gives the detection obtained using all the cells. The non-significant (with threshold of 0.001 significance level) detections are marked with a star.

Sess.	cell1	cell2	cell3	cell4	cell5	cell6	cell7	cell8	Av.	Max	PSTH2
b1	42.9	27.2 *	25.2 *						31.8	42.9	43.0
b2	45.0	50.3	51.9	31.5					44.7	51.9	61.3
b3	60.2	48.5	48.0	61.7	37.5	47.3	60.1	46.5	51.2	61.7	93.0
b4	55.1	58.1	47.9	48.5	64.7	48.5			53.8	64.7	87.4
b5	33.7	36.6	39.5	42.4	46.1	36.5	46.6	36.6	39.8	46.6	65.8
b6	19.7 *	34.8	41.4	42.9					34.7	42.9	56.1
b7	42.6	56.9	48.2	28.9 *	46.1	51.8	48.2	72.1	49.4	72.1	84.3
b8	42.9	33.5	42.0	44.8	30.3 *				38.7	44.8	56.7
b9	39.9	44.6	34.2	44.5	41.7	36.5			40.2	44.6	54.9
c1	44.3	37.7	44.0	38.6	35.4				40.0	44.3	59.4
c2	34.3	39.3	48.3	26.4	51.2				39.9	51.2	63.2
c3	56.0	58.9	49.6	55.1	70.0				57.9	70.0	81.4
c4	55.5	42.7	32.6	32.6	42.4				41.1	55.5	65.3
c5	40.6	49.3	42.2	55.5					46.9	55.5	71.6
c6	58.7	42.2	44.0	49.8	59.6				50.9	59.6	67.3
c7	34.9	36.6	48.8	40.7	49.9	45.1			42.7	49.9	65.1
c8	53.8	44.6	29.3*	33.3	53.8	53.0	31.3		42.7	53.8	68.7
c9	44.5	43.1	42.3	45.1	45.4				44.1	45.4	65.9
c10	44.4	47.6	53.9	61.9					51.9	61.9	75.0
Av.									44.3	53.6	67.7

From the table it is clear that both the average and maximal detection per recording sessions are significantly less than the detection achieved using all the cells.

4.4.2 Dependency of detection by the number of cells

In order to visualize the contribution of groups of cells we examined the detection rate using all possible subsets of cells. Starting from single cells, moving on to pairs of cells, and ending with all the cells used together. We compared this type of dependency in three scenarios:

- Recording cortical activity under the delayed response paradigm.
- Recording cortical activity under the box puzzle paradigm.
- Recording basal ganglia activity under the delayed response paradigm.

Typical examples of these three cases is presented in figure 16A,B,C.

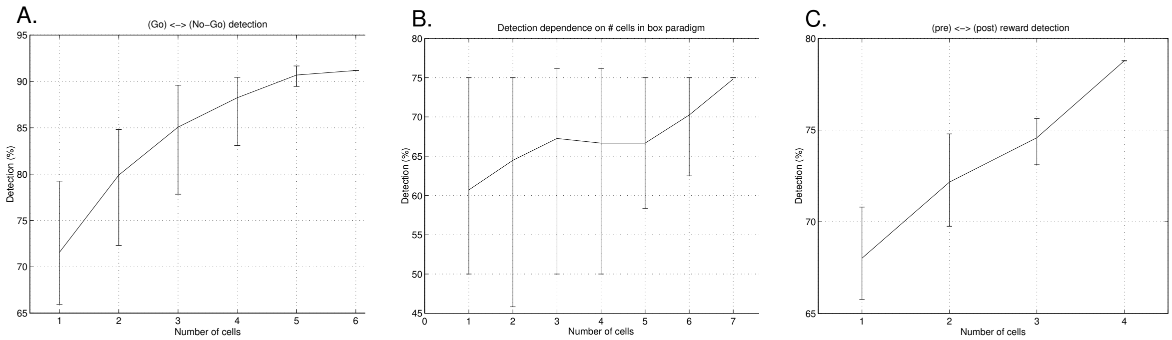

Figure 16: The dependency of the detection rate on the number of cells used. For each number of cells all the permutations were tried and used for the detection problem. The error bars indicate the maximal and minimal detection value for each group of cells. A. Detection in the cortical activity of delayed response. B. Detection in cortical activity in box puzzle. The detection task examined was to differentiate between Box 1 and Box 3 (see section 2.2.2). C. Detection in basal ganglia activity of delayed response.

It can be seen that in parts A and C, as the number of cells increases the detection improves, whereas in part B, the detection of the best cell equals the detection of all the cells together. It is interesting to note that in the box opening paradigm the identity of the best cell varied with the detection goal. For example, in trying to differentiate the opening of two types of boxes, one may find that the best cell for detection was cell number 1. However when trying to differentiate between two *other* types of boxes we found that the only contributing cell was cell number 2.

A more detailed examination of the amount of cooperation is presented later in section 5.

4.5 Relationship between firing rate and detection

Using the single trial detection tool we looked into the relationship between the firing rate of the cell and it's detection power. The hypothesis was that the higher the firing rate of the cell, the higher the detection rate.

Computing the correlation coefficient between the two variables, we reached the value of 0.29. We can conclude from this value that some positive correlation exists between the firing rate and the detection rate - but this correlation is a weak one. A typical example is presented in figure 17

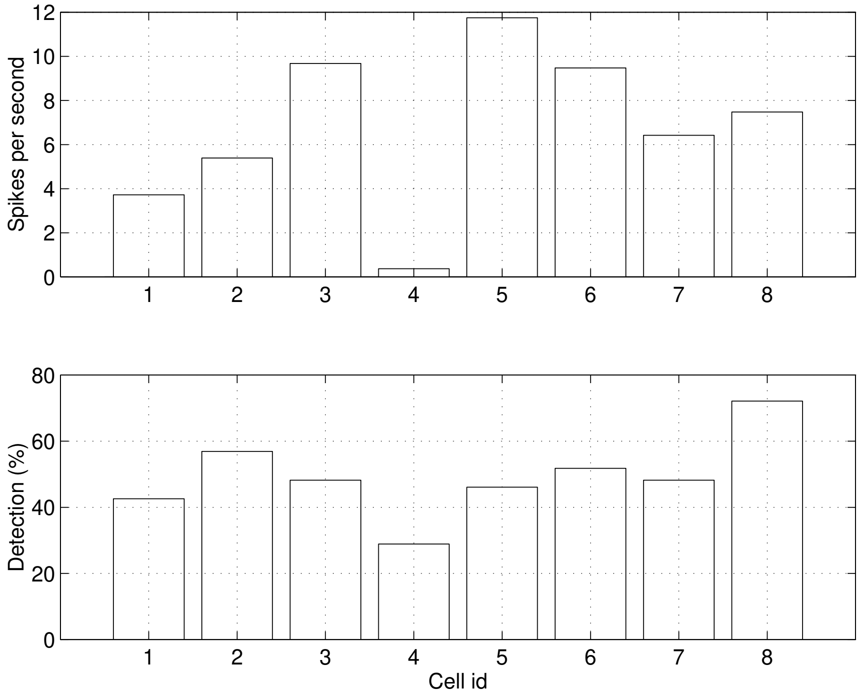

Figure 17: An example of the relationship between firing rate and detection (data taken from session b7). The upper part shows the average firing rate in the delay period for each of the cells. The lower part shows the detection rate computed for each cell separately. The correlation coefficient in this case is 0.46, which is higher than the average correlation coefficient.

Examining the correlation exhibited in figure 17 it can be seen that:

- The cell with the high firing rate (cell number 5) is only fifth when the cells are ranked according to detection rate.
- The cell with highest detection rate (cell number 8) is only fourth in it's firing rate.
- A cell with very low firing rate (cell number 4) is also unable to supply us with reasonable detection.

4.6 Jacks' knife algorithm - the problem of over-fit

The Jacks' knife algorithm is the method for dealing with a small training set that also suffice the test examples. According to this method, whenever a test example is examined, the models are re-computed by subtracting the current test example from them. In this section we will show the influence of **not** using this algorithm on the cortical data. The first example is based on the PSTH algorithm, and the second on the fine resolution HMM. The importance of using the Jacks' knife algorithm was demonstrated throughout all the detection algorithms.

4.6.1 Fixed bin size PSTH

The result is shown in figure 18. In this figure, the detection is shown for bin sizes ranging from 20 ms to 4000 ms. For each such bin size, the detection is given for the non over-fit algorithm (that computes the new PSTH each time without the example in question), and the algorithm which over-fits the data (that uses the pre-computed PSTH models).

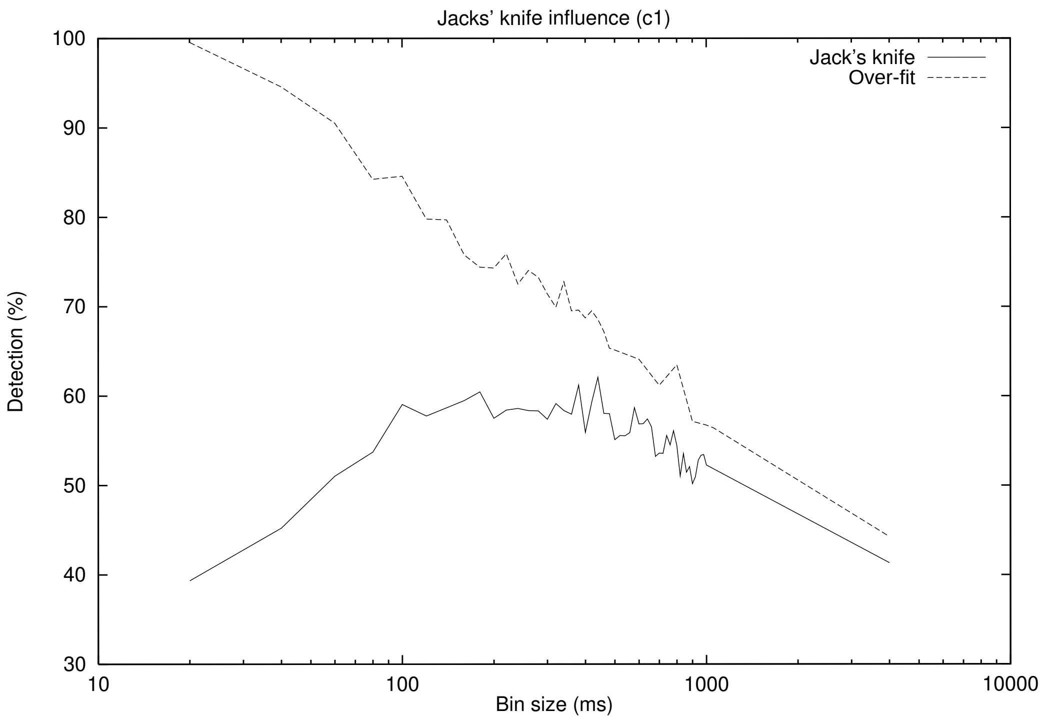

Figure 18: Detection as a function of bin size. The continuous line represents the non over-fit algorithm (with Jacks' knife), whereas the dashed line shows the outcome of over-fit. In this example the number of segments ranged from 50 to 57 in each mode.

It is clear that the over-fitting of data changes the results quite dramatically when the

Table 7: The fine-resolution HMM detection rate using the over-fit method in the initialization, versus. the full Jacks' knife method.

Session	Mean over-fit(%)	Mean Jack-knife (%)
b7	76.2	45.0
c3	77.8	58.0

bin size is short. This is due to the amount of parameters to be estimated. Note, however, that the effect is quite evident even when the number of parameters is rather small as in the case of homogeneous Poisson process, using bin width of 4000 ms.

4.6.2 Fine resolution HMM

Another example is the case of the fine resolution HMM. This example is important because at first sight it might be considered as implementing the Jacks' knife algorithm. In this algorithm the following steps should be taken:

1. A new set of data including all the train examples should be created, excluding the current test example.

2. A new model should be initialized with random initialization.

In this section we will show an example in which the second stage was omitted for the purpose of computational speed [86]. The new data set is, therefore, created and trained without over-fit, but *only* the initialization is based on a train phase that includes the test example. The results of applying the fine-resolution HMM to two recording sessions are shown in table 7. The first column shows the detection results obtained with the Jacks' knife correction, and the second column shows the results of the algorithm that lacks the random initialization.

The change in detection results between these cases is quite dramatic. It is important to note that this is *not* due to the shape of the error function that the HMM is trying to minimize in the learning process [76], but rather to the amount of parameters and the over-fit to the data due to the starting point. Nevertheless, it **is** a weak point of such an algorithm that a small change in the initialization due to over-fit can change the results in such a dramatic manner.

5 Synergy definition

Measuring ways by which several neurons in the brain participate in a specific computational task can shed light on fundamental neural information processing mechanisms. While it is unlikely that complete information from any macroscopic neural tissue will ever be available, some interesting insight can be obtained from simultaneously recorded cells in the cortex of behaving animals. The question we address in this study is the level of synergy, or the level of cooperation, among brain cells, as determined by the information they provide about the observed behavior of the animal.

Our measure of synergy level among cells is information theoretic and was recently proposed by Brenner *et. al.*[30] for analysis of spikes generated by a single neuron. This is the first application of this measure to the quantification of cooperativity among neurons.

5.1 Synergy and redundancy

A fundamental quantity in information theory is the mutual information between two random variables X and Y. It is defined as the cross-entropy (Kullbak-Liebler divergence) between the joint distribution of the variables, $p(x, y)$, and the product of the marginal distributions $p(x)p(y)$. As such, it measures the statistical dependence of the variables X and Y. It is symmetric in X and Y and has the following familiar relations to their entropies[32]:

$$\begin{aligned} I(X;Y) &= D_{KL}[P(X,Y)|P(X)P(Y)] = \sum_{x,y} P(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \\ &= H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \end{aligned} \tag{12}$$

When given three random variables X_1, X_2 and Y, one can consider the mutual information between the joint variables (X_1, X_2) and the variable Y, $I(X_1, X_2; Y)$ (notice the position of the semicolon), as well as the mutual informations $I(X_1; Y)$ and $I(X_2; Y)$. Similarly, one can consider the mutual information between X_1 and X_2 *conditioned* on a given value of $Y = y$,

$$I(X_1; X_2|y) = D_{KL}[P(X_1, X_2|y)|P(X_1|y)P(X_2|y)], \tag{13}$$

as well as its average, the conditional mutual information,

$$I(X_1; X_2|Y) = \sum_y p(y) I_y(X_1; X_2). \tag{14}$$

Following Brenner *et. al.*[30] we define the *synergy level* of X_1 and X_2 with respect to the variable Y as

$$Syn_Y(X_1, X_2) = I(X_1, X_2; Y) - (I(X_1; Y) + I(X_2; Y)), \tag{15}$$

with the natural generalization to more than two variables X. This expression can be rewritten in terms of entropies and conditional information as follows:

$$\begin{aligned} & Syn_Y(X_1, X_2) = \\ = \;& H(X_1, X_2) - H(X_1, X_2|Y) - ((H(X_1) - H(X_1|Y)) + (H(X_2) - H(X_2|Y))) \\ = \;& \underbrace{H(X_1|Y) + H(X_2|Y) - H(X_1, X_2|Y)}_{\textbf{Depends On } Y} + \underbrace{H(X_1, X_2) - (H(X_1) + H(X_2))}_{\textbf{Independent of } Y} \\ = \;& I(X_1; X_2|Y) - I(X_1; X_2). \end{aligned} \tag{16}$$

When the variables exhibit positive synergy value, with respect to the variable Y, they jointly provide more information on Y than when considered independently, as expected in *synergetic* cases. Negative synergy values correspond to *redundancy* - the variables do not provide independent information about Y. Zero synergy value is obtained when the variables are independent of Y or when there is no change in their dependence when conditioned on Y. We claim that this is a useful measure of cooperativity among neurons, in a given computational task.

It is clear from Eq.(16) that if

$$I_y(X_1; X_2) = I(X_1; X_2) \;\; \forall y \in Y \Rightarrow Syn_Y(X_1, X_2) = 0, \tag{17}$$

since in that case

$$\sum_y p(y) I_y(X_1; X_2) = I(X_1; X2). \tag{18}$$

In other words, the synergy value is not zero only if the statistical dependence, hence the mutual information between the variables, is affected by the value of Y. It is positive when the mutual information increases, on the average, when conditioned on Y, and negative if this conditional mutual information decreases. Notice that the value of synergy can be both positive and negative since information, unlike entropy, is not sub-additive in the X variables.

6 Synergy among neurons

Our measure of synergy among the units is based on the ability to detect the behavioral mode from the recorded activity, as we discuss below. As discussed above, synergy among neurons is possible only if their statistical dependence changes with time. An important case in which synergy is **not** expected is pure "population coding"[44]. In this case the cells are expected to fire independently, each with its own fixed tuning curve. Our synergy value can thus be used to test if the recorded units are indeed participating in a pure population code of this kind, as hypothesized for certain motor cortical activity.

Theoretical models of the cortex that clearly predict nonzero synergy include attractor neural networks (ANN)[13] and SynFire chain models(SFC)[1]. Both these models predict changes in the collective activity patterns, as neurons move between attractors in the ANN case, or when different SynFire-chains of activity are born or disappear, in the SFC case. To the extent that such changes in the collective activity depend on behavior, nonzero synergy values can be detected. It remains an interesting theoretical challenge to estimate the quantitative synergy values for such models and compare it to observed quantities.

6.1 Time-dependent cross correlations

In our previous studies[43] we demonstrated, using hidden Markov models of the activity, that the pairwise cross-correlations in the same data can change significantly with time, depending on the underlying collective state of activity. These states, revealed by the hidden Markov model, depend in turn on the behavior, and enable its' prediction. Dramatic and fast changes in the cross-correlation of cells has also been shown by others[11]. These findings indicate directly that the statistical dependence of the neurons can change (rapidly) with time, in a way correlated to behavior. This clearly suggests that nonzero synergy should be observed among these cortical units, relative to this behavior. In the present study this theoretical hypothesis is verified.

6.2 Redundancy cases

If, on the other hand, the conditioned mutual information equals zero for all behavioral modes, i.e.

$$I_y(X_1; X_2) = 0 \;\; \forall y \in Y, \tag{19}$$

while $I(X_1;X_2) > 0$, we expect to get negative synergy, or redundancy among the cells, with respect to the behavior variable Y. We observed clear redundancy in another part of the brain, the basal ganglia, during the same experiment, when the behavior was the pre-reward and post-reward activity. In this case, different cells provide exactly the same information, which yields negative synergy values.

7 Experimental results

7.1 Synergy measurement in practice

To evaluate the synergy value among different cells, it is necessary to estimate the conditional distribution $p(y|x)$ where y is the current behavior and x represents a single trial of spike trains from the considered cells. Estimating this probability, however, requires an underlying statistical model. Otherwise there is never enough data, since spike trains can never be reproduced exactly. In this work, we chose the rate representation, which is the simplest to evaluate. The estimation of $p(y|x)$ goes as follows:

- For each of the M behavioral modes $(y_1, y_2.., y_M)$ collect spike train samples (the *training* data set).
- Using the training sample, construct a Post Stimulus Time Histogram (PSTH), i.e. the rate as a function of time, for each behavioral mode.
- Given a spike train, outside of the training set, compute its probability to be a result of each of the M modes.
- The spike train is considered correctly classified if the most probable mode is in fact the true behavioral mode, and is otherwise considered incorrectly classified.
- The fraction of correct classification, for all spike trains of a given behavioral mode y_i, is taken as the estimate of $p(y_i|x)$, and denoted P_{c_i}, where c_i is the identity of the cells used in the computation.

In the case of only two categories of behavior and for a uniform distribution of the different categories, the value of the entropy $H(Y)$ is the same for all combinations of cells, and is simply $H(Y) = -\sum_y p(y)\log_2(p(y)) = \log_2 2 = 1$. The full expression (given in units of bits) for the synergy value can thus be written as follows:

$$\sum_x p(x) \left[-\sum_y P_{c_1,c_2} \log_2(P_{c_1,c_2}) \right] \stackrel{?}{>} \tag{20}$$
$$1 + \sum_x p(x) \left[-\sum_y P_{c_1} \log_2(P_{c_1}) \right] + \sum_x p(x) \left[-\sum_y P_{c_2} \log_2(P_{c_2}) \right] ,$$

If the first expression is larger than the second than there is (positive) synergy and vice versa for redundancy. However there is one very important caveat. As we saw, the computation of the mutual information is not exact, and what one really computes is only a lower bound. If the bound is tighter for multiple cell calculation, the method could falsely infer positive synergy, and if the bound is tighter for the single cell computation, the method could falsely infer negative synergy. We have shown in great detail above (see section 4) that the method we use for this estimation is quite reasonable and robust. We therefore believe that we have an estimate of synergy that might even be considered conservative (i.e. less positive).

7.2 Observed synergy values

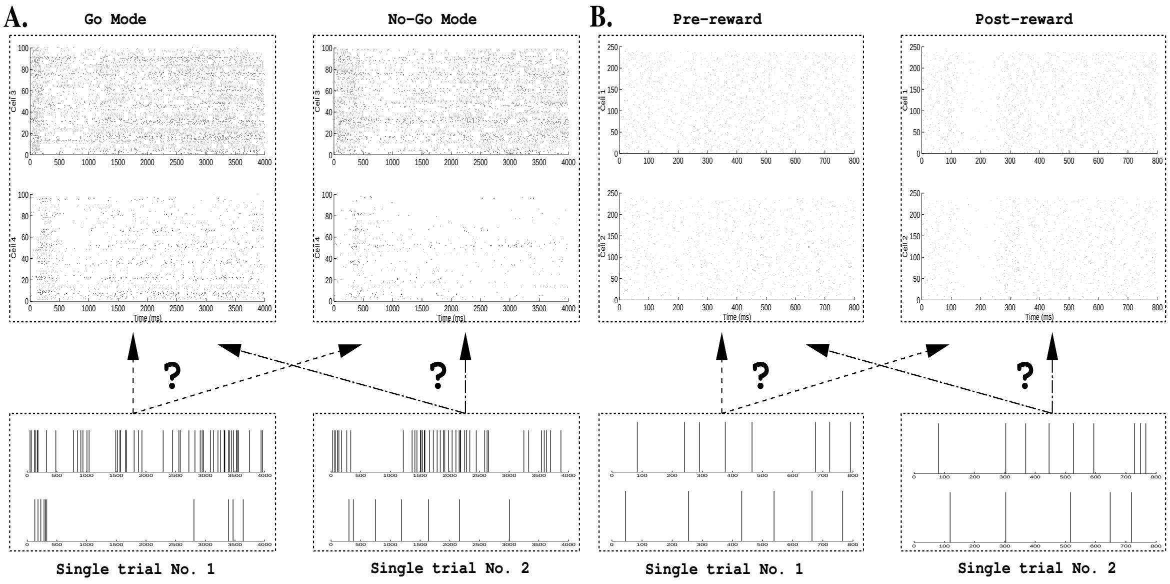

Figure 19: Raster displays of simultaneously recorded cells in the 2 different areas. In each area there were 2 behavioral modes.

In the first set of experiments we tried to detect the behavioral mode during the delay-period of correct trials. In this case the two types of behavior were the "Go" and the "No-Go" described in the introduction. An example of this detection problem is given in figure 19A. In this figure there are 100 examples of multi-electrode recordings of spike trains during the

Table 8: Examples of synergy among cortical neurons. For each example, the mutual information of each cell separately is given together with the mutual information of the pair. The matching detection probability (average over $p(y|x)$) is given in parenthesis. The last column gives the percentage of increase from the mutual information of the single cells to the mutual information of the pair. The table gives only those pairs for which the percentage was larger than 20% and the detection rate higher than 60%.

Session	Cells	Cell1	Cell2	Both cells	Syn (%)
b3	5,6	0.068 (64.84)	0.083 (66.80)	0.209 (76.17)	38
b4	1,4	0.201 (73.74)	0.118 (69.70)	0.497 (87.88)	56
b4	3,4	0.082 (66.67)	0.118 (69.70)	0.240 (77.78)	20
b6	0,3	0.062 (62.63)	0.077 (66.16)	0.198 (75.25)	42
b6	1,2	0.030 (60.10)	0.051 (63.13)	0.148 (72.22)	82
c7	2,3	0.054 (62.74)	0.013 (61.50)	0.081 (68.01)	20
c9	0,2	0.074 (65.93)	0.058 (63.19)	0.160 (73.08)	21
c9	0,4	0.074 (65.93)	0.042 (62.09)	0.144 (71.98)	24
c9	3,4	0.051 (62.09)	0.042 (62.09)	0.111 (69.23)	20
c10	0,1	0.070 (65.00)	0.063 (64.44)	0.181 (74.44)	36

delay period. On the left is the “Go-mode” data and on the right the “No-Go mode”, for two cells. In the lower part there is an example of two single spike trains that need to be classified by the mode models.

Table 8 presents some examples of detection results obtained by using 2 cells independently, and by using their joint combination. It can be seen that the synergy is positive and significant. We examined 19 recording sessions of the same behavioral modes for two different animals and evaluated the synergy value. In 18 out of the 19 sessions there was at least one example of significant positive synergy among the cells.

For comparison, we analyzed another set of experiments in which the data was recorded from the striatum in the basal ganglia. An example of this detection is shown in figure 19B. The behavioral modes were the “pre-reward” vs. the “post-reward” periods. Nine recording sessions for the two different monkeys were examined, using the same detection technique. Although the detection results improve when the number of cells increases, no positive synergy value was found in any of these recordings. For most of the data, the synergy value was close to zero, i.e. the mutual information among two cells jointly was close to the sum of the mutual information of the independent cells, as expected when the cells exhibit (conditionally) independent activity.

The prevailing difference between the synergy measurements in the cortex and in the TANs’ of the basal ganglia is also strengthened by the different mechanisms underlying these cells. The TANs are assumed to be *global* modulators of neuronal interactions in the striatum, a relatively simple task, whereas the information processed in the frontal cortex in this task is believed to be much more collective and complicated. Here we suggest a first handle for quantitative detection of such different neuronal activities.

Part III

Spotting a template

8 Introduction

Our goal in this work is the spotting of a specific pattern of multi-neural spike trains within the ongoing neural activity of these neural cells. We assume, for the sake of simplicity, that the pattern is a given before the search, and we do not look here for ways of self-constructing such patterns (a possibility which also exists).

Given such a pattern, a fundamental question is: How unique or characteristic is this pattern to this particular stimulus? In other words, how specific is this “code word” in the ongoing activity of these cells? To answer this question one should try to detect, or *spot*, this specific pattern within a *single* recording, independently of any stimulus. This is a rather difficult pattern recognition task, considering the low firing rate of these cells.

The decision theoretic problem of spotting a pattern of sequenced data in a “noisy background” is classically considered a binary statistical hypothesis test, where the two possible sources are a statistical model of the pattern and a background statistical model. Given a complete description of the two sources, there is a fully understood statistical approach, the Likelihood-Ratio-Test (LRT), first derived by Neyman and Pearson (see e.g. [62]). When one or more of the possible sources are not fully defined, the LRT is not applicable.

In many practical cases all one knows is that the sources belong to a certain parametric family, and is given a training sequence emitted from one of the sources. For this problem, also known as the two-sample problem [62], a decision is made about whether the training sequences and a candidate test sequence were both emitted from the same source. An asymptotically optimal information theoretic approach to the two-sample problem was proposed by Ziv in [100] and extended by Gutman in [48]. Their method relies on the information theoretic notion of “typicality” of the empirical sequences and is proven to provide the best exponential decay of the probability of classification error. Strictly speaking, their methods are restricted to finite alphabet Markov sources, for which the empirical entropy can be efficiently estimated via universal compression. The above information theoretic approach clearly suggests, however, that an optimal detection method is possible without knowing the specific statistical model of the background, using the "empirical type" of the classified sequence. The main drawback of these methods lies in the need to estimate the probability over the finite alphabet from the data.

In the next section we will present the basic setup and definition of this work. The statistical assumption, together with the exact details of treating the spike trains. The third section will show the problems that may occur when using the wrong background model for spotting a pattern in such data. The fourth section will present a more plausible background model. The calculation for this background model will be given in detail.

9 Definitions

9.1 Statistical assumptions

In this work we will concentrate on detecting a neural response in multi-unit recordings. An example of such a response is given in figure 20A and B. These figures show several of the problematic features of neuronal data, e.g. the low firing rate on the one hand, and the fast changes in rate on the other. It can be seen that a prominent response is observed in this data (seen in part C). Nevertheless, it is far from trivial to observe the same response in a single trial.

The response is mainly manifested in elevation in firing rate (between 150 and 220 ms after the alignment point), followed by a reduction in rate (until 400 ms), and another elevation (until 500 ms). Due to the low firing rate of the cells – a few spikes per second – this whole pattern may be determined by the existence (or absence) of a single spike.

For the statistical description of this data we work under the common assumption that the spike trains are the outcome of a non-homogeneous Poisson process [1]. According to this view the spikes within a short time (a bin) are obeying a Poisson distribution. A pattern to be searched in such data is therefore parameterized by a sequence of M firing rates $(\lambda_1, \lambda_2, ..., \lambda_M) = \underline{\lambda}$. The value of $\underline{\lambda}$ are the estimated expected number of spikes in the corresponding bins of the pattern. Notice that when the bin sizes are sufficiently small such a pattern can account for arbitrarily precise spike timing. When the bins are wider – larger variability and randomness in the pattern is allowed but there are fewer parameters to estimate. This pattern is therefore of sufficient generality, where the bin sizes are determined,

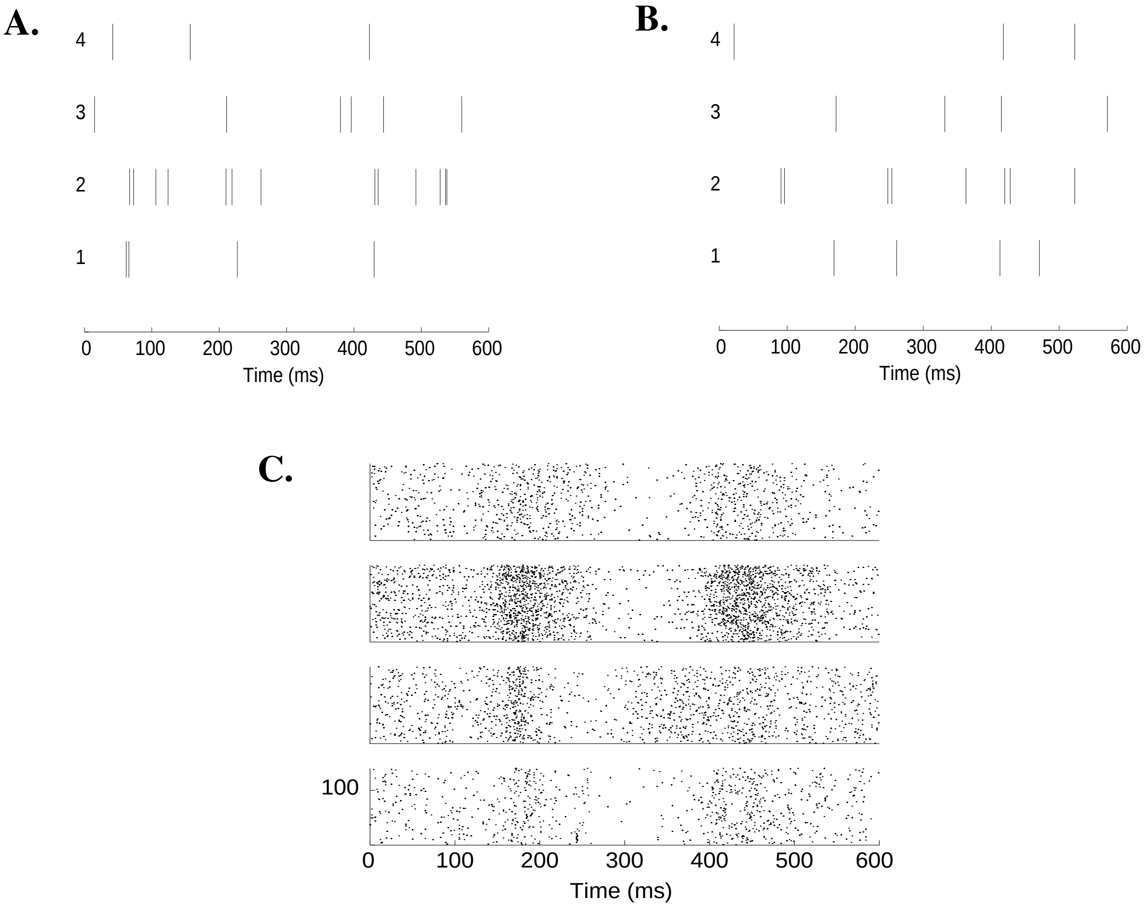

Figure 20: Spike trains of single trial and alignment of many repetitions. A. B. show examples of 4 cells recorded simultaneously from the basal ganglia. Each line represents the action potential of one cell. C. shows the alignment of many repetitions according to some external stimulus.

such that the spike-rate parameters are reliably estimated from the training data. The $\underline{\lambda}$ rates are usually estimated by alignment of several spike trains around an external stimulus. An example of such an alignment was shown in figure 7.

An observed firing sample is a vector of spike counts (number of spikes found in each bin *in a single trial*) $(n_1, n_2, ...n_M) = \underline{n}$. This vector is the outcome of a counting process which splits the data into bins and within each bin, counts the number of spikes. The observed firing sample will be used in this work as the data point to be searched using the pattern defined above. Examples of slicing the data into bins and computing the count vectors are shown in figure 21.

9.2 The log-likelihood distribution

The basic statistical relationship between the data points and the pattern is the likelihood, which is the statistical value that defines how likely a pattern is, given a data point. Given

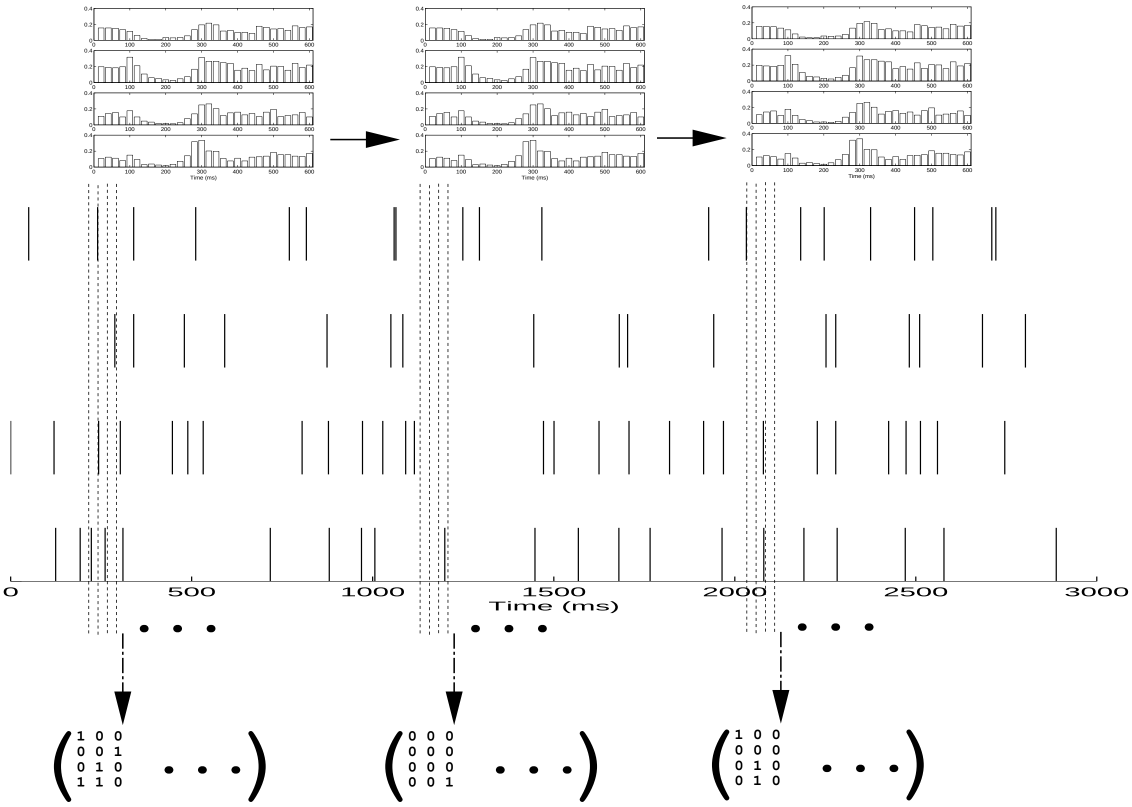

Figure 21: An example of the spotting process. The template (shown above) is shifted on the spike trains. Each shift produces a parsing of the spike trains into bins, where the number of spikes is counted in each bin . The output matrices are shown in the lower part. The task of the spotting is to assign probabilities to the combination of counting matrices and the template being sought.

a variable rate Poisson pattern with M bins

$$(\lambda_1, \lambda_2, ..., \lambda_M) = \underline{\lambda}\ , \qquad \sum_{i=1}^{M} \lambda_i = \Lambda\ , \tag{21}$$

and a multi-cell spike train, we can produce the sample sequence of counts

$$(n_1, n_2, ..., n_M) = \underline{n}\ , \qquad \sum_{i=1}^{M} n_i = N, \tag{22}$$

by slicing the spike trains into bins with the same width as the bins used to estimate the rate Poisson pattern. The likelihood of the observed counts under the pattern is:

$$p_{\underline{\lambda}}(\underline{n}) = \prod_{i=1}^{M} e^{-\lambda_i} \frac{{\lambda_i}^{n_i}}{n_i!}\ . \tag{23}$$

When summing this probability over *all* possible count sequences with the same total number of spikes, N, one obtains,

$$\sum_{\underline{n}} p_{\underline{\lambda}}(\underline{n}) = \sum_{\underline{n}} \prod_{i=1}^{M} e^{-\lambda_i} \frac{{\lambda_i}^{n_i}}{n_i!} = e^{-\Lambda} \frac{\Lambda^N}{N!}\ , \tag{24}$$

which is the well known compositionally rule for Poisson distributions. In other words, the Poisson distribution remains invariant under changes of the bin sizes when summing over all possible internal partitions.

In the current formulation we do not give special treatment to the relationship between bins from the same cell. We assume all bins to be independent of each other no matter which cell they were recorded from.

9.3 Spotting paradigm

Using the building block defined in this section we can now construct the search for a pattern in a continuous data. We first need to have a statistical description of the pattern, which holds the following details:

- The length of the pattern.
- Number of cells used.
- The bin width.
- The values of $\underline{\lambda}$ in those bins.

We can now build a window whose width is the width of the pattern and slide it upon the continuous spike trains. This sliding is performed with short pre-defined shifts (shorter than the PSTH bin width), and creates frames of data. We can now compute the likelihood of each such frame and conclude whether it is similar to the original pattern sought. The main issue that will be discussed in the following sections is the exact details of such computation, or, to be more specific, what we should consider as the background model for our data. A background model should describe the activity of the cells when the pattern is not presented. As described above, we do not have a reasonable background model at hand, and we require a plausible one for this search. Figure 21 gives an example of such a spotting paradigm.

10 Selection of background model

10.1 Using an incorrect background model

The LRT can be shown to be optimal given a correct background model. In this section we will briefly show the consequences of using an incorrect background model. The simplest assumption regarding the background may be that it is constant over all the data. The LRT in this case is reduced to the likelihood of the examined data point (Eq. 23).

The drawback of this method is quite evident when we look at a simple example. Suppose that we want to search for one of the patterns given in figure 7 . It can be seen from our formulation that the frame with the highest likelihood for that pattern is the frame that contains no spikes at all. This is due to the fact that for a Poisson variable of value less than 1 the most likely value is 0. Applying this procedure to data with high variability would supply us only with the frames in the data whose spike count is the lowest. This can be demonstrated by selecting different similarity thresholds for the spotting under this background model. When the threshold is set higher (so that the probability of crossing it becomes lower) it can be seen that the total number of counts in the spotted examples goes down monotonically, as shown in figure 22. Nevertheless, the problem shown here is not limited to cases in which the expected number of counts in each bin is very small. Rather, it is a symptom of using an inadequate background mode.

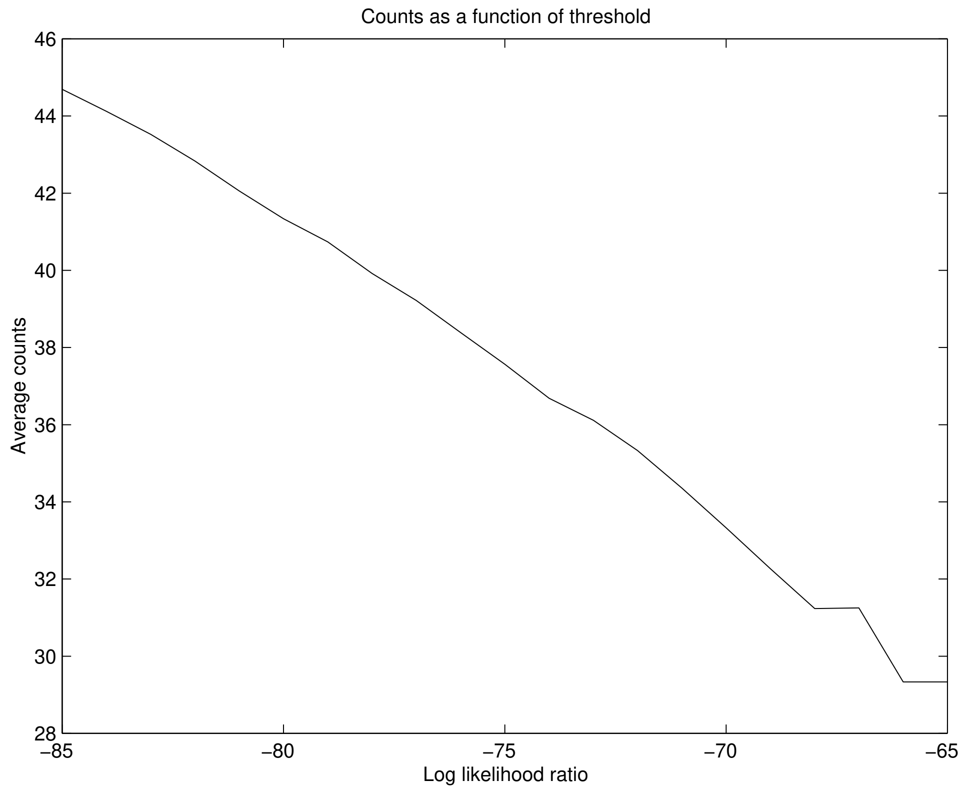

Figure 22: The average number of spikes in examples that crossed the threshold, decreases monotonically as the threshold of similarity is increased. The data presented here was computed using the following process: We chose a uniformly distributed set of $\underline{\lambda}$, and created 10,000 simulations of random Poisson spike trains according to the correct $\underline{\lambda}$. For each of these simulations we counted the number of spikes and computed the likelihood with the assumption of constant background.

10.2 Adversary background model

We suggest using an adversary background model. This model is adversary in the sense that it has a first order statistics, as in our data, and uses this data to confuse us. It produces examples with probability according to the given distribution family with parameters taken from the first order statistics. The background model cannot distinguish between the different examples (all having the same first order statistics) – and therefore they all have the same probability.

The following spotting problem is an example of the use of an adversary background model. Suppose we want to spot the hand-written digit "2" - in a natural writing environment, i.e. the person can write whatever she wishes and we attempt to spot only the cases of "2". For that purpose we can collect many examples of how people write the digit – but it is a much harder task to produce a model that describes the default scribbling of a person. From this point onward we will assume that we have a statistical model for the writing of "2", and we were also able to estimated its parameters using our training examples.

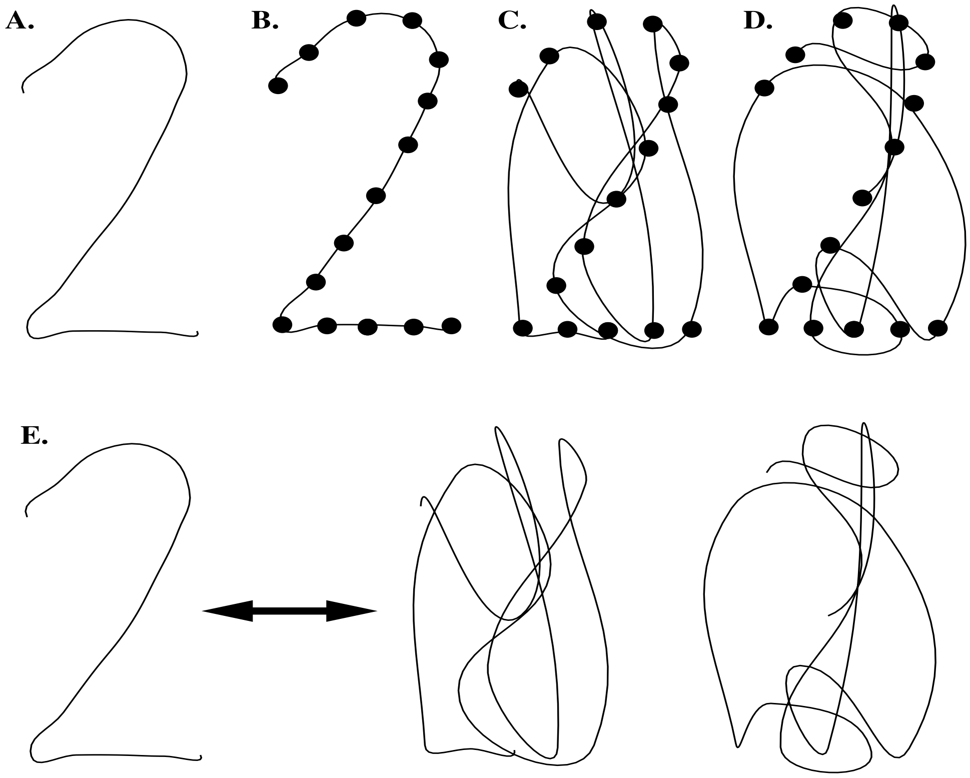

Figure 23: An adversary background model for spotting the digit "2" in an unknown background. In the first row we see a data example of "2" (the figure to the left), followed by indication of the first order statistics being used (the points on the digit). We show two examples of our background adversary model which has this first order statistics. The second row shows that the LRT is computed between the original example (and our statistical model), and the background examples.

Given a scribbling, we would divide it into N segments and would call the position of the end-points of these segments – our first order statistics. The adversary background model would therefore be composed of all the scribbling that goes through the same end-points as our original example. An example can be seen in figure 23.

In figure 23A we see the example presented to the spotter. The first order statistics defined in this example are given in 23B, and consist of a finite set of points defined for the scribbling. Furthermore, we see two examples of the adversary background which imitate the scribbling up to a first order statistics, i.e. they go through the same points defined on the scribbling (23C,D). The ultimate goal is to compute the LRT of the example with the background, as seen in 23E, and to differentiate between 23A and 23C,D.

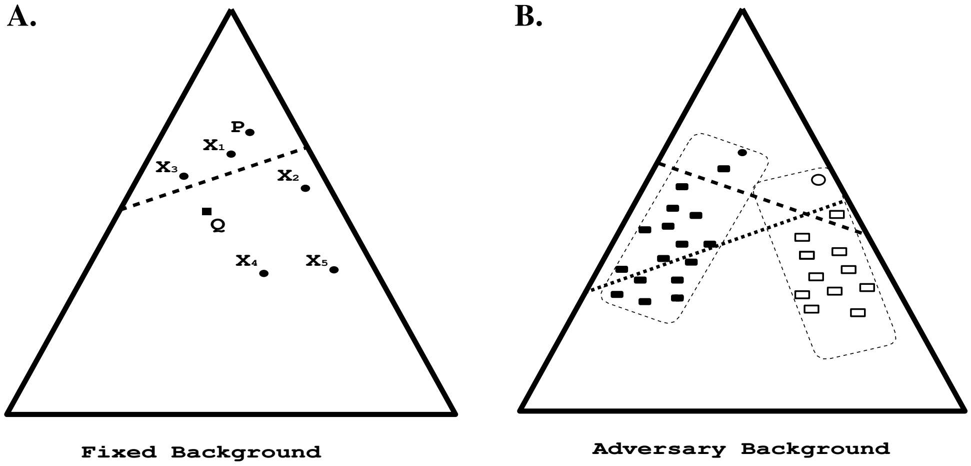

Figure 24: The spotting via adversary background model shown in the probability simplex. A. shows the regular spotting mechanism using the probability simplex. We have to models - marked as **P** and **Q**, and given a set of data points - each is spotted according to the LRT of the models – which is represented as a linear separator between P and Q. B. Using the adversary background model - we have, for each data point, a *different* background model, and therefore a different separator. The different examples are marked either with filled rectangles (example 1) or with empty ones (example 2).

The adversary model is optimal if all that is known about the background is the first order statistics. This use of uniform distribution over the permutations of the data is optimal due to the principle of maximum entropy. The background model is the maximum entropy distribution subject to the partial information of the first order statistics. Any other classifier which does not impose a uniform distribution can be seen as adding some structure to the background model. If this structure does not stem from correct information about the background it would on average *increase* the probabilistic distance between the background and the pattern. Such an increase creates a decision boundary which is less tight, and therefore the decision would be less accurate.

In general what we have here is a case in which the LRT, which can be seen as a linear separator in the probability simplex (see [32]) can be extended according to our data points. An example of this phenomenon is shown in figure 24. In this figure, we see two linear separators, each for a different background model. This figure shows how some of our data points would not have been spotted using a wrong background model.

Using the adversary background model, one still have to choose the first order statistics to be used, i.e. to what extent can the adversary background imitate the data points. We also need to choose the parameters of these statistics, i.e. the number of points used in the case described above. The building of an adversary background model is in general a difficult problem, and in this work we solve it for a specific case.

11 Computing the adversary background model

11.1 Adversary background model and types

In the case of spotting a neuronal response in an unknown background activity we choose our statistic as the *type class* of our data point. The type class [32] is defined as the class of all spike patterns that have the same empirical distribution of bin-counts, i.e., c_0 empty bins, c_1 bins with one spike, c_2 with 2 spikes, etc. The type class is completely characterized by the vector of numbers $c_0, c_1, c_2, ...$ Given a data point, we therefore need to calculate the LRT between the data and the background model consisting of all the possible manifestations of the same type class. In other words, we need to compute the probability of the data point according to the distribution of the type class.

In order to do that, we will show in the following sections what the distribution of the type class is, and how to estimate its parameters. We will start by showing the distribution of the log-likelihood distribution (23), and later on show how to break-down this distribution into the much narrower type class distributions. Finally we will present the method for computing the LRT of a given data example according to the type class distribution.

11.2 The log-likelihood distribution

Let us examine the logarithm of the likelihood, $\log p_{\underline{\lambda}}(\underline{n})$, in Eq.(23). This expression is a sum over M independent random variables,

$$\begin{aligned} \log p_{\underline{\lambda}}(\underline{n}) &= \sum_{i=1}^{M} \left(-\lambda_i + n_i \log \lambda_i + \log(n_i!)\right) \\ &= -\Lambda + \sum_{i=1}^{M} \left(n_i \log \lambda_i + \log(n_i!)\right) \ . \end{aligned} \tag{25}$$

When N and M are large enough, this is the sum over many independent random terms, and by the central limit theorem the distribution of this sum approaches a normal (Gaussian) distribution, centered around its mean. The mean is simply given by:

$$\begin{aligned} \mu_{\underline{\lambda}} &= \langle \log p_{\underline{\lambda}}(\underline{n}) \rangle_{\underline{\lambda}} \\ &= -\Lambda + \sum_{i=1}^{M} \lambda_i \log \lambda_i - \sum_{i=1}^{M} \langle \log(n_i!) \rangle_{\underline{\lambda}} \end{aligned} \tag{26}$$

and can be easily determined from the model rate vector $\underline{\lambda}$. The last term can be evaluated numerically for each n_i (notice that the n_i are small and the Stirling approximation is not valid). To illustrate this, suppose that $\lambda_i = 0.15$, then $\langle \log(n_i!) \rangle_{\underline{\lambda}}$ is governed by the first four values for $n!$, i.e.,

$$\langle \log(n_i!) \rangle_{\underline{\lambda}} \approx 0 \times e^{-0.15} + 0 \times e^{-0.15} 0.15 + \log(2) \times e^{-0.15} \frac{0.15^2}{2} + \log(6) \times e^{-0.15} \frac{0.15^3}{6} \ . \tag{27}$$

The evaluation of the variance of the sample log-likelihood, (Eq. 25), under the model distribution, is similar:

$$\sigma^2_{\underline{\lambda}} = \mathbf{Var}(\log p_{\underline{\lambda}}(\underline{n})) = \mathbf{Var}\left(-\Lambda + \sum_{i=1}^{M} n_i \log(\lambda_i) + \log(n_i!)\right) \tag{28}$$

The first term, $-\Lambda$, is constant and its variance is 0 and we are left with

$$\sum_{i=1}^{M} \mathbf{Var}\left(n_i \log(\Lambda) - \log(n_i!)\right) \ . \tag{29}$$

Using the expression for $\mu_{\underline{\lambda}}$ the variance of each term in this sum can be written as

$$\langle (n_i \log \lambda_i)^2 \rangle \quad + \quad \log^2(n_i!) - 2 \log \lambda_i n_i \log(n_i!) \rangle - \left(\lambda_i \log \lambda_i - \langle \log(n_i!) \rangle\right)^2$$

$$
\begin{aligned}
= (\lambda_i \log \lambda_i)^2 \quad &+ \quad \langle \log^2(n_i!)\rangle - 2\log\lambda_i \langle n_i \log(n_i!)\rangle \\
&- \quad \left(\lambda_i^2 \log^2 \lambda_i - 2\lambda_i \log \lambda_i \langle \log(n_i!)\rangle + \langle \log(n_i!)\rangle^2\right) \\
= \lambda_i \log^2 \lambda_i \quad &+ \quad \mathbf{Var}(\log(n_i!)) - 2\log\lambda_i \langle n_i \log(n_i!) - \lambda_i \langle \log(n_i!)\rangle\rangle \ .
\end{aligned} \tag{30}
$$

The model and its sample log-likelihood distributions are shown in figure 25.

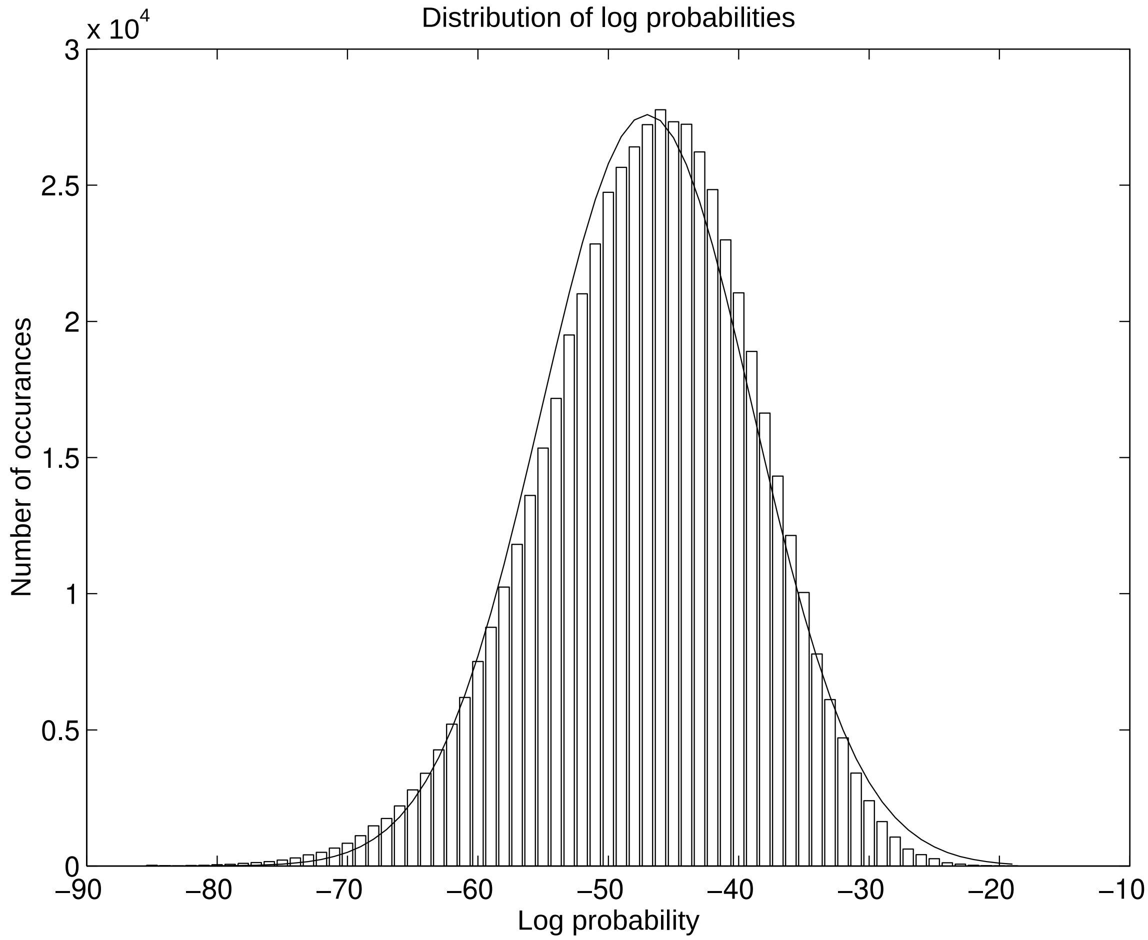

Figure 25: The distributions of the model log-likelihood $\log p_{\underline{\lambda}}(\underline{n})$ for samples generated by the model (the bars) compared with the estimation (solid line).

11.3 Computing the log-likelihood distribution of a sample class

Denoting by $T(\underline{n})$ the type class of the count vector $\underline{n}$ and by $c_i(\underline{n})$ the number of bins with i spike counts in $\underline{n}$, clearly, for patterns with $M = \sum_i c_i$ bins, the size of $T(\underline{n})$ is given by:

$$
|T(\underline{n})| = \frac{M!}{\prod_{i=1}^{M} c_i(\underline{n})!}, \tag{31}
$$

where $|\ |$ is the number of elements in a variable.

Since sample type classes are disjoint sets, the sample log-likelihood distribution is uniquely broken-down into a sum of the log-likelihood distributions for each type. That is,

$$P_{\underline{\lambda}}(\log p_{\underline{\lambda}}(\underline{n})) = \sum_{T(\underline{n})} P_{\underline{\lambda}}(T(\underline{n})) \times P_{\underline{\lambda}}(\log p_{\underline{\lambda}}(\underline{n})|T(\underline{n})) \ . \tag{32}$$

When given a sample, its type is easily computed. The idea, therefore, is to use only one term in this sum for detection, rather than the whole sum. In other words, the sample type serves as a sufficient statistic, together with the sample likelihood, for each term in this mixture distribution. It only remains to evaluate the different distributions in this process of breaking-down.

The probability of the type class $T(\underline{n})$ under the model $\underline{\lambda}$ can be determined through

$$P_{\underline{\lambda}}(T(\underline{n})) = \sum_{n' \in T(\underline{n})} p_{\underline{\lambda}}(\underline{n'}) = |T(\underline{n})| \langle p_{\underline{\lambda}}(\underline{n}) \rangle_{T(\underline{n})} \approx |T(\underline{n})| e^{\langle \log p_{\underline{\lambda}}(\underline{n}) \rangle_{T(\underline{n})}}, \tag{33}$$

where $\langle ... \rangle_{T(\underline{n})} = \frac{1}{|T(\underline{n})|} \sum_{n' \in T(\underline{n})} ...$ denotes an average over all the members of the type class $T(\underline{n})$, i.e., an average over all the $|T(\underline{n})|$ permutations of bins.

The last exponential approximation in Eq.(33) is based on the fact that $p_{\underline{\lambda}}(\underline{n})$ is a product over many positive statistically independent terms (each with a well behaving distribution) and thus distributed by a $log - normal$ distribution for a large enough M (its' log "self-average").

The type probabilities, $P_{\underline{\lambda}}(T(\underline{n}))$, should be normalized to 1, i.e.,

$$\sum_{T(\underline{n})} P_{\underline{\lambda}}(T(\underline{n})) = 1 \ , \tag{34}$$

so each term can be considered as the type prior.

Where T is the number of configurations of counts into bins, $Type(t)$ is the number of possibilities to put the t configuration into the M bins, and the remaining Poisson probability is just the probability of the configuration. A configuration is defined as an infinite set $c_0, c_1, ...$, where c_0 is the number of empty bins, c_1 is the number of bins with one count, etc. Therefore

$$P(Type(t)) = c_0 e^{-\lambda_i} \times c_1 e^{-\lambda_i} \lambda_i \times c_2 e^{-\lambda_i} \frac{\lambda^2}{2} \times c_3 e^{-\lambda_i} \frac{\lambda^3}{6} ... \tag{35}$$

11.4 The log-likelihood distribution in each type

Next we evaluate the parameters of the distribution of $\log p_{\underline{\lambda}}(\underline{n})$ in each type. As can be seen from the histogram of the log-likelihood in the type, figure 26.

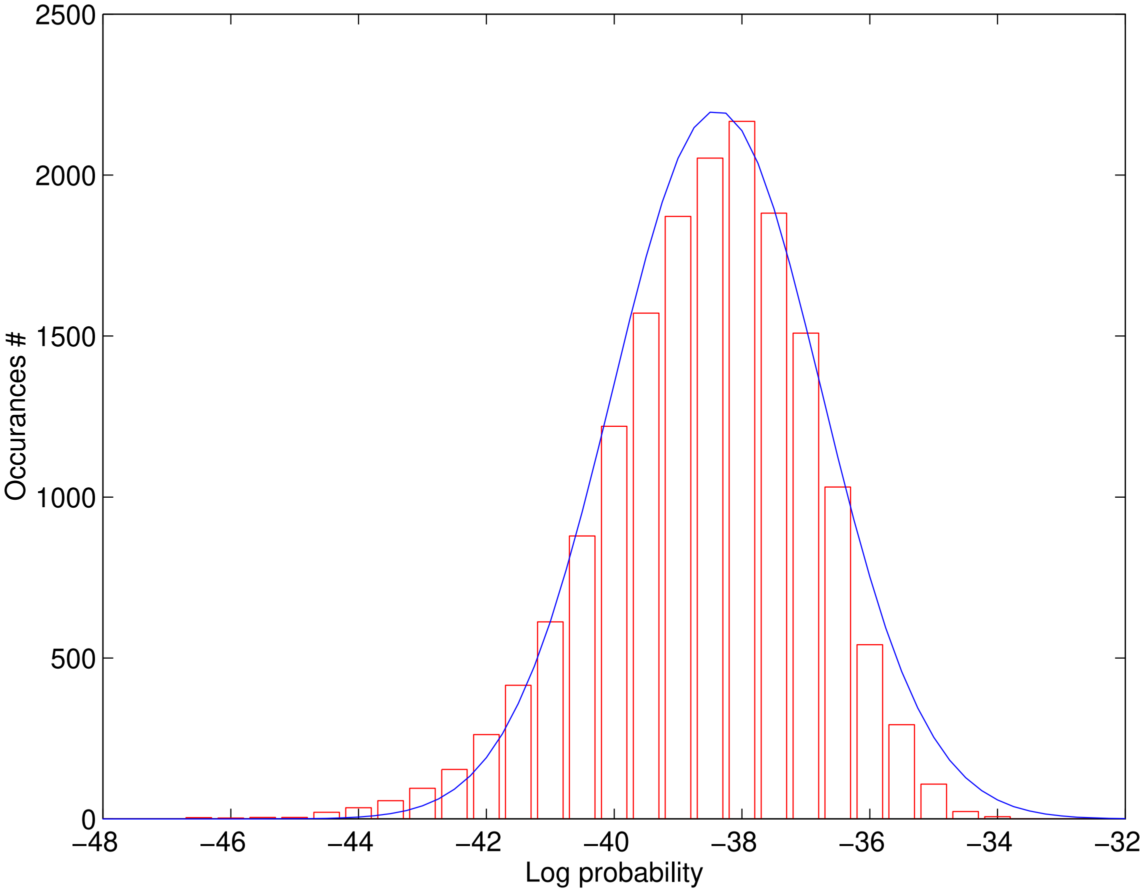

Figure 26: Histogram of the model log-likelihood in a single type with the Gaussian approximation. The bars are the histogram on simulated data, and the solid line is drawn from calculations.

We now calculate the mean and variance of this normal distribution, or the mean and variance of $\log(p_{\underline{\lambda}}(\underline{n}))$ *in each type.*

The log-likelihood, $\log p_{\underline{\lambda}}(\underline{n})$, can be written as

$$\log p_{\underline{\lambda}}(\underline{n}) = \sum_{i=1}^{M} [-\lambda_i - \log(n_i!) + n_i \log \lambda_i] = -\Lambda - L(\underline{n}) + \sum_{i=1}^{M} n_i \log \lambda_i \ , \tag{36}$$

with the notation

$$L(\underline{n}) \stackrel{def}{=} \sum_{i=1}^{M} \log(n_i!) \ . \tag{37}$$

Notice that in each type class, the counts $\underline{n}$ are fixed and the only randomness is in the permutation of the bins, i.e., a permutation $\pi(i)$ between the index of n_i and λ_i. Since all these permutations are equally likely in each type (which is also the assumption about the adversary background) the type mean, $\mu_{T(\underline{n})}$, is readily evaluated as follows,

$$\mu_{T(\underline{n})} = \langle \log p_{\underline{\lambda}}(\underline{n}) \rangle_{T(\underline{n})} = -\Lambda - L(\underline{n}) + \langle \sum_{i=1}^{M} n_{\pi(i)} \log(\lambda_i) \rangle_{\pi \in T(\underline{n})} \ . \tag{38}$$

Since the last term is averaged over all the permutations of the sequence $\{n_i\}$, each count n_i is multiplied by the all the rates $\log\lambda_i$ the same number of times. Thus,

$$\mu_{T(\underline{n})} = -\Lambda - L(\underline{n}) + \langle\underline{n}\rangle\sum_{i=1}^{M}\log\lambda_i = -\Lambda - L(\underline{n}) + \frac{N}{M}\sum_{i=1}^{M}\log\lambda_i\ . \tag{39}$$

Namely, to compute the mean of the normal distribution of the model log-likelihood in a specific type one should compute the sum of all rates $-\Lambda$ (which is fixed for the model), the sum of $\log(n_i!)$ (fixed for the type), and the sum of the log rates of the bins $l(\underline{\lambda}) \stackrel{def}{=} \sum_{i=1}^{M}\log(\lambda_i)$ (fixed for the model) multiplied by the mean number of spikes in the bins.

The evaluation of the variance is somewhat more tricky. We first compute the mean-square:

$$\begin{aligned}\mu^2_{T(\underline{n})} &= \langle\log p(\underline{n})\rangle^2 \\ = \Lambda^2 &+ L^2(\underline{n}) + \left(\frac{N}{M}\right)^2 l^2(\lambda) + 2\Lambda L(\underline{n}) - 2\Lambda\frac{N}{M}l(\lambda) - 2L(\underline{n})\frac{N}{M}l(\lambda)\ ,\end{aligned} \tag{40}$$

then the average of the square of $\log p_{\underline{\lambda}}(\underline{n})$ over the type permutations,

$$\begin{aligned}\langle\log^2(p(\underline{n}))\rangle_{T(\underline{n})} = \langle(-\Lambda - L(\underline{n}) + \frac{N}{M}l(\lambda))^2\rangle = \\ = \langle\Lambda^2 + L^2(\underline{n}) + \left(\frac{N}{M}\right)^2 l^2(\lambda) + 2\Lambda L(\underline{n}) - 2\Lambda\frac{N}{M}l(\lambda) - 2L(\underline{n})\frac{N}{M}l(\lambda)\rangle_{T(\underline{n})}.\end{aligned} \tag{41}$$

The only terms that contribute to the variance are those that contain both $\{n_i\}$ and $\{\lambda_i\}$ and we are left with

$$\mathbf{Var}(\log p_{\underline{\lambda}}(\underline{n})) = \langle\sum_{i,j} n_i n_j \log\lambda_i\log\lambda_j\rangle_{T(\underline{n})} - \left(\frac{N}{M}\right)^2 l^2(\lambda). \tag{42}$$

The first sum can be divided into two parts - the first for $i = j$ and the second for $i \neq j$. Since both i and j undergo the same permutation,

$$\begin{aligned}\langle\sum_{i,j} n_{\pi(i)}n_{\pi(j)}\log\lambda_i\log\lambda_j\rangle_{\pi\in T(\underline{n})} \\ = \langle\sum_i n^2_{\pi(i)}\log^2\lambda_i\rangle_{\pi\in T(\underline{n})} + \langle\sum_{i\neq j} n_{\pi(i)}n_{\pi(j)}\log\lambda_i\log\lambda_j\rangle_{\pi\in T(\underline{n})}\ .\end{aligned} \tag{43}$$

Averaging, as before, on all the permutations in the type this term can be written as

$$\frac{\sum_{i=1}^{M} n_i^2}{M}\log^2\lambda_i + \frac{\sum_{i\neq j} n_i n_j}{M^2 - M}\sum_{i\neq j}\log\lambda_i\log\lambda_j\ , \tag{44}$$

or as

$$\frac{1}{M^2 - M}\left((\sum_i n_i)^2 - \sum_i n_i^2\right)\left(l^2(\lambda) - \sum_i \log^2\lambda_i\right) \ . \tag{45}$$

Finally, the type log-likelihood variance can be written simply as:

$$\sigma^2_{T(\underline{n})} = \frac{1}{M-1}\left(\sum_i n_i^2 - \frac{N^2}{M}\right)\left(\sum_i \log^2(\lambda_i) - \frac{l(\lambda)^2}{M}\right) \ , \tag{46}$$

or, through the variances of the counts and log rates, as

$$\sigma^2_{T(\underline{n})} = \frac{M^2}{M-1}\mathbf{Var}(\{n_i\})\mathbf{Var}(\{\log\lambda_i\}) \ . \tag{47}$$

The last equation clearly shows that when either the $\{n_i\}$ or the $\{\lambda_i\}$ are uniform, the variance of type log-likelihood vanishes and the log-likelihood is sharply centered on the type mean.

Using the above expressions for the mean and variance of the type log-likelihood distribution we obtain the type break-down of $P_{\underline{\lambda}}$, Eq.(32), as:

$$\begin{aligned} P(\log p_{\underline{\lambda}}(\underline{n})) &= \sum_{T(\underline{n})} P_\lambda(T(\underline{n}))P(\log p_{\underline{\lambda}}(\underline{n})|T(\underline{n})) \\ &= \sum_{T(\underline{n})} e^{(\log|T(\underline{n})|+\mu_{T(\underline{n})})} e^{\left(-\frac{(\log p(\underline{n})-\mu_{T(\underline{n})})^2}{2\sigma^2_{T(\underline{n})}} - \frac{1}{2}\log(2\pi\sigma^2_{T(\underline{n})})\right)} \end{aligned} \tag{48}$$

An example of such break-down is given in figure 27. In this figure we show how the log-likelihood distribution is made up of a narrower distribution of type classes.

The last term is the log normal probability for which we have just computed the mean and the variance, and the first part is:

$$|T(\underline{n})| \exp(< \log(p(\underline{n})) >_{T(\underline{n})}), \tag{49}$$

or alternatively:

$$\exp(\log(|T(\underline{n})|)+ < \log(p(\underline{n})) >_{T(\underline{n})}). \tag{50}$$

For a given spike sample both the type and the model log-likelihood are known and our final probabilistic score would be:

$$S_{\underline{\lambda}}(\underline{n}) = \log(|T(\underline{n})|) + \mu_{T(\underline{n})} - \frac{(\log p(\underline{n}) - \mu_{T(\underline{n})})^2}{2\sigma^2_{T(\underline{n})}} - \frac{1}{2}\log(2\pi\sigma^2_{T(\underline{n})}) \ , \tag{51}$$

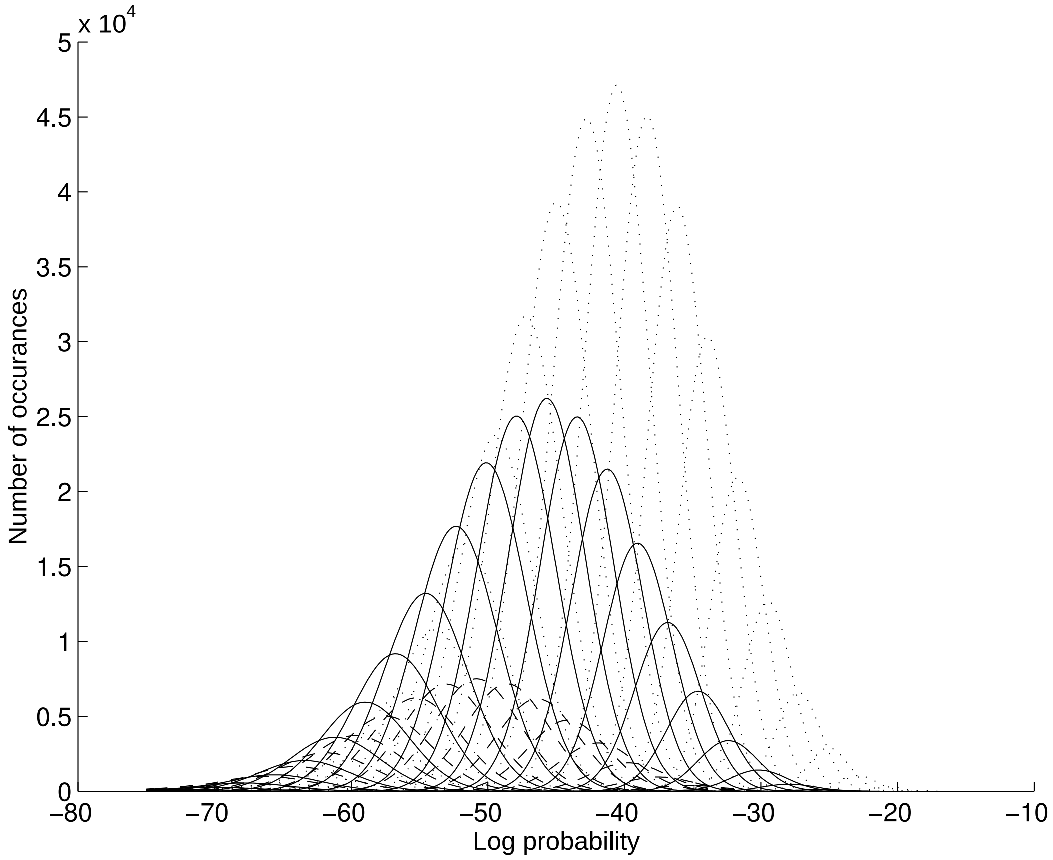

Figure 27: The decomposition of the model log likelihood into sample types. The different line styles represent different families of types. The dotted lines are types not including more than one count in each bin. The solid lines hold exactly one bin with two counts, and the dashed lines have 2 bins with 2 counts. Data taken from simulations of the activity of 4 cells.

Where both $\mu_{T(\underline{n})}$ and $\sigma^2_{T(\underline{n})}$ depend both on the type $T(\underline{n})$, and the model $\underline{\lambda}$.

12 Reward response of TANs

Neurobiological reward systems are of major importance in our understanding of learning and drug abuse mechanisms. The dopaminergic system is the most closely studied reward system, and the role of dopaminergic transmission in appetitive mechanisms has been demonstrated in many behavioral, pharmacological and physiological studies [82]. It has been suggested that dopamine neurons are feature detectors for the discrepancy between environmental events and the animal's predictions [84]. The spontaneous firing rate of dopaminergic neurons remains unchanged when no unpredicted events occur, increases when an event is better than predicted, and decreases when an event is less appetitive than predicted.

The striatal neurons may be classified according to their spiking activity in behaving animals [34, 50, 58] Tonically active neurons (TANs) have a broad action potential, a spontaneous firing rate of 3-15 Hz and, after training, show strong and robust responses to cues predicting future rewards. The response pattern is characterized by a reduction in firing rate (pause), often flanked by brief elevation of the firing rate [16, 17, 78]. Recent studies indicate that TANs are the cholinergic interneurons of the striatum [57, 97, 15, 25]. While cholinergic interneurons represent only 1-5% of the total population of striatal neurons [98], anatomical studies [29, 47] and clinical studies [19, 90] strongly suggest that they play a major role in the functions of the basal ganglia. The first aim of this study is to find whether there is a continuous reward-related activity (henceforth referred to as expectation signal) of TANs in the normal monkey. As with midbrain dopaminergic neurons this background activity might indicate that the environment is as good as predicted. Moreover, we tested whether the rate of the expectation signals increases or decreases when the environment is better or worse than predicted respectively.

The hallmark of Parkinson disease is extensive death of midbrain dopaminergic neurons, leading to substantial reduction in the dopaminergic transmission to the striatum. The reward response of TANs is significantly reduced in the MPTP primate model of Parkinsonism ([14, 78] or following local application of dopamine antagonists [96], without significant change in their tonic (spontaneous) firing rate. The second aim of this study is to differentiate between the two possible mechanisms for the reduction of the phasic response. The first mechanism is an overall reduction in the release of a neuronal expectation signal. In the second, neuronal expectation signal is still emitted but is no longer time locked to the reward event. We therefore searched for the expectation signal of the TANs in the striatum of monkeys treated with the dopaminergic neurotoxin - MPTP.

13 Building the reward template

In this analysis we used all recording sessions in which 3 or more TANS were recorded simultaneously. The array of four (or three) post-stimulus time histograms (PSTH) aligned to the reward cue was adopted as a template. An example of such a rate template computed for four cells is given in figure 28A. We then searched the data for matches to this rate template. The search was conducted by using a sliding window of 600 ms which was sliding in 5 ms steps. The data was thereby transformed into a sequence of overlapping frames. We computed the similarity probability of each such frame to the rate template. The result of the search was therefore a sequence of probabilities, which describe the probability of the neural activity in each frame to be derived from rate modulations as described by the template.

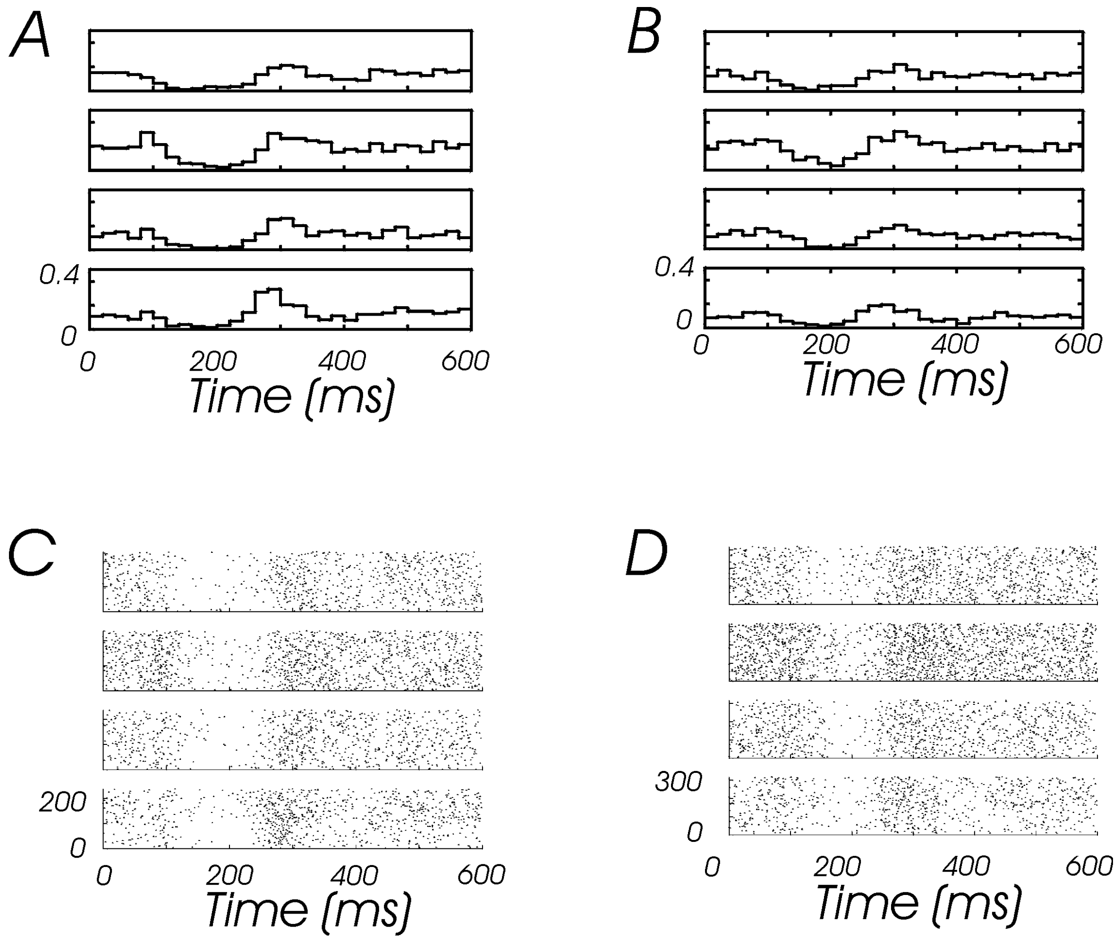

Figure 28: Peri stimulus histograms and raster displays of the template and the spotted events A. An example of a template used in the search for reward response. This template was calculated from the peri stimulus histograms of four simultaneously recorded TANs (HN06a). Reward was hinted at zero lag time (reward cue). The bin size used was 20 ms (238 trials were used to compute the histogram). The Y-axis values are counts/bin per trial. B: An example of peri-stimulus histogram computed from the spotted events of the previously defined template. The spotted events used to construct the histogram do not include segments from the vicinity of the reward cue. 400 spotted events were used. The bin size and Y-axis as in A. C. A raster display of the reward response of four units shown in A. Y-axis is trial number. D. A raster display of the spotted events (excluding those shown in C). Y-axis values are as in C.

The spotting of reward rate template was performed as devised in section 11. The outcome of the spotting procedure supplies us with a sequence of probabilities for the consecutive data frames described above. In order to select the frames with the greatest similarity to the reward response, we set a threshold that accepted only frames whose probability was above the threshold. In our analysis we used a threshold that would leave us with the top 0.1% of the frames, e.g. from a typical recording session lasting 3600 seconds and therefore holding 720,000 frames, we were left with the most likely 720 frames. In order to ensure that the

Table 9: The percentage of time-locked expectation signals out of the total number of expectation signals is weakly dependent on the threshold of detection. The fraction of expectation signals that are time locked to the reward out of the total number of detections is presented for each recording session for a range of thresholds. The average of all recording sessions is presented in the right column. As expected, the fraction of time-locked detections increased with higher threshold (e.g., selection of a smaller fraction of the events). However the changes are moderate. Note that with very high thresholds ($< 0.05\%$), the number of detections is very small, and therefore the results are noisier than in the lower range

Session	hn06a	hn06b	hn07	in05	in13	Average
# Cells	4	3	3	4	3	
Thresh (%)						
1.0 %	3.4	4.1	4.8	12.4	7.9	6.5
0.5 %	3.5	5.5	7.3	15.9	7.7	8.0
0.1 %	5.8	7.7	13.0	23.1	11.0	12.1
0.05 %	7.0	9.3	17.0	35.9	7.0	15.2
0.01 %	7.7	13.6	20.0	33.3	15.0	17.9

results are not due to this arbitrary threshold, we tested a number of different thresholds, and compared their physiological findings (e.g. fraction of time-locked events) see Table 9.

14 Spotting the reward template

14.1 The expectation signal of TANs is not time-locked to the reward cue

For each of the recording sessions we calculated a 600 ms multi-rate template (figure 28A) and searched for matches of the parallel activity of the recorded TANs to this template. Matches were sought in frames lasting 600 ms and shifted over data in steps of 5 ms. Only the best 0.1% of frames were selected (see Methods). This search provided us with frames of spike trains, to be referred to henceforth as the spotted frames. We excluded from this set the subset of frames found in the vicinity of the reward cue and examined the resemblance of the spotted frames to the original reward signature. As can be expected, in all the five sessions examined, the resulting PSTH was very similar to the original reward rate template. An example of this resemblance can be seen by comparing figure 28B (the reconstructed multi-PSTH) to figure 28A (the original rate template). This example shows both the PSTH (figure 28A,B) and the raster display (figure 28C,D) of the neuronal responses to the reward cue (figure 28A,C) and the spotted frames (figure 28B,D). The raster display of the spotted frames shows that the complex (on-off-on) response appeared in every spotted frame. We can therefore, rule out the possibility that it was formed from excitation in some frames and suppression of firing in others. The probability distribution of the matching of the template among the spotted frames was very similar to the probability distribution of reward locked frames.

The reconstructed PSTH was composed of those spotted frames which did not appear in the vicinity of the reward cue, suggesting that co-variations (on-off-on) of TANs' discharge were not time locked to any of the recorded behavioral events. A major question is the percentage of time locked vs. non-time locked events, i.e. how many of the frames spotted were actually in the vicinity of the reward cue? The result of this test showed that, using a 0.1% threshold level, the percentage of time-locked spotted frames ranged from 6% to 23% of all spotted frames for the different sessions, with an average of 12%. The percentage anticipated by chance (taking into account the duration of the post reward-cue epoch out of the total recording duration) is equal to 0.028%[1]. We may conclude therefore, that the spotting mechanism significantly found the neuronal signal that it was seeking, i.e. the time-locked response to the reward cue.

Despite the significant number of the exception signals in the vicinity of the reward cue, the data show that most of the detections were not time-locked to the reward cue. Figure 29 shows an example of this phenomenon: the rate of the detection is maximal near the reward cue, and yet most of the detections are found in different time periods of the task. With a 0.1% threshold, 77%-94% of the expectation signal are not time locked to the reward cue, and are probably the result of a continuous emission of the expectation signal

The robustness of the percentage of time-locked events out of all events, to arbitrary setting of the threshold level, is critical for an objective estimate of the biological relevance of the spotting algorithm. When we varied the threshold in the range of 1% to 0.01%, the percentage of time locked events remained steady (see Table 9). For example, when we set the threshold at 0.01% level, the percentage of reward cue time locked frames was 18% on average, with a range of 8-33%. This may indicate only a mild jitter of the shape of spontaneously emitted signals compared with the time-locked responses that were used for the construction of the template.

[1]Due to the fact that bins were 20ms and the shifts 5ms, we filtered the results so as not to include successive bins in the spotting.

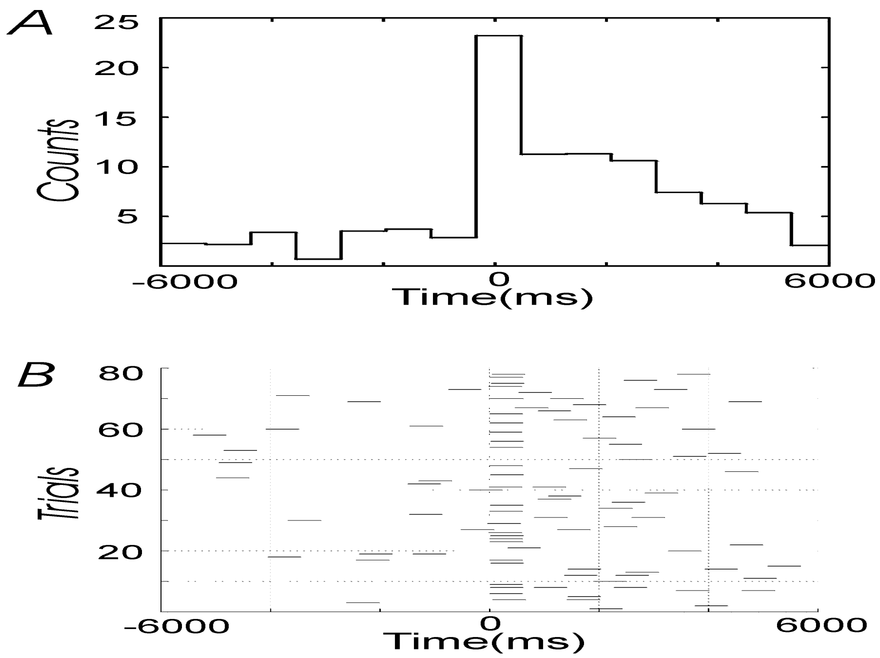

Figure 29: The rate of the neural expectation signal increased after the reward cue. A: Time histogram of the TANs1 expectation signal. Each bar represents the number of times the expectation signal occurred in a certain delay with respect to the reward cue. The bin size used was 800 ms. Y axis is counts/bin. B: Raster display of the occurrences of the expectation signal. Each short horizontal line represents a single spotted expectation signal. The spotted signals are arranged in trials aligned with the reward cue at lag 0. Data was taken from IN05.

14.2 The expectation activity is not found in surrogate spike trains

The spotted neuronal event was composed of elevation, depression and subsequent re-elevation of firing rate (which occurs simultaneously in all the TANS). We call this on-off-on firing pattern the reward expectation activity. Although the spotting method is based on sound statistical grounds, it is possible that the appearance of short bursts and/or pauses of activity (one of the most frequently observed neural phenomena in the brain) might be falsely detected as an indicator for the searched event. Having parallel recordings of several TANS, enabled us to conduct two validity tests. The first was based on shuffling the spike trains and the second on using other templates.

In the first validity test, we calculated the significance of the number of spotted frames. This was carried out by shifting the spike trains with respect to each other (figure 30A), where the amount of the shift was uniformly distributed between 0 and the length of the recording file (on the order of 100 seconds). We performed the same searching algorithm on the surrogate data. The absolute value of the threshold used in the search for the expectation signal in the shifted data was the same as in the original data; i.e. it was not tuned so as to filter out a certain proportion of frames out of the shifted data. The program performed 100 different shifts, and the distribution of the number of spotted frames in the shifted data was compared to the number of such frames in the original data. A comparison of this type is presented in figure 30B and reveals a large difference between the number of spotted frames in the original data and in the shifted data. We refer to this difference as the significance of the expectation signal detection and quantify it in units of standard deviations.

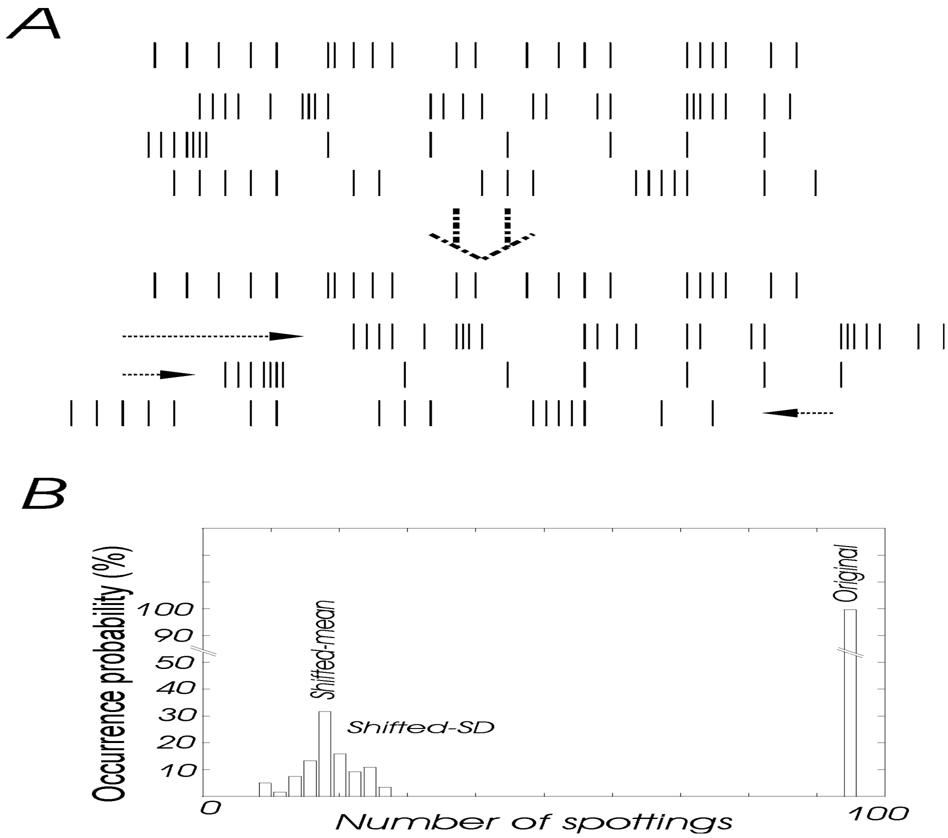

Figure 30: Computing the spotting significance by random shifting of the multiple spike trains. A. An example of the shifting procedure: The simulated activity of four simultaneously recorded neurons is shown. All the spikes of the second neuron (from top) are time shifted to the right by a large shift. The spikes of the third neuron are shifted by a small shift to the right, and the spikes of the fourth neuron are shifted to the left by a different amount of time. The shifting was performed in a cyclical manners as not to leave empty gaps as in the figure. B. We repeat the shifting with random selection of the duration of shifts for 100 repetitions, and calculate the number of spotting of the template in the shifted data. We than calculate the mean and the standard deviation of the number of spotting in the shifted examples. The significance of the spotting was defined as the difference between the number of detections in the original data and mean number of detections in the shifted data divided by the standard deviation of the shifted data. Differences greater than 3 standard deviations (SD) are considered as significant deviations from the null hypothesis. The results shown here were taken from monkey I in recording session IN06, where four cells where used.

The second validity test was based on the usae of null templates. We constructed several types of null templates and searched for matches for these templates in the original and shifted data. The significance level of the detection of null templates was computed in the same manner, as was the significance of the detection of the expectation signal. The null templates used were a flat template (in which all the bins of a single PSTH had exactly the same value equal to the average counts of that neuron). A second null template was estimated from the data segments of 2 seconds before the reward cue, i.e. from the pre-cue or delay intervals. Finally, the third null shuffled template was created by random permutations of the bins of the reward response template. Examples of these types of null templates are given in figure 31A,B and C respectively.

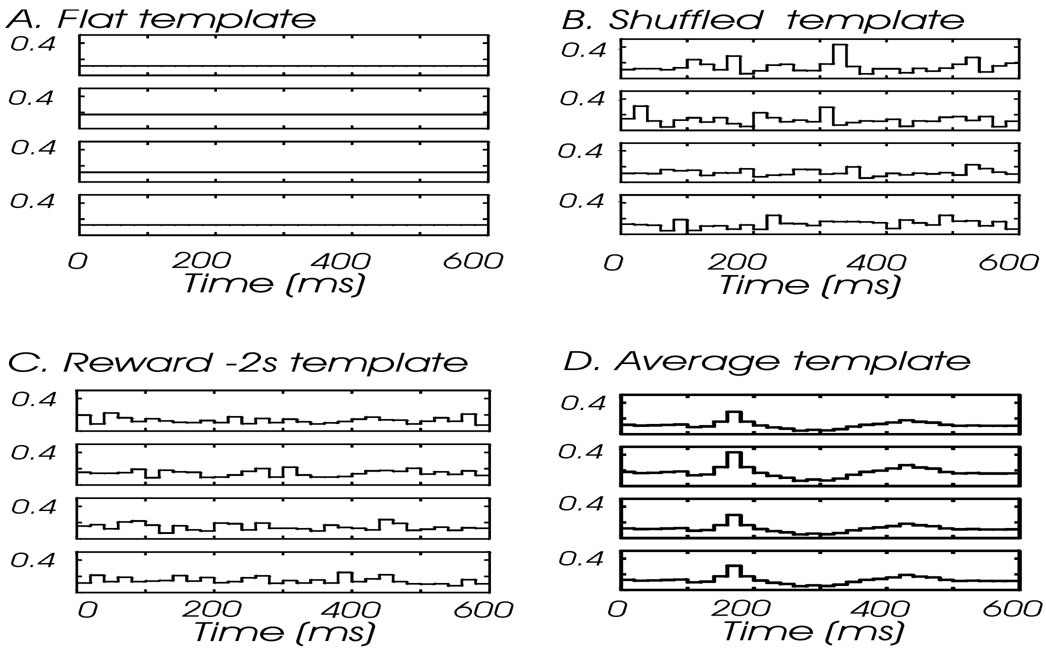

Figure 31: The Null hypothesis1 templates. Examples of the null template used in the validity tests of the detections of the expectation signals. A. The flat template was calculated from the average firing rate of four simultaneously recorded TANs (HN06a). B. The shuffled template was created by random shuffling of the original template (figure 28A) of that recording. C. The reward minus 2 seconds template calculated from the peri stimulus histograms of four simultaneously recorded TANs (HN06a) aligned on the time point of 2 seconds before reward (130 trials were used to compute the histogram). D. The average (prototypical) template adjusted to HN06b by the average firing rate of each of the recorded neurons. In all cases the bin size used is 20 ms. The Y-axis values are counts/bin per trial.

The results of the last two validity tests are summarized in figure 32. We considered a detection of a template to be significant if it was of three or more standard deviations. In the first validity test, it was found that the expectation signal was detected in a highly significant manner in all the recording sessions. The significance levels observed were in the range of 9 to 18 standard deviations. On the other hand, none of the null templates was detected in a significant manner in any of the recording sessions. Another interesting finding shown in this figure is the dependency of the significance level upon the number of cells being used. In the two sessions where the number of simultaneously recorded cells was 4 (HN06a and IN05) the significance level was higher than in the three sessions where the number of cells was only 3.

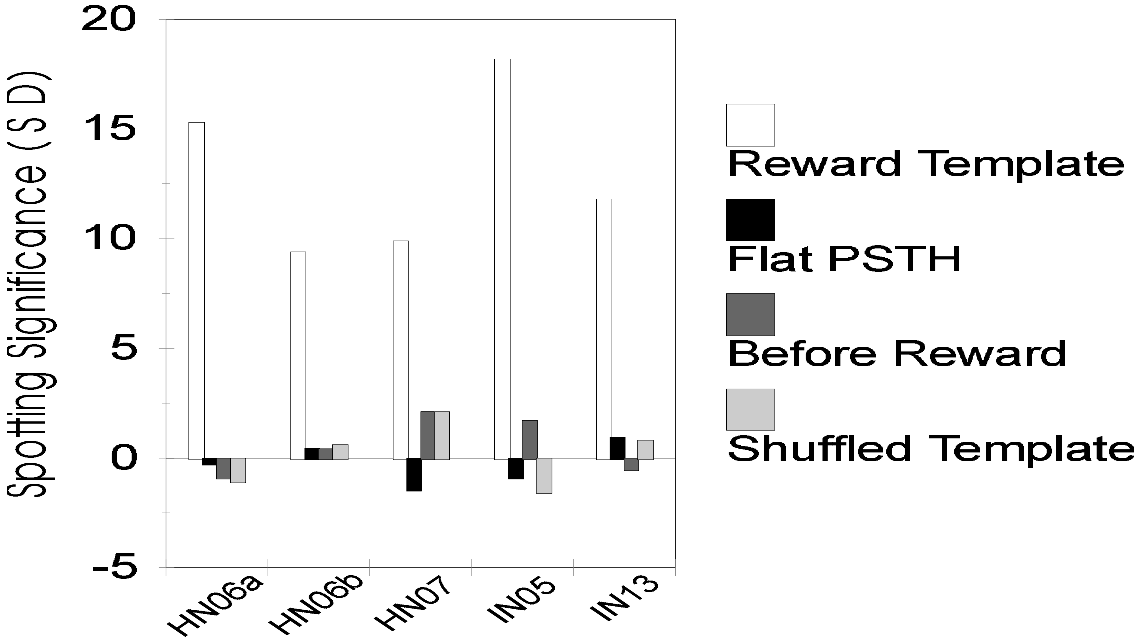

Figure 32: The neural expectation signal is significantly detected in the normal state. The X-axis labels give the animal's identity and the session number. We used several null templates to verify the significance of spotting the reward response (calculated separately for each of the recording sessions). The templates used were a flat PSTH, a PSTH computed from segments aligned 2 seconds before the reward cue, and a shuffled template calculated by random mixing of the bins of the reward response template.

14.3 The rate of the expectation signal increased after correct trials and decreased following errors

The rate of the spotted frames was not constant over all behavioral periods. As can be expected the rate of spotted frames reached a peak in the reward cue period. Nevertheless, we observed that the level of spotted frames remained high in the 2-4 seconds following the reward cue. A typical example of this situation is shown in figure 29A and B. In order to quantify these results, we counted the number of spotted frames found between the second and third second after the reward cue. These counts were compared to the expected count (average count) both in the original session, and in the shifted data. The results of this comparison are shown in the left part of table 10 and in figure 33. The number of the spotted frames following the reward cue was higher in all recording sessions both compared to the average count in the original session and to the average count in the shifted data.

In a similar manner, we computed the rate of expectation signal occurring following erroneous trials. In such trials, the animal did not finish the trial correctly and consequently was not rewarded. However, we could still count the number of spotted frames, which occurred after the animal was informed that the trial was erroneous. As for the correct

Table 10: The number of expectation signals in one-second epochs increases after correct trials and decreases after erroneous trials. The number of expectation signals in one-second epoch is shown in the Found column. The average number of expectation signals was computed based on the threshold, the total length of the recording session, and the total length of the epochs examined. The average number of expectation signals per one second calculated over all recorded periods is shown in the Average column, and the average number of expectation signals in one second after random shifting of the spike trains is shown in the Shifted column. The epoch for correct trials starts two seconds after the reward cue and for the erroneous trials two seconds after the switching-off of the ready light indicating to the monkey that the trial was aborted due to a behavioral error.

Session	Correct trials			Erroneous Trials		
	Found	Average	Shifted	Found	Average	Shifted
h1	21	13	3.3	0	2.5	0.7
h2	11	9	3.5	2	5	0.6
h3	9	7	3.2	3	7	2.6
i1	13	9	3	8	9	3.3
i2	20	18	7	7	15	4.2

trials, we calculated the number of expectation signals in between 2nd and 3rd seconds after the point in time when the animal was informed that the trial was aborted. The number of spotted frames in such erroneous trials is presented in the right part of Table 10 and figure 33. In all of the sessions, the counts in the erroneous trials were less than the average count in the original data and more than the average count in the shifted data.

Reverse correlation of the spotted frames with the other behavioral events in the task (e.g., the ready-on, visual cues, Go signal, release and touch of central and target keys and the licking movements) failed to reveal any time lock (the only significant correlation was found with the reward cue). This failure further suggests that the elevation of the rate of spotted frames from the background level is associated only with the reward cue.

14.4 A search with a prototypical template detected the expectation signal in the normal monkey, but not after MPTP treatment

The reward response of TANs is significantly reduced in the MPTP primate model of Parkinsonism, without significant change in their tonic (spontaneous) firing rate ([14, 78]. Therefore, in order to gain more insight into the behavior of TANs in the dopamine depleted state, we searched for a prototypical template in the normal data.

It was previously shown that the responses of different distant TANs to reward predicting cues are very similar [47]. We calculated a prototypical reward template for the response of a single TAN. This template was constructed by averaging the responses to the reward cue of

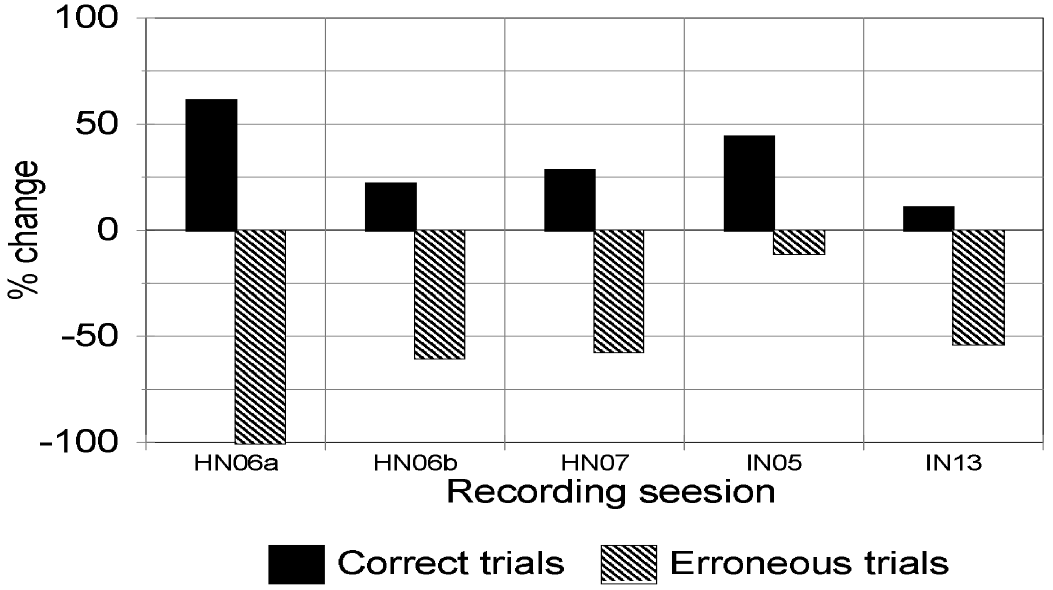

Figure 33: The rate of expectation signals increases after correct trials and decreases following behavioral errors. Filled bars represent the proportional increase in the rate of expectation signals in one-second epochs starting 2 seconds after reward cue compared to the average rate of the expectation signal. The hatched bars represent the proportional decrease of the rate of expectation signal 2 seconds after abortion of erroneous trials.

23 TANs with characteristic responses of monkey H. In this reconstruction of the prototypical response, we also used recording sessions in which the number of simultaneously recorded TANs was less than 3. The prototypical template was then normalized by each neuron's average firing rate, but no other adaptation with specific data was done. An example of the normalized prototypical template (according to the session firing rate) used in one of the sessions is shown in figure 31D. The search for the prototypical template was conducted in a similar manner to the search for the other templates, and provided us with the significance level of the detections in each session.

We computed the significance level of detections of the prototypical template both in the normal and in the Parkinsonian states. The monkeys were treated with systemic injections of the dopaminergic neuro-toxin MPTP. Following the treatment, the monkey developed severe parkinsonian symptoms, including akinesia and low frequency tremor. The monkeys were unable to perform the task, and recordings were carried with the monkey facing the same behavioral stimuli as in the normal state but rewarded with a drop of water at the end of all trials. The results of the search for the prototypical template are summarized in figure 34. This figure shows that in the normal state, the prototypical template was spotted significantly. Moreover, the prototypical template, which was based on data from monkey

H, was also effective in finding a significant level of spotted frames in Monkey I. However, when the same prototypical template was sought in the MPTP-treated animals the number of spotted frames was not significantly higher than in the shifted data.

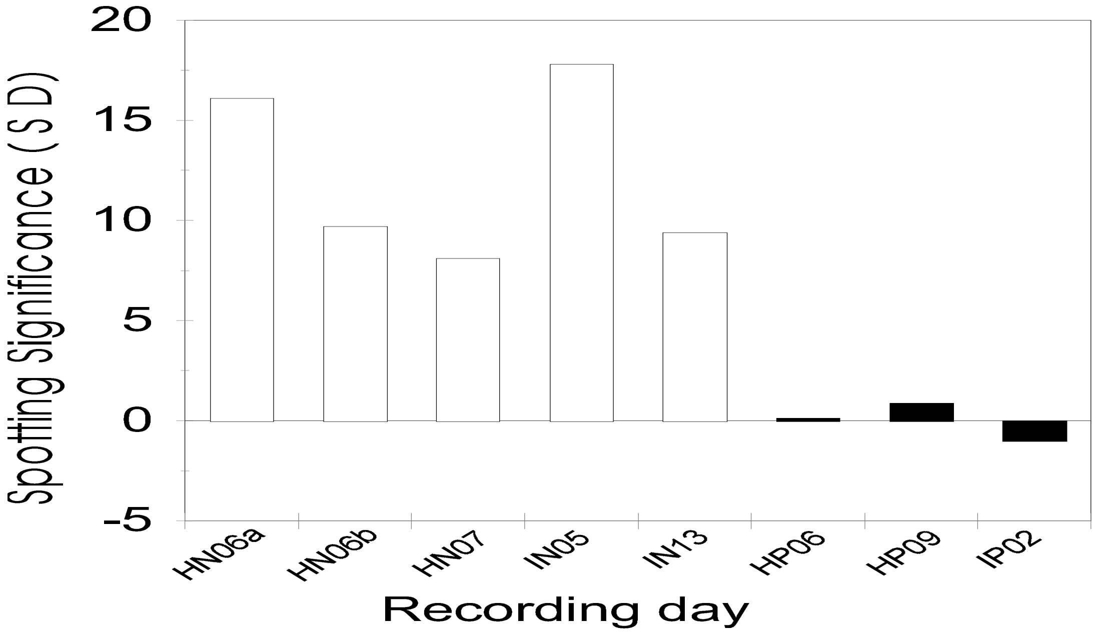

Figure 34: The neural expectation signal is significantly detected in the normal state , but not in the MPTP state. We searched for the expectation signal using the average template in the normal state and after MPTP. For each of the recording sessions the significance of spotting an average reward template was computed. This computation was performed in the normal state (empty bars) and in the MPTP treated (Parkinsonian) animals (filled bars). In both of them the detection threshold was set according to the original data and tested also on the shifted data.

14.5 The expectation signal is a network signal

Our template is a multi-neuronal template, suggesting that it detects coordinated rate changes in all the neurons. However, it is possible that only one or two neurons dominate the responses. To assess the contribution of each neuron, we carried out a modified significance test. In the previous tests we examined the number of spotted events in the real data compared with the number of spotted events in the shifted data. For this purpose, we shifted the entire spike train of each neuron by a random shift. In order to examine the role of single neurons, we tested what would happen if we shifted only part of the cells. We examined the modified significance test in the two recording sessions with four simultaneously recorded cells (HN06a and IN05).

As we shifted more and more spike trains, the significance level (the number of SD between the distribution of number of events in the shifted data and the original one) increased (figure

35). The average spotting significance was higher in the cases where we shifted two cells, than in the cases where only one cell was shifted, and the same happened when we compared shifting of three to shifting of two cells. Shifting three spike trains is equivalent to shifting all four because only the relative time among the spike trains counts. Shifting of any single cell did not produce the high level of significance that shifting of **all** of them produced. Moreover, in one case (cell 3 in HN06) a single cell is not crucial to the spotting, and when it is shifted alone we still did not get significant spotting. The combination of that cell and any other recorded cell still produced a lower SD than any other combination. In general, the contribution of any group of cells was proportional to the contribution of its' ingredients, i.e. a group made up of the two cells which contributed the most, was usually found to contribute more than all the other groups of pairs. We therefore conclude that the TANs expectation signal is a network signal. However, different cells contribute differently to the signal.

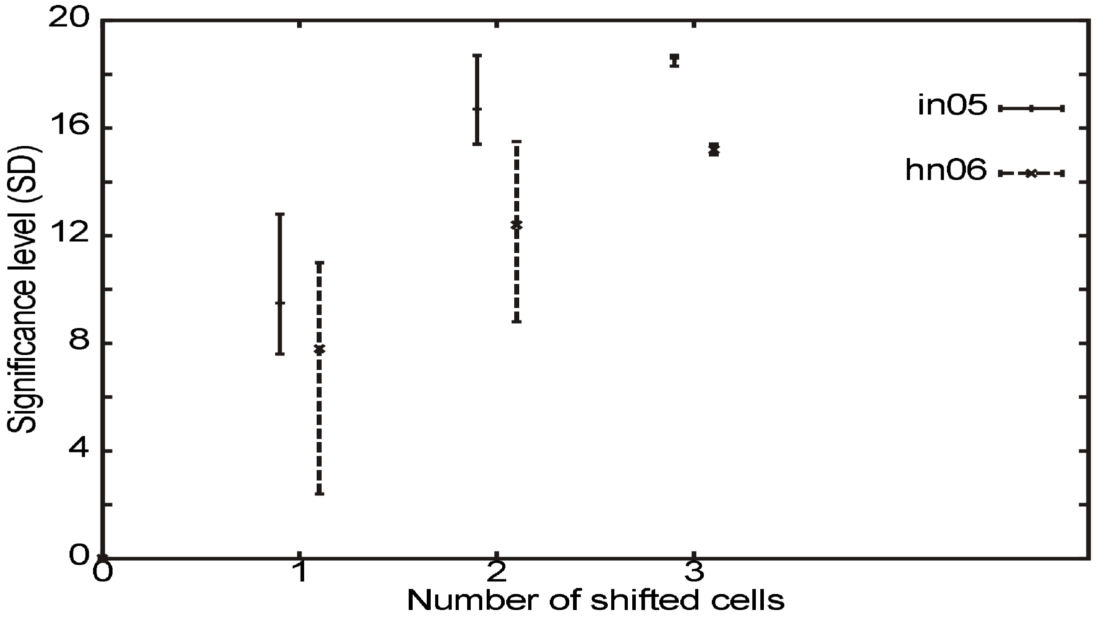

Figure 35: The significance detection of the reward signal is a cooperative action. The results are given for the two recording sessions in which there were 4 cells recorded simultaneously. The X-axis gives the number of cells being shifted ranging from 0 (the original data without any shift) to 3 cells (which is equal to shifting all the cells). The Y-axis gives the number of SD's that indicate the significance for each test. A large number of SD indicates that the shifted data produced much less spotting than the original data. The SD's range is presented in the form of a mean value (marked by a cross) and bars to the minimum value and maximum value indicating the range of results.

Part IV

Detection of Firing patters

15 Introduction

15.1 Goal of this work

The existence and role of fine temporal structure in neural activity is an open question in brain research. The goal of the method presented here is to provide a general framework to identify excessively repeating firing patterns. "A coding scheme that involves such temporal structures is rich and sufficiently flexible to facilitate a rapid organization of cortical neurons into functional groups. The results can be accounted for by the SynFire chain model, which suggests that cortical activity is mediated by synchronous activation of neural groups in a reverberatory mode" [75]. The analysis of firing patterns is another building block in treating the spike trains trial by trial, and exploitation of their high dimensionality.

15.2 Previous work

Reports exist regarding neurons that fire in coordinated fashion with very precise spike timing. One of the first reports regarding such activity concerns presenting a cat with visual stimulus[91]. Among other cases in which exact synchrony of two neurons was found are: motor cortex of behaving monkeys as reported by [12, 49], and visual cortex in a cat [68]. When research was aimed at events which are more complex (i.e. three spike events or more) many such instances were found: in cats' visual cortex[37, 61], in the inferotemporal cortex of behaving monkeys [95], auditory cortex of behaving monkeys[94], and primary motor cortex of behaving monkeys [81].

The notion of precisely timed spike trains has been addressed lately by [70]. In this work the amount of information given by precisely timed spike trains is compared to the information gained just from the rate function. The authors did not find anything was gained by using precise timed data. Furthermore they show that the same amount of precise patterns of spikes can be found in surrogate data. The results reported there are lacking in certain aspects:

- The definition of patterns as every triplet of cells that repeat at least twice in a single response, lack several of the basic ideas shown here.

- The fact that all "patterns" in the work were treated in the same way and no effort was made to check their significance level.
- The work was carried out on a single spike train, and therefore might miss some of the crucial ideas concerning precise timing.
- The area under investigation and the behavioral paradigm may not necessarily involve precise timing.

The method presented here is based on ideas previously published [75]. In this work, however, we tested all the statistical assumptions underlying the search mechanism. The testing was accompanied by many data simulations, often more than one type of simulation for testing the same issue. The suggested algorithm solves all the problems that were found in previous algorithms. The two drawbacks that remains in this version are that many simulations are needed in order to run it, and, on the way, some of the data is discarded (see below).

The question of finding a significant firing pattern is far from trivial. How can we distinguish between an excessive number of patterns and one that results merely from two or three bursts of neurons? What are the exact statistical assumptions underlying such a search and how can they be justified? In this report we try to answer these questions.

15.3 Definition of a firing pattern

In this work we follow the definition of a firing pattern given in [74]. A firing pattern is defined by its composition (the units that emitted the spikes) and the time delay between the spikes. For example, a spike emitted by unit S_1, then after t_1 ms, a spike emitted by unit S_2, and after t_2 ms (from the first event, the spike of S_1) a spike emitted by unit S_3, is a firing pattern. A schematic illustration is shown in figure 36.

Obviously, one should consider only those FPs which repeat *significantly* above the chance level. In order to be able to test the hypothesis whether the firing patterns found are significant we carried out the following steps:

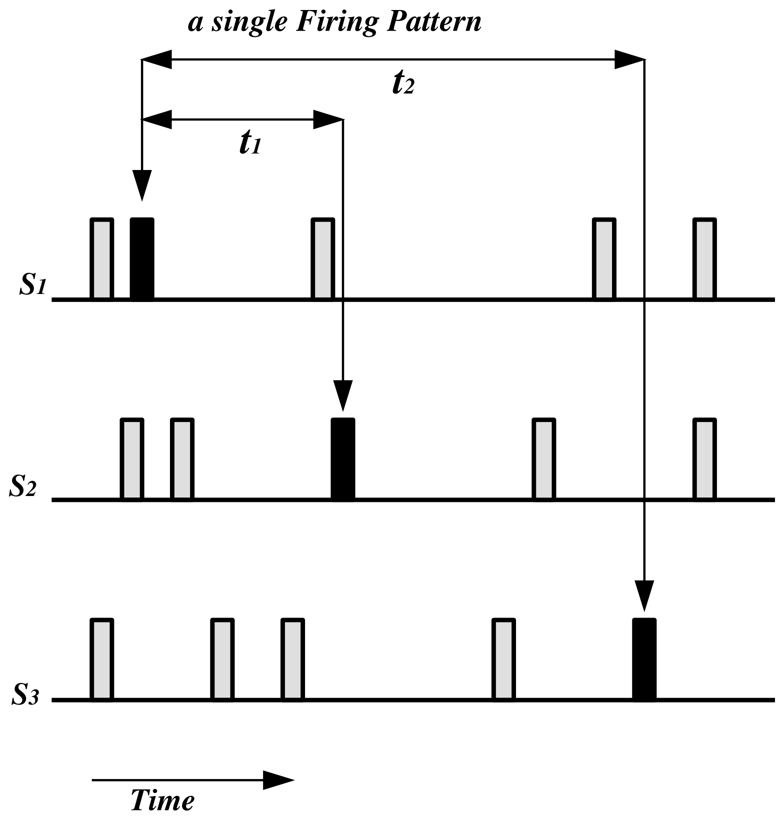

Figure 36: Illustration of a single FP. Three spike trains of three different units, which generate a firing pattern (FP). This firing pattern (FP) is defined by its unit composition (here S_1, S_2, and S_3), and the time interval between these units (here t_1 and t_2). Other spikes may appear between the spikes that are part of a single FP (black squares).

15.4 General scheme of identifying firing patterns

The first step in the method is to generate an estimator of the probability of seeing so many repetitions of the same firing sequence. Each firing pattern is therefore assigned a probability value. In the second step, we estimate the probability of observing so many small probability values. If the probability estimated in the second stage is below a predefined significance threshold, the small probability firing patterns are defined as significant.

To validate the acceptance we use surrogate data. We produce many random examples of the surrogate data and extract the same statistics for the surrogate data. We plot the distribution of values of these statistics. If the values for the real data are well below the small tail of the distributions (e.g. 0.05), then they can be accepted as representing precise firing sequences with the same confidence level (e.g. 0.05).

In the following sections we will show in greater detail both the process of computing the probability values, and the significance estimation based on them. Finally we will describe different methods used to produce surrogate data.

16 Computing the probability of each firing pattern

The process of computing the probability of each firing pattern includes the following sub-stages:

1. Computing the three-fold correlation [2] of the given triplet of cells. This correlation computation produces a two dimensional matrix which will henceforth be referred to as the *count matrix.*

2. Estimating the expected values for the three-fold correlation based *only* on the marginals of the count matrix. The value of this estimation would be henceforth referred to as the *expected matrix.*

3. Based on the previous two entities, the probability of each firing pattern is computed. The value of this estimation will henceforth be referred to as the *probability matrix.*

16.1 Computation of three-fold correlation

We start by constructing a three fold correlation among spike trains of three single units: S_1, S_2 and S_3. In this computation, the X-axis represents the delay between a spike of S_1 and a spike of S_2, and the Y-axis represents the delay between a spike of S_1 and a spike of S_3. The X-Y plan is parceled to N^2 bins, and at each bin we count how many times we observed a sequence $< S_1, S_2, S_3 >$ with the appropriate delays. The number of bins (N) should be specified in advance (we used a value of 100). The bin width represents the accuracy of the firing pattern we seek.

In our search, we used the value of 3 ms for the bin width. The outcome of this process is therefore a matrix $C_{i,j}$ whose size is N^2. Figure 37 illustrates such a three fold correlation matrix for a case of 3 tonically active neurons which were strongly correlated with each other. For cortical neurons the firing rates are lower and correlations are much weaker. The example of Figure 37 is useful for illustrating the point that selection of the bins with the highest count may be misleading, as a bin with 50 counts at (200, 50) may be highly significant, yet much below the non-significant bins near (0, 0) with 80 counts.

[2]Throughout this presentation the term three-fold correlation is assigned to a count matrix whose exact details are given in 16.1. It is not a full implementation of a correlation function as it is merely the starting point for that function. We do not reduce the average, nor do we divide by the measurement time or by the variance.

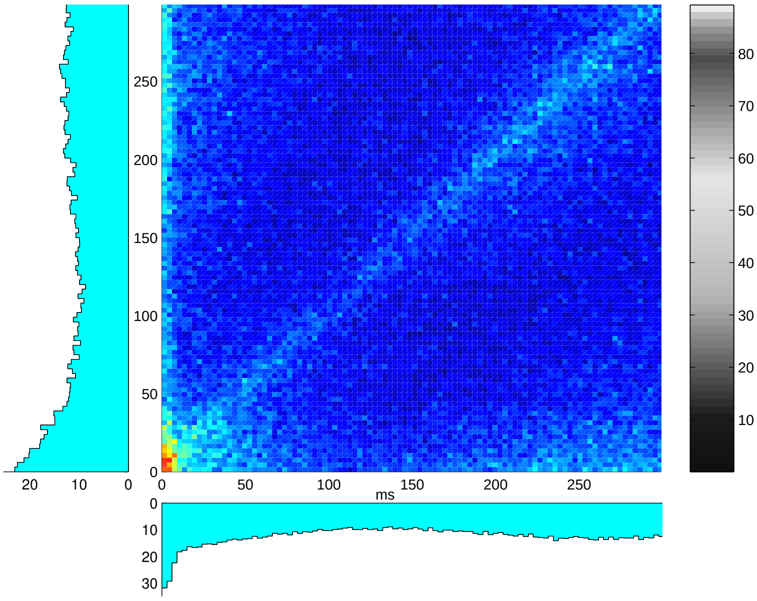

Figure 37: Three cell correlation. An example of correlation between 3 neurons with very strong correlations among all three. The gray-level codes the number of cases in which a triplet of spikes whose delay falls in each bin occurred. The data is for tonically active neurons in the striatum. Cortical neurons show much fewer correlations. Courtesy of H. Bergman and E. Raz.

We claim that the counting process is Poisonian. Note that this is not a claim about the spike-trains themselves, which may be very remote from Poisonian. This claim is based on the observation that at any time t, the probability of finding a particular firing sequence is:

$$P(counts) = \lambda_1(t)\lambda_2(t)\lambda_3(t)\delta t DD \tag{52}$$

where λ_1, λ_2, and λ_3 are the firing rates of the three single units, δt is the time resolution of data recording (1 ms in our data), D is the bin size for the correlation histogram (3 ms in our analysis). For three neurons firing at 3 spikes per second this probability will be $3e^{-9}$. Therefore, in this case, the Poisson approximation (which holds for $P(counts) \ll 1$) holds. We try and count at each bin many times: $T/\delta t$, where T is the total measurement time. If these two assertions are true, then the counting process is approximately Poisonian. The advantage of a Poisonian counting process is that even if the firing rates change drastically with time, the over all counts accumulated in every bin will still be Poisonian.

To test the validity of this claim we constructed a large number of simulated spike trains with various firing rates. The firing rates were estimated from real data, and spike trains were randomly chosen according to Poisson distribution with those variable rates. For each simulation, we computed the three fold correlation histogram. From the correlation matrices we were able to assess the Poisonian assumption directly, i.e. build the distribution of counts in one specific bin across many simulations and examine the match for Poisson distribution. In figure 38 two such cases are presented.

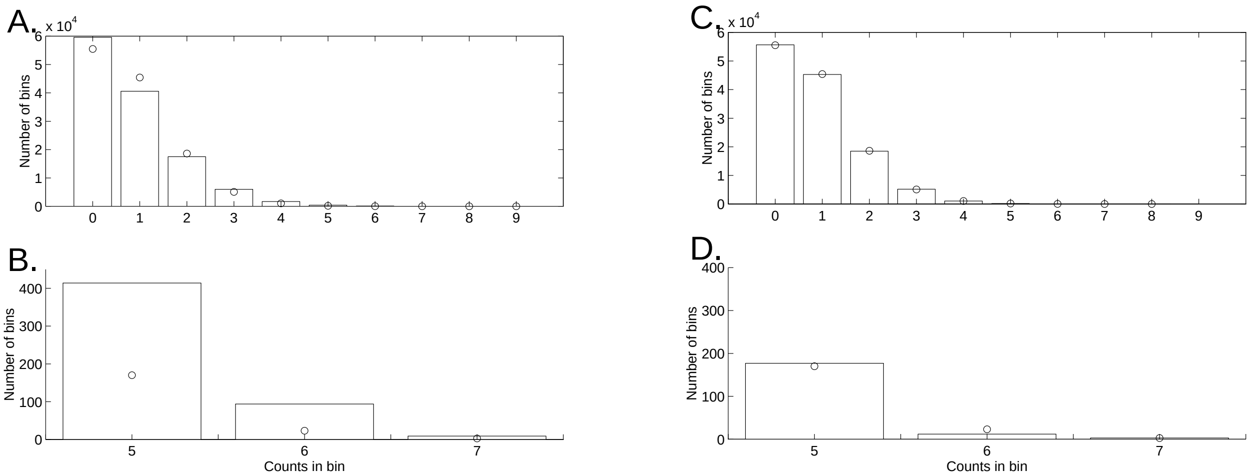

Figure 38: The deviation from Poisson distribution.A. The histograms of the distribution of values in count matrices of simulated data. The circles show the expected value according to Poisson distribution. B. A magnification of the histogram tail that shows the large deviation from Poisson. C. The same as A. just for Poisson randomly filled matrices. D. The same as B. just for Poisson randomly filled matrices. The large deviation found in A. and B. was also mirrored in the χ^2 and Kolmogorov-Smirnov tests that gave highly significant mismatch results.

In figure 38 parts A & B we see an example of a bin whose distribution cannot be approximated accurately using the Poisson distribution. This is due to the high rates found in the data for which $P(counts)$ was no longer negligible compared with 1. We mainly show the counts in the tail of the distribution - because there the deviation from Poisson is more pronounced. We also examined this match using χ^2 and **Kolmogorov-Smirnov** match tests, and according to these tests the deviation was highly significant. In C & D of the figure we see, by comparison, the same distribution in random Poisson variables.

Another method being used to check the Poisson assumption was based on the fact that for Poisonian counting processes, the variance should be equal to the mean. Figure 39 shows the ratio of the variance to mean for these simulations. The top two curves show that as the firing rate increases, the ratio departs more and more from one. Thus when the product $\lambda_1(t)\lambda_2(t)\lambda_3(t)$ increased, the counting process was **not** Poisonian (although the simulated spike trains were).

However, the Poisson approximation can be restored if the spike trains are diluted, such that when two spikes come closer than Δt, the second one is ignored. A schematic example of such dilution is shown in figure 40. Such a dilution effectively leaves only the first spike in a

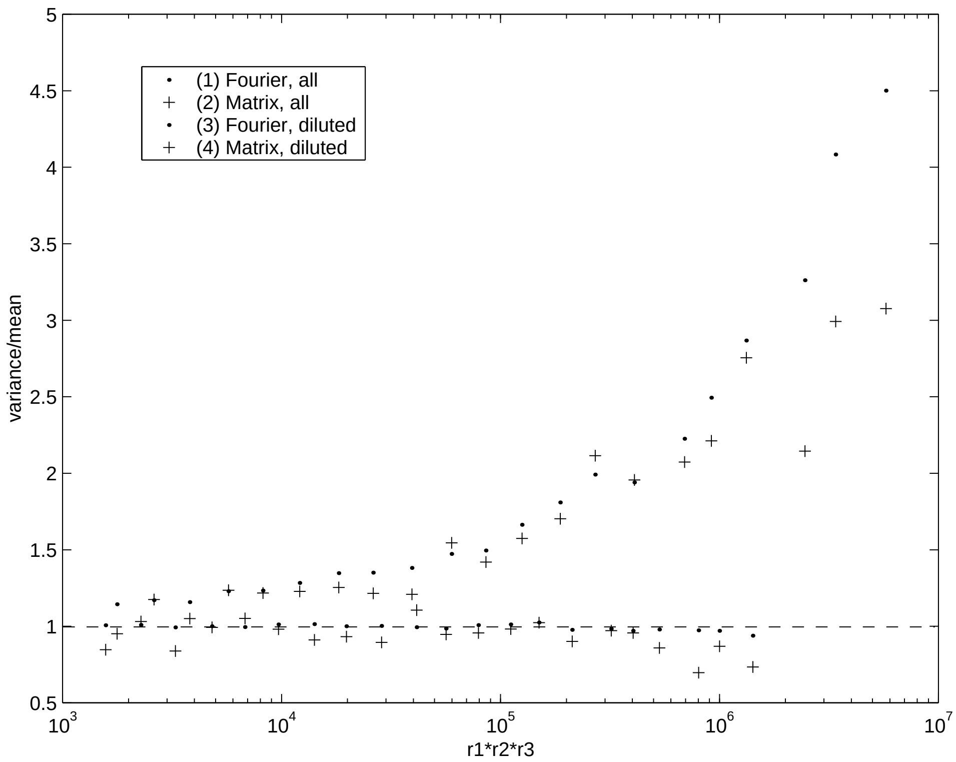

Figure 39: Variance of counts in 3 fold correlation. The ratio variance/mean for 21 simulations of Poisonian spike trains for which the correlation-histograms were computed with 3 ms per bin. The simulations differed in the their rates. The product of the 3 firing rates is shown on the abscissa. Two modes of computing variance were used: the first show the results of direct computation. Dots show estimation by computing the Fourier transform of the counting process and extrapolating its power to the value at 0 Hz. (We thank D. R. Brilinger for suggesting this method as a more proper one.) The top two traces show the results of using all the spikes in the train. Clearly the counting process is **not** Poisonian. The bottom traces show the results for diluted spike trains, in which, whenever two spikes came closer then 3 bins, the second one was deleted. The rates for these points are the rates of the diluted spike trains When the spike trains are diluted, the counting process behaves as if it is Poisonian. We also see that computing variance directly yields lower values then the Fourier transform method.

burst of spikes. The lower two curves of Figure 39 show that such dilution is indeed effective over a wide range of firing frequencies. The value of Δt which was found to be suitable (from simulations) was 10 ms. Therefore, the maximal rate permitted in these experiments cannot exceed 100 spikes per second.

16.2 Estimating the expected matrix

In order to calculate the probability of having a certain number of counts in a bin, we need to estimate the *expected* number of counts in the same bin. The term "expected" refers here to the number of counts that one should see if the spikes were only governed by a rate process and taking into account the pair-wise correlations.

The goal of the estimation is, therefore, to find the most likely matrix $E_{i,j}$ that has the same marginals and volume (total number of counts) as the original count matrix. In our

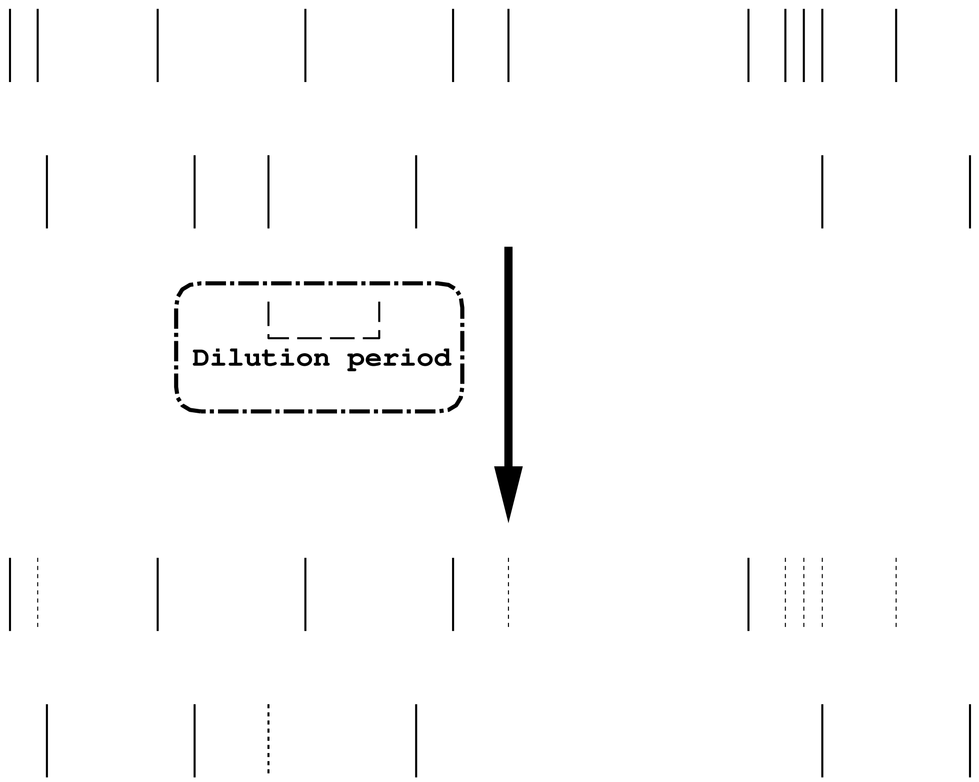

Figure 40: Illustration of dilution. The original spike trains of two cells are shown before dilution. The dilution window is given below. In the outcome spike trains, the spikes diluted out are marked by dotted lines. Pay attention to the fact that the time difference is always measured from the last spike, *even* if it was diluted. See for example that the last spike of the first cell is diluted although its difference from the last spike that remains is *more* than the dilution period.

case, marginals include marginals along constant Y, constant X, and constant Y-X values. These correspond to the pair-wise cross-correlations of S_1 with S_2, S_1 with S_3, and S_2 with S_3. The problem can be seen as solving an equations set where the number of variables N^2 (the number of elements in the matrix) is much larger than the number of equations $(4N)$. In this situation we have an infinite number of solutions, and our goal is to find the most probable of all. Our first choice was to build an expected histogram with minimal entropy [8], with respect to a set of constraints (the marginal distribution). However the, large number of constraints $4N$, rendered this a numerically unstable solution.

Instead, we used an iterative correction algorithm. This algorithm starts with an initial guess regarding the expected matrix, and, with each step, sequentially updates the difference between the true marginals and marginals of the current guess back into the guess. Such algorithms have been used in the past in CAT scans [55, 87]. The goal of such an algorithm is to reduce the error between the marginals of the estimated expected matrix and the true marginals to minimum. It can be shown that in each step of the algorithm the error function decreases monotonically. Nevertheless, such an algorithm supplies us only with a local minimum in the error function space. This local minimum depends on the value of the initial guess for the expected matrix. We used an initial guess that was a cross product of all three marginals and kept the volume of the original count matrix.

Unfortunately, this algorithm does not supply us with the *true* expected matrix. This is due to the small amount of data we have for estimation purpose and the inevitable over-fitting of the data. A more detailed description of this phenomenon will be given in section 17. In order to overcome this difficulty, and to further validate of our technique, we used several methods for producing surrogate data. These data then underwent the same computation process as the real data, and our results are based mainly on comparison between the two. The different methods used to create surrogate data are described in section 18.

16.3 Probability estimation

At this point of the calculation we have the three fold correlation matrix and the "expected" matrix. We wish to estimate the probability in each bin of getting this many counts if the expected value is given by the matching bin of the expected matrix. As shown above, the distribution in each bin obeys a Poisson probability and we therefore assign each bin with the following probability:

$$P_{i,j} = \sum_{n=n_{i,j}}^{\infty} P_{\lambda_{i,j}}(n) = \sum_{n=n_{i,j}}^{\infty} \frac{e^{-\lambda_{i,j}} \lambda_{i,j}^{n}}{n!} \tag{53}$$

where $\lambda_{i,j}$ is the expected number of counts in a bin, and $n_{i,j}$ is the number of counts for that bin. Calculating the probability matrix in such a fashion assumes that the computation of $P_{i,j}$ relies only on the values of $C_{i,j}$ and $E_{i,j}$. We will address this assumption of independence later on when discussing the surrogate data.

In principle, when one computes this tail probability for samples drawn from the *correct distribution* one expects the values (the tail probabilities) to be uniformly distributed. This is due to the fact that ons samples uniformly from the true distribution. In such a case, for example, we would expect around half of the cases to have accumulative values of less than half. In later stages of the computation we would like to asses what is the probability of seeing so many bins with probabilities below some p (say 0.001). This will be possible if the probabilities for random data are uniformly distributed between 0 and 1 and independent. Nevertheless, we observed that this is not true for discrete value distributions such as the

Poisson distribution. This is illustrated in Figure 41. In part A we see the tail distributions for values drawn from a distribution with one mean value. It can be seen that the deviation from uniform is quite prevalent. In part B we see that this deviation is also observed when averaging over a range of λ values.

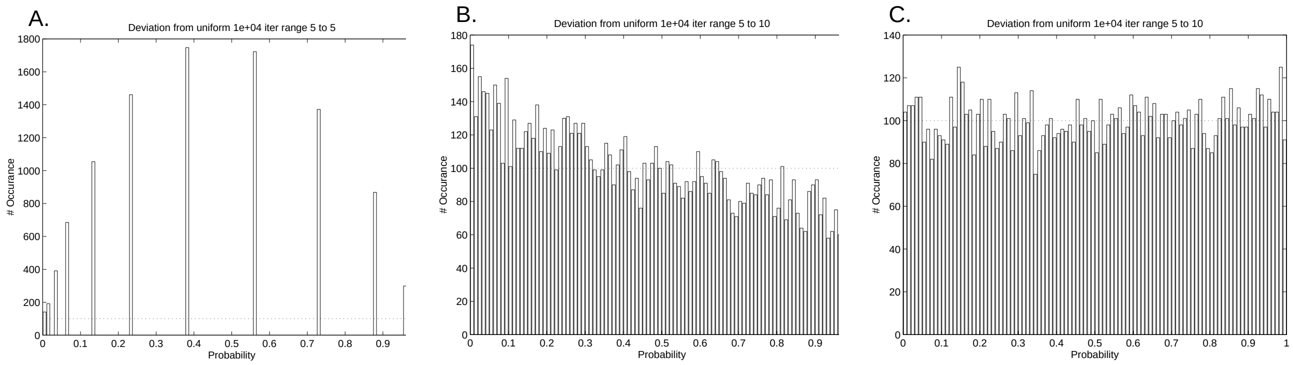

Figure 41: Computing the tail of discrete probabilities. 10,000 λ mean values were uniformly drawn from a given range. For each λ a Poisson distributed variable was selected. The histogram shows the tail distribution of the variables according to the matching means. A. The values of λ were all the same (5). B. The range from which the λ were drawn was 5 to 10. C. Adding the correction of Eq 54.

In order to solve the problem more accurately, one may either estimate the tail distribution, or change the tail computation in such a manner as to restore its uniformity. The first solution is both tedious and heavily relies on knowledge of the prior distribution of λ values in the expected matrix. We did not have access to this knowledge and therefore we chose the second solution. According to this method we replace the probability given in Eq. 53 by:

$$P_{i,j} = l \sum_{n=n_{i,j}}^{\infty} P_{\lambda_{i,j}}(n) + (1-l) \sum_{n=n_{i,j}+1}^{\infty} P_{\lambda_{i,j}}(n) \tag{54}$$

where l is a random number uniformly distributed between 0 and 1. Figure 41C shows that these new values are evenly distributed between 0 and 1. We note that these new values are not true probabilities any more, so we call them *P values* and relate only to the fact that they are expected to be evenly distributed between 0 and 1. We also note that for the same histogram, on different trials, we may get somewhat different P values. How we overcome this difficulty will be explained later.

17 Is it significant?

Having reached a smooth estimate of the expected counts, one can compute the P values for all the bins and look for an excess of bins with small P values. Given a threshold P_{thres}, one can compute the number of P values that are smaller, and compare that to what was expected. The expected number of bins with such probability is just $P_{th}N^2$. This is due to the uniform distribution discussed in the section 16.3.

Unfortunately, due to the over-fit problem discussed in section 16.2 the number of P values is *less* than $P_{th}N^2$ even for random data. The following example shows this: construct a histogram of expected values and a histogram for random (Poisonian) counts. Now recompute the expected histogram from this random count, according to the algorithm above, and use it to estimate the P values. Figure 42A,B shows that this, too, results in a shortage of small P values. The reason is that the reconstruction from the marginals of the very same data, over-fits the data. Thus, if by chance a certain bin has a large excess of counts, the marginals are likely to have a little higher bin at the corresponding delay, and their cross-product would predict a higher expected value at the very place where the bin had a high count. Such over-fitting can be largely reduce by smoothing the marginals. We found that by taking, for each point, the average of two values just before and just after the point yields a fair distribution of P values (Figure 42C). We note that this procedure will not work properly when there are sharp peaks in the pairwise cross correlations.

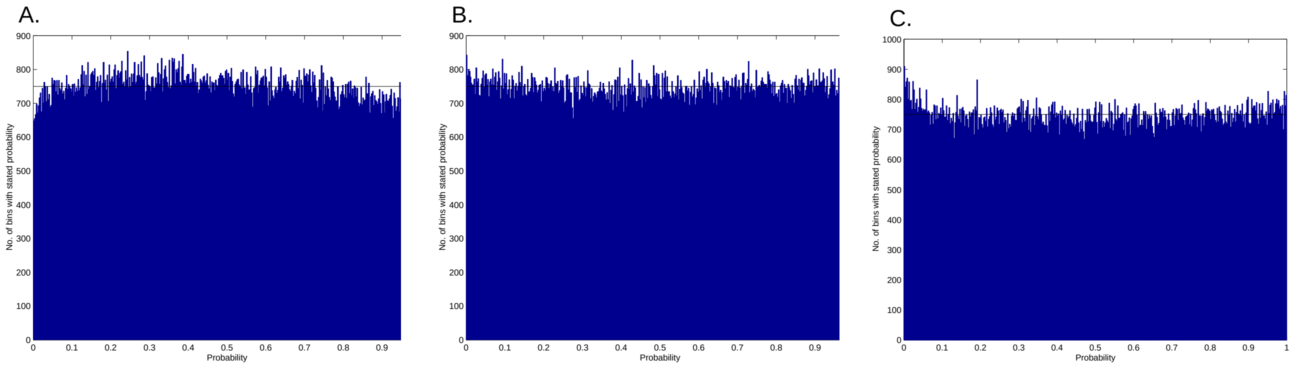

Figure 42: Histograms of P values. A. For 30 3-fold cross-correlations each of which computed for spike trains from 3 different units. A clear lack of small P values is seen. This becomes more severe for very small P values. B. For random data. An expected matrix for each of the 30 histograms was computed. Then random Poisonian counts were picked, around these expected values. These were treated as real counts. Then new marginals were computed and, based on them, new expected values were computed. These were used again to compute the P values. The density of these P values showed once more lack of small values, although less severe than for the real data. C. For the same data as in A, but the marginals were smoothed. Now an excess of small P's is seen. As approximately one third of these histograms showed excess of PFSs we believe that this excess represents the tendency of triplets of spikes to repeat excessively.

Till now we assumed that the value of P_{th} is fixed in advance. However, this may not be desirable due to the following reason. Two extreme cases can be envisaged: there is only one bin with extremely small P value (e.g. one bin with $P = 1e-8$ in a 10,000 bins histogram), or there might be many bins with bigger P (e.g 130 bins with $P \leq 0.01$ in the same histogram). In addition there could be all kinds of intermediate cases. In order to cover this spectrum of possibilities we try and look for several critical levels of P_{th}. For each level we estimate the probability of seeing so many bins with small P by chance.

For this end, we assume that the number of bins with a small P value is Poisson distributed. This is reasonable, since the probability of seeing a bin with small P is small, and we try to find it many times (N^2 in a histogram with N bins in each axis). For a critical level p_{th} the expected number of bins with P value below p_{th} is $p_{th}N^2$ (It should be remembered that although P values are not exactly probabilities, the distribution of their values is flat in the range 0 to 1.) We call the probability of having found m such bins P_{excess}. which is

$$P_{excess} = \sum_{m'=m}^{\infty} P(m'|p_{th}N^2). \tag{55}$$

If P_{excess} is smaller then 0.01 we reject the null hypothesis that the P values are distributed at random. If not, we move to the next p_{th}. We repeat this for 5 critical levels, so the probability of rejecting the null hypothesis is smaller than 0.05. This probability is an upper limit reached only when the computation for each of the p_{th} is independent of each other. In reality, the values are **not** independent and therefore the actual probability would be even smaller.

The last problem to solve is that P values have some random element in them as they were chosen as a random number (see section 16.3). To overcome this problem, we repeat the estimation of the P values four times and accept the count matrix as showing excess of bins with small P values only if it was significant on all 4 repetitions. The bins with excess of counts are only those that appeared on all four repetitions, and an estimator of the chance of seeing so many bins by chance is the geometric mean of the four P_{excess}.

18 Surrogate data

Due to the over-fit problem of the expected matrix, and for general assessment of the algorithm we need to compare our results with ones obtained on surrogate data. We have used several types of surrogate data which will be presented in detail in the following section.

18.1 Randomly shifting spikes within bins

The first method that can be used in order to produce surrogate data is slicing the data into equi-width bins and within each bin, randomly placing the spikes using uniform distribution. An example of this process is presented in figure 43A. The activity of one cell is shown together with the surrogate data produced from it.

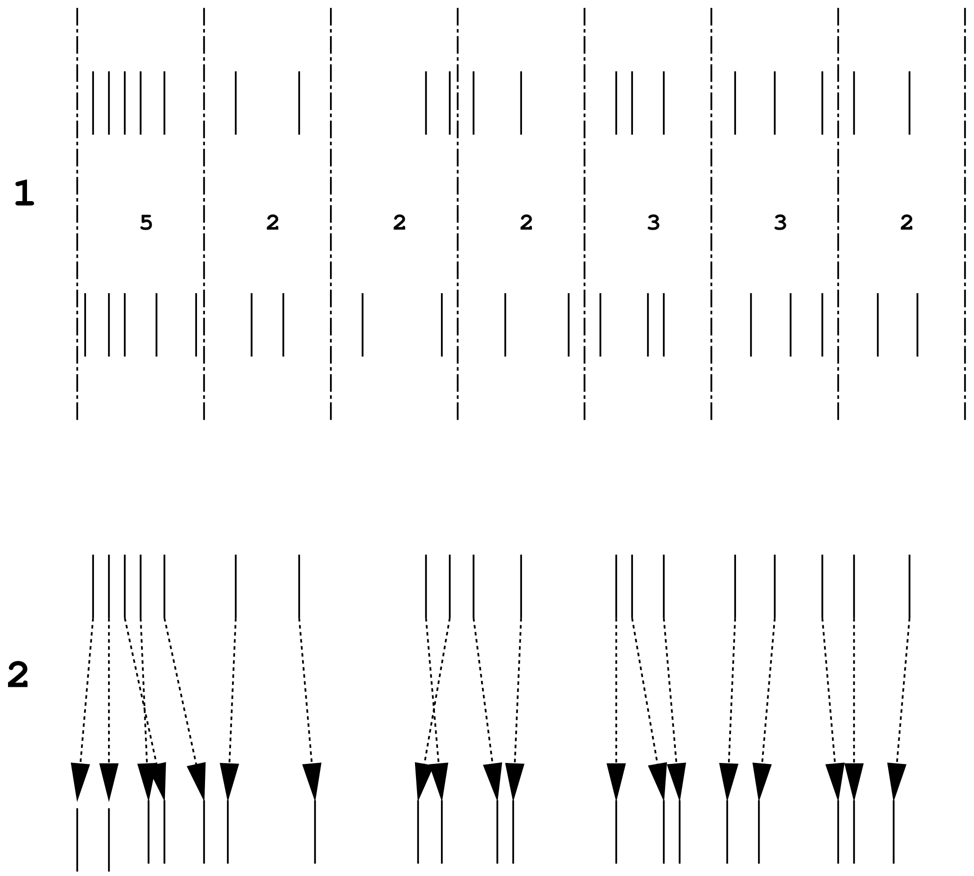

Figure 43: Two methods for producing surrogate data. In the upper part we see the slicing of the data into bins, with random shifting of the spikes within each bin. In the lower part each spike is being shifted by a small jitter.

The method can be applied in two ways: either shift the spikes of the original data and dilute it later, or shift only the diluted data. In the first option we may change the number of spikes being used later in the data - due to different dilution. In the second method, we can keep the number of spikes exactly the same - but we need to randomly choose spikes in such a manner that they are not too close to each other (and should therefore have been diluted before). This should also be taken into consideration when crossing the bins boundaries.

18.2 Jittering the data

A different method for producing surrogate data was presented in [35, 36]. According to this method each spike is shifted by a random jitter. As in the previous method, one can either jitter all the spike trains and then dilute them, or do it the other way around. An example for such a jitter is shown in figure 43B.

18.3 Jittering the triplets

The disadvantage of all the previous surrogate methods lies in the fact that our assumption of independence between bins does not hold. An example of this is given in figure 44. In this example the spike trains of 3 cells are given, and one can see that two counts are produced due to this activity in the count matrix. The two counts are produced based on the same first two spikes (of cell1 and cell2). Therefore, these counts are not truly independence of each other as we assume in the calculation done above.

It is important to note that the surrogate data suffers from the same problem, as it is trying to shift the spike trains and compute the count matrix in the same manner. We suggest using a surrogate data that skips the phase of jittering of the spike trains - but directly jitters the counts. This can be done in the following manner: the process of computing the count matrix involves a passage over the data and a search for occasions of the triplet of cells within the predefined range. We suggest that whenever such a count occurs, we also jitter the three spikes producing it many time (using the method described in the previous section). Each such jitter would be written for a different count matrix.

In this manner we can overcome the problem shown in figure 44C because the two counts shown there would be shifted in the surrogate data independently.

19 Current state of algorithms

The pattern detection method described above summarizes the effort mad in this sphere. The method obtained was carefully examined and each stage was analytically proven. The tests it passed were both theoretical, in the understanding of statistical problems, and by using many simulations.

The algorithm presented here shows many interesting statistical features which are highly

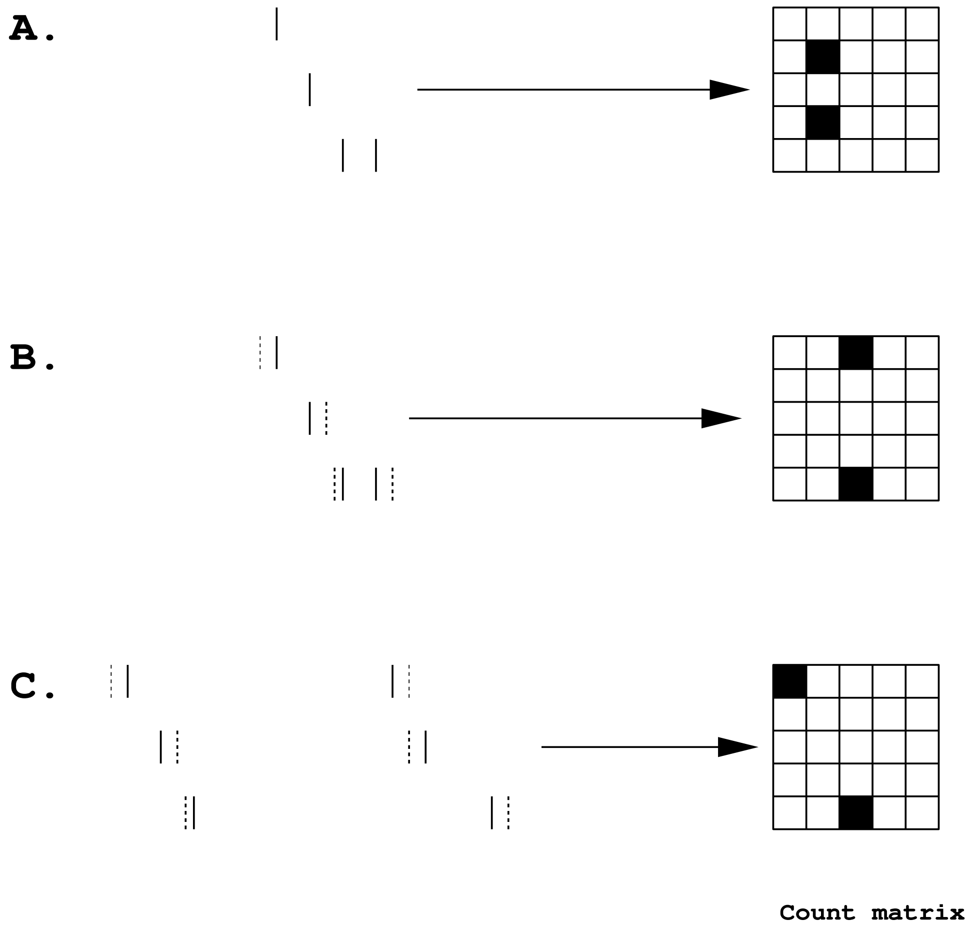

Figure 44: Surrogate data without correlation. Computing the count matrix for regular spike trains produces correlation between the counts (part A). This correlation is preserved in the regular jitter mechanism (part B), where the spikes after jitter are marked with dotted lines. It can be eliminated if the jitter is applied to each occurrence of three cells separately (part C). Two cases are than created and each of them is shifted independently.

important for the implementation of any such algorithm. It was therefore decided to include the algorithm, even though the results of using such an algorithm are yet to be presented.

Part V

Summary

20 Work summary

The summary of this work is given first for each of its components separately. In the last section, a global overview is presented.

20.1 Classifying the neural response

The most impressive result of this work is the finding that a single segment of simultaneously recorded spike trains can predict with good accuracy the behavioral task of an animal. This is a non-trivial result, taking into consideration the low firing rate of the cells, and the high variability from trial to trial. Building the statistical models describing the data, required the use of aligned data. Nevertheless, this result stands in contrast with many of the common analysis methods which in order to get reliable results align a large number of data segments, and average them. The impressive power of detection shown here may lead to new examination of this paradigm.

Based on these high detection figures, we have examined several detection methods under several conditions. These examinations showed that not only can the detection be achieved, it can also highlight some intrinsic features of the data. The comparison between detection methods (as summarized in table 4) shows that:

The detection methods that were based on parametric distribution function supplied us with the best detection rates. The parametric function used was based on the assumption that the spike trains are distributed according to non-homogeneous Poisson distribution. The methods that used non-parametric estimation of the probability functions had significantly worse results. The combination of these facts provides further support for the non-homogeneous Poisson statistical assumption regarding the activity of neurons [1].

The hidden Markov model outperformed the simplified non-homogeneous Poisson model. This possible advantage of the HMM over the PSTH are:

1. Dynamical time modeling – the ability to wrap the time scale for every example.

2. Building a high-dimensional mode and thus addressing the fact that several cells were recorded simultaneously.

The fact that the HMM was proven superior to the PSTH suggest by using these features, the description of neural data is more accurate.

The ability of the pair-wise cross-correlation methods to detect the behavioral mode is even more surprising than the regular detection scheme. One should take into consideration that in the correlation computation of a single segment, the results of correlating around 20 spikes with 20 other spikes (4 seconds of 5 spikes-per-second on the average), were put into a vector of 150 bins (2 seconds with bin size of 13 ms). The data is *highly* sparse.

The correlations ability to detect the behavioral mode may be the outcome of stimulus locked activity, or the outcome of invariant correlation between the cells. In order to check this possibility, we examined the connection between the PSTH detection and the pair-wise correlation. The results were not conclusive, but they suggest that in some cases, all the correlation detection abilities are the outcome of locked activity (and therefore are contributed for in the PSTH), and in some cases they evolve from true change of correlation between the different behavioral modes.

The suggestion that correlation may represent a different type of information about neuronal activity, is very interesting. It may also be used in order to gain a deeper understanding of the nature of neuronal processes [3].

The detection mechanism allows us to weigh the contribution of each cell separately, and compare it to the combination of all cells. The results show that in all the cortical delayed response paradigm recording sessions examined, the detection was a cooperative achievement and was not only based on one single cell. In the recordings done in the TANs' of the basal ganglia, the same phenomenon was observed. Nevertheless, in the same cortical area, when recording cells under different paradigms a different behavior was observed: the differentiation between tasks performed by all cells was as good as the differentiation of the performance performed of the best single cell.

This finding supports the use of simultaneous recording as a superior method for understanding the neural computation. It highlights it's importance and suggests that more cells would allow better detection, and more reliable prediction of the animal's actions. This result can be seen from yet another point of view. According to this view the HMM achieved better results because the exploitation of the multi-unit activity is an integral part of such a model.

The amount of data used here for modeling was quite limited. The number of examples for each different mode ranged from a few tens to several hundreds (a fact which reflects the state of the art ability in simultaneous recordings). This number does not permit the splitting of the data into separate training and testing groups, while still getting reliable estimation of the detection percentage. In order to overcome this limitation, the method of Jacks' knife was introduced. According to this method, the test group holds one example, and the training group is formed by eliminating this example from all the examples.

Throughout this work we have shown the importance of using this type of correction (and avoiding the insertion of the test examples into the train). We showed that the detection results change dramatically when this correction algorithm is not used. It may seem surprising that the results should be totally different even when the number of examples is relatively large (around 100 examples for each mode). This can be explained by the sparseness of this type of data, and its irregularity. These features construct a situation in which one example can have a large contribution to the model. This is true especially when the modeling techniques used are rich enough with parameters. This richness allows for an over-fitting of the model to the data that it is trying to describe. This over-fitting prevents the model from generalizing from the data.

Another trap that this work tried to avoid was the over-fitting of the modeling technique to the test set. Because of the limited number of examples, we did not hold a validation set and a training set, as might seem required. Instead we performed almost no optimization of our algorithms. Such optimization could have been carried out if the model parameters were chosen in order to maximize the detection on the test set. We tried each time to set the parameters of the technique in advance and allowed very little change to the initial results.

The single segment detection algorithms have proven to give reliable predictions of the behavioral task, as well as to reveal some interesting features of the neuronal process involved in the task. These persuasive results, together with previous works [42] shown on unsupervised modeling of the same data, give hope that a combined model can be suggested. In such a model the data could be modeled in an unsupervised manner, where the inherent differences between the modes could be used for self-learning algorithms. We believe that through careful combination of the two ends - the highly successful supervised detection, and the self-emerging features of the unsupervised modeling, a better model for the neuronal activity may be constructed. Such a model may give more insights regarding the nature of the neuronal computation.

20.2 Spotting via adversary background summary

In this work we present a novel method for pattern spotting in a harsh environment. Our goal was to spot a pattern within a single trial consisting of multi-unit recordings of a specific pattern of activity. The environment we aimed at consists of multi-cellular recording of spikes in the brain of behaving monkeys. Statistical modeling of such data is difficult due to the following characteristics:

- The rate of spike discharge per single unit is very low – around 5 to 10 spikes per second.
- The change of activity may be relatively high. A whole pattern of activity can be encapsulated within 600 ms, where this pattern consists of a rise in firing rate, a depression and another rise. One should remember, however, that due to the low firing rate, one may expect to see only 6 spikes within this pattern.
- The background activity is *not* a uniform firing rate – but rather supplies us with similar fluctuations in the activity, and, furthermore, the average firing rate is similar to the pattern searched.

Due to these difficulties, the common method in using this kind of data involves aligning several repetitions of the same behavioral event together and thus receiving more reliable statistics regarding the neural code. We showed in section 4 the advantages of using single trial modeling, and the new questions that may be answered using this mechanism. In this work we tried to confront the spotting neuronal response in a single trial.

We have shown the importance of selecting a reasonable background model for the spotting algorithm. When inadequate background models were used, the spotting did not supply us with reasonable results. For example, putting a threshold on the likelihood function which assumes a constant rate background, supplied us basically with the frames of data in which the number of spikes was the smallest.

The core idea of the method presented here, is that we are trying to spot the neuronal pattern in the adversary environment, in which all *shuffles* of a given pattern are equally probable. When implementing this break-down process we showed that the sample type and its log-likelihood serve as sufficient statistics. Since we are interested in patterns with a large number of bins, the total log-likelihood distribution in each sample type can be well approximated by a Gaussian, whose mean and variance may be *analytically* evaluated.

Based on this Gaussian approximation, we derived an exact expression for a score which does not rely on the asymptotic properties of the direct application of the method of types. While this break-down, shown in figure 27, clearly breaks the log-likelihood distribution into narrower pieces, the mean of each piece depend very weakly, (see Eq. (39)) on the model rate parameters. The type weight, however, as given by Eq. (33) is exponential in the type log-likelihood mean, thus much more sensitive to the models' rates. For these reasons the type conditional (Gaussian) distributions separate the model patterns from the background very well.

We also examined a related solution to this problem based on the method of types [48]. Direct application of this technique for detecting multi-neural spike patterns can be carried out by treating the empirical type of every bin in a template model (PSTH histogram) of the training data [54]. In this case, one simply estimates the probability that the training data and the test sample are taken from the same source, for every bin or combination of bins in the template model. As shown above, this method suffers from insufficient training data and therefore performed rather poorly on our data, although in principal this method is able to overcome the difficulties related to this kind of pattern spotting.

Our method was implemented on data from the TAN's (cholinergic inter-neurons of the striatum), and we tried to spot the neural pattern usually related to the outcome of an action. A thorough description of the data with the significance of the results is shown in section 14. We believe that these results could not have been obtained without such a method of spotting the reward pattern on the basis of a single trial.

There are several possible extensions to the work presented here. The most straightforward one is to extend the types used here into ones containing information on higher correlation of the data. In our application we work against an adversary environment that consists of all the shuffles of the given pattern in single bins. Second order correlation would have to be an adversary environment that also keeps the order within pairs of bins. In order to perform such spotting we need to recompute the mean and variance inside each such new type. In general, we can work with each order of correlation, and thus would be able to cope with even more difficult spotting cases. When we extend the correlation order, the background becomes increasingly similar to the original pattern under investigation.

In general, we believe that the idea of breaking down the likelihood into sub-parts could be found useful. This could allow us to work in environments in which we cannot assume the usual flat distribution of the background, and we can perform reliable spotting without building a specific model for it. In our case, the decomposition into types seems sufficient and produced the required results.

20.3 Spotting the reward response summary

In this report, we used the novel tool we developed especially for the detection of a specific multi-neuronal response in a noisy background. This detection was performed on the continuous spike trains without any need for aligning the data around some given stimuli. The main advantage of this tool lies in the modeling of the unknown noisy background. We assume an adversary background model, which imitates the response sought up to a first order statistics. The basic validity check of these results is therefore supplied by the sound method being used. Our spotting procedure is further validated by a variety of statistical tests, and by the difference in the results in the MPTP monkey and the normal monkey. The use of this method might change the way of interpreting the relations between neuronal responses and external stimuli in many other neurophysiological studies.

A previous study [15] that tested only the *pause* of a single TAN response, concluded that the pause is an active suppression of the TAN firing beyond the average inter spike interval. In-vitro studies of synaptic regulation of the discharge of the cholinergic inter-neurons of the striatum [25, 26] also suggested that the pause of TANs' firing is caused by dopamine mediated enhancement of the post-spike after hyper polarization. Our results emphasize the role of the entire complex of the neuronal response, including the flanking excitations, as the main neuronal message of these neurons. It is therefore very intriguing that in many cases the dopaminergic response to reward cues is flanked by periods of elevated firing.

The findings of loose time locking of the TAN expectation signal, suggest that striatal cholinergic and mid-brain dopaminergic systems [84] have similar properties. These properties consist of continuously emitted signals that serve as a background activity level. As with the dopaminergic system, elevation from the background activity indicates an unpredicted appetitive event. The two systems can therefore display complicated behavior indicating both the continuous conditions where there are no unpredicted events and the arrival of an unpredicted reward. We therefore suggest that the neural signals of unpredicted rewards are used to modify the striatal network for better future performance [67]. The continuous background activity is a necessary signal instructing the plastic neural networks of the basal ganglia to stay in their current state ("no news is good news").

The rate of the expectation signal increases when the monkey receives a cue, which predicts reward, and decreases when the monkey performs an error. Our finding of reduction in the rate following an error is important, since it occurred following erroneous trials, in some of which the absence of reward is not a disappointment to the monkey. Recently, [79] debated the connection between dopaminergic activity and reward expectation and suggested that this activity is related to reallocating attentional resources to unexpected salient events. Although more elaborate behavioral tasks are needed in order to discriminate between neural mechanisms related to attention-shift or reward expectation, we believe that the similarities between the TANs and dopaminergic neurons support the hypothesis that the basal ganglia play a significant role in reinforcement learning.

It was shown that, following MPTP treatments, TANs lose their specific response to reward cues [14, 78]. This could be due either to the fact that the continuous background level of the neural expectation signal is reduced, or to the inability of the biological system to detect the reward cue. Our failure to detect any significant expectation signal in the activity of TANs in the MPTP treated monkeys, supports the view that the on-going level of this neural signal is reduced. We therefore hypothesize that, as in the case of the dopaminergic system, the continuous decrease in the expectation signal of TANs in the MPTP treated animals indicates a continuous state of predicted rewards that fail to occur. This pathological state may lead to impaired reorganization of the basal ganglia circuitry in Parkinson's disease.

21 Epilogue

The main contribution of this work (together with the earlier report [42]) lies in suggesting statistical views for the single trial multi-unit activity. Such tools allow us to investigate more elaborate questions regarding the neural code. All of these methods are motivated by the current ability to record from a small network of cells. This ability allows us a glimpse into the complicated computations that are carried out by the brain. These computations are believed to be gaining a lot from the large number of cells, and the large number of connections between the cells.

We believe that as the number of cells recorded simultaneously increases, there will be more need for such statistical methods. By and by such recording will also allow us to ask questions that are inaccessible today due to the small number of cells recorded simultaneously. It will be fascinating to find out how much information can be gained from one subset of cells on the other.

Our hope is that the kind of statistical methods discussed here will increase the global knowledge about the neural code and how the brain functions.

References

[1] M. Abeles. *Corticonics.* Cambridge University Press, 1991.

[2] M. Abeles. *Local Cortical Circuits - an electrophysiological study. In: Studies of Brain Function.* Berlin: Springer-Verlag, 1991.

[3] M. Abeles, H. Bergman, I. Gat, I. Meilijson, E. Seidemann, N. Tishby, and E. Vaadia. Cortical activity flips among quasi-stationary states. *Proc. Natl. Acad. Sci.*, pages 8616–8620, 1995.

[4] M. Abeles, H. Bergman, E. Margalit, and E. Vaadia. Spatio temporal firing patterns in the frontal cortex of behaving monkeys. *J. Neurophysiology*, 1993.

[5] M. Abeles and G.L. Gerstein. Detecting spatiotemporal firing patterns among simultaneously recrded single neurons. *J. Neurophysiology*, 6:909–924, 1988.

[6] M. Abeles, Y. Prut, H. Bergman, E. Vaadia, and A. Aertsen. Integration, synchronicity and periodicity. In A. Aertsen, editor, *Brain Theory*, pages 149–181. Elsevier Publishers, 1993.

[7] M. Abeles, E. Vaadia, and H. Bergman. Firing patterns of single units in the prefrontal cortex and neural-network models. *Network*, 1:13–25, 1990.

[8] N. Agmon, Y. Alhassid, and R. D. Levine. An algorithm for finding the distribution of maximal entropy. *Journal of Computational Physics*, 30:250–258, 1979.

[9] E. Ahissar, E. Vaadia, M. Ahissar, H. Bergman, A. Arieli, and M. Abeles. Dependence of cortical plasticity on correlated activity of single neurons and on behavioral context. *Science*, 257:1412–1415, 1992.

[10] M. Ahissar, E. Ahissar, H. Bergman, and E. Vaadia. Encoding of sound-source location and movement : Activity of single neurons and interactions between adjacent neurons in the monkey auditory cortex. *J.Neurophysiology*, 67:934–952, 1992.

[11] M. Ahissar, E. Ahissar, H. Bergman, and E. Vaadia. Encoding of sound-source location and movement : Activity of single neurons and interactions between adjacent neurons in the monkey auditory cortex. *J.Neurophysiology*, 67:934–952, 1992.

[12] J.H. Allum, M.C. Hepp-Reymond, and R. Gysin. Cross-correlation analysis of interneuronal connectivity in the motor cortex of the monkey. *Brain Res.*, 231(2):325–34, 1982.

[13] D.J. Amit. *Modeling Brain Function, The world of attractor neural networks.* Cambridge University Press, 1989.

[14] T. Aosaki, A.M. Graybiel, and M. Kimura. Effect of the nigrostriatal dopamine system on acquired neural responses in the striatum of behaving monkeys. *Science*, 265:412–415, 1994.

[15] T. Aosaki, M. Kimura, and A.M. Graybiel. Temporal and spatial characteristics of tonically active neurons of the primate's striatum. *J.Neurophysiol.*, 73:1234–1252, 1995.

[16] T. Aosaki, H. Tsubokawa, A. Ishida, K. Watanabe, A.M. Graybiel, and M. Kimura. Responses of tonically active neurons in the primate's striatum undergo systematic changes during behavioral sensorimotor conditioning. *J.Neurosci.*, 14:3969–3984, 1994.

[17] P. Apicella, E. Legallet, and E. Trouche. Responses of tonically discharging neurons in the monkey striatum to primary rewards delivered during different behavioral states. *Exp.Brain Res.*, 116:456–466, 1997.

[18] P. Apicella, T. Ljungberg, E. Scarnati, and W. Schultz. Responses to reward in monkey dorsal and ventral striatum. *Exp. Brain Res.*, 85:491–511, 1991.

[19] A. Barbeau. The pathogenesis of parkinson's disease: A new hypothesis. *Canad.Med.Ass.J.*, 87:802–807, 1962.

[20] H. Barlow. The biological role of the neocortex. In A. Aertsen and V. Braitenberg, editors, *Information Processing in the Cortex*, pages 53–80. Springer-Verlag, 1992.

[21] H. Barlow. The neuron doctrine in perception. In M. Gazzaniga, editor, *The Cognitive Neuroscience*, pages 415–435. Boston: MIT Press, 1994.

[22] H.B. Barlow. Single units and sensation: a neuron doctrine for perceptual psychology. *Perception*, 1:371–394, 1972.

[23] R.E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[24] R.E. Bellman and S.E. Dreyfus. *Applied Dynamic Programming.* Princeton University Press, 1962.

[25] B.D. Bennet and C.J. Wilson. Synaptic regulation of action potential timing in neostriatal cholinergic interneurons. *The Journal of Neuroscience*, 18:8539–8549, 1998.

[26] B.D. Bennett and C.J. Wilson. Spontaneous activity of neostriatal cholinergic interneurons in vitro. *J.Neurosci.*, 19:5586–5596, 1999.

[27] W. Bialek, F. Rieke, R.R. de-Ruyter-van Steveninck, and D. Warland. Reading a neural code. *Science*, 252(5014):1854–7, 1991.

[28] J. P. Bolam and B. D. Bennet. Molecular and cellular mechanisms of neostriatal function. *R.G. Landes Company, Georgetown, Texas*, 1995.

[29] J.P. Bolam and B.D. Bennet. *Microcircuitry of the Neostriatum*, chapter Molecular and cellular mechanisms of neostriatal function, pages 1–19. R.G. Landes Company, 1995.

[30] N. Brenner, S.P. Strong, R. Koberle, W. Bialek, and R. de Ruyter van Steveninck. The economy of impulses and the stiffnes of spike trains. Technical report, NEC Research Institute, 1998.

[31] C.J. Bruce. Single neuron activity in the monkey's prefrontal cortex. In P. Rakic and W. Singer, editors, *Neurobiology of Neocortex*, pages 297–329. Wiley, 1988.

[32] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley, 1991.

[33] D.R. Cox. *Renewal Theory.* Methuen London, 1962.

[34] M.D. Crutcher and M.R. DeLong. Single cell studies of the primate putamen. i. functional organization. *Exp.Brain Res.*, 53:233–243, 1984.

[35] A. Date, E. Bienenstock, and S. Geman. On the temporal resolution of neural activity. Unpublished manuscript Division of Applied Mathematics, Brown University, May 1998.

[36] A. Date, E. Bienenstock, and S. Geman. A statistical technique for the detection of fine temporal structure in multi-neuronal spike trains. *Society for Neuroscience Abstracts*, 25, 1999.

[37] J.E. Dayhoff and G.L. Gerstein. Favored patterns in spike trains. ii. application. *J. Neurophysiol.*, 49(6):1349–63, 1983.

[38] R. Desimone, S.J. Schein, J. Moran, and L.G. Ungerleider. Contour, color and shape analysis beyond the striate cortex. *Vision. Res.*, 25(3):441–52, 1985.

[39] R. Frost. *The road not taken.*, chapter Mountain interval. Henry Holt, New York, 1916.

[40] J.M. Fuster. Behavioral electrophysiology of the prefrontal cortex. *TINS*, pages 408–414, 1984.

[41] J.M. Fuster. Prefrontal cortex and temporal integration. In A. Peters and E.G. Jones, editors, *Cerebral Cortex*, pages 151–177. Plenum Press, 1985.

[42] I. Gat. Unsupervised learning of cell activities in the associative cortex of behaving monkeys, using hidden markov models. Master's thesis, The Hebrew University, Jerusalem, January 1994.

[43] I. Gat and N. Tishby. Hidden markov modelling of simultaneously reocrded cells in the associative cortex of behaving monkeys. *Nework*, 8(3):297–332, August 1997.

[44] A.P. Georgopoulos, A.B. Schwartz, and R.E. Kettner. Neuronal population coding of movement direction. *Science*, 233:1416 – 1419, 1986.

[45] A.P. Georgopoulos, M. Taira, and A. Lukashin. Cognitive neurophysiology of the motor cortex. *Science*, 260(5104):47–52, 1993.

[46] C. Ghez. Voluntary movement. In J.H. Schwartz E.P. Kandel and T.M. Jessell, editors, *Principles of Neural Science*, pages 609–625. Wiley, 1988.

[47] A. M. Graybiel, T. Aosaki, and M. Kimura A. W. Flaherty. The basal ganglia and adaptive motor control. *Science*, 265:1826–1830, 1994.

[48] M. Gutman. Asymptotically optimal classification for multiple tests with empirically observed statistics. *IEEE Trans. Info. Th*, 35:401–408, 1989.

[49] N. G. Hatsopoulos, C. L. Ojakangas, L. Paninski, and John P. Donoghue. Information about movement direction obtained from synchronous activity of motor cortical neurons. *Proc. Natl. Acad. Sci.*, 95:15706–15711, 1998.

[50] O. Hikosaka, M. Sakamoto, and S. Usui. Functional properties of monkey caudate neurons. i. activities related to saccadic eye movements. *J.Neurophysiol.*, 61:780–798, 1989.

[51] J.J. Hopefield. Neural network and physics systems with emergent selective computational properties. *Proc. Natl. Acad. Sci.*, 79:2554–2558, 1982.

[52] D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction and functional architecture in cat's visual cortex. *J. Physiol.*, 160:106–154, 1962.

[53] D.H. Hubel and T.N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J. Physiolgy*, 195:215–243, 1968.

[54] D. H. Johnson, C.M. Gruner, K. Baggerly, and C. Seshagiri. Information-theoretic analysis of the neural code. Download from http://www-dsp.rice.edu/ dhj, February 1998.

[55] A.C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging.* IEEE Press, New York, 1988.

[56] I. Kanter and H. Sompolinsky. Associative recall of memory without errors. *Phys. Rev.*, A35:380, 1986.

[57] Y. Kawaguchi, C.J. Wilson, S.J. Augood, and P.C. Emson. Striatal interneurons: chemical, physiological and morphological characterization. *TINS*, 18:527–535, 1995.

[58] M. Kimura. The role of primate putamen neurons in the association of sensory stimuli with movement. *Neurosci.Res.*, 3:436–443, 1986.

[59] T.W. Kjaer, J.A. Hertz, and B.J. Richmond. Decoding cortical neuronal signals: Network models, information estimation and spatial tuning. *J. Comput. Neurosci*, 1:109–139, 1994.

[60] S. Kullback. *Information theory and statistics.* Wiley & sons New York, 1959.

[61] C.R. Legendy and M. Salcman. Bursts and recurrences of bursts in the spike trains of spontaneously active striate cortex neurons. *J. Neurophysiol.*, 53(4):926–39, 1985.

[62] F.L. Leheman. *Testing Statistical Hypotheses.* Wiley, New York, 1959.

[63] S.E. Levinson, L.R. Rabiner, and M.M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell Syst. Tech*, 62(4):1983, 1035-1074.

[64] J. Lin. Divergence measures based on the shannon entropy. *IEEE Trans. Info. Th.*, 37:145–151, 1991.

[65] J.W. McClurkin, L.M. Optican, B.J. Richmond, and T.J. Gawne. Concurrent processing and complexity of temporally encoded neuronal messages in visual perception. *Science*, pages 675–677, 1991.

[66] Y. Miyashita. Neuronal correlate of visual associative long-term memory in the primate temporal cortex. *Nature*, 335(6193):817–20, 1988.

[67] P.R. Montague, P. Dayan, and T.J. Sejnowski. A framework for mesencephalic dopamine systems based on predictive hebbian learning. *J.Neurosci.*, 16:1936–1947, 1996.

[68] J.I. Nelson, P.A. Salin, M.H. Munk, M. Arzi-M, and J. Bullier. Spatial and temporal coherence in cortico-cortical connections: a cross-correlation study in areas 17 and 18 in the cat. *Vis. Neurosci.*, 9(1):21–37, 1992.

[69] M.T. Newsome, K.H. Britten, and J.A. Movshon. Neuronal correlates of a perceptual decision. *Nature*, 341(6237):52–4, 1989.

[70] M.W. Oram, M.C. Winter, and B.J. Richmond. Stochastic nature of precisly timed spike patterns in visual system neuronal responses. *J. Neurophysiol*, 81(6):3021–3033, June 1999.

[71] D.N. Pandya and E.H. Yeterian. Architecture and connections of cortical association areas. In A. Peters and E.G. Jones, editors, *Cerebral Cortex*, pages 609–624. Plenum Press, 1985.

[72] D.H. Perkel, G.L. Gerstein, and G.P. Moore. Neuronal spike trains and stochastic point processes. ii. simultaneous spike trains. *Biophys. J.*, 7:419–440, 1967.

[73] A.B. Poritz. Hidden markov models: A guided tour. In *Proc. ICASSP-88*, pages 7–13, 1988.

[74] Y. Prut. *Firing Patterns as a sensitive estimator of cortical information processing.* PhD thesis, Hebrew university of Jerusalem, August 1995.

[75] Y. Prut, E. Vaadia, H. Bergman, I. Haalman, H. Slovin, and Moshe Abele. Spatiotemporal structure of cortical activity: Properties and behavioral relevance. *J. Neurophysiology*, 79(6):2857–2874, June 1998.

[76] L.R. Rabiner. Tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, 1989.

[77] L.R. Rabiner, B.H. Juang, S.E. Levinson, and M.M. Sondhi. Recognition of isolated digits using hidden markov models with continuous mixture densities. *AT&T Tech*, 64(6):1211–1222, 1986.

[78] A. Raz, A. Feingold, V. Zelanskaya, E. Vaadia, and H. Bergman. Neuronal synchronization of tonically active neurons in the striatum of normal and parkinsonian primates. *J.Neurophysiol.*, 76:2083–2088, 1996.

[79] P. Redgrave, T.J. Prescott, and K. Gurney. Is the short-latency dopamine response too short to signal reward error? *TINS*, 22, 1999.

[80] B.J. Richmond, L.M. Optican, M. Podell, and H. Spitzer. Temporal encoding of two-dimensional patterns single units in primate inferior temporal cortex. i. response characteristics. *J. Neurophysiol.*, 57(1):132–46, 1987.

[81] A. Riehle, Sonja Grn, M. Diesmann, and A. Aertsen. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, 12(278):1950–1953, December 1997.

[82] T.W. Robbins and B.J. Everitt. Neurobehavioural mechanisms of reward and motivation. *Curr.Opin.Neurobiol.*, 6:228–236, 1996.

[83] K. Sakai and Y. Miyashita. Neural organization for the long-term memory of paired associates. *Nature*, 354(6349):152–5, 1991.

[84] W. Schultz. Predictive reward signal of dopamine neurons. *J.Neurophysiol*, 80:1–27, 1998.

[85] E. Seidemann, M. Abeles, I. Meilijson, and E. Vaadia. Simultaneously recorded single units in the frontal cortex go through sequences of discrete and stable states in monkeys performing a delayed localization task. *J. Neurophysiology*, pages 752–768, 1996.

[86] E. Seidman. Prediction on monkey's behavior from simultaneously recorded single unit spike trains. Master's thesis, Tel-Aviv University, September 1993.

[87] L.A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, 1:113–122, 1982.

[88] W. Singer. Time as coding space in neocortical processing: A hypothesis. In G. Buzsaki, R. Llinas, W. Singer, A. Berthoz, and Y. Christen, editors, *Temporal Coding in the Brain*, pages 51–79. Springer-Verlag, 1994.

[89] H. Slovin. *Cognitive Impairments and Neural Activity in the Frontal Cortex of Low-Dose MPTP Treated Monkeys*. PhD thesis, Hebrew University, Jerusalem, July 1997.

[90] E.S. Tolosa and C. Marin. *Medical Management of Parkinsonian Tremor*, pages 333–350. Marcel Dekker, Inc., New York, 1995.

[91] K. Toyama, M. Kimura, and K. Tanaka. Cross-correlation analysis of interneuronal connectivity in cat visual cortex. *J. Neurophysiology*, 46(2):191–201, 1981.

[92] J.D. Victor and K.P. Purpura. Nature and precision of temporal coding in visual cortex: A metric-space analysis. *J. Neurophysiol*, 76(2):1310–1326, August 1996.

[93] J.D. Victor and K.P. Purpura. Metric-space analysis of spike trains: theory, algorithms and application. *Network*, 8(2):127–164, May 1997.

[94] A. E. P. Villa, I. V. Tetko, B. Hyland, and A. Najem. Spatiotemporal activity patterns of rat cortical neurons predict responses in a conditioned task. *Proc. Natl. Acad. Sci.*, 96:1106–1111, 1999.

[95] A.E. Villa and J.M. Fuster. Temporal correlates of information processing during visual short-term memory. *Neuroreport*, 3(1):113–6, 1992.

[96] K. Watanabe and M. Kimura. Dopamine receptor-mediated mechanisms involved in the expression of learned activity of primate striatal neurons. *J.Neurophysiol.*, 79:2568–2580, 1998.

[97] C.J. Wilson, H.T. Chang, and S.T. Kitai. Firing patterns and synaptic potentials of identified giant aspiny interneurons in the rat neostriatum. *J.Neurosci.*, 10:508–519, 1990.

[98] J. Yelnik, G. Percheron, C. Francois, and A. Garnier. Cholinergic neurons of the rat and primate striatum are morphologically different. *Prog.Brain Res.*, 99:25–34, 1993.

[99] H. Yoon and H. Sompolinsky. The effect of correlation on the fisher information of population codes. *Neural Information Processing Systems*, 11:167–173, 1999.

[100] J. Ziv. On classification with empirically observed statistics and universal data compression. *IEEE Trans. Info. Th.*, 34:278–286, 1988.