Manuscript Number: 2105

Spotting Neural Spike Patterns using an Adversary Background model

Itay Gat & Naftali Tishby

Institute of Computer Science and Center for Neural Computation
Hebrew University
Jerusalem, 91904 Israel
Email: {itay,tishby}@cs.huji.ac.il

February 21, 2001

Abstract

The detection of a specific stochastic pattern embedded in an unknown background noise is a difficult pattern recognition problem, encountered in many applications such as word spotting in speech. A similar problem emerges when trying to detect a multi-neural spike pattern in a single electrical recording, embedded in the complex cortical activity of a behaving animal. Solving this problem is crucial for the identification of neuronal "code words" with specific "meaning". The technical difficulty of this detection is due to the lack of a good statistical model for the background activity, which rapidly changes along the recording conditions and activity of the animal. This work introduces the use of an adversary background model. This model assumes that the background "knows" the pattern sought, up to a first order statistics, and this "knowledge" creates a background composed of all the permutations of our pattern. We show that this background model is tightly connected to the "type-based" information theoretic approach. Furthermore, we show that computing the likelihood ratio, is actually decomposing the log-likelihood distribution according to "types" of the empirical counts. We demonstrate the application of this method for detection of the "reward" patterns in the Basal-Ganglia of behaving monkeys, yielding some unexpected biological results.

1 Introduction

One of the intriguing questions in brain research concerns the neural code, i.e. the code used by neurons in the brain to process, store, and convey information. There is strong evidence for the theory that this code is conveyed by action potentials (usually referred to as spikes trains), but we lack a full understanding of how these spike trains hold the information needed. The understanding of this code is a key issue for understanding how the brain works.

In order to gain more information about this code, researches usually attempt to associate the spike trains taken from extra-cellular recordings with the stimuli the animal was presented with at the time. The next step is to characterize this association under different conditions. In principle, this association can be discovered in one of two ways. The first involves computing the response of cells near a known stimuli. The second, which involves working out the proximity of a given response to the stimuli, is the commonly used method. This method has proven to be quite successful in the past and revealed some important basic mechanisms of neuronal code (e.g. see (Georgopoulos, Schwartz, & Kettner, 1986),(Graybiel, Aosaki, & A. W. Flaherty, 1994)).

It does, however, suffer from some severe limitations: the alignment problem and the non-continuous search, to name only two. The alignment problem stems from the assumption that every time the animal is presented with the same stimulus, it goes through the same thinking process. This is, clearly, a problematic assumption. We are not able to construct the exact same process in the brain, as we are unable to control, directly, many relevant variables. The non-continuous problem refers to the fact that the search is not conducted in *all* the continuous recordings taken from the animals' brain. The search is conducted only a short time *before* and *after* an event we consider important has occurred. It is possible that the response we are looking for appeared in a non-stimuli-locked fashion. It might appear in unexpected places – which are not searched using the current methods.

The main problem that prevented researches from conduction this "reverse search" lies in the nature of the spike trains, which hide the neural code. The spike trains (especially in areas involved in complicated tasks) are known to be of a low rate (5-20 spikes per second), and of high variability (Abeles, 1991). The combination of these facts creates a signal with a very high variance. Most of the research was therefore concerned with reducing this variance by aligning several repetitions of the same task and calculating an average response.

In the work presented here, when choosing a method, we took "the one less traveled by"(Frost, 1916), of searching for a single trial response in a continuum of data. We made use of the fact that modern technology enables us to record from several cells simultaneously, and therefore to reduce the variance of response. Our problem is, therefore, the detection 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 exists prior to the search. No attempt was made to look here for ways of self-constructing such patterns.

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. In such test 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 (Leheman, 1959). When one or more of the possible sources are not fully defined, the LRT is not applicable.

In many cases, the researcher knows only that the sources belong to a certain parametric family, and receives training sequence emitted from one of the sources. This problem, also known as the two-sample problem (Leheman, 1959), requires a decision as to 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 (Ziv, 1988) and extended by Gutman (Gutman, 1989). Their method relies on the information theoretic notion of "typicality" of the empirical sequences. It has been shown 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 present 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. The fifth section will show an application of this technique for recordings performed in the basal ganglia of behaving monkeys. The last section will discuss the method presented in this work, and will also show possible extensions of it.

2 Definitions

2.1 Statistical assumptions

In the work presented here we will concentrate on detection a neural response in multi-unit recordings. An example of such a response is given in figure 1A and B. These figures show several of the problematic features of neuronal data, e.g. the low firing rate on one hand and the fast changes in rate on the other hand. 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.

INSERT FIGURE 1 HERE.

The response is mainly manifested in elevation in firing rate (between the 150 and 220 ms after the alignment point), followed by a reduction in rate (till 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 influenced by the existence (or absence) of a single spike.

The statistical description of this data is based on the common assumption that the spike trains are the outcome of a non-homogeneous Poisson process (Abeles, 1991). According to this view, the spikes within a short period of 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 may 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, 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. The example of the pattern in figure 2 represents the spike activity of four neurons recorded in the Basal-Ganglia of a behaving monkey. A detailed description of this specific pattern will be presented in section 5.

INSERT FIGURE 2 HERE

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 3.

INSERT FIGURE 3 HERE

2.2 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 a variable rate Poisson pattern with M bins

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

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

$$(n_1, n_2, ..., n_M) = \underline{n} \;, \qquad \sum_{i=1}^{M} n_i = N, \tag{2}$$

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:

$$p_{\underline{\lambda}}(\underline{n}) = \prod_{i=1}^{M} e^{-\lambda_i} \frac{{\lambda_i}^{n_i}}{n_i!} \;. \tag{3}$$

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

$$\sum_{\underline{n}} p_{\underline{\lambda}}(\underline{n}) = \sum_{\underline{n}} \prod_{i=1}^{M} e^{-\lambda_i} \frac{{\lambda_i}^{n_i}}{n_i!} = e^{-\Lambda} \frac{\Lambda^N}{N!} \;, \tag{4}$$

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.

2.3 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, 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 would 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 start out with reasonable background model, and we require a plausible one for this search. Figure 3 gives an example of such a spotting paradigm.

3 Selection of background model

3.1 Using a wrong 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 a background model that is wrong. 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. 3).

The drawback of this method is evident when example is examined. Let us assume that we wish to search for the pattern shown in figure 2. It can be seen from our formulation that the frame with the highest likelihood for this pattern is the frame that contains no spikes at all. This is the consequence of 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), the total number of counts in the spotted examples goes down monotonically, as shown in figure 4. The problem shown here is not, however, limited to cases in which the expected number of counts in each bin is very small. Rather, it results from using an inadequate background mode.

INSERT FIGURE 4 HERE

3.2 Adversary background model

We suggest using an adversary background model. This model is adversary in the sense that it has first order statistics on 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 can not distinguish between the different examples (all having the same first order statistics) – and therefore they are all assigned 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 this purpose we can collect many examples of how people write the digit. It is, however, 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 that we were also able to estimate its parameters using our training examples.

INSERT FIGURE 5 HERE

Given a scribble, 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 is shown in figure 5.

Figure 5A shows the example presented to the spotter. The first order statistics defined in this example are given in 5B, and consist of a finite set of points defined for the scribbling. Two examples of the adversary background are also shown. They imitate the scribbling up to a first order statistics, i.e. they go through the same points defined on the scribbling (5C,D). The ultimate goal is to compute the LRT of the example with the background, as seen in 5E.

INSERT FIGURE 6 HERE

The adversary model is optimal if all that is known about the background is the first order statistics. This is due to the use of uniform distribution over the permutations of the data, which stems from 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 less tight decision boundary, and therefore the decision would be less accurate.

In general what we have here is a case in which the LRT, which may be seen as a linear separator in the probability simplex (Cover & Thomas, 1991) can be extended according to our data points. An example of this phenomenon is shown in figure 6. In this figure, we see two linear separators each for a different background model. This figure shows that some of our data points could not have been spotted had the wrong background model been used.

Using the adversary background model, one still has to choose the first order statistics to be used. The question is to what extent is the adversary background able to 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.

4 Computing the adversary background model

4.1 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 (Cover & Thomas, 1991) 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 (3), and later on show how to decompose 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.

4.2 The log-likelihood distribution

Let us examine the logarithm of the likelihood, $\log p_{\underline{\lambda}}(\underline{n})$, in Eq.(3). This expression is a sum over M independent random variables,

$$\begin{aligned} \log p_{\underline{\lambda}}(\underline{n}) &= \sum_{i=1}^{M} (-\lambda_i + n_i \log \lambda_i + \log(n_i!)) \\ &= -\Lambda + \sum_{i=1}^{M} (n_i \log \lambda_i + \log(n_i!)) \ . \end{aligned} \tag{5}$$

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:

$$\begin{aligned} \mu_{\underline{\lambda}} &= \langle \log p_{\underline{\lambda}}(\underline{n}) \rangle_{\underline{\lambda}} \\ &= -\Lambda + \sum_{i=1}^{M} \lambda_i \log \lambda_i - \sum_{i=1}^{M} \langle \log(n_i!) \rangle_{\underline{\lambda}} \end{aligned} \tag{6}$$

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.,

$$\langle \log(n_i!) \rangle_{\underline{\lambda}} \approx 0 \times e^{-0.15} + 0 \times e^{-0.15} 0.15 + \log(2) \times e^{-0.15} \frac{0.15^2}{2} + \log(6) \times e^{-0.15} \frac{0.15^3}{6} \ . \tag{7}$$

The evaluation of the variance of the sample log-likelihood, Eq.(5), under the model distribution, is similar:

$$\sigma^2_{\underline{\lambda}} = Var(\log p_{\underline{\lambda}}(\underline{n})) = Var\left(-\Lambda + \sum_{i=1}^{M} n_i \log(\lambda_i) + \log(n_i!)\right) \tag{8}$$

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

$$\sum_{i=1}^{M} Var\left(n_i \log(\Lambda) - \log(n_i!)\right) \ . \tag{9}$$

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

$$\begin{aligned}
\langle (n_i \log \lambda_i)^2 &+ \log^2(n_i!) - 2\log\lambda_i n_i \log(n_i!)\rangle - (\lambda_i \log\lambda_i - \langle \log(n_i!)\rangle)^2 \\
= (\lambda_i^2 + \lambda_i)\log^2\lambda_i &+ \langle \log^2(n_i!)\rangle - 2\log\lambda_i \langle n_i \log(n_i!)\rangle \qquad (10) \\
&- \left(\lambda_i^2 \log^2\lambda_i - 2\lambda_i \log\lambda_i \langle \log(n_i!)\rangle + \langle \log(n_i!)\rangle^2\right) \\
= \lambda_i \log^2\lambda_i &+ Var(\log(n_i!)) - 2\log\lambda_i \langle n_i \log(n_i!) - \lambda_i \langle \log(n_i!)\rangle\rangle \ .
\end{aligned}$$

The model and its sample log-likelihood distributions are shown in figure 7.

INSERT FIGURE 7 HERE

4.3 Computing the log-likelihood distribution of a sample class

$T(\underline{n})$ denotes the type class of the count vector $\underline{n}$, and $c_i(\underline{n})$ demote 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:

$$|T(\underline{n})| = \frac{M!}{\prod_{i=1}^{M} c_i(\underline{n})!}. \qquad (11)$$

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

$$P_{\underline{\lambda}}(\log p_{\underline{\lambda}}(\underline{n})) = \sum_{T(\underline{n})} P_{\underline{\lambda}}(T(\underline{n})) \times P_{\underline{\lambda}}(\log p_{\underline{\lambda}}(\underline{n})|T(\underline{n})) \ . \qquad (12)$$

When given a sample, its type is easily computed. The idea is, therefore, 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 decomposition.

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

$$P_{\underline{\lambda}}(T(\underline{n})) = \sum_{n' \in T(\underline{n})} p_{\underline{\lambda}}(\underline{n'}) = |T(\underline{n})| \langle p_{\underline{\lambda}}(\underline{n}) \rangle_{T(\underline{n})} \approx |T(\underline{n})| e^{\langle \log p_{\underline{\lambda}}(\underline{n}) \rangle_{T(\underline{n})}}, \quad (13)$$

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.(13) 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.,

$$\sum_{T(\underline{n})} P_{\underline{\lambda}}(T(\underline{n})) = 1 \ , \quad (14)$$

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 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

$$P(Type(t)) = c_0 e^{-\lambda_i} \times c_1 e^{-\lambda_i} \lambda_i \times c_2 e^{-\lambda_i} \frac{\lambda_i^2}{2} \times c_3 e^{-\lambda_i} \frac{\lambda_i^3}{6} ... \quad (15)$$

4.4 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, $P_{\underline{\lambda}}(\log p_{\underline{\lambda}}(\underline{n}))$. As can be seen from the histogram of the log-likelihood in the type (figure 8), the central limit theorem applies even better to this distribution, which can thus be considered as normal.

INSERT FIGURE 8 HERE

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

$$\log p_{\underline{\lambda}}(\underline{n}) = \sum_{i=1}^{M} [-\lambda_i - \log(n_i!) + n_i \log \lambda_i] = -\Lambda - L(\underline{n}) + \sum_{i=1}^{M} n_i \log \lambda_i \ , \quad (16)$$

with the notation

$$L(\underline{n}) \stackrel{def}{=} \sum_{i=1}^{M} \log(n_i!) \ . \quad (17)$$

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,

$$\mu_{T(\underline{n})} = \langle \log p_{\underline{\lambda}}(\underline{n}) \rangle_{T(\underline{n})} = -\Lambda - L(\underline{n}) + \langle \sum_{i=1}^{M} n_{\pi(i)} \log(\lambda_i) \rangle_{\pi} \ . \quad (18)$$

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

$$\mu_{T(\underline{n})} = -\Lambda - L(\underline{n}) + \langle \underline{n} \rangle \sum_{i=1}^{M} \log \lambda_i = -\Lambda - L(\underline{n}) + \frac{N}{M} \sum_{i=1}^{M} \log \lambda_i \ . \quad (19)$$

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:

$$\begin{aligned} \mu^2_{T(\underline{n})} &= \langle \log p(\underline{n}) \rangle^2 \quad (20) \\ = \Lambda^2 &+ L^2(\underline{n}) + \left(\frac{N}{M}\right)^2 l^2(\lambda) + 2\Lambda L(\underline{n}) - 2\Lambda \frac{N}{M} l(\lambda) - 2L(\underline{n}) \frac{N}{M} l(\lambda) \ , \end{aligned}$$

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

$$\langle \log^2(p(\underline{n}))\rangle_{T(\underline{n})} = \langle (-\Lambda - L(\underline{n}) + \frac{N}{M} l(\lambda))^2 \rangle = \tag{21}$$

$$= \langle \Lambda^2 + L^2(\underline{n}) + \left(\frac{N}{M}\right)^2 l^2(\lambda) + 2\Lambda L(\underline{n}) - 2\Lambda \frac{N}{M} l(\lambda) - 2L(\underline{n}) \frac{N}{M} l(\lambda) \rangle_{T(\underline{n})}.$$

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

$$Var(\log p_{\underline{\lambda}}(\underline{n})) = \langle \sum_{i,j} n_i n_j \log \lambda_i \log \lambda_j \rangle_{T(\underline{n})} - \left(\frac{N}{M}\right)^2 l^2(\lambda). \tag{22}$$

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,

$$\langle \sum_{i,j} n_{\pi(i)} n_{\pi(j)} \log \lambda_i \log \lambda_j \rangle_\pi = \tag{23}$$

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

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

$$\frac{\sum_{i=1}^M n_i^2}{M} \log^2 \lambda_i + \frac{\sum_{i \neq j} n_i n_j}{M^2 - M} \sum_{i \neq j} \log \lambda_i \log \lambda_j \ , \tag{24}$$

or as

$$\frac{1}{M^2 - M} \left((\sum_i n_i)^2 - \sum_i n_i^2 \right) \left(l^2(\lambda) - \sum_i \log^2 \lambda_i \right) \ . \tag{25}$$

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

$$\sigma^2_{T(\underline{n})} = \frac{1}{M-1} \left(\sum_i n_i^2 - \frac{N^2}{M} \right) \left(\sum_i \log^2(\lambda_i) - \frac{l(\lambda)^2}{M} \right) \ , \tag{26}$$

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

$$\sigma^2_{T(\underline{n})} = \frac{M^2}{M-1} Var(\{n_i\}) Var(\{\log \lambda_i\}) \ . \tag{27}$$

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 decomposition of $P_{\underline{\lambda}}$, Eq.(12), as:

$$
\begin{aligned}
P(\log p_{\underline{\lambda}}(\underline{n})) &= \sum_{T(\underline{n})} P_\lambda(T(\underline{n}))P(\log p_{\underline{\lambda}}(\underline{n})|T(\underline{n})) \qquad (28)\\
&= \sum_{T(\underline{n})} e^{(\log|T(\underline{n})|+\mu_{T(\underline{n})})} e^{\left(-\frac{(\log p(\underline{n})-\mu_{T(\underline{n})})^2}{2\sigma^2_{T(\underline{n})}} - \frac{1}{2}\log(2\pi\sigma^2_{T(\underline{n})})\right)}
\end{aligned}
$$

INSERT FIGURE 9 HERE

An example of such decomposition is given in figure 9. In this figure we show how the log-likelihood distribution is formed out 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:

$$
|T(\underline{n})| \exp(< \log(p(\underline{n})) >_{T(\underline{n})}), \qquad (29)
$$

or alternatively:

$$
\exp(\log(|T(\underline{n})|)+ < \log(p(\underline{n})) >_{T(\underline{n})}). \qquad (30)
$$

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

$$
S_{\underline{\lambda}}(\underline{n}) = \log(|T(\underline{n})|) + \mu_{T(\underline{n})} - \frac{(\log p(\underline{n}) - \mu_{T(\underline{n})})^2}{2\sigma^2_{T(\underline{n})}} - \frac{1}{2}\log(2\pi\sigma^2_{T(\underline{n})}) \ , \qquad (31)
$$

Where both $\mu_{T(\underline{n})}$ and $\sigma^2_{T(\underline{n})}$ depend on the model $\underline{\lambda}$ up to a permutations of λ_i.

4.5 Implementation issues

The score computed in Eq 30 can be used directly when trying to spot a neural template. Its use is derived from the maximum likelihood principle. The procedure includes the parceling of neuronal data into frames. These frames would usually be successive and might also overlap. For each frame, the score is computed taking into consideration the sample type of the frame. The score of all frames is compared to a threshold and the frames whose score was above the threshold are declared to be the "spotting" of the algorithm.

In the current implementation of the adversary background model we use all the bins as though they were drawn from one source. When calculating the score for a given frame of data the adversary background is blind to the identity of the cell a specific bin was drawn from. The only information is the type class which count the number of bins with no spike, with one spike etc. This implementation was motivated by the features of our neuronal data (as explained in the application section). We also refer to different adversary knowledge in the discussion.

One important issue that must be addressed when using this adversary background model, is how one goes about choosing the type classes, or how broad they should be. One extreme is to define the type so broadly that (almost) all spike patterns are part of the same type, conversely one can define the type so narrowly that a typical type has only one pattern. These extremes do not serve the purpose of this work and it is clear that some value intermediary should be used. The two extremes described above are mainly governed by the bin size being used in parceling the neuronal template. An example of how to select an appropriate bin width will be given in the application section.

5 Application: Detecting the "reward pattern" in the Basal-Ganglia

5.1 Description of the data used

The application of the new detection method was shown on the reward signal of the tonically active neurons (TANs) in the Basal-Ganglia. A more detailed description of the data being used together with more results is given elsewhere (Gat, Raz, Abeles, Tishby, & Bergman, 2000). 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 (Aosaki et al., 1994). This response will henceforth be referred to as the "reward signature" of the TANs. 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 (Kawaguchi, Wilson, Augood, & Emson, 1995), they have a major role in the functions of the basal ganglia (Bolam & Bennet, 1995). Another interesting feature of these cells is the fact that all the TANs show the same response simultaneously.

The data used here consisted of recordings from two vervet monkeys (Monkeys H and I; Cercopithecus Aethiops Aethiops) that were trained to perform a visual-motor task (Abeles et al., 1995). The activity of 1-4 TANs was recorded simultaneously while the monkeys performed the behavioral task. The recording were carried out in the Higher Brain Function Laboratory at the Haddassah medical school. These experiments were conducted by E. Raz and A. Feingold as part of their PhD work under the supervision of H. Bergman (Raz, Feingold, Zelanskaya, Vaadia, & Bergman, 1996).

We defined the tone signaling of the reward (Monkey H), or the visual cue predicting the reward (Monkey I) as the *reward cue.* The responses of the 3-4 simultaneously-recorded TANs to the reward cues were used to create a rate-template for a specific recording session (Fig 10A). This rate template was computed over equi-width bins (of 20 ms in our example) using the method of peri stimulus time histogram (PSTH) model. The selection of bin width (and therefore the broadness of our type definition) was based on previous work (Gat & Tishby, 1997) in which we examined the optimal bin for detecting a neuronal response. In general it is important to choose a bin width that captures the changes in firing rate on the one hand, and enables a reasonable estimation of the firing rate on the other hand. Selecting a bin width that is too large may hide an important response. Bins that are too short may result in poorly estimated neuronal templates. The rate template was used as the model and was sought over *all* the trials of the recording session.

INSERT FIGURE 10 HERE

5.2 Reward signature is spotted significantly

In the work presented here we addressed the question of whether our detection procedure actually detected the reward signal. We found that the PSTH of the reward signal was very similar to the original one (see figure 10B). Thus, the detection scheme works properly. It is possible, however, that given enough data one can detect any reasonable pattern. To address that question we designed a significance computation procedure. In this procedure we performed the same detection on shuffled spike trains: we used the fact that we had multi-cell recordings and shifted the spike trains of the cells relative to each other.

The results of this significance test are as shown in the third column of table 1. The data there shows that the number of detected reward signals is significantly greater than in the shuffled data. We then tried to detect two other "signatures" in the same data. One signature was a constant (in which all the bins had exactly the same value). The other signature was constructed from different segments of the recording session (the 2 seconds *before* the reward was signaled). The differences between the number of detected signals in the real data and in the shuffled data can be seen in the fourth and fifth columns of table 1. It can be seen clearly that for these new signatures there were no significant results whatsoever.

Session	No. of cells	(Found-Expected)/ (Standard Deviation) Reward response	Flat PSTH	Before reward
H-1	4	15.3	-0.25	-0.91
H-2	3	9.4	0.44	0.43
H-3	3	9.9	-1.45	2.10
I-1	4	18.2	-0.88	1.70
I-2	3	11.8	0.96	-0.53

Table 1: The significance of spotting the neural reward signal. The first column gives the animal identity and the session number. The number of simultaneously recorded TANs in each session is given in the second column. We used several templates to verify the significance of spotting: the reward response (calculated separately for each of the recording sessions), a flat PSTH, and a PSTH computed from segments aligned 2 seconds before the reward cue. A significance level was computed for each of these experiments. This level was the difference between the number of detections in the original data and the average number of detections in the shuffled data. This difference was measured in standard deviation (SD) of the shuffled data. Differences greater than 3 SDs are considered as significant deviations from the null hypothesis.

5.3 Reward signal is not only confined to the reward cue

Previously it was hypothesized that the reward signature is a binary signal which can be found only in close proximity to the reward cue. This hypothesis was based on the fact that the reward signature was discovered after averaging many reward cue epochs. Using the detection method presented here we were able to test this hypothesis directly. This was done by locating the reward signals in relation to the behavioral events of the trials. An example of such a location is shown in figure 11.

INSERT FIGURE 11 HERE

Fig. 11A shows two interesting phenomena of the spotted template. Nat-

Session	Correct trials	Erroneous trials
H-1	61.7	-100
H-2	22.3	-60.6
H-3	28.7	-57.4
I-1	44.7	-12.7
I-2	12.5	-54.2

Table 2: The deviation from average level in correct trials and erroneous ones. The change from average level of reward signal is given both for the segment of 1 second starting 2 seconds after correct trials, and for the same segment in erroneous ones. The change is given in percentage of change from average level.

urally, the frequency of spotted events is maximal in the epoch locked to the reward cue. We also found that the number of spotted events was significant compared to the number expected by chance. However, the template is significantly spotted in all periods of the trial, and the majority of spotted events was not found in the vicinity of the reward cue. The frequency of the spontaneously emitted reward signal was not constant over all behavioral periods. As expected, this frequency reached a peak at the reward cue period, but remained high in the 2-4 seconds following the reward cue (Fig. 11 A,B). Reverse correlation of the spotted signals with the other behavioral events in the task failed to reveal any time locking, further suggesting that the elevation of the frequency of reward signal from the background level is associated only with the reward cue. A clear tendency toward reduction in this frequency is observed following erroneous trials. This tendency was observed in all recording sessions being examined, and is summarized in table 2.

5.4 Comparison with other "type" detection method

We compared the results obtained by our method to those achieved using a different type-based method (Johnson, Gruner, Baggerly, & Seshagiri, 1998), henceforth referred to as the Jensen-Shannon (*JS*) divergence (Lin, 1991). In this technique, a statistic is computed. It has been shown elsewhere (Gutman, 1989) that this statistic is optimal in a certain sense. It was also shown that *given enough training and testing data*, this statistic will produce error probabilities having the *same* exponential rate as the likelihood ratio classifier that clairvoyantly knows the underlying statistical model for the training data.

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: $R_l \in r_1, ..., r_K$. When given a test set $\mathbf{R} = R_1, ...R_{L_R}$, the statistics are:

$$G_j = \frac{L_T}{L_R}\mathcal{D}(\hat{P}_{\mathbf{T}_j}||\bar{P}_{\mathbf{T}_j,\mathbf{R}}) + \mathcal{D}(\hat{P}_{\mathbf{R}}||\bar{P}_{\mathbf{T}_j,\mathbf{R}}) \tag{32}$$

$$\bar{P}_{\mathbf{T}_j,\mathbf{R}} = \frac{L_T\hat{P}_{\mathbf{T}_j} + L_R\hat{P}_{\mathbf{R}}}{L_T + L_R} \tag{33}$$

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

$$\hat{P}_{\mathbf{T}_j}(r_k) = \frac{\#\ times\ r_k\ occurs\ in\ \mathbf{T}_j}{L_T} \quad \hat{P}_{\mathbf{R}}(r_k) = \frac{\#\ times\ r_k\ occurs\ in\ \mathbf{R}}{L_R} \tag{34}$$

The method can be implemented directly by building the histogram for all response values. For more details see (Johnson et al., 1998). The number of entries in the histogram (type) would, however, usually prevent us from such an implementation. Having N neurons whose responses are measured across B bins, the number of possible response value 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, it is necessary to define stationary segments of data, and to quantize the possible value of responses. As a result, the optimality promise is abandoned, but the computation becomes feasible.

In the examples presented here, we divided the 600 ms after the reward into 6 stationary 100 ms segments, and computed the histogram for each such segment. Spotting via this method suffered from two crucial problems:

- The highly probable spotted segments were not locked to the reward cue.
- The average detected segment was much lower than the original reward response, i.e. similar to the assumption of flat background (see figure 4).

We assume that the inability of this "type" method to serve as a feasible spotting method has to do with the nature of the data. Although the JS method is optimal given enough data, our case represents a dramatically different scenario. Our data is rather sparse and the use of a parametric model such as the Poisson distribution seems to be highly beneficial. Furthermore, when trying to learn the true nature of the firing histogram, the empty bins (which form the bulk of the data) dominate the JS method.

6 Discussion

In this work we present a novel method for pattern spotting in a harsh environment. Our goal was to be able to spot the 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 features:

- The rate of spike discharge per 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. Furthermore, the average firing rate is similar to the pattern searched.

Due to these difficulties, the common method when 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 have shown elsewhere (Gat & Tishby, 1997), 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 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 produce reasonable results. For example, in a situation where the threshold on the likelihood function assumes a constant rate background, the result was frames of data in which the number of spikes was the smallest.

The core idea of the method presented here, is that spotting of the neuronal pattern is attempted in adversary environment, in which all *shuffles* of a given pattern are equally probable. When implementing this decomposition, 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 can be *analytically* evaluated.

Based on this Gaussian approximation we derive 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 decomposition, shown in figure 9, clearly breaks the log-likelihood distribution into narrower pieces, the mean of each piece depends very weakly, (see Eq. (19)) on the model rate parameters. The type weight, however, as given by Eq. (13) is exponential in the type log-likelihood mean. It is 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 (Gutman, 1989). 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 (PST histogram) of the training data (Johnson et al., 1998). 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, albeit the optimality promise.

Our method was implemented on data from the TAN's (cholinergic interneurons of the striatum), and an attempt was made to spot the neural pattern usually related to the reward mechanism. This paper presented some results of this spotting - a more thorough description of the data with the significance of the results can be found in (Gat et al., 2000). We believe that these results could not have been obtained without this method of spotting the reward pattern on the basis of single trial.

There are several possible extensions to the work presented here. One such extension involves a slight change in the definition of $T(\underline{n})$. In this paper the type $T(\underline{n})$ was defined as a permutation over all the bins in all the cells together. For multi cell recording where one has cells with very different rates, it is better to choose a type where $T(\underline{n})$ is defined as the subset of all vectors where the permutation is being done for each cell *separately*. This definition make use of the fact that the data comes from N neurons from which one has measured for B time bins. In our example, however, we took advantage of the fact that TAN cells are known to respond in a similar manner (both in the response profile and in their firing rates). By doing so we were able to overcome the sparseness of our data, and to enrich our background model. This sparseness is probably the major cause for the failure of the JS divergence method presented here.

A different extension is to change the types used here into types 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. A 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 may be able to cope with more complex spotting cases. When we extend the correlation order, the background becomes increasingly similar to the original pattern under investigation.

In the current work we found it sufficient to produce the concept of adversary background model, and show how it can be obtained for first order. Fortunately this model was quite appropriate for the data in hand, i.e. spotting the reward signal in TAN. We do not rule out the possibility that higher order adversary background model might be needed for different data modeling.

In general we believe that the idea of decomposing the likelihood into subparts could be useful. This could allow us to work in environments in which we can not assume the usual flat distribution of the background. It would enable us to perform reliable spotting without building a specific model. In our case, the decomposition into types seems sufficient, and produced the required results.

Acknowledgments

Enlightening discussions with William Bialek, Moshe Abeles, and Shai Fine are greatly appreciated. We thank Hagai Bergman and Aeyal Raz for providing us with the TAN-cells recordings. This work was partially supported by a grant from the Israeli Science Foundation

References

Abeles, M. (1991). *Corticonics.* Cambridge University Press.

Abeles, M., Bergman, H., Gat, I., Meilijson, I., Seidemann, E., Tishby, N., & Vaadia, E. (1995). Cortical activity flips among quasi-stationary states. *Proc. Natl. Acad. Sci.*, 8616-8620.

Aosaki, T., Tsubokawa, H., Ishida, A., Watanabe, K., Graybiel, A., & Kimura, M. (1994). Responses of tonically active neurons in the primate's striatum undergo systematic changes during behavioral sensorimotor conditioning. *J.Neurosci.*, *14*, 3969-3984.

Bolam, J. P., & Bennet, B. D. (1995). Molecular and cellular mechanisms of neostriatal function. *R.G. Landes Company, Georgetown, Texas.*

Cover, T., & Thomas, J. (1991). *Elements of information theory.* Wiley.

Frost, R. (1916). The road not taken. In *The road not taken.* New York: Henry Holt.

Gat, I., Raz, A., Abeles, M., Tishby, N., & Bergman, H. (2000). *The neural signature of reward expectation signal in the striatum of normal and parkinsonian primates.* (Preprint)

Gat, I., & Tishby, N. (1997). *Comparative study of different supervised detection method of simultaneously record spike trains.* (Preprint)

Georgopoulos, A., Schwartz, A., & Kettner, R. (1986). Neuronal population coding of movement direction. *Science*, *233*, 1416 - 1419.

Graybiel, A. M., Aosaki, T., & A. W. Flaherty, M. K. (1994). The basal ganglia and adaptive motor control. *Science*, *265*, 1826-1830.

Gutman, M. (1989). Asymptotically optimal classification for multiple tests with empirically observed statistics. *IEEE Trans. Info. Th*, *35*, 401-408.

Johnson, D. H., Gruner, C., Baggerly, K., & Seshagiri, C. (1998, February). *Information-theoretic analysis of the neural code.* (Download from http://www-dsp.rice.edu/ dhj)

Kawaguchi, Y., Wilson, C., Augood, S., & Emson, P. (1995). Striatal interneurons: chemical, physiological and morphological characterization. *TINS*, *18*, 527-535.

Leheman, F. (1959). *Testing statistical hypotheses.* New York: Wiley.

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Trans. Info. Th.*, *37*, 145-151.

Raz, A., Feingold, A., Zelanskaya, V., Vaadia, E., & Bergman, H. (1996). Neuronal synchronization of tonically active neurons in the striatum of normal and parkinsonian primates. *J.Neurophysiol.*, *76*, 2083-2088.

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

Figure 1:

Spike trains of a single trial, and the 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.

Figure 2:

A PSTH model of spike pattern. The histogram of 4 cells recorded simultaneously is presented. The activity of the cells was measured in bins of 20 ms. The estimation of the firing rate estimation is given by bars whose height is proportional to the rate. The rate units are counts per bin.

Figure 3:

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

Figure 4:

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, under the assumption of constant background.

Figure 5:

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 an 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.

Figure 6:

The spotting via adversary background model shown in the probability simplex. A. shows the regular spotting mechanism using the probability simplex. We have two models - marked as **P** and **Q**. Given a set of data points, each is spotted according to the LRT of the models. This 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.

Figure 7:

The distributions of the model log-likelihood $\log p_{\underline{\lambda}}(\underline{n})$ for samples generated by a non-homogeneous Poisson model. The histogram of the simulated data (given in bars) is compared with the estimated normal distribution (solid line).

Figure 8:

Histogram of the model log-likelihood in one sample type with the Gaussian approximation.

Figure 9:

The decomposition of the model log likelihood into sample types. The different line styles represent different families of sample types. The dotted sample types are types which do not include more than one count in each bin. The solid sample types hold exactly one bin with two counts, and the dashed sample types have 2 bins with 2 counts.

Figure 10:

It is possible to detect the neural signature of reward even when it is not locked to external reward. 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. Reward was hinted at zero lag time (reward cue). The bin size used is 20 ms (130 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 do not include segments from the vicinity of the reward cue. The bin size and Y-axis as in A.

Figure 11:

The frequency of the neural reward signal increased after the reward was indicated. Time histogram of the reward-like response. Each bar represents the number of times the reward signal occurred in a certain delay with respect to the reward cue. The bin size used was 800 ms.

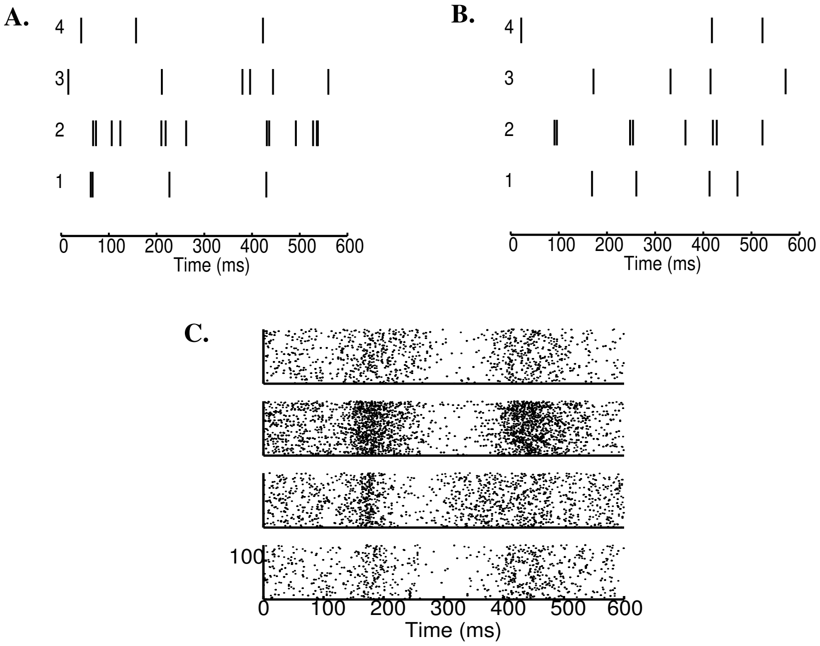

Figure 1: Gat, Manuscript: 2105

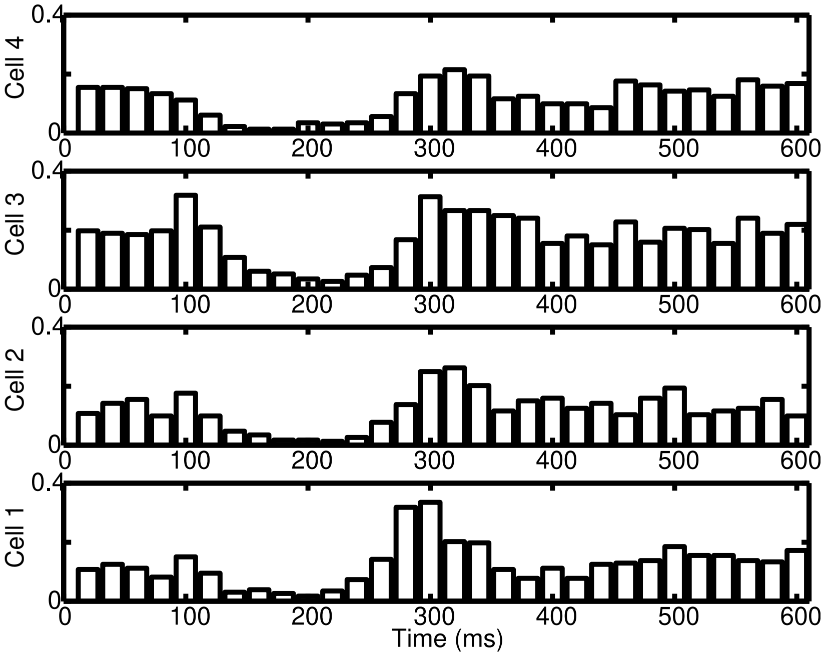

Figure 2: Gat, Manuscript: 2105

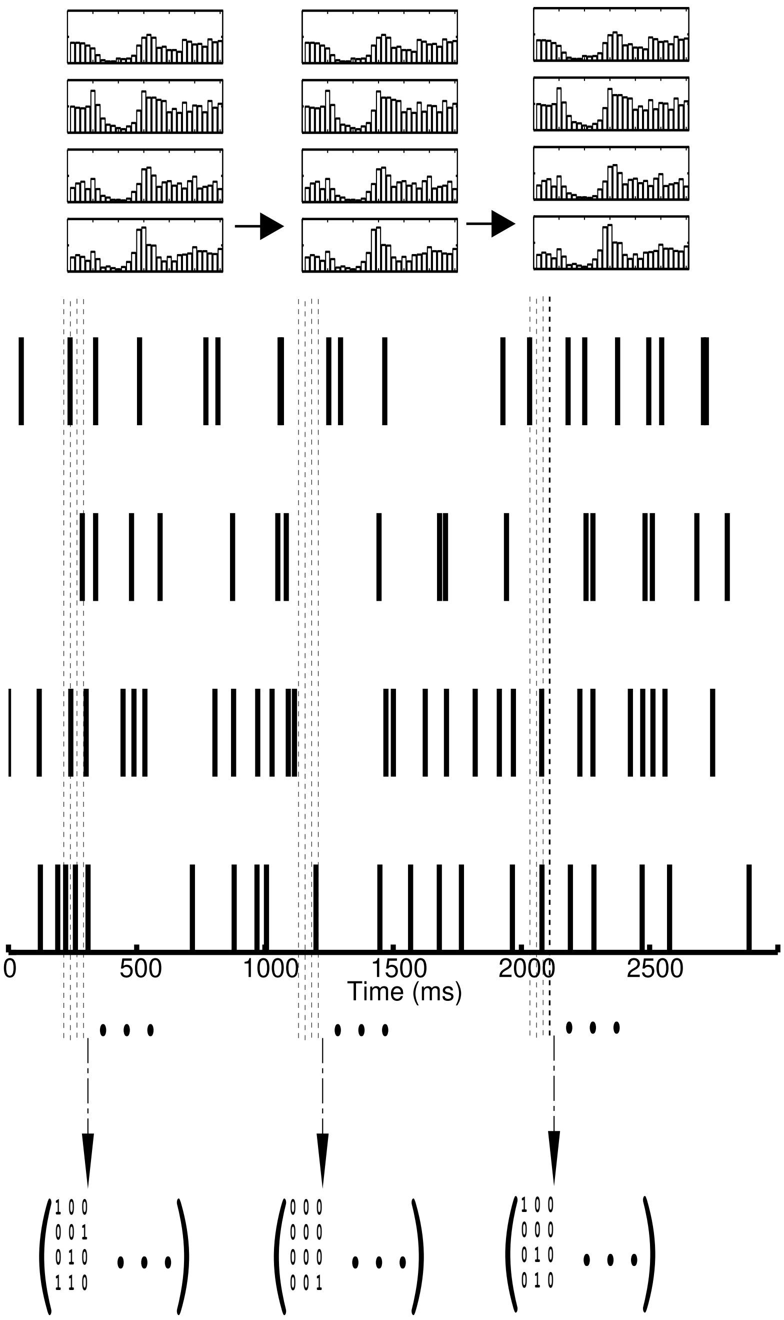

Figure 3: Gat, Manuscript: 2105

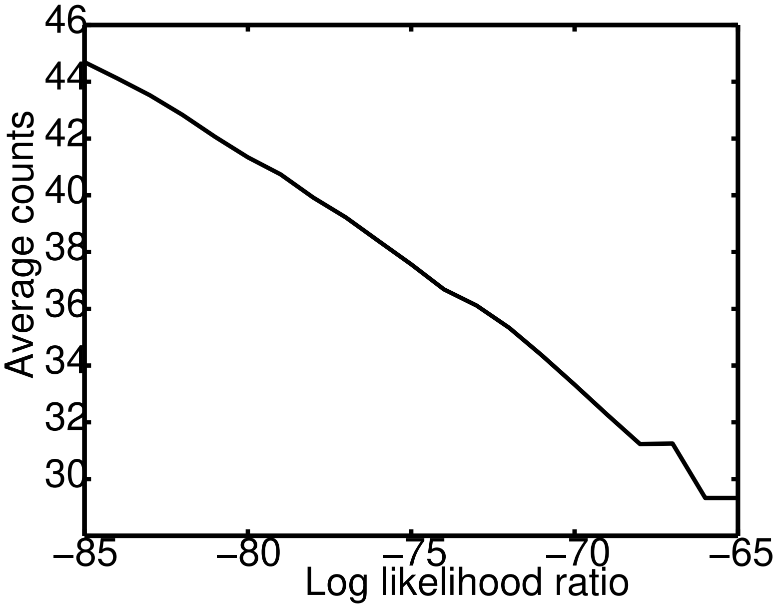

Figure 4: Gat, Manuscript: 2105

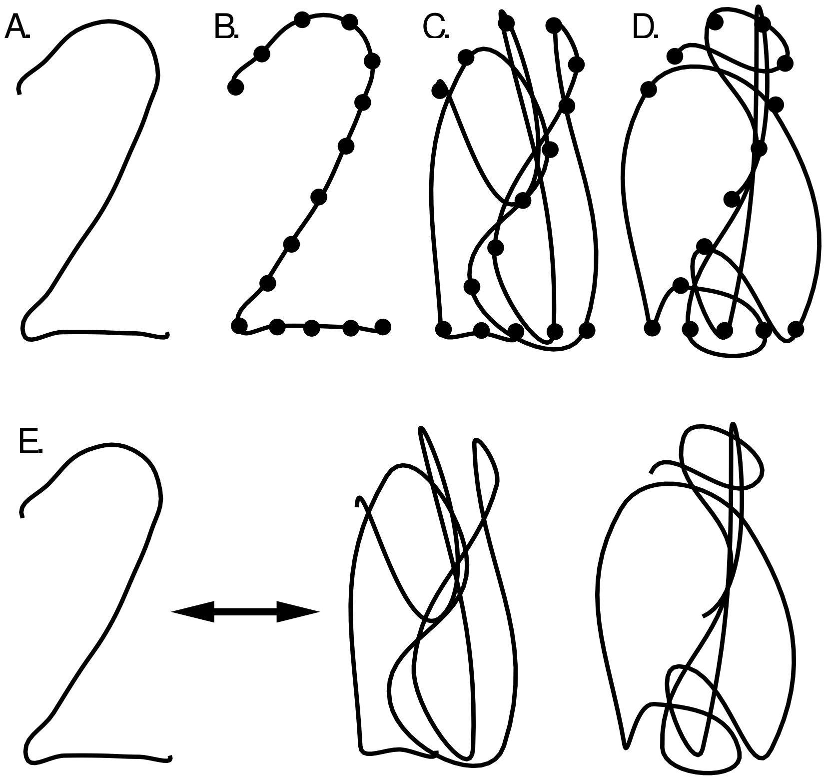

Figure 5: Gat, Manuscript: 2105

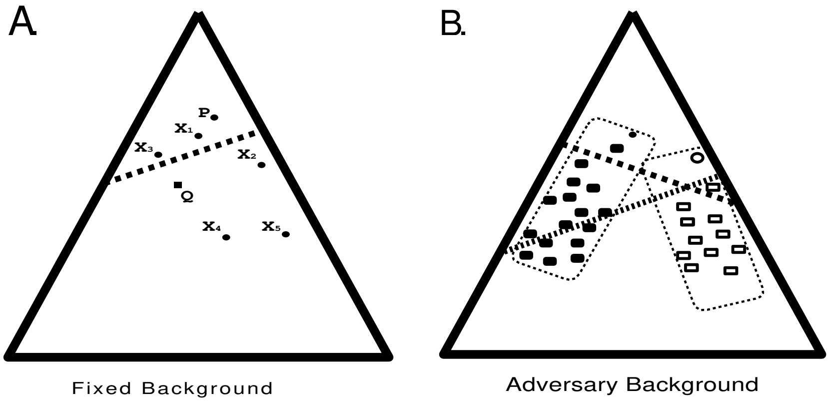

Figure 6: Gat, Manuscript: 2105

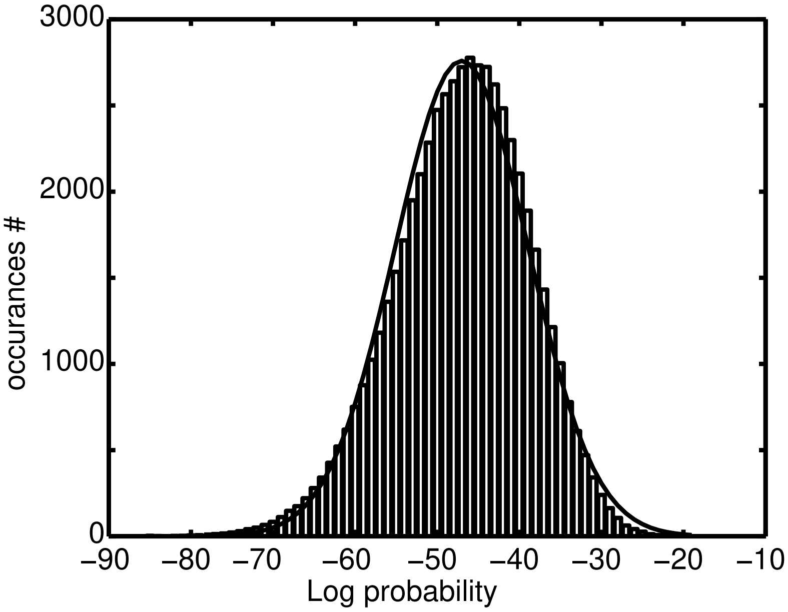

Figure 7: Gat, Manuscript: 2105

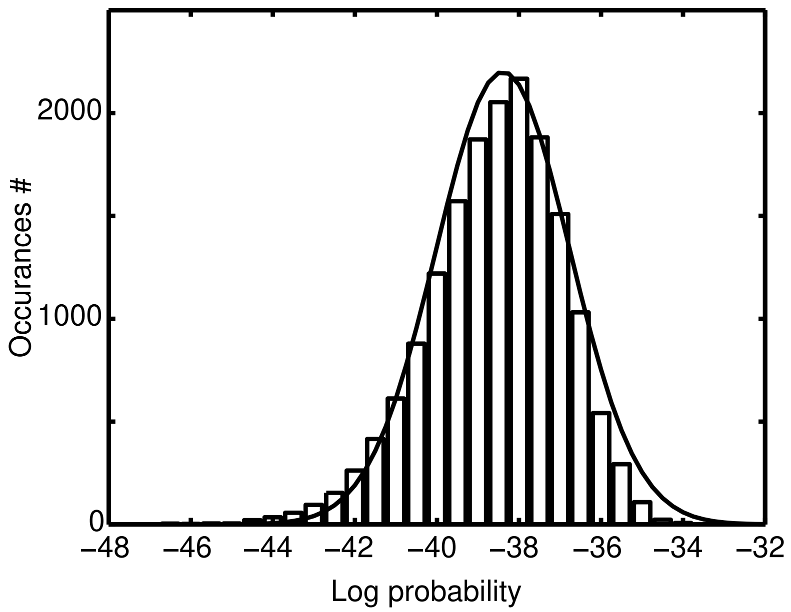

Figure 8: Gat, Manuscript: 2105

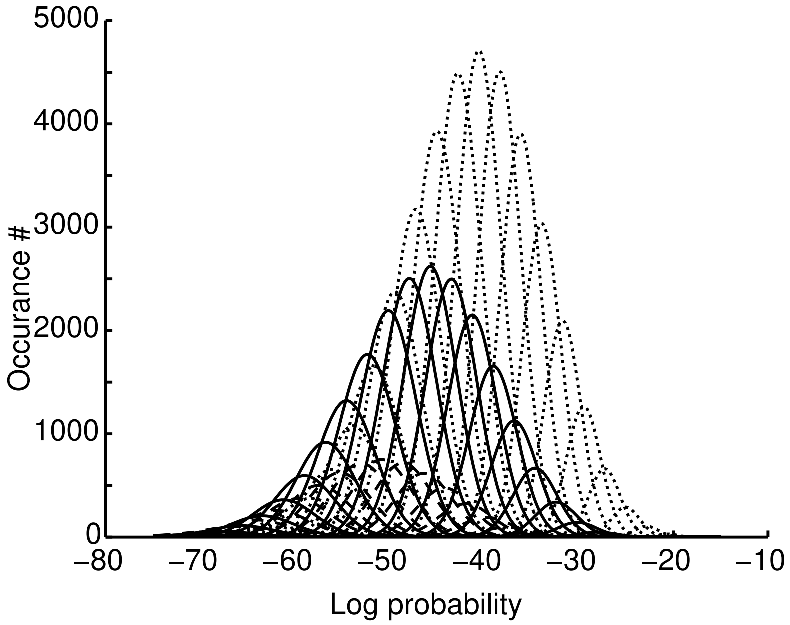

Figure 9: Gat, Manuscript: 2105

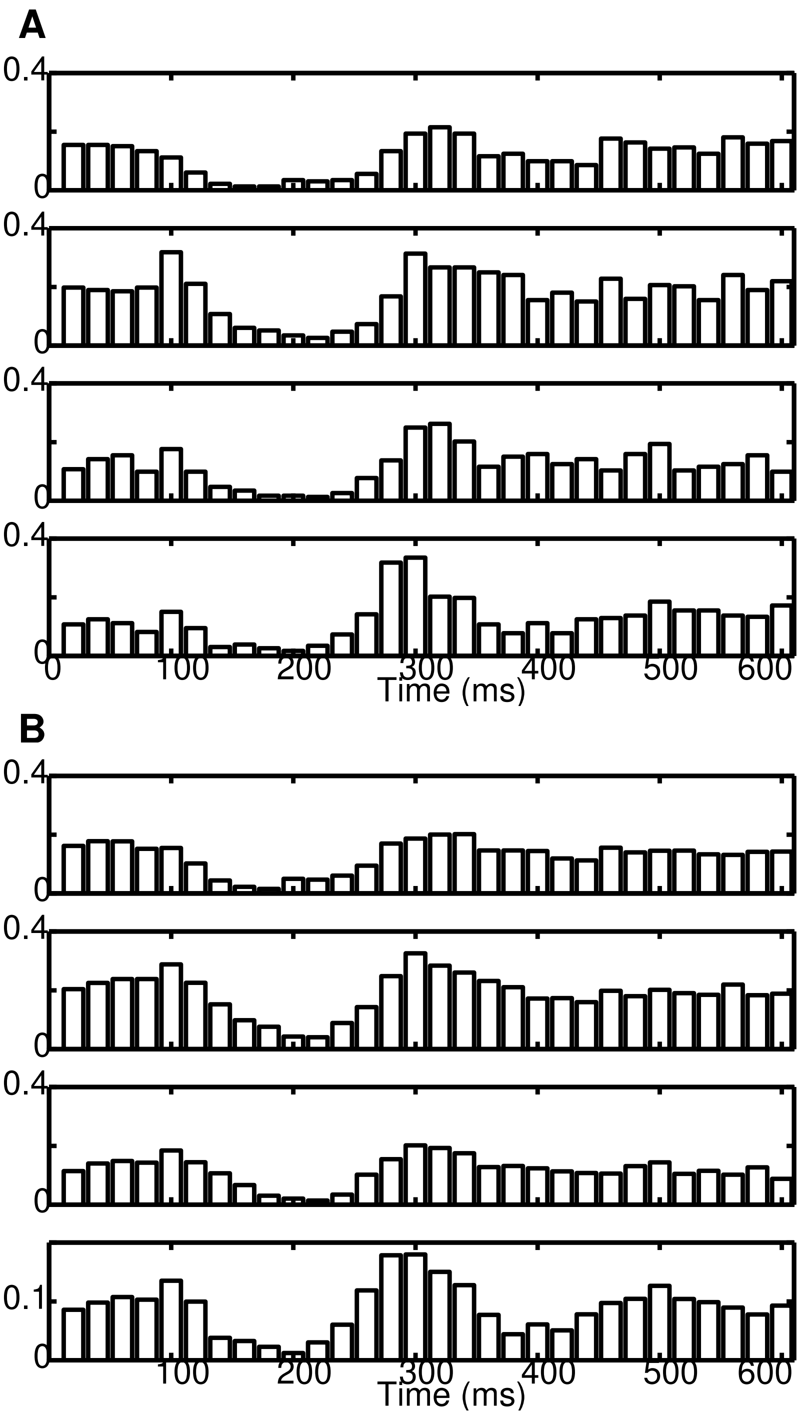

Figure 10: Gat, Manuscript: 2105

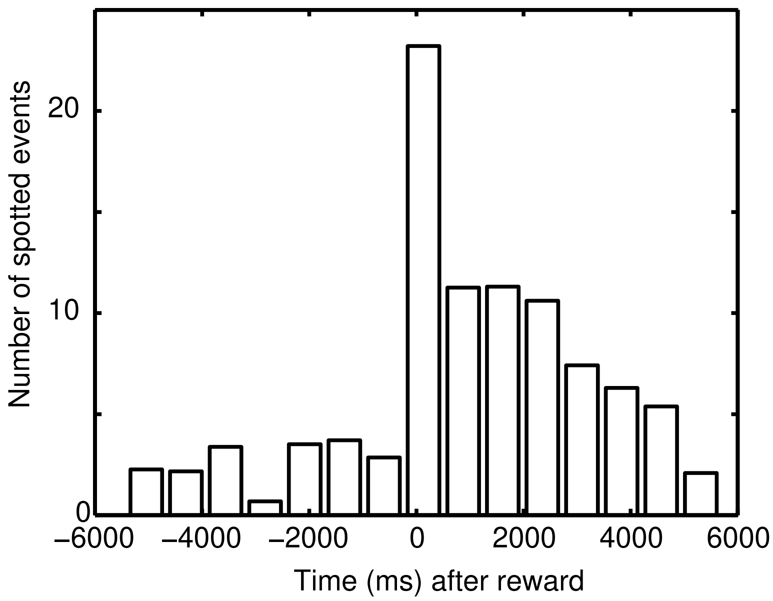

Figure 11: Gat, Manuscript: 2105