Hidden Markov Modeling of simultaneously recorded cells in the Associative cortex of behaving monkeys

Itay Gat†, Naftali Tishby† § and Moshe Abeles‡

† Institute of Computer Science and Center for Neural Computation Hebrew University, Jerusalem 91904,Israel

‡ Haddasa School of Medicine and Center for Neural Computation Hebrew University, Jerusalem 91904, Israel

Abstract.

A widely held idea regarding information processing in the brain is the cell-assembly hypothesis suggested by Hebb in 1949. According to this hypothesis, the basic unit of information processing in the brain is an assembly of cells, which can act briefly as a closed system, in response to a specific stimulus. This work presents a novel method of characterizing this supposed activity using a Hidden Markov Model. This model is able to reveal some of the underlying cortical network activity of behavioral processes. In our study the process in hand was the simultaneous activity of several cells recorded from the frontal cortex of behaving monkeys. Using such a model we were able to identify the behavioral mode of the animal and directly identify the corresponding collective network activity. Furthermore, the segmentation of the data into the discrete states also provides direct evidence for the state dependency of the short-time correlation functions between the same pair of cells. Thus, this cross-correlation depends on the network state of activity and not on local connectivity alone.

1. Introduction

The cell-assembly hypothesis, first introduced by Hebb in the late forties, has since gained support in later works. According to this hypothesis, *"It is proposed first that a repeated stimulation of specific receptors will lead slowly to the formation of an "assembly" of association-area cells, which can act briefly as a closed system after stimulation has ceased; this prolongs the time during which the structural changes of learning can occur and constitutes the simplest instance of a representative process (image or idea)"* [Heb49]

This hypothesized cell-assembly may include thousands of cells in a highly interconnected network. The cell-assembly is further characterized by stronger than average connections between the cells which comprise it. The organization of the cell-assembly is achieved via these strong connections, i.e. there is a mechanism that creates these assemblies by strengthening the connection between cells. Hebb further hypothesized that this mechanism is related to the ordered activation of cells, that is, whenever two cells are activated one after the other, the connection between them is reinforced. The cell-assembly is a group of interconnected cells, whose participation in

§ To whom correspondence should be addressed.Email: tishby@cs.huji.ac.il

the cell-assembly is not determined by proximity. One cell may participate in several cell-assemblies, and it is possible for cells which are next to each other not to be part of the same cell-assembly.

The definition of a cell-assembly is, therefore, more of a functional than an anatomical one, as the cell-assembly is not defined by specific physiological locations. The information processing in the cortex is dynamic in time, and possibly tends to alternate between different activation states of one cell-assembly, and between activations of several cell-assemblies. The same cell-assembly may be utilized in different computational tasks. Thus, a typical computation process consists of a sequence of activations of these functional units, and will henceforth be referred to as a sequence of *cognitive states*.

The notion of the cell-assembly has been formalized since Hebb first suggested it [vdM85, Pal82]. Furthermore, the cell-assembly hypothesis matches the present knowledge about the anatomical structure of the cortex, especially the structure of the high cortical areas [GBA89, BS91]. The cell-assembly hypothesis shifts the focus from the single neuron to the complete network activity. This view has led several laboratories to develop technology for simultaneous multi-cellular recording from a small region in the associative cortex [Abe91, Kru83]. Extracellular electrophysiological measurements have so far obtained simultaneous recordings from just a few randomly selected neurons (about 10) in the cortex, a negligible number compared to the size of the hypothesized cell-assembly. These are recordings of simultaneous firing activities of randomly selected cortical neurons of a *behaving animal*. *Behaving animal* is the term used to describe a conscious animal engaged in a specific, predefined, activity. The recorded discretized action potentials are called henceforth the *spike trains* of cells.

In order to process this multi-cellular recording information and to relate the data to the hypothesized cell-assembly, several techniques have been developed. The best known of them involves computing the pair-wise cross-correlation function between cells [PGM67, PAG88].

This technique analyzes the spike trains of a pair of cells for possible temporal relations, by computing the cross-correlation function of the two spike trains. Departures from background in the correlogram are taken as an indication of *functional connectivity* between the two cells [AGHP89, VABL91]. The term *functional connectivity* (also referred to as *effective connectivity*) [MPS66, PGM67] is used in this context because of the cell-assembly hypothesis which implies that this correlation is due to the activity of the whole cell-assembly.

The idea of applying hidden Markov models (HMM) to cortical data was first introduced by us in 1992 [GT93], where we showed the basic idea of using multivariate Poisson HMM for multi-electrode recordings. Following our work, Radon *et al* [RBDK94] applied the same technique to data recorded from the visual cortex and obtained similar segmentation of the data, with little concrete biological conclusions. Later on, Seidemann *et al* [SAMV96] used a similar technique, applied to the same experimental data, in a finer temporal resolution. While their work supported the general picture we presented, it showed sharper state flips by allowing state transition every 1ms. It was limited, however, to very short segments of the data following an external stimulus, and the HMM used was essentially a left-to right model, unlike our more general ergodic unsupervised approach. The two works give a consistent picture of flips between collective cell activity states which clearly depend on the behavior and activity of the behaving animal [ABG+95].

The main contribution of the present work, in addition to the detailed exposition of our technique, is in the relationship found between the segmentation of the data into states of collective activity and the pairwise correlations between the cells. Whereas the HMM segmentation is based on the average firing rate of each cell, measured in temporal resolution of 100ms, the pairwise correlograms were measured in a fine resolution of less than 10ms. The fact that we find cross-correlations that clearly vary from state to state, for the *same pair of cells* is the first direct evidence that the HMM states indeed reflect different underlying collective activity. Furthermore, it supports the idea that the cross-correlations depend on the collective cell-assembly activity and not only on the synaptic connection between the cells, by itself in important biological consequence of our analysis.

The next section reviews the biological origin of the data and the current experimental methodology used so far for this data. In section 3 we describe our new statistical modeling method , and in section 4 – the details of our hidden Markov model. The results of our analysis are given in section 5, with emphasis on the evidence found for the different activation of the cell assemblies. A discussion and suggested further work is given in section 6.

2. Biological methodology

2.1. Origin of the data

Modern technology allows the parallel recording of spike trains from more than one cell. This allows us to investigate the relationship between several cells and to relate their activity to the overall network activity. The data used in the present analysis were recorded from the frontal cortex (FC) of Rhesus monkeys, in the Higher brain function laboratory at the Hadassah Medical School. These experiments were conducted by Y. Prut, I. Halman and H. Slovin as part of their Ph.D work under the supervision of M. Abeles, H. Bergman and E. Vaadia.

The FC has been traditionally divided into three broad regions: the motor cortex (MC), the premotor cortex (PMC) and the prefrontal cortex (PFC). The MC deals directly with the voluntary muscles. The PMC is assumed to deal with the preparation of movement and the ability to develop an appropriate strategy for movement [Ghe88]. The PFC is considered to be a “higer” association area, i.e., an area which is not directly connected to sensory or motor systems, and has been divided to numerous subregions. 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"* [Fus84]. This area is also involved in the composition and execution of plans [Bru88]. Two of these areas were investigated in this work, the PMC and the PFC that lies rostral to it[Ghe88, PY85]. These areas are characterized by multiple connections with many other cortical areas, most of which are associative areas.

In this work, recordings were taken from two Rhesus monkeys (*Macaca Mulatta*), trained to perform a spatial delayed-response task, in which they had to switch between two different behavioral modes. A diagram of these behavioral modes is presented in figure 1. This task consisted of the following stages:

(i) The monkey initiated the trial by pressing a central key, and a fixation light was turned on in front of it.

(ii) After 3-6 seconds, a visual stimulus appeared, in the form of a cue light, either on the left or on the right. The stimulus was presented for 200 ms.

(iii) The stimulus was followed by a delay period of 1-32 seconds, in factors of 2 (i.e. 1,2,4,8,16,32 seconds).

(iv) Then, the fixation light was dimmed, and the monkey was required to touch the visual cue light ("Go" mode), or keep its hand on the central key regardless of the external stimulus ("No-Go" mode).

(v) The monkey was rewarded for the correct behavior with a drop of juice.

After 4 such correct trials, all the lights in front of the monkey were turned on for 4 seconds, signaling to the monkey to change its behavioral mode - so that if it had started in the "Go" mode it now had to switch to "No-Go" mode, and vice versa.

The training period for each monkey lasted 2-4 months. At the end of the training, the monkeys were able to perform this complicated task with low error rates (around 10%).

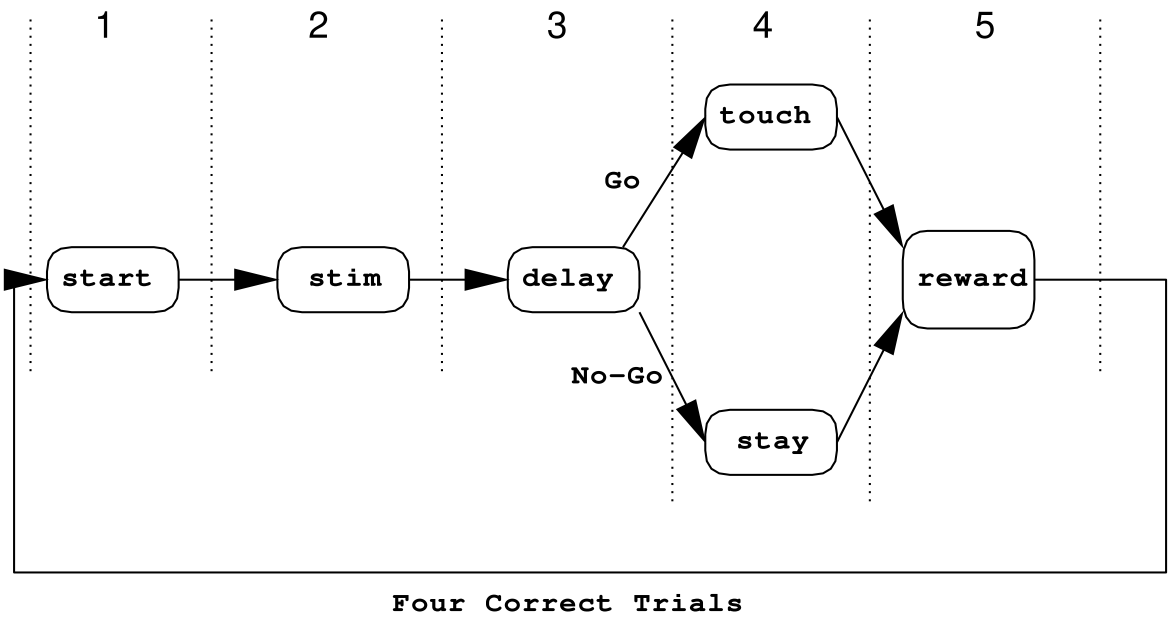

Figure 1. A block diagram of the behavioral modes of the monkey

At the end of the training period, the monkeys were prepared for electro-physiological recording. Surgical anesthesia was induced. A recording chamber was fixed above a hole drilled in the skull to allow for a vertical approach to the frontal cortex. After recovery from surgery, recording sessions began, and the activity of cells in the cortex was recorded while the monkey performed the previously-learned routine. The recording was extra-cellular, and was performed using an array of 6 microelectrodes. The output of the electrodes was fed into pre-amplifiers, and than band pass filtered. Several spike detection devices [Pru95] were used to detect and identify the action potential discharge of the cells (the *spike trains*). Each recording session was held on a different day and typically lasted 2-4 hours. The output of these recordings is the discrete times of the spike trains. Typically at each such recording there were 5-8 such spike trains. It is important to note that the results of these recording were carefully examined in order to reduce artifacts due to insufficient separation between spike shapes, unstable electrodes, etc. It is also important to notice that these recordings do not reflect the learning process, but rather the cortical activity of the well-trained monkey while performing its task.

The quality check described above, was applied to 8 recording sessions, out of which 6 were found to include *sufficient data* for further processing. *Sufficient data* in this context was defined as follows:

(i) More than 5 well identified and isolated cells.

(ii) The intervals chosen held more than 5000 *seconds* of stable and stationary recording of all neurons.

Following the quality check, the data were divided into trials made in the "Go" mode and the "No-Go" mode. From this point on, the spike trains of these behavioral modes were processed separately. The result of this selection can be seen in table 1.

Table 1. Presentation of the recording sessions used throughout the experiments. For each recording session, the number of well-isolated cells used is indicated, as is the total length of the session (in seconds) and the cortical area from which the data were recorded. These time tables are presented separately for the "Go" mode and the "No-Go" mode.

Session name	# Cells	# States	"Go" Length	"No-Go" Length	Total Length	Area
bl16	8	6	3907	4097	8004	PMC
bl25	8	7	6202	5131	11333	PMC
bl29	8	8	2557	2508	5065	PMC-MC
bl39	6	7	5987	5864	11854	PMC
cl77	6	6	9946	6566	16512	PFC
cl85	8	6	3908	4654	8562	PFC
Total	44	40	32507	28820	61327	

Notice that the cells were not chosen according to any criterion of responsiveness - some of the cells chosen were responsive to an external stimulus while others showed only weak responses to the stimulus.

2.2. Two widespread analysis methods

2.2.1. Peri-Stimulus time histogram The usual way of computing the activity of a single cell is by the peri-stimulus-time histogram (PSTH), which is an average measure of the firing rate probability relative to the stimulus presented. In high cortical areas it was found that in order to attain a reasonably smoothed PSTH it was necessary to average the activity over many repeated activations (typically at least 30 to 50). This fact by itself shows that the requisite information cannot be coded in the activity of a single cell alone [SHS82]. Still, the PSTH is a valuable tool for measuring the firing rate of cells, and can give a flexible estimation in the time domain.

An example of PSTH and the raster display which was the source of computation is shown in figure 2. In this example the raster display shows 233 trials aligned horizontally according to some pre-defined conditions. Each trial is shown as one line in this display and the dots in that line represent the time in which the cell emitted action potentials (spike trains). The PSTH is drawn above the spike trains, and in this example, a bin width of 50 ms was used, i.e. in the 9000 ms presented in this figure, 180 such bins exist. The computation process involves counting the spikes in each such bin and dividing the result by the number of repetitions for each bin. The result is transformed into units of spikes-per-second by dividing it by the time duration of the bins. The histogram is automatically scaled according to the maximum firing rate value, which is given on the upper left side of the histogram, in units of spikes-per-second. In this example, the maximum firing rate is 4 spikes-per-second.

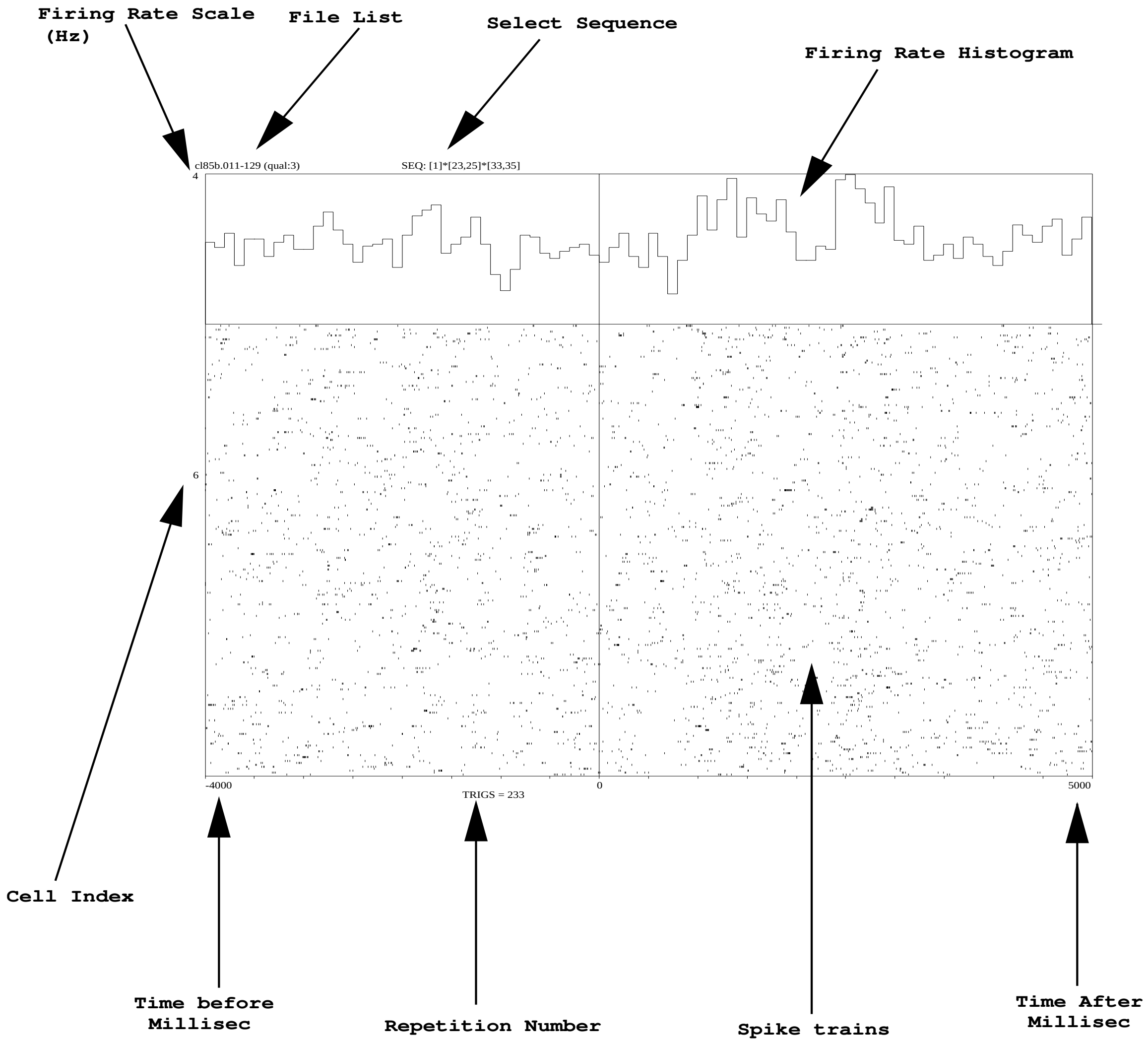

Figure 2. An example of a raster-display, with PSTH drawn above. The PSTH in this example shows no significant response to the stimulus.

2.2.2. Cross-correlations analysis Another important tool used to analyze the spike trains of cells is the cross-correlation tool. The temporal cross-correlation function between cells measures the similarity of the spike trains of two cells and can be computed within bins (windows) of variable length. The figures representing the cross-correlation function with a specific bin width are called *cross-correlograms* [VABL91].

The temporal cross-correlogram between two cells i and j at time bin l is defined as follows:

$$R_{ij}(l) = \frac{1}{N_t \Delta t} \sum_{t=1}^{N_t} S_i^{'}(t) S_j^{''}(t+l) \ . \tag{1}$$

Where $S_i^{'}(t)$ is the event that cell i fired in the time interval $(t, t+0.001)$, and $S_j^{''}(t+l)$ is the event that cell j fired in the time bin $(t + l\Delta t, t + (l+1)\Delta t)$. The absolute time mentioned above is in seconds, i.e. 0.001 is 1 ms. N_t is the number of spikes of cell i (that are used as "triggers" for the averaging) and Δt is the time bin width.

The shape of the cross-correlogram provides indications for the *functional connectivity* between neurons. Typical shapes of cross-correlograms feature peaks or troughs near the origin. When the peaks or troughs extend to both sides of the origin, they are usually interpreted as reflecting a shared input to the two neurons. Narrow peaks or troughs, restricted to one side of the cross-correlogram, are usually interpreted as reflecting a direct synaptic interaction.

An example of a cross-correlogram is presented in figure 3, and contains the following information:

- The indexes of the cells used to compute the correlogram.
- The average firing rate of each cell.
- The time scale of the cross-correlogram is shown on the abscissa. In this example, from -500 ms to +500 ms
- The firing rate is shown on the ordinate in units of spikes-per-second.
- The total duration of spike trains used to compute this correlogram.
- Bin size of Gaussian smoothing applied to the counting of the correlation function.
- An average line which is the average firing rate of the second cell.
- Significance lines with a confidence level of 0.005 for each side of the average, computed on the *tails* of the correlation function, i.e. from -500 to -400 ms, and from +400 to +500 ms.

The correlation seen in this figure is clearly non-flat and shows a hump around time zero. This hump is an indication of shared input between the cells under investigation.

2.2.3. Limitations of these analysis methods Both the PSTH and cross-correlation tools suffer from the need to align a large number of repetitions of data together in order to get reasonable statistics. This amounts to *labeling* of segments of the data, as samples of specific activity. In learning theory, algorithms that use labeled data are known as *supervised* learning algorithms, while procedures that use unlabeled data are called *unsupervised*. All the methods of analyzing the data collected from the cortex described above, were processed, therefore, in a supervised manner. The labeling used in these processes was the timing and modes of the external events of the experiment.

The basic advantage of using an unsupervised method is that it enables the direct modeling of the data without making assumptions as to the nature of the data. For example, the modeling techniques shown above assume a direct connection between

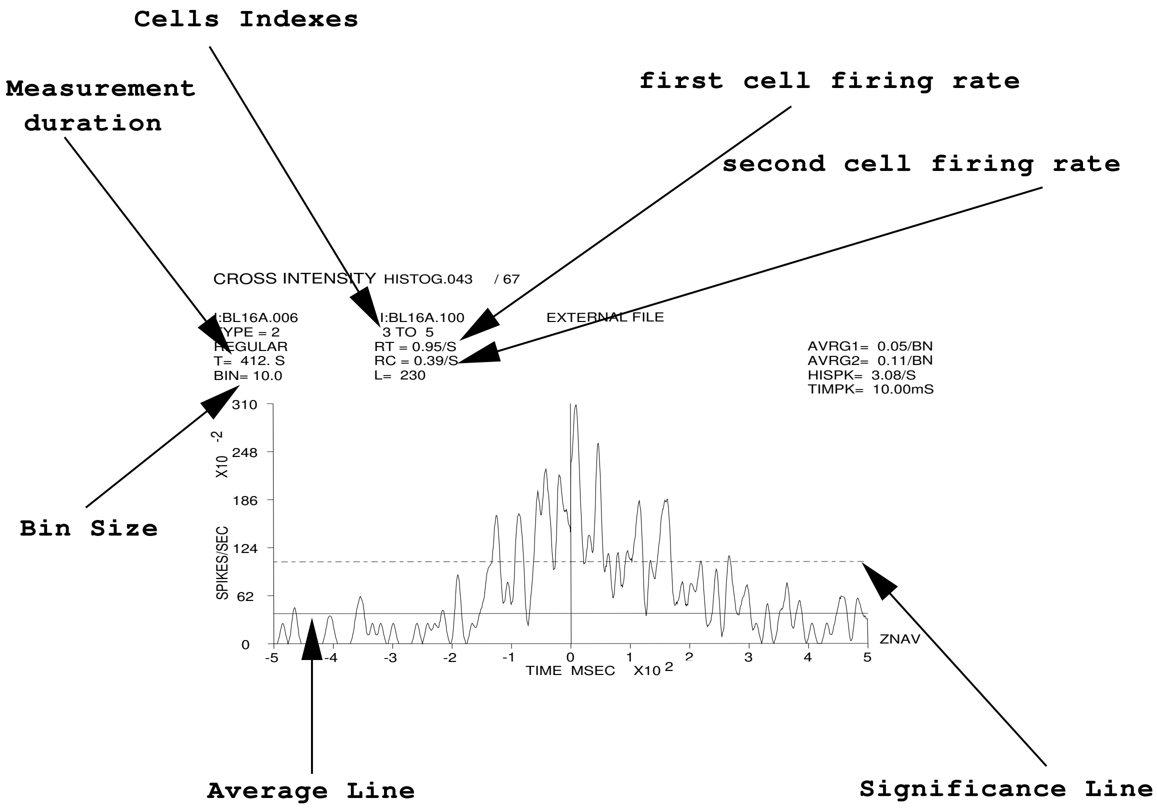

Figure 3. An example of a cross-correlation function (correlogram) of two cells.

events of the experiment (stimuli to the animal, or behavioral actions) and the recorded cortical data, an assumption which is not always justified. An attempt to model that data in an unsupervised manner, on the other hand, may reveal a direct connection between the recorded **spike trains** of several cells and the collective activity of the recorded cortical tissue.

Furthermore, the PSTH and cross-correlation methods fail to provide a sufficient quantitative measure of the *collective* cell assembly activity. Clearly such a small number of recorded cells can give only a glimpse into the full cell-assembly activity. These spike trains should rather be considered as a vector of stochastic measurements *conditioned* on the underlying collective network state.

3. Statistical modeling of spike trains

The limitations of current analysis methods motivated the data analysis paradigm of this work. This paradigm considers the recordings as the stochastic observations of a hidden state. This setup fits precisely the framework of the hidden Markov models (HMM), originally developed for speech recognition and analysis.

3.1. *Hidden Markov Modeling*

HMM is a stochastic modeling technique for the study of complex time series [Por88, CT91, Rab89]. It is essentially a stochastic function of a Markov chain, i.e., a conditioned probability distribution on the possible output observations is attached to each state of the Markov process. 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 [LRS74, RJLS86].

The general HMM is constructed of the following elements:

- The states of the Markov process change according to specific transition probabilities.
- The states cannot be observed directly. Only the stochastic output of the state at each point in time is observable. The resulting model is a double embedded stochastic process with an underlying stochastic process that is not observable.
- The available data is the a stochastic output of the Markov process. For convenience reasons, this output is usually transformed into a *feature space*, in which the essential information found in the data is preserved, while its size is reduced. The transformed data are referred to as the the *observations* of the model.

One type of such a process is the first order HMM, in which the transition between states is governed by the current state and the next state only. An example of such a process can be seen in figure 4. The double-embedded feature of that modeling is shown in that figure in the following manner: the states are shown to be hidden by the dash-dotted horizontal line presented in the upper part of the figure, where the observation of each state is shown as a dotted arrow emerging from the state and crossing that horizontal line.

A HMM is characterized, therefore, with the following elements:

- N, The number of states in the model.
- M, The number of distinct observation symbols, i.e., the discrete alphabet size. In the case of continuous observation, M denotes the number of parameters of the distribution function.
- A, The state transition probability distribution matrix.
- B, The observation symbol probability distribution matrix. In the case of continuous observations, this matrix is replaced by the distributions functions.
- π, The initial state distribution vector.

Given appropriate values of N,M,A,B and π, the HMM can be used as a generator to give an observation sequence $O = o_1, o_2, o_3 \cdots o_T$, and an un-observable sequence of states $Q = q_1, q_2, q_3 \cdots q_T$. where each observation O_t is one of the symbols, and T is the number of observations in the sequence.

3.2. *The basic issues of hidden Markov modeling*

Usage of the HMM require the solution to the following computational problems [Rab89].

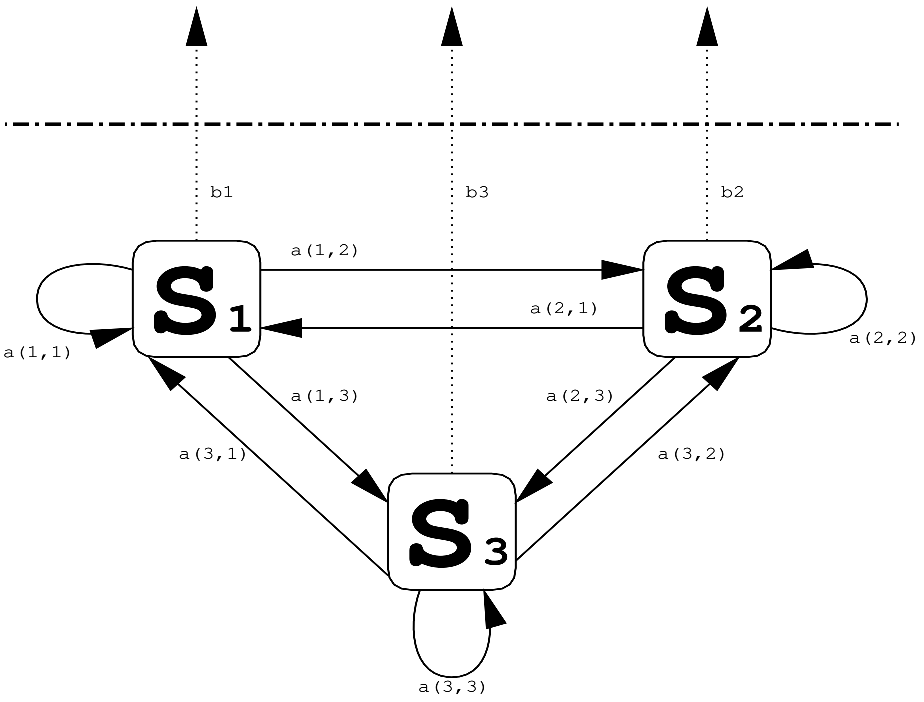

Figure 4. A simple Hidden Markov Model with 3 states. The S_i are the different states of the model, the $a(i,j)$ are the transition probabilities from state S_i to state S_j, and the b_j are the observations probabilities in each state S_i.

(i) Given the observation sequence O and a model $\Lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\Lambda)$, i.e. the probability of the observation sequence, given the model? This problem is referred to as the *evaluation problem*.

(ii) Given the observation sequence O and a model $\Lambda = (A, B, \pi)$, how do we choose a corresponding state sequence Q which is optimal in some meaningful sense (i.e., best "explains" the observations)? This problem is referred to as the *decoding problem*.

(iii) How do we adjust the model parameters $\Lambda = (A,B,\pi)$ to maximize $P(O|\Lambda)$ (likelihood function)? This problem is referred to as the *learning problem*, or the *estimation problem*.

Efficient solution of these problems is critical for the practical implementation of the model. HMM's can be used for classification. When several alternative models exist and an observation sequence is presented, the evaluation problem is used to determine the most probable model for the data. After finding the best model for the data, the solution to the decoding problem helps reveal the underlying structure of the observations, i.e. the hidden states as a function of time. The learning problem maybe solved by training the models on given observation sequences.

Both evaluation and decoding can be solved in a time linear in the length of the observations and quadratic in the number of states of the model. This implementation is based on *dynamic programming* [Bel57], which is an algorithmic technique used for the analysis of sequential decision problems based on Bellman's principle of optimality.

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 [BD62]. The use of this technique assures the correctness and optimality of the solutions.

3.3. Estimation of HMM parameters

The estimation problem (finding the model parameters which maximize the likelihood function), on the other hand, cannot be solved analytically in a time polynomial in the observation length. It is possible, however, to find local maxima of this function, and two of the algorithms performing this task are:

(i) The Baum-Welch method, also known as the **EM** (expectation maximization) method [DLR77, BP66, Bau72].

(ii) The Segmental K-means technique [LRS74, JR90, RWJ86].

The Baum-Welch algorithm provides an estimate which locally maximizes the likelihood function of the given observation sequence, and the segmental K-means algorithm provides an estimate which locally maximizes the joint likelihood of the observation sequence and the most likely state sequence.

The likelihood function of a continuous observation HMM can be rewritten as:

$$P(O|\Lambda) = P(\overline{o}|\overline{\lambda}) = P(o_1, o_2, \cdots, o_K|\lambda_1, \lambda_2, \cdots, \lambda_K) = \frac{N!}{\prod_{i=1}^{K} o_i!}\prod_{i=1}^{K} \lambda_i^{o_i} \quad (2)$$

where

$$N = \sum_{i=1}^{K} o_i \ . \quad (3)$$

Taking the logarithm of both sides, and using Stirling's approximation, we get:

$$\begin{aligned}
&\log P(\overline{o}|\overline{\lambda}) = N\log(N) - N - \sum_i o_i \log(o_i) + \sum_i o_i + \\
&\sum_i o_i \log(\lambda_i) + o(N) = \\
&= N\log(N) - \sum_i o_i \log(o_i) + \sum_i o_i \log(\lambda_i) + o(N) \ . \quad (4)
\end{aligned}$$

(due to equation 3)

Defining a new set of observations p_i as:

$$p_i \stackrel{def}{=} \frac{o_i}{N} \ , \quad (5)$$

equation 4 can be rewritten as:

$$\begin{aligned}
&-\sum_i o_i \log\left(\frac{o_i}{N}\right) + \sum_i o_i \log(\lambda_i) + o(N) = \\
&-N\left[\sum_i p_i \log\left(\frac{p_i}{\lambda_i}\right)\right] + o(N) \ . \quad (6)
\end{aligned}$$

The probability measure of equation 2 can be written as:

$$P(\overline{o}|\overline{\lambda}) \approx e^{-ND[p||\lambda]} \ , \tag{7}$$

where $D[p||\lambda]$ is defined as:

$$D[p||\lambda] \stackrel{def}{=} \sum_{All x} p(x) log\left(\frac{p(x)}{\lambda(x)}\right) \ . \tag{8}$$

This expression is called the Kullbak-Liebler divergence measure (KL) [Kul59]. It is also sometimes referred to as the *cross-entropy* of the distributions [ME91], *discrimination information* or the *information divergence.*

An example of the different divergence measures may be seen in the following example. It can be shown that the divergence of the Gaussian distributions with the same variances, turn out to be the Euclidean distance measure:

$$D[\mathcal{N}(\mu_1, 1)||\mathcal{N}(\mu_2, 1)] = (\mu_1 - \mu_2)^2 \ , \tag{9}$$

whereas for Poisonian distributions the divergence measure is quite different:

$$D[\mathcal{P}(\lambda_1)||\mathcal{P}(\lambda_2)] = \lambda_2 - \lambda_1 + \lambda_1 \log \frac{\lambda_1}{\lambda_2} \ . \tag{10}$$

It should be noted that the later divergence measure is of non-symmetric nature, in contrast with the normal Gaussian example. This fact is the outcome of the non-symmetric nature of the likelihood function.

4. Application of the model

The computation of the hidden Markov model was carried out in three stages. The first one consisted of the transformation of the spike trains into the firing rate feature space. The second and the third stages were the computation of a limited model and a full model. The outcome model was furthermore used to compute the pair-wise cross-correlation.

4.1. Transforming the spike train into firing rates

In this implementation of the hidden Markov model we assume that each of the spike trains are distributed according to a Poisson distribution [Abe91]. The distances between spikes are distributed exponentially and their number in each time frame, n, depends only on the mean firing rate λ, through the distribution

$$P_\lambda(n) = \frac{e^{-\lambda}\lambda^n}{n!} \ . \tag{11}$$

The simultaneous recorded spikes trains are modeled as a multivariate Poisson process, such that:

$$P_{\overline{\lambda}}(\overline{n}) = \prod_{i=1}^{N} \frac{e^{-\lambda_i}\lambda_i^{n_i}}{n_i!} \ . \tag{12}$$

where N is the number of cells in the experiment.

The spike trains of the cells were therefore transformed into the firing rates feature space. This transformation was achieved by first slicing the spike trains into consecutive over-lapping segments. The number of spikes in each such segment was used as the observation of the hidden Markov model. This computation produced a vector for each such segment, therefore, transforming the day's firing activity into a series of vectors, each N dimensional. The length of the segment used was 500 ms, whereas the shift from segment to segment was 100 ms.

An example of such transformation is presented in figure 5. This figure depicts the firing times of a 4 second segment, holding the simultaneous recorded spike trains of 8 cells. For each of these spike trains, a base line is plotted, and above it the short-time firing rate of the spike train is plotted as a dashed line. The height of the line (above the base line) is proportional to the computed firing rate. Two consecutive computation windows are also plotted in the figure, each is represented as a horizontal brace. One of these windows is shown above the spike trains, and the other is below. The results of the computation in those windows is presented to the left of the spike trains, as two sets of bars. The height of these bars is proportional to the number of spikes found in each of the windows. In this example, total duration of 4 *seconds* was used. This computation therefore produces 40 vectors, each of them 8 dimensional. These vectors were the point-wise measurements for the next stages of computation.

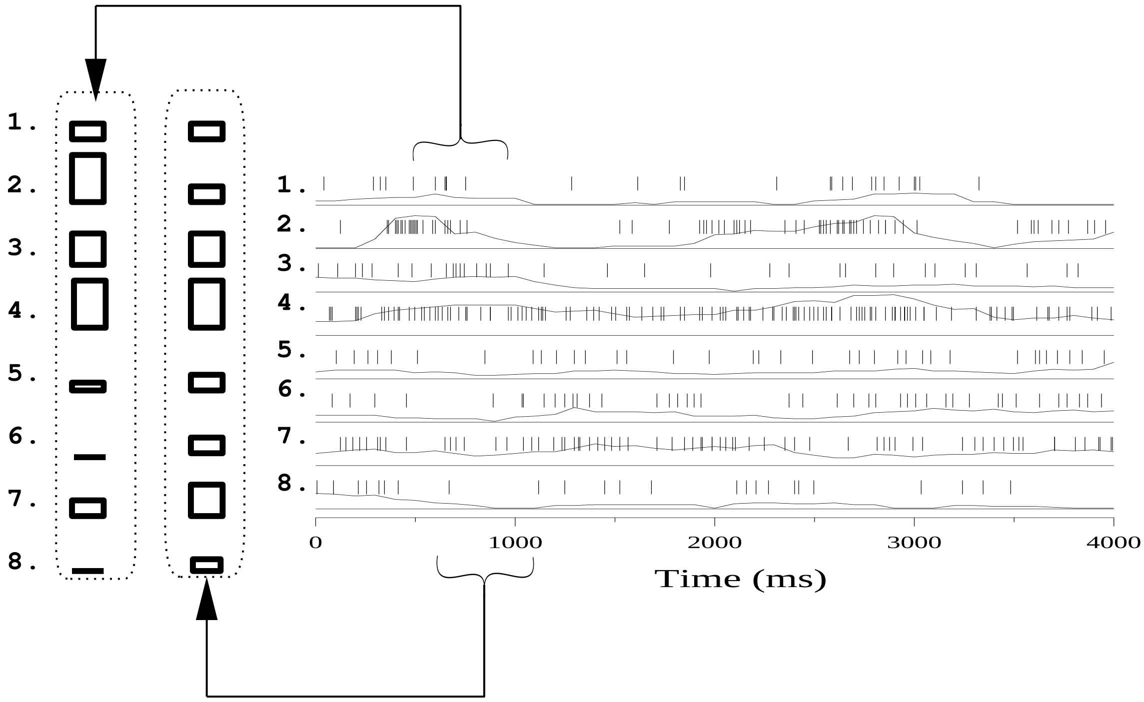

Figure 5. An example of 8 spike trains of simultaneously recorded cells, with the short time firing rate computation example. The firing rate is computed using the segments shown on the spike trains, and the result of computation is shown to the left of the data. The firing rates are scaled so that the width of each strip represents 30 spike/second.

4.2. Vector Quantization of the observations

In the second step of the modeling a modified version of the K-means algorithm [DH73], which is a special case of the EM algorithm [BP66, Bau72], was used to quantize the feature space. In such algorithm the continuous feature space is quantized into a discrete number of regions each characterized by a representative. These representative, which in our case are the firing rate vectors, will henceforth be referred to as *centroids*. The number of centroids has to be pre-defined for the algorithm.

These kinds of algorithms include two phases: initialization, and re-estimation. The initialization of the algorithm, which consists of choosing the first set of centroids, was implemented by randomly choosing K observations (firing rate vectors) from the full set of observations. In the second phase of the K-means algorithm an attempt was made to minimize the total divergence measure (error measure) of the data with respect to the model parameters. This two phases computation would be referred to henceforth as an iteration of the algorithm. In this implementation the divergence measure used was the KL-divergence described above in section 3.3.

The centroids computation carried out by this algorithm dependents upon the initialization centroids. This dependency may be very strong in some cases, i.e. small changes in the initial centroids can produce a totally different result. In order to overcome this problem, the execution of the algorithm was iterated several times, starting from different initial centroid values. The number of iterations was dependent on the convergence values, and upon the centroids reached. This dependency was according to the following conditions:

The algorithm was first repeated 5-6 times, the halting condition was decided upon based on the following steps:

(i) Each iteration produced, beside the centroids, a measurable result. This result was the average divergence of all the observation vectors, each one to its closest centroids.

(ii) These results were ordered from lower to higher.

(iii) If the range of the lower 60% of the scale (of results) was below a threshold, the process was halted. The threshold used was 0.5.

(iv) Five more iterations of the algorithm were carried out, and all the results were checked beginning with step No. 1.

4.3. Full hidden Markov model

The result of the quantization phase (the centroids), were used in the initialization of a hidden Markov model. The modeling continued by computing all the model parameters under the initialization inspired by the centroids. The parameters of the full model include a set of centroids and a matrix of transition probabilities. The training of the model was performed using the segmental K-means algorithm referred to in section 3.3.

After reaching convergence, the states of the model were ordered according to their time length (the total time that the system spent in each state). This was done in order to have a natural order and to be able to compare between the results of different models.

A typical example of such modeling can be seen in figure 6. This figure shows modeling of spike trains using 4 different states, each of which is characterized by different set of centroids. Eight cells were used to train that model, thus each state hold eight dimensional firing rate vectors. The states are connected via arrows, whose width is proportional to the transition probability between the different states.

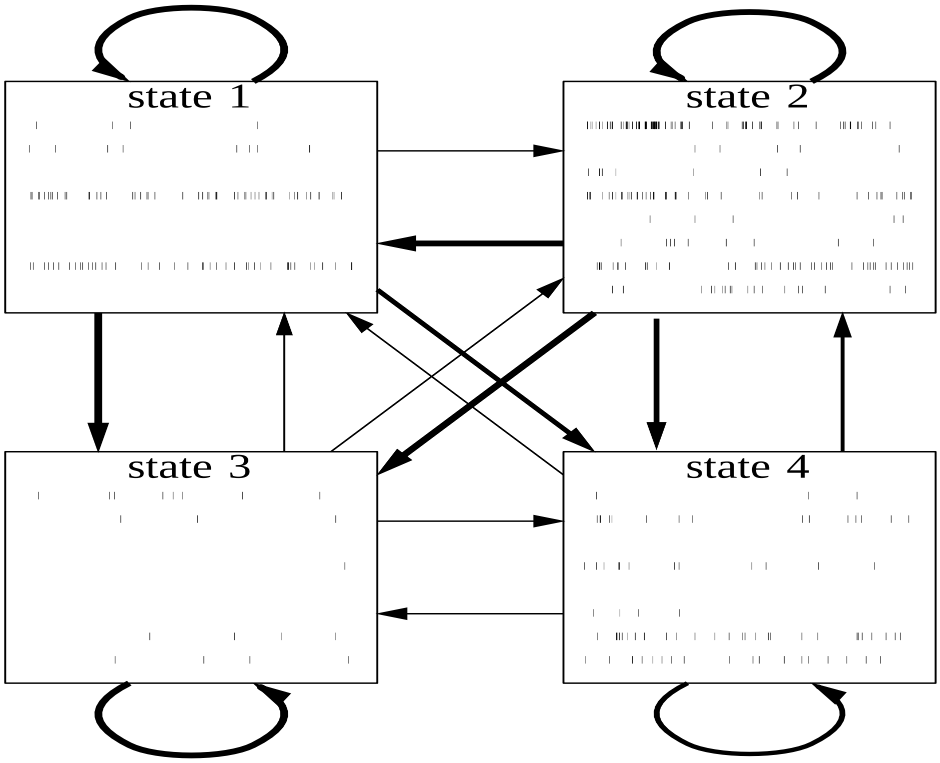

Figure 6. Typical modeling of 8 simultaneously recorded spike trains with Hidden Markov Modeling. In this example only four states were used (compared with 6-8 used in the real modeling of the data), each represented by 4 seconds of spike trains that were the observation of the state.

4.4. Cross-correlations computation

The computation of the hidden Markov model also produce a temporal segmentation of the spike trains, into states. This segmentation can be found out using the decoding problem, which tells for a given model and a sequence of observation what is the most probable sequence of states that produced it. An example of such a segmentation is presented in figure 7, in which the spike trains of 8 cells are shown. The temporal segmentation of the model is drawn over them (the vertical lines).Consecutive basic time units (each consisting of 100 *ms*) belonging to the same state constitute a time segment. The indexes of these segments are shown above the spike trains, at the top of the figure. These indexes will henceforth be called the *labels* of the segments.

We have used this segmentation to compute the pair-wise cross-correlation in different states of the model. This computation was performed as follows:

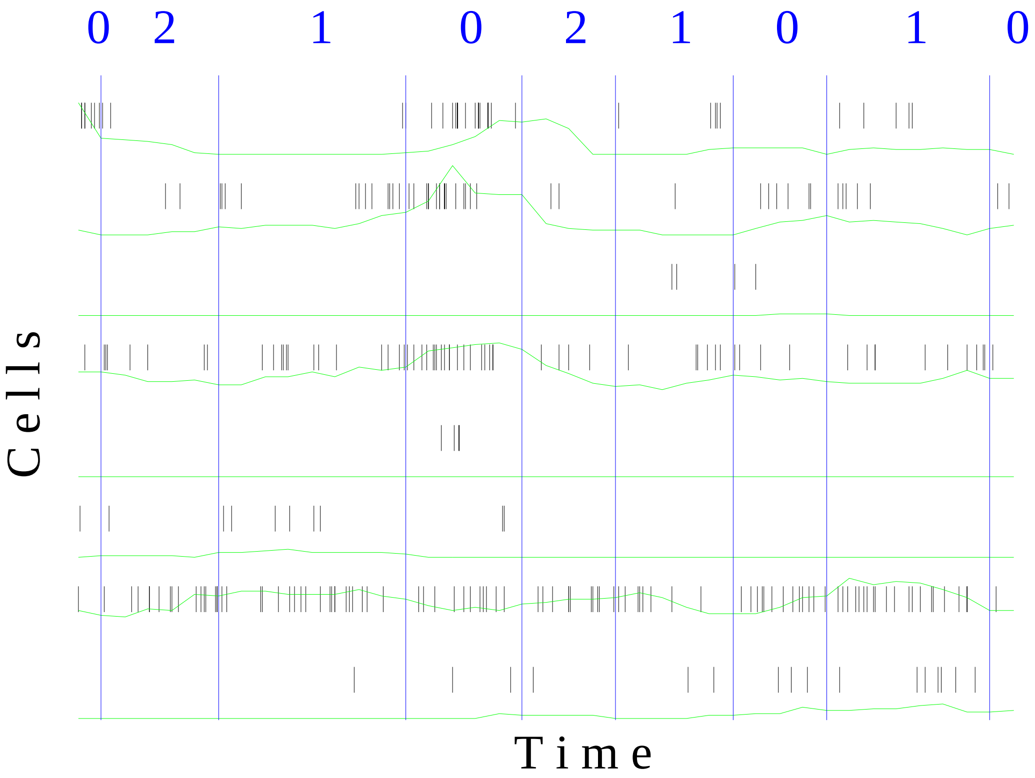

Figure 7. An example of a typical segmentation of spike trains of 8 cells.

(i) All the data from a recording session were segmented and labeled according to the model.

(ii) All the spike trains found in the segments of a given state were collected together.

(iii) From the spike trains so selected, only segments with a length of more than 1000 ms were used to compute the cross-correlation functions.

(iv) The computation of the cross-correlation functions was carried out as described in section 2.2.2.

(v) The results of this computation were collected into figures. Each figure included the correlograms of a specific pair of cells in all different states of the model. These figures were printed and scanned manually.

5. Results

5.1. Modeling results

The modeling described above was repeated several times for each of the recording sessions. Each repetition differed in its initial conditions, and yielded different models. Thus **multiple** sets of parameters were computed for each recording session.

It was found that all these multiple sets converged to a nearly identical parameter sets with almost the same error measure. This was true when the data were taken from the same behavioral mode. This fact was seen clearly when different recording sessions were taken, and also when models of data from the same recording day, but from different behavioral modes, were examined.

Two examples of models' centroids are presented in figure 8. This figure shows the results of modeling the data with 6 states, for a recording session that holds the spike trains of 7 cells. Two sets of centroids (**a** and **b**) are the output of data from the "Go" mode in this recording session. These two sets are the outcome of different parts of the data. The other set of centroids (**c**) is the output of data from the "No-Go" mode. In each of the centroids a set of bars is shown, whose height is proportional to the firing rate of the cells.

The upper two centroid sets show a high similarity, whereas the lower set of centroids differs from them. It can be seen therefore that the ordering procedure does bring the results into a natural order. An example of this difference can be seen in cell No. 5 in centroid No. 3. In sets **a** and **b** its firing rate is very low, whereas in set **c** its firing rate is very high. The differences between the sets from different behavioral modes exhibit some interesting phenomena:

(i) Some of the centroids display almost identical firing rates (for example, centroids No. 5 and No. 2 are almost identical in all sets).

(ii) For the centroids that show differences between the behavioral modes, the difference is usually seen in the firing rate of one or two cells. In centroid No. 3, for instance, only the firing rate of cell No. 5 changes between different behavioral modes.

(iii) The average firing rate of cells does not change dramatically, but rather, the change is seen in the combination of firing rates in different centroids. Cell No. 6, for example, does not change its average firing rate between different behavioral modes, but the combination of high firing rate of cell No. 6 and moderate firing rate of cell No. 1 exists only in set **c** and not in sets **a** or **b**.

The results presented above point to an interesting finding: the different models which characterize cells' behavior in one behavioral mode were very similar, whereas models of **different** behavioral modes were different from each other. This phenomenon holds for all the recording sessions examined. It enables us to reverse the process, i.e. to identify the behavioral mode of the monkey through analysis of the electro-physiological data from cells in the cortex. This identification can be carried out using the evaluation problem.

The next question to be examined was the accuracy of the restricted model (without transition probabilities and firing rate cross-correlations) in describing the data. The quantitative measure of accuracy, used in this model, was the average divergence measure of all firing rate vectors from their centroids. The degree of

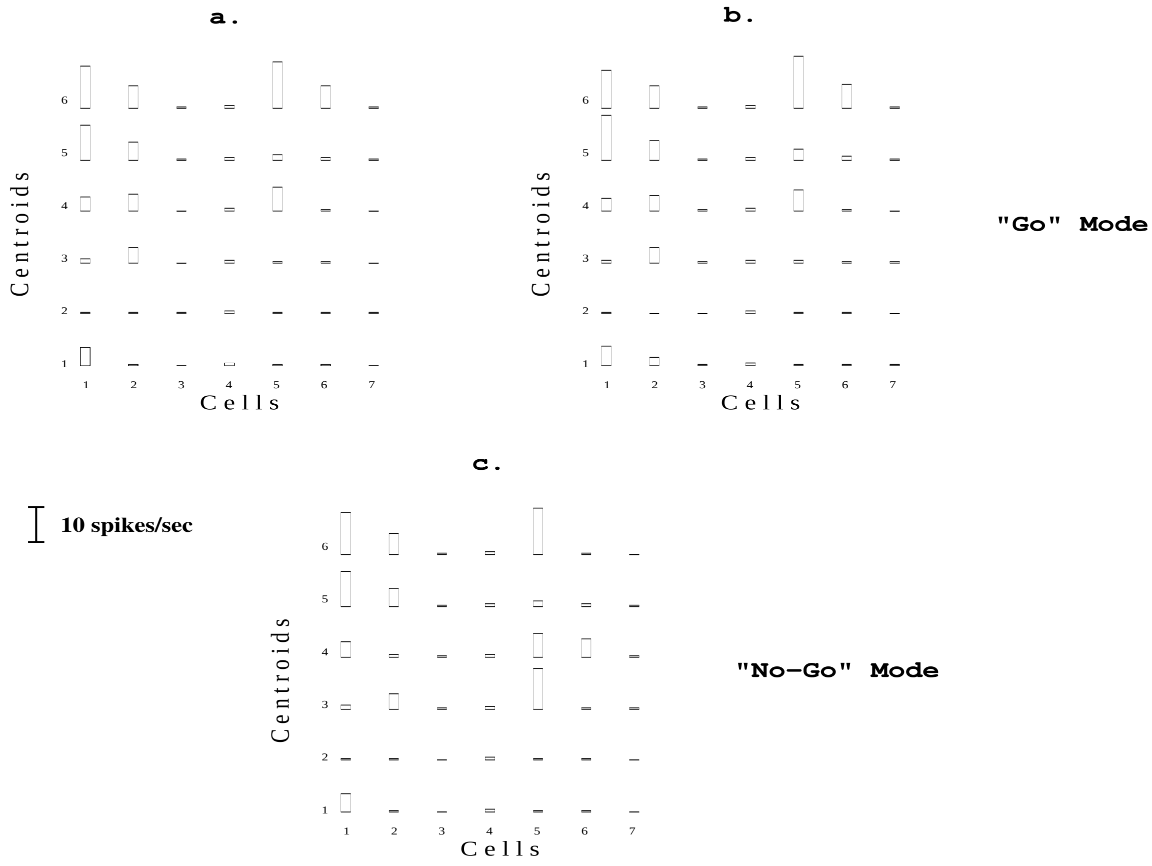

Figure 8. An example of three sets of centroids. The upper sets are from the same behavioral mode, the lower set is from a different behavioral mode. All the sets were taken from the same recording session.

accuracy will henceforth be called the *coherency* of the model. The better the model explains the data, the more coherent it is, and the smaller the measure of error. This coherency is measured by the average KL-divergence, and is presented in log scale.

In figure 9 the behavior of this coherency is depicted for all of the 6 days that were checked. This figure shows that the coherency measure differentiates between the different behavioral modes in all of the recording sessions. In 4 recording sessions the difference between the behavioral modes is significant (the threshold used here was a difference of more than 0.5 divergence). In three of them (**bl16** and **cl77** and **bl39**) significance is not reached, but the directionality is maintained.

The findings presented above show that a difference exists between the coherence of spike trains of different behavioral modes. Based on this result, it is possible to predict the behavioral mode from the coherence measure for a given set of data. The prediction can be made in two ways:

(i) Given models for different behavioral modes in the same recording session, and a set of spike trains from and **unknown** behavioral mode, it is possible to compute the coherence measure of this set and to relate it to one of the behavioral modes.

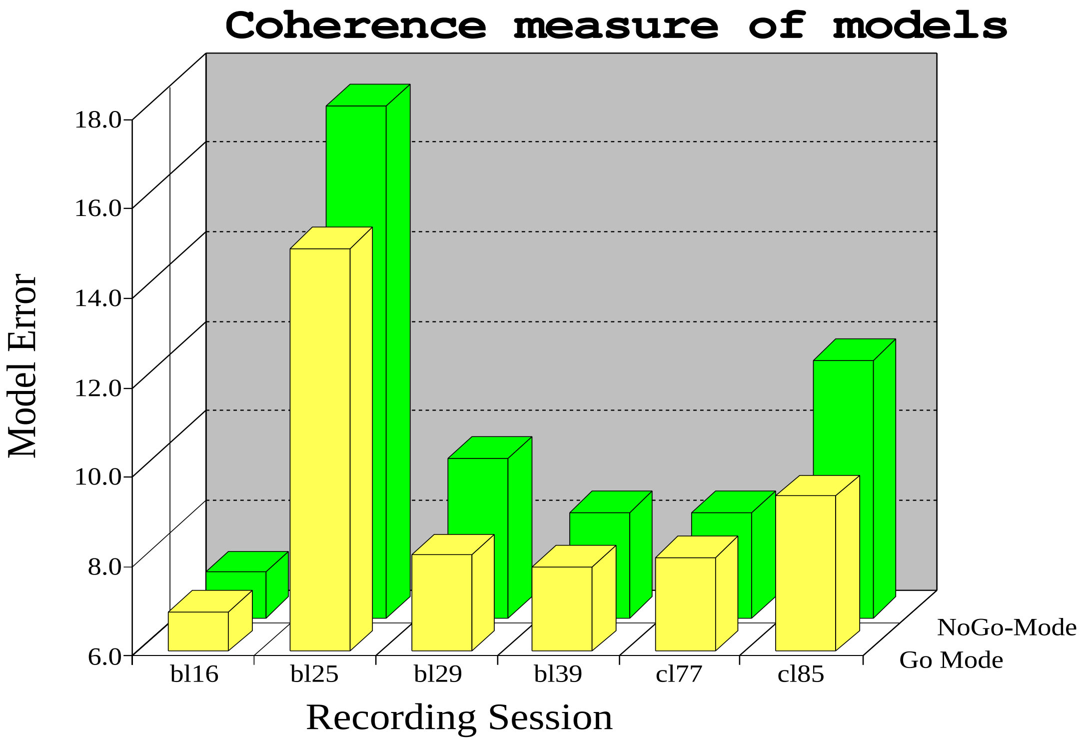

Figure 9. The coherency of the best model for the "Go" mode and the "No-Go" mode at each of the recording sessions.

(ii) Given two sets of spike trains for different behavioral modes from the same recording session, and without appropriate models, it is possible to differentiate the set of spike trains from "Go" mode versus "No-Go". This differentiation is achieved by computing the optimal model for each set of spike trains, and the coherence of the set according to its model. The set of spike trains with greater coherence with respect to its model can be assumed to be the product of the "Go" mode.

This type of analysis was not able to establish a significance correlation between the hidden states of the model, and the exact actions taken by the animal. The states were able to identify the general mode performed by the animal (as described above), but did not suffice us with finer resolution predictions.

5.2. Temporal cross-correlation

5.2.1. Cross-correlation selection The computation of the temporal cross-correlation was carried out as described in section 4.4. This stage of computation include also the selection of states whose length was above a certain threshold. In this test the threshold used was 1000 *ms*. As a result of this selection, only part of the data could be used for further analysis. The percentage of data used is shown in table 2. The information included in the table is as follows:

(i) The recording session code.

(ii) The total length of data in the recording session (in *sec*).

(iii) The length of all states used (in *sec*).

(iv) The percentage of data used from the total length.

Table 2. The percentage of data used for the temporal cross-correlation, from the total length of recorded data. This data were furthermore parceled into behavioral modes.

Session name	Total Length	Data Used	Percentage
bl16	8004	3149	39
bl25	11333	2372	21
bl39	11854	8664	73
cl77	16512	4632	28
cl85	8562	2688	31
Total	56262	21505	38

The second step in analyzing the pair-wise cross-correlation function is to check whether they have reasonable number of counts in their bins to warrant further analysis. Table 3 shows the information about the correlograms that include sufficient data, which is:

(i) The recording session code.

(ii) The number of well-isolated cells used in each recording session.

(iii) The number of cross-correlograms computed for each of the different states of the HMM. This number is $N(N-1)/2$, where N is the number of cells used in the recording session under examination.

(iv) The number of cell-pairs which included sufficient data. This number may be taken as an indication of the activity of cells in the recording session under examination.

Table 3. The number of sufficient data cross-correlograms, among the cross-correlograms of the chosen cells.

Session name	# Cells	# Total correlations	# Sufficient Data
bl16	8	28	28
bl25	8	28	22
bl39	6	15	13
cl77	6	15	11
cl85	8	28	25
Total	36	114	99

This table shows that although the firing rate is usually low, most of the correlograms hold sufficient information to achieve results.

After defining which correlograms hold sufficient data, the next step was to examine the data in these correlograms. We are interested in correlograms that show departures from background and thus indication some correlation between the activity of the cells. The process of selecting the non-flat correlograms was based mainly on 1% significance lines computed on each correlogram.The number of non-flat correlograms computed by that method in the recording sessions is shown in table 4. The table holds the following information:

(i) The recording session code.

(ii) M - The number of states used in the modeling of each recording session

(iii) N - The number of sufficient data pairs of cells, as found in the previous analysis. The total number of correlograms checked is $M * N$, and is shown in parenthesis in this column.

(iv) The number of pairs of cells in which non-flat correlograms were found, and the number of non-flat correlograms found (out of $N * M$ total correlograms). The latter number is shown in parenthesis. These numbers were computed separately for the "Go" mode and for the "No-Go" mode.

(v) The percentage of pairs of cells that show non-flat correlogram and the percentage of non-flat correlograms from the total number of correlograms (shown in parenthesis). These percentages were computed separately for the "Go" mode and for the "No-Go" mode.

Table 4. The number of non-flat correlograms among the sufficient data correlogram.

Session name	States (M)	Sufficient Data (N)	Non-Flat		Percentage	
			Go	No-Go	Go	No-Go
bl16	6	28 (168)	5 (15)	1 (3)	17.9 (8.9)	3.6 (1.8)
bl25	7	22 (154)	4 (7)	2 (3)	18.2 (4.5)	9.1 (1.9)
bl39	7	13 (91)	4 (12)	1 (1)	30.8 (13.2)	8.5 (1.1)
cl77	6	11 (66)	7 (21)	2 (5)	63.6 (31.8)	18.2 (7.6)
cl85	6	25 (150)	6 (16)	5 (15)	24 (10.7)	20 (10)
Total	36	99 (629)	26 (71)	11(27)	26.2 (11.3)	11.1 (4.3)

It can be seen from this table that a non-negligible number of correlograms reveal a tendency to deviate from a flat cross-correlation. This fact is crucial for the next steps, because if all the correlograms had shown flat correlations there would be no point in checking the changes between states. The results presented in table 4 show that a functional connectivity exists between pairs of cells in specific states of the model. This result may be due to the modeling of the data, or it may be the result of a prominent functional connectivity that exists throughout the whole recording session.

This table also shows that the percentage of non-flat correlograms is much higher in the "Go" mode than in the "No-Go" mode. In the "Go" mode 11.3% of the correlograms were non-flat, whereas in the "No-Go" mode only 4.5% were non-flat. This phenomenon is also seen in the percentage of pairs of cells that show any correlation. In the "Go" mode they constitute 26.2%, whereas in the "No-Go" mode they constitute only 11.1%.

5.2.2. Modification of pair-wise cross-correlation functions between states The Modifiable Pair-wise Cross-correlation function between States (MPCS) is defined as all pairs of cells that have different types of functional connectivity in different states of the model, i.e. in one of the states one type of functional connectivity is observed, whereas in other states a different functional connectivity is seen. In any case of doubt regarding the existence of a functional connectivity, the criterion was as strict as possible. The MPCS would not be counted if either:

- It was not clear that at least in one state a functional connectivity existed.
- It was not absolutely clear that at least in one (other) state, the correlogram was flat.

A typical example of the MPCS phenomenon is shown in figure 10. In this example, the number of states of the HMM is 7, so correlograms numbered from 0 to 6 of the same pair of cells (No. 5 and No. 8), are depicted. Each of these correlograms was computed for the spike trains of the matching state. In the correlogram of state No. 1, a clear, non-flat, cross-correlation function exists between the pair of cells, whereas in the correlogram of state No. 6 the correlogram is of the flat type. The cross-correlation function in state No. 1 is of a shared input type, i.e. the rise of the cross-correlation function above the significance line occurs on both sides of zero time. The duration of this rise is in the order of two hundreds ms. In the correlograms of states 0, 2 and 3, a similar, shared input phenomenon is shown. In state 3, however, the departure from background and crossing of the significance line are much weaker compared with states 0-2. On the other hand, in the correlograms of states 4, 5 and 6 no such departure is seen.

In table 5 the results of this analysis are shown. The table presents the number of MPCS occurring in each of the behavioral modes, compared with the number of non-flat correlations.

Table 5. The number of MPCS occurrences - classified into "Go" trials and "No-Go" trials. For each of these occurrences the number of non-flat correlations is shown.

Session name	"Go" mode		"No-Go" mode	
	MPCS	Non-flat	MPCS	Non-flat
bl16	5	5	1	1
bl25	4	4	2	2
bl39	4	4	1	1
cl77	6	7	2	2
cl85	6	6	4	5
Total	25	26	10	11

This table shows that in almost all of the cases where a non-flat correlation exists, an MPCS occurred. Furthermore, it can be seen that a non-negligible number of cell pairs show occurrences of MPCS, and that this percentage is much higher in the "Go" mode compared to the "No-Go" mode.

The MPCS example given in figure 5 also shows that the changes in correlation shape can not be explained only by changes in the firing rate. The same firing rates were seen in states 0 and 6 (around 10), whereas the correlation shape changes dramatically. Another example can be seen by comparing states 1 and 4 which share a different firing rate (around 0.5), but yet shows dramatic changes in the correlation shapes.

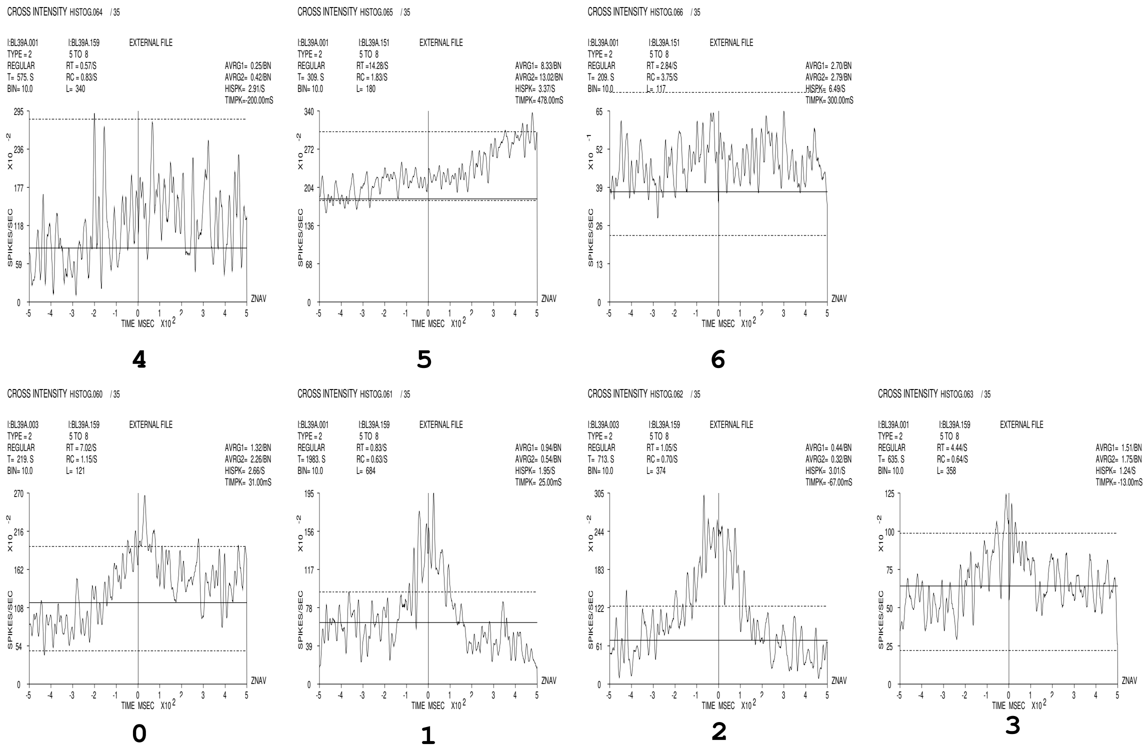

Figure 10. All the correlograms of a pair of cells in different states.

It is important to note also that random segmentation made to the same recording session didn't yield in any of the cases examined a phenomenon similar to the one seen so clearly above. That is in none of these cases was change observed in the cross-correlation function between the random states.

6. Discussion

The aim of the work presented here was to construct a model relating the hypothesized cell-assemblies and the attractors introduced in ANN, with the spike trains of several cells recorded simultaneously. Unlike most previous work we treated the simultaneous recordings as a vector random process of synchronous spike trains rather than individual point processes. Moreover, rather than creating histograms of activity (PSTH) by alignment of many repeated sequences by an external stimulus, we folded the recorded sequences in an unsupervised manner, based only on the statistical similarity of the vector process. This method of analysis provides the following advantages and consequences:

- It utilizes all the recorded activity, as long as the recording is stable, and not just the portions next to stimuli. This by itself often extracts more useful data out of a single recording day.

- All active cells contribute to the statistical discrimination not only cells with activity modulation around specific events. It might, in fact, be the case that cells that have slow firing modulation affect the collective activity in a way that cannot be detected by short aligned PSTH.
- The analysis induces an unsupervised segmentation of the data into "states" of similar activity, based on the statistical similarity of the firing activity of all the recorded cells. The number of states that can be resolved in this way is determined by the firing rates, number of recorded cells, and total amount of recorded data in a single session. This number should not be confused with the "real" number of states of collective activity in the data. The number of discernible states is essentially a modeling issue and should be determined by cross-validation or a similar model order estimation method, for the given data.
- The most striking consequence of the temporal segmentation, provided by our analysis, is its relation to the finer resolution structure, as observed by the pairwise correlations. Our finding that the cross-correlations between the same pair of cells can vary significantly from state to state, and can change significantly during 100ms, provides a compelling evidence to the following claims:
  (i) The cross-correlation is a consequence of the collective dynamics of the cell-assembly and not just the connectivity of the two cells.
  (ii) The states found by the model are in fact reflections of real internal dynamic processes which are collective and recurrent, such as dynamic attractors or syn-fire chains. To the best of our knowledge, this is the first direct electro-physiological evidence for such recurrent cortical dynamical processes.

The existence of dynamical changes in the cross-correlation function has already being shown in previous works [AVA$^+$92].

The statistical model used was a hidden Markov model (HMM), whose observations were the firing rate vectors of several cells recorded simultaneously. Our model assumption was that, given the states, the cells are conditionally independent,Poisson processes. The rates were estimated in a sliding window of 500ms, which gave rather poor temporal resolution. However, we suspected that shorter windows would not give a reasonable estimate of the rates, at least with our estimation technique. There is clearly a need for improvement on this score. Utilizing more sophisticated rate estimation methods, and this is currently being implemented.

The Markovian assumption behind the HMM should not be taken as an assumption about the actual cortical activity, nor should its states be considered as "states of mind" or the "true" cortical activity states. These are merely modeling assumptions which enable a direct glimpse into the actual cortical dynamics, clearly a much more complex process. The HMM states are characterized by internal consistency, and carry significance beyond their place in the model. This significance manifested itself in two ways:

- The model was able, within certain limitations, to predict the behavior of the animal from its computed states.
- The model originally dealt with the estimated parameters of processes which were presumed to take place in the cortex. The findings, however, seem to reveal underlying, **primary**, processing elements in the cortex.

This confirms the view that the neural computation is inherently a complicated process, which can only be glimpsed through different measurable parameters. These parameters may be the averaged firing rate of cells, the short-time cross-correlation, the firing patterns emerging from synchronous activity and so on. A better understanding of the neural computation can only be achieved by combining all these parameters in a united framework, and modeling them together. The Markovian state machine offers such a possibility.

6.1. Relations to other works

We are aware of two other works applying HMM to cortical recordings of multiple neurons, both followed our initial report in 1992 [GT93]. The first work, by Radon *et al* in 1994 [RBDK94], compared the non-homogeneous Poisson model (NHPM), which is essentially a PSTH, with a left-to-right HMM. They claimed that the NHPM gives a more appropriate description of the neuronal data, at least in the visual cortex. However, this work did not utilize the vector activity, as we suggested, and thus didn't take advantage of the main power of the HMM.

The other work, by Seidmann *et al* [SAMV96], was similar in several ways to our work. It was applied to a similar data, recorded from frontal and prefrontal cortex, but was used in a supervised way following a stimulus, in a much finer temporal resolution. The trained HMM was then used for the detection of one of the four behavioral modes. A comparison with NHPM in this work suggested that the NHPM was more powerful than the HMM, for the detection problem, but the HMM was better in the later stages of activity. Our understanding of this finding is that it is not the disadvantage of the HMM as much as it is a result of the temporal misalignment of the different trials in the NHPM, further away from the stimulus. The main advantage of the HMM is in its ability to analyze each trial separately. That work, however, suffers from several methodological flaws, which put in question many of its conclusions. Its main flaw is that it did not distinguish carefully enough between training and testing data, in fact most of the reported results were on the training data and as such exhibit severe over-fitting. The second problem is that the HMM used is not the best possible HMM for this data. In fact, when applying our HMM approach on precisely the same behavioral detection events used by Seidmann *et al* the HMM is found to be superior to the NHPM [GT97], unlike their reported results.

Acknowledgments

We would like to thank Hagai Bergman, and Eilon Vaadia for sharing their data with us, and for numerous stimulating and encouraging discussions of our approach. This research was supported in part by a grant from the Unites States Israeli Binational Science Foundation (BSF), and the Israeli Academy of Sciences.

References

[Abe91] M. Abeles. *Corticonics*. Cambridge University Press, 1991.

[ABG+95] 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.

[AGHP89] A. Aertsen, G.L. Gerstein, M.K. Habib, and G. Palm. Dynamics of neuronal firing correlation: Modulation of "effective connectivity". *J. Neurophysiology*, 61:900–917, 1989.

[AVA+92] 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.

[Bau72] L.E. Baum. An inequality and associated maxmization technique in statistical estimation of probabilistic functions of markov processes. *Inequalities*, 3(1):1–8, 1972.

[BD62] R.E. Bellman and S.E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, 1962.

[Bel57] R.E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[BP66] L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Stat.*, 37:1554–1563, 1966.

[Bru88] 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.

[BS91] V. Braitenberg and A. Schuz. *Anatomy of the Cortex*. Springer-Verlag, 1991.

[CT91] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[DH73] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Stat. Soc*, 39(1):1–38, 1977.

[Fus84] J.M. Fuster. Behavioral electrophysiology of the prefrontal cortex. *TINS*, pages 408–414, 1984.

[GBA89] G.L. Gerstein, P. Bedenbaugh, and A. Aertsen. Neuronal assemblies. *IEEE Trans. on Biomed. Eng.*, 36(1):4–14, 1989.

[Ghe88] 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.

[GT93] I. Gat and N. Tishby. Statistical modeling of cell-assemblies activities in associative cortex of behaving monkeys. *Neural Information Processing Systems*, 5:945–953, 1993.

[GT97] I. Gat and N. Tishby. Comparative study of different supervised detection methods of simultaneously recorded spike trains. Work in final stages., 1997.

[Heb49] D.O. Hebb. *The Organization of Behavior : a neurophysiological theory*. Wiley, 1949.

[JR90] B.H. Juang and L.R. Rabiner. The segmental k-means algorithm for estimating parameters of hidden markov models. *IEEE trans. on signal processing*, ASSP-38(9):1639–1641, 1990.

[Kru83] J. Kruger. Simultaneous individual recordings from many cerebral neurons: Techniques and results. *Rev. Phys. Biochem. Pharmacol*, 98:177–233, 1983.

[Kul59] S. Kullback. *Information theory and statistics*. Wiley & sons New York, 1959.

[LRS74] 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.

[ME91] N. Merhav and Y. Ephraim. Hidden markov modeling using the most likely state sequence. In *Proc. ICASSP-91*, volume 1, pages 469–472, 1991.

[MPS66] G.P. Moore, D.H. Perkel, and J.P. Segundo. Statistical analysis and functional interpretation of neuronal spike data. *Ann. Rev. Phsiol*, 18:493–522, 1966.

[PAG88] G. Palm, A. Aertsen, and G.L. Gerstein. On the significance of correlations among neuronal spike trains. *Biol. Cybern*, 59:1–11, 1988.

[Pal82] G. Palm. *Neural Assemblies - Studies of Brain Functions*. New York: Springer, 1982.

[PGM67] 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.

[Por88] A.B. Poritz. Hidden markov models: A guided tour. In *Proc. ICASSP-88*, pages 7–13, 1988.

[Pru95] Yifat Prut. *Firing Patterns as a sensitive estimator of cortical information processing*. PhD thesis, Hebrew university of Jerusalem, August 1995.

[PY85] 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.

[Rab89] 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.

[RBDK94] G. Radons, J.D. Becker, B. Dulfer, and J. Kruger. Analysis, classification, and coding of multielectrode spike trains with hidden markov modles. *Biol. Cybernetics*, page 359, 1994.

[RJLS86] 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.

[RWJ86] L.R. Rabiner, J.P. Wilson, and B.H. Juang. A segmental k-means training procedure for connected word recognition. *AT&T Tech*, pages 21–40, 1986.

[SAMV96] 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.

[SHS82] G.L. Shaw, E. Harth, and A.B. Scheibel. Cooperativity in brain function assemblies of approximately 30 neurons. *Exp. Neurol.*, 77:324–358, 1982.

[VABL91] E. Vaadia, E. Ahissar, H. Bergman, and Y. Lavner. Correlated activity of neurons: a neural code for higher brain functions. In J.Kruger, editor, *Neural Cooperativity*, pages 249–279. Springer-Verlag, 1991.

[vdM85] C. von der Malsburg. Nervous structures with dynamical links. *Ber Bunsenges. Phys. Chem.*, 89:703–710, 1985.