Supplementary Methods

Testing the Efficiency of Sensory Coding with optimal stimulus ensembles

by Christian K. Machens, Tim Gollisch, Olga Kolesnikova, and Andreas V.M. Herz

Contents

1 Introduction

In the following, we present the technical details necessary to find stimulus ensembles that maximize the mutual information. For a small set of predetermined stimuli, the optimal stimulus ensemble (OSE) can be found offline (Section 2; cf. Fig. 1D,E of main text). For larger stimulus spaces, we approximate the OSE online in an iterative procedure (Section 3; cf. Fig. 2–5 of main text). Some of the reliability and consistency issues of the online algorithm, such as estimation bias, problems with latencies and adaptation are discussed in Sections 3 and 4. Finally, we describe how to estimate the information rates shown in Fig. 6D,E (Section 5).

1.1 Information capacity

As explained in the main text, we define an optimal stimulus ensemble (OSE) as a stimulus ensemble that maximizes the information transfer of a given system. In this section, we recall the technical definition of OSEs.

Consider a stochastic input-output system with inputs s and outputs r. For simplicity, we will assume that both are given by discrete sets. The conditional probability distribution $p(r|s)$ describes how the system relates inputs to outputs. The amount of information I that is conveyed by such a system (i.e., the mutual information between s and r) depends on both $p(r|s)$ and the prior distribution of inputs $p(s)$,

$$I[p(\cdot)] = \sum_r q(r) \log_2 q(r) - \sum_s p(s) \sum_r p(r|s) \log_2 p(r|s) \tag{1}$$

where $q(r) = \sum_s p(r|s) \cdot p(s)$. The mutual information is bounded by the information capacity

$$C = \max_{p(\cdot)} I[p(\cdot)] \tag{2}$$

of the system (Cover and Thomas, 1991; Shannon and Weaver, 1949), where the maximum is determined with respect to all possible input distributions. We call any distribution $p(s)$ for which the system reaches its information capacity, an optimal stimulus ensemble and denote it as $p_{\text{opt}}(s)$. In the next section, we will show how an optimal ensemble can be estimated numerically, if $p(r|s)$ is known.

2 Information maximization OFF-line

A central result of information theory holds that the mutual information has a single maximum in the space of all probability distributions $p(s)$ (Cover and Thomas, 1991). This maximum can be degenerate, forming a plateau rather than a single peak. Nonetheless, we need not worry about the existence of local maxima which considerably simplifies the problem of determining $p_{\text{opt}}(s)$ for a given $p(r|s)$. To find an optimal stimulus ensemble $p_{\text{opt}}(s)$, we use the Blahut-Arimoto algorithm, as described next.

2.1 Blahut-Arimoto algorithm

Given a conditional response distribution $p(r|s)$, the Blahut-Arimoto algorithm allows to determine an optimal stimulus ensemble $p_{opt}(s)$. At step $n = 1$, we construct an initial stimulus ensemble $p_n(s)$ with $p_n(s) > 0$ for all s (for instance, a uniform distribution). We then iterate the equation

$$p_{n+1}(s) = c \cdot p_n(s) \cdot \exp \Big[\sum_r p(r|s) \log_2 \frac{p(r|s)}{q_n(r)} \Big] \tag{3}$$

where $q_n(r) = \sum_s p(r|s)p_n(s)$. The proportionality factor c in (3) is determined by the normalization condition $\sum_s p_{n+1}(s) = 1$. The term in the exponent corresponds to the Kullback-Leibler distance between $p(r|s)$ and $q_n(r)$ and will become large if the two distributions differ strongly. Consequently, the Blahut-Arimoto algorithm increases the probability $p(s)$ of an informative stimulus s, i.e., a stimulus s whose elicited responses, as described by $p(r|s)$, are distinct from the overall response distribution $q_n(r)$. By the same token, the algorithm decreases the probability $p(s)$ of an uninformative stimulus s, i.e., a stimulus s whose conditional response distribution $p(r|s)$ is similar to $q_n(r)$. For $n \to \infty$, the algorithm converges to an optimal stimulus ensemble, $p_n(s) \to p_{\mathrm{opt}}(s)$ (Blahut, 1972; Cover and Thomas, 1991). The information maximum can be degenerate and many different stimulus ensembles can be optimal. In these cases, the Blahut-Arimoto algorithm will return only one of the optimal ensembles. Which one is found, depends on the initial stimulus ensemble $p_1(s)$.

2.2 Smoothness of the OSE

When applying the algorithm to real data, we found that the resulting OSEs are highly susceptible to minor variations in the estimates of the conditional probabilities $p(r|s)$. Two slightly different sets of data may then lead to completely different OSEs. The results of this naive procedure therefore do not generalize well, a classical signature of overfitting. Since the OSE assigns a probability to every stimulus, there are as many parameters as stimuli. The naive approach usually results in a spiky distribution in which only a subset of stimuli have finite probabilities and the rest zero probability of occurring.

To circumvent this problem, we need to constrain the set of all possible OSEs. One possibility is to parameterize the OSE with a small set of parameters and then find their optimum values; this is the approach we take in the online algorithm. Another possibility is to introduce explicit “regularization” constraints on the objective function such as a penalty for spiky distributions (Hastie et al., 2001).

In the spirit of the latter approach, we used an ad hoc smoothing procedure whose main justification is that it leads to OSEs that generalize well (in the sense that slightly different data sets result in the same OSEs). Since we expect that the conditional probabilities $p(r|s)$ change smoothly with s, there should be an optimal stimulus ensemble $p_{opt}(s)$ that changes smoothly as well. Using the stimulus ensemble that results from the Blahut-Arimoto algorithm, we eliminated the spurious, spiky structure of the ensemble by smoothing the distribution with a Gaussian kernel. The width of the kernel was maximized under the constraint that the information rate of the smoothed ensemble lie within 99% of the maximum rate. This is the procedure that underlies the red curve in Fig. 1E.

3 Information maximization ON-line

The high dimensionality of the input space and the limited amount of data available in an experiment make it difficult to calculate $p_{\mathrm{opt}}(s)$ a posteriori and offline. Here, we handle the high-dimensionality of the space as well as the potential overfitting problem by parameterizing the stimulus ensemble. We then approximate $p_{\mathrm{opt}}(s)$ in an iterative procedure in which we choose new experimental stimuli according to the most recent estimate of $p_{\mathrm{opt}}(s)$ (Machens, 2002). In short, the algorithm works as follows (see also Fig. 2 in the main text):

1 **Initialization:** We arbitrarily choose an initial stimulus ensemble $p_n(s)$ with $n = 1$, e.g., the black ellipse in Fig. 2A. The technical details of the parameterization of the stimulus ensemble are explained in Section 3.1.

2 **Testing:** We draw ten stimuli from the distribution $p_n(s)$ and apply them repeatedly to the system. The stimuli and spike train responses are shown in Fig. 2B. Every

stimulus is characterized by its sample mean and standard deviation and is plotted as a dot in Fig. 2A. Our method of selecting the stimuli is explained in Section 3.2.

3 **Updating:** We use the Blahut-Arimoto algorithm to determine the contribution of each stimulus to the current maximum information rate; these contributions are indicated by the size of the dots in Fig. 2C. We update the stimulus ensemble $p_n(s)$, $n \to n+1$, by shifting it towards the more important stimuli (gray ellipse in Fig. 2C). The parameter updating procedures are described in Sections 3.3 and 3.4

4 **Iterating:** We continue with Step 2, using the updated stimulus ensemble. Note that all previous stimuli and the recorded responses remain in the data set used for the updating procedure.

In our experiments, we let the algorithm run for a fixed number of iterations ($n = 50$) or until the cell was lost. For experimental systems with very long recording times, a formal convergence criterion can be used (Machens, 2002).

3.1 Parameterization of stimulus ensembles

In the present study, a stimulus snippet $\mathbf{s}$ was defined as an 80-msec-long amplitude modulation of a sine wave carrier (characteristic frequency of the receptor neuron, 2.5 kHz in the example of Fig.2). Using a cut-off frequency of 250 Hz, a stimulus snippet $\mathbf{s}$ is thus specified uniquely by $N = 40$ sampling points, $\mathbf{s} = (s_1, s_2, \cdots, s_N)$. We characterize each snippet by its sample average, $a(\mathbf{s})$, and its sample standard deviation $b(\mathbf{s})$,

$$a(\mathbf{s}) = \frac{1}{N}\sum_j s_j \tag{4}$$

$$b(\mathbf{s}) = \sqrt{\frac{1}{N-1}\sum_j (s_j - a(\mathbf{s}))^2} \quad . \tag{5}$$

The stimulus ensemble, in turn, is characterized as a distribution over these two stimulus variables, $a(\mathbf{s})$ and $b(\mathbf{s})$. We define the distribution of stimuli for the online algorithm as

$$p(\mathbf{s}) \propto G(a(\mathbf{s}), b(\mathbf{s})) \quad . \tag{6}$$

where $G(a, b)$ denotes a Gaussian distribution over the mean a and standard deviation b of the snippets,

$$G(a, b) = \frac{1}{2\pi\sigma_\alpha\sigma_\beta} \exp\left[\frac{-(a-\alpha)^2}{2\sigma_\alpha^2}\right] \cdot \exp\left[\frac{-(b-\beta)^2}{2\sigma_\beta^2}\right]. \tag{7}$$

with parameters α, σ_α, β, and σ_β that describe the mean (α) and standard deviation (σ_α) of the snippets' sample average and the mean (β) and standard deviation (σ_β) of the snippets' standard deviation, respectively.

Accordingly, the stimulus ensemble is characterized by the four parameters α, σ_α, β, and σ_β. An advantage of this distribution is that we can easily estimate its four parameters using maximum-likelihood methods as explained below. Note that all stimuli $\mathbf{s}$ with identical mean and standard deviation have the same probability of occurrence.

The choice of parametrization allows us to independently assess the effects of two time scales: on a fast time scale (2 ms), i.e., within a snippet, the stimulus structure is determined by the statistics of the stimulus fluctuations (parameters β, σ_β); on a slow time scale (80 ms), the stimulus is controlled by changes in the mean of the snippets (parameters α, σ_α). The values of the parameters can be visualized as an ellipse that is drawn with a center (α, β) and half axes σ_α, σ_β (see Fig. 2A). During the course of the experiment, the values for these parameters will be updated such that information transmission is maximized.

3.2 Drawing stimuli

To draw snippets $\mathbf{s}$ from the distribution in Eq. (6), we resorted to the following simple approximation: First, mean a and standard deviation b are drawn from the two-dimensional Gaussian distribution $G(a, b)$. In the rare cases when b turned out to be negative, corresponding to negative standard deviations, we simply set b to zero. Second, a "white" noise snippet $\mathbf{y} = (y_1, y_2, \ldots, y_{40})$ is drawn from a 40-dimensional, normalized Gaussian distribution with zero mean and a diagonal covariance matrix with unit standard deviation. Finally, the experimentally applied snippet $\mathbf{s}$ is obtained by shifting and scaling $\mathbf{y}$ by the selected mean a and standard deviation b,

$$\mathbf{s} = b\mathbf{y} + a \quad . \tag{8}$$

Note that this procedure is not unique: different combinations of a, b, and $\mathbf{y}$ can lead to the same snippet $\mathbf{s}$. Accordingly, the probability of drawing $\mathbf{s}$ is not exactly represented by Eq. (6). Our drawing procedure therefore only approximates the stimulus ensemble in Eq.6. (Given the high dimensionality of the stimulus space, the approximation is very good, though.) For the online algorithm, this does not matter. At worst, the approximation may slow down the speed of convergence.

3.3 Updating the parameters—rate code

If a given set of stimuli $\{\mathbf{s}_i, i = 1 \ldots K\}$ has been tested on the system, we can estimate the conditional probabilities $p(r|\mathbf{s}_i)$ where r denotes the response—here the firing rate of the neuron. Given this information, we can use the Blahut-Arimoto algorithm and iterate the equation

$$q_{m+1}(\mathbf{s}_i) \propto q_m(\mathbf{s}_i) \cdot \exp\Big[\sum_r p(r|\mathbf{s}_i) \log_2 \frac{p(r|\mathbf{s}_i)}{q_m(r)}\Big] \tag{9}$$

with $q_m(r) = \sum_i p(r|\mathbf{s}_i) q_m(\mathbf{s}_i)$. The resulting optimal distribution $q(\mathbf{s}_i)$ is only defined on the stimuli measured so far; we can visualize the values $q(\mathbf{s}_i)$ as weights that specify how important each stimulus is for the transmitted information (see dots in Fig. 2C; see also Fig. 5).

In the next step, we use these weights to update the parameters of the stimulus ensemble $p(\mathbf{s})$. For that purpose, we use maximum likelihood estimation to find parameters such that the probabilities $p(\mathbf{s}_i)$ best match the weights $q(\mathbf{s}_i)$. This is done by maximizing the weighted likelihood function

$$\log L = \sum_i q(\mathbf{s}_i) \log p(\mathbf{s}_i) \tag{10}$$

$$= \sum_i q(\mathbf{s}_i) \log G\Big(a(\mathbf{s}_i), b(\mathbf{s}_i)\Big) \quad . \tag{11}$$

The likelihood maximization then results in the following set of update equations:

$$\alpha = \sum_i q(\mathbf{s}_i) a(\mathbf{s}_i) \tag{12}$$

$$\sigma_\alpha^2 = \sum_i q(\mathbf{s}_i)(a(\mathbf{s}_i) - \alpha)^2 \tag{13}$$

$$\beta = \sum_i q(\mathbf{s}_i) b(\mathbf{s}_i) \tag{14}$$

$$\sigma_\beta^2 = \sum_i q(\mathbf{s}_i)(b(\mathbf{s}_i) - \beta)^2 \tag{15}$$

The new parameters specify the updated stimulus ensemble (see second ellipse in Fig. 2C). In the next iteration, the new ensemble is used to draw new stimuli (as explained in Section 3.2) and all the steps are repeated, thus iterating the algorithm.

In each iteration, the algorithm uses all available data (and not just those from the last iteration) to calculate new estimates for the parameters of the stimulus ensemble, Eqs. (12)–(15). During the first few iterations, however, the available data is relatively sparse which can compromise the estimation of the parameters. This can be counterbalanced by using a more conservative updating rule for the first few iterations, e.g., by updating the parameters "only" to $\frac{1}{2}[x_{\text{new}} + x_{\text{old}}]$. In our experiments, we applied this rule during the first ten iterations.

3.4 Updating the parameters—timing code

In the timing code, the responses r are taken to be binary strings or "words" which are constructed by binning the spike train into 2 msec bins; every bin takes either the value one (if a spike is present) or zero (if no spike is present). The size of the bins was chosen to be small enough that no bin contained two spikes.

For 80-msec-long stimuli, we obtain 40-bit words so that the response set has 2^{40} entries. Unfortunately, this number is far too large to explore a response set of that size in the electrophysiological experiments. We therefore resorted to the following short-cut: Assuming that we can neglect correlations in the response above 20 msec[1], we approximate the conditional response distribution by its factorization

$$p(r|\mathbf{s}) = p(r_1|\mathbf{s}_{20,1})p(r_2|\mathbf{s}_{20,2})p(r_3|\mathbf{s}_{20,3})p(r_4|\mathbf{s}_{20,4}) \quad .$$

Here r_1, r_2, r_3, and r_4 are four 10-bit words whose concatenation is r; likewise, $s_{20,1}, \ldots, s_{20,4}$ are four 20-msec stimuli whose concatenation is $\mathbf{s}$. Since the stimulus ensemble can be decomposed similarly, the joint distribution of responses and stimuli can be approximately factorized as well, and the information conveyed about an 80-msec-long stimulus is roughly the same as the summed information about the four 20-msec-long stimulus fragments. Using this short-cut, we can perform the updating procedure on 20-msec-long stimuli and 10-bit words. Since the response set is now reduced to 2^{10} different symbols, its exploration during the course of the experiment is feasible. To make full use of the data, we used sliding 20-msec windows (advanced in steps of 2 msec) to assemble stimuli and responses.

With this approximation, the updating procedure is the same as for the rate code (Section 3.3). The stimulus weights $q(\mathbf{s}_{20,i})$ were computed using the Blahut-Arimoto algorithm, Eq. (9), and the parameters were updated according to Eq. (12–15).

3.5 Examples for the evolution and convergence of parameters

As explained in the main text, the analysis can be performed for different readout modes. We used two different modes, a rate code, where the response is specified by counting spikes in 80-msec-long windows corresponding to the length of the snippets, and a timing code, where the response is given by 20-msec-long stretches of the spike train binned at 2 msec, yielding 10-bit binary words.

For the case of a rate code, Figure S1 shows the evolution of the model parameters and the achieved transmitted information as a function of the number of iterations. The values of the model parameters are again visualized as ellipses that outline the standard deviations of the Gaussian $G(a, b)$ from which the mean and standard deviation of the snippets are drawn for the subsequent iteration. The final ensemble is denoted by a thick black ellipse. Here, three runs of the algorithm obtained from the same neuron with different choices

[1] Note that the investigated receptor neurons integrate the sound amplitude over a few milliseconds only

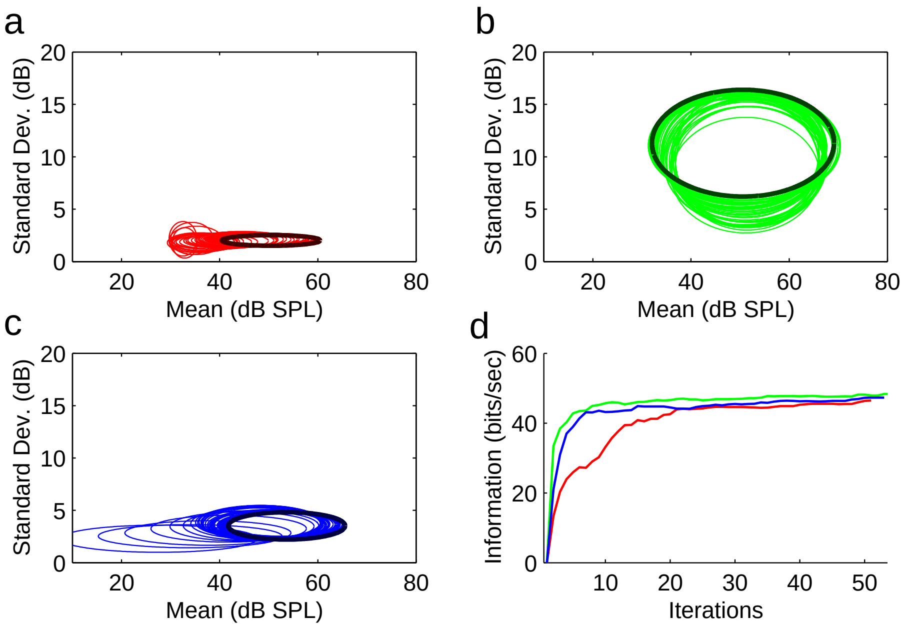

Figure S1: Online iterations for three different initial conditions from the same cell, using a rate-code read-out. (a–c) The different initial conditions of the stimulus ensembles result in different fits of the final optimal ensembles (black ellipses). (d) In all cases, the information rate grows with each iteration until it saturates after about 20 iterations. Independently of the initial conditions, all runs achieve about the same information rates ($I \approx 50$ bits/sec).

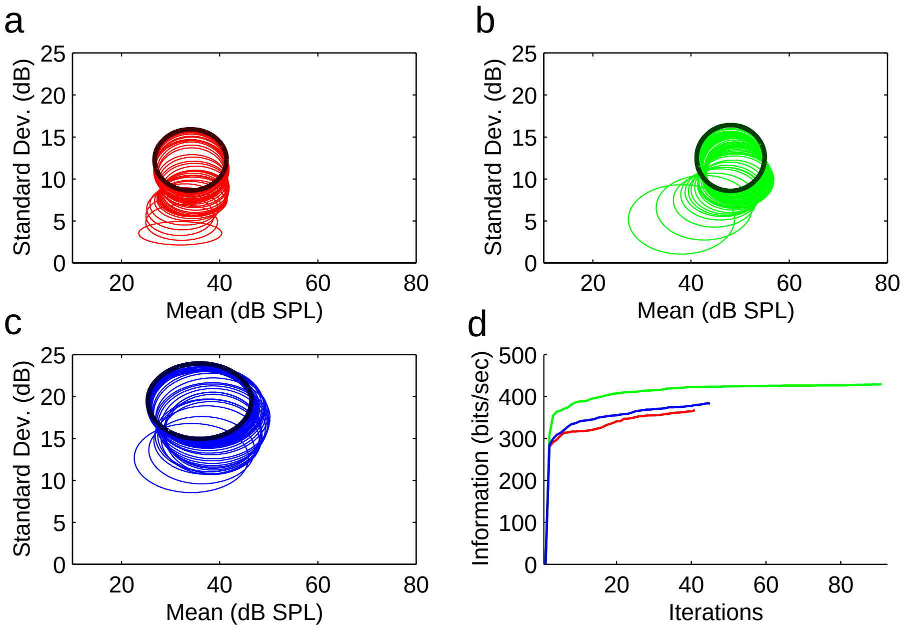

Figure S2: Online iterations for three different initial conditions for three different cells using a timing-code read-out.

for the initial stimulus ensembles are shown. Although the runs start from different initial conditions and converge to different final ensembles, the three experiments achieve about the same information rates (Fig. S1d), suggesting that each ensemble succeeds in maximizing the information transfer.

To illustrate how the model parameters develop for the timing code, series of such iterations are shown in Fig. S2 for three different cells. The obtained information rates are about one order of magnitude larger, approaching $I_{\mathrm{opt}} = 400$ bits/sec.

4 Estimation bias and other technical concerns

4.1 Effect of bias on the maximization algorithm

All values of the transmitted information that we used in this work were obtained from entropy estimates of the form $H(r) = -\sum_r p_{\mathrm{est}}(r) \log_2 p_{\mathrm{est}}(r)$, where the sum runs over all observed responses r and where $p_{\mathrm{est}}(r)$ denotes the empirical estimator of the probability of finding r,

$$p_{\mathrm{est}}(r) = \frac{\#\text{ observations of } r}{\#\text{ trials}}. \tag{16}$$

It is well known that this naive estimator of H is prone to systematically underestimate the true entropy thus leading to a bias in the information rates (Treves and Panzeri, 1995; Strong et al., 1998; Nemenman et al., 2004; Paninski, 2003). Two issues are therefore to be considered: how does this bias influence the information estimates and, more importantly, how does it influence the obtained OSEs?

To gain some insight into the latter problem, it is helpful to consider the extreme case of only one repetition per stimulus. In this case, the noise entropy (second term in Eq. (1)) is always zero, and the maximization algorithm maximizes output entropy only. Hence, the algorithm does not distinguish between noisy and reliable regions in stimulus space, it simply moves into whatever region leads to the greatest variety of reponse symbols. If we increase the number of repetitions, the estimate of the conditional response distribution $p(r|s)$ will provide more and more information about the reliability of the symbols. Accordingly, the noise entropy term will become more and more important, until, in the event of infinitely many repetitions, the conditional estimates are perfect, and we maximize the information rate. Hence, any optimization based on a finite number of repetitions can be seen as a compromise between merely maximizing the output entropy and truly maximizing the mutual information.

In the offline analysis, we can address the question of the influence of bias induced by finite data, e.g., by performing the analysis on only a fraction of the trials. Calculating the information for a fixed set of optimal weights $p_{\mathrm{opt}}(s)$, but using only a fraction of trials allows an extrapolation to infinite data (inverse data fraction = 0), and thus an estimate of the true information (Strong et al., 1998). Fig. S3a and S4a show that there is indeed a substantial overestimation of information and that the corrections to the information values are of the order of 5% for both the rate and the timing code.

The same procedure can be applied to evaluate the bias on the estimation of the parameters of the OSE. Recalculating the parameters for fractional data, one observes that the variance of the estimates for the parameters outweighs the bias in most cases (Fig. S3c–f and S4c–f). Accordingly, the estimation bias for the optimal parameters is less severe than that of the information rates. An explanation for this may be that the bias, even if substantial, is similar for all stimulus ensembles near maximal information transmission. Thus the height of the information maximum is certainly affected by the estimation bias, but less so its location. These results suggest that the parameter values obtained from the online algorithm are rather robust and that the number of trials allows for a sufficiently precise

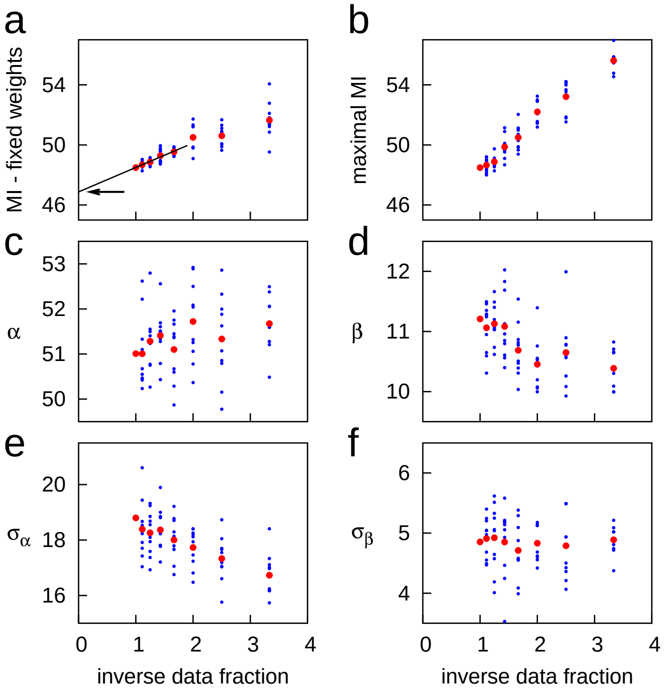

Figure S3: Finite size effects: mutual information and parameters as a function of the data size (shown as inverse fraction of the measured data) for one sample cell and the rate code paradigm. (a) With the weights $p_{\mathrm{opt}}(s)$ fixed at the values obtained when using the complete data set, the mutual information (MI) was calculated for different fractions of the data by choosing a subset of trials at random. The blue points depict results from individual calculations, and the red points average values from 10 calculations. The black line shows an extrapolation to infinite data (inverse data fraction = 0) obtained from the five leftmost red points, which can be used to estimate the true information (arrow), in this case ∼46.9 bits/s. (b) Values of the mutual information with weights $p_{\mathrm{opt}}(s)$ not held fixed, but recalculated for each fractional data set for comparison. This approach leads to an information estimate of ∼46 bits/sec. (c–f) Values of the optimal parameters α, σ_α, β, σ_β obtained for different fractions of the data. Blue points again show individual calculation, red points averages over 10 calculations for a fixed number of left-out trials. In contrast to the values for the mutual information, differences of the parameter values for different calculations are dominated by the variance, not the bias, as shown by the large variability of the data even for small values of the inverse data fraction.

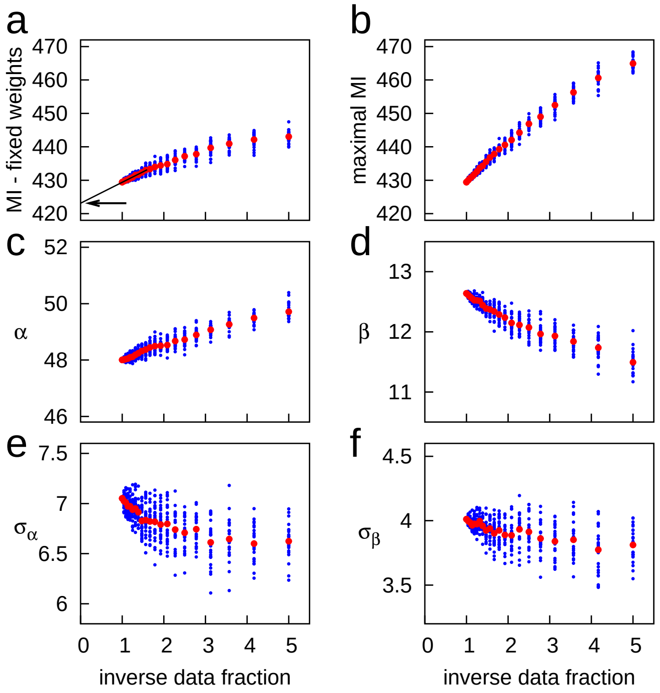

Figure S4: Finite size effects: mutual information and parameters as a function of the data size (shown as inverse fraction of the measured data) for one sample cell and the timing code paradigm. The figure format is equivalent to that of Fig. S3. (a) The extrapolated value of the true information (arrow) with weights $p_{\text{opt}}(s)$ fixed at the values obtained when using the complete data set was found to be 423.1 bits/s. (b) Information for fractional data with recalculated weights. (c–f) Parameter values computed for fractional data. Blue points show individual calculation with subsets of trials chosen at random, red points averages over 25 calculations for a fixed number of left-out trials. The effect of bias is most severe for estimation of the parameter β, indicating that this value may be underestimated. For the other parameters, the bias is of the order of a few per cent and similar to the accuracy of the data given by the spread of the individual results.

estimate of the conditional response distribution. The analysis with fractional data allows us to assess the size of the statistical error for the values of the model parameters. Leaving out one trial of all the data, e.g., yields a spread of values of the order of 10%.

4.2 Response latency

Since the investigated system has an intrinsic response delay between stimulus presentation and measured response, stimuli and responses could potentially be misaligned, especially when cut into 20-msec-long windows as in the timing-code read-out. We therefore corrected for this response delay. The resulting total latency of the signal, due to propagation of the acoustic signal, the biomechanical processes at the receptor, and the finite conductance velocity along the axon, was estimated to have a value of around 4 msec. The assignment of responses to particular stimuli was therefore shifted correspondingly.

4.3 Adaptation

The conditional probabilities $p(r|\mathbf{s})$ may depend on the past history of the system. Indeed, the system under study adapts on longer time scales ($>$ 50 msec) and thus the estimated conditional probabilities $p(r|\mathbf{s})$ will also reflect the specific stimulus history of the system. If adaptation seeks to maximize the mutual information about the stimuli, as has been suggested (Brenner et al., 2000; Fairhall et al., 2001), then the algorithm and the neuron may meet halfway. Anecdotal evidence suggests that the dependence on initial conditions (in particular, the position and extent of the ensembles along the x-axis) may in some instances be partly a result of adaptation. This point, however, will have to investigated in more detail in the future.

5 Estimating model probabilities

The conditional probabilities $p(r|\mathbf{s}_i)$ provide us with knowledge about the system's input-output relation on select points $\mathbf{s}_i$ in stimulus space. Can we use this knowledge to reliably estimate information rates for arbitrary stimulus ensembles $p_\vartheta(\mathbf{s})$ where $\vartheta = (\alpha, \sigma_\alpha, \beta, \sigma_\beta)$? In general this will not be possible; however, if the bulk of the distribution $p_\vartheta(\mathbf{s})$ falls into the part of stimulus space where we do know the conditional probabilities $p(r|\mathbf{s}_i)$, we can.

We can compute the probabilities $p_\vartheta(\mathbf{s}_i)$ for the tested stimuli according to Eq. (6). In estimating the information rates, we need to keep in mind that the stimuli s_i were not drawn from this distribution. Rather, in the online procedure, the stimuli were drawn from the distribution (sampling density)

$$\rho(\mathbf{s}) = \frac{1}{N} \sum_{i=1}^{N} p_{\theta_i}(\mathbf{s}) \tag{17}$$

where θ_i denotes the parameters of the stimulus ensemble at the i-th iteration of the experiment and we average over all N iterations. To correct for this fact, we introduce the corrected distribution

$$\tilde{p}_\vartheta(\mathbf{s}_i) \propto \frac{p_\vartheta(\mathbf{s}_i)}{\rho(\mathbf{s}_i)} \tag{18}$$

which in turn can be used to estimate the mutual information according to the equation

$$I = \sum_{i,r} p(r|\mathbf{s}_i) \tilde{p}_\vartheta(\mathbf{s}_i) \log_2 \frac{p(r|\mathbf{s}_i)}{q(r)} \tag{19}$$

and $q(r) = \sum_i p(r|\mathbf{s}_i)\tilde{p}_\vartheta(\mathbf{s}_i)$. The value of I is then an estimate for the mutual information conveyed by the stimulus ensemble $p_\vartheta(\mathbf{s})$.

References

Blahut, R. E. (1972). Computation of channel capacity and rate-distortion functions. *IEEE Trans. Inform. Theory*, IT-18(4):460–473.

Brenner, N., Bialek, W., and de Ruyter van Steveninck, R. R. (2000). Adaptive rescaling maximizes information transmission. *Neuron*, 26:695–702.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory.* Wiley.

Fairhall, A. L., Lewen, G. D., Bialek, W., and de Ruyter van Steveninck, R. R. (2001). Efficiency and ambiguity in an adaptive neural code. *Nature*, 412(22):787–792.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning theory.* Springer.

Machens, C. K. (2002). Adaptive sampling by information maximization. *Phys. Rev. Lett.*, 88:228104.

Nemenman, I., Bialek, W., and van Steveninck, R. R. R. (2004). Entropy and information in neural spike trains: Progress on the sampling problem. *Phys. Rev. E*, 69:056111.

Paninski, L. (2003). Estimation of entropy and mutual information. *Neural Comp.*, 15:1191–1254.

Shannon, C. E. and Weaver, W. (1949). *The mathematical theory of communication.* University of Illinois Press.

Strong, S. P., Koberle, R., de Ruyter van Steveninck, R. R., and Bialek, W. (1998). Entropy and information in neural spike trains. *Phys. Rev. Lett.*, 80(1):197–200.

Treves, A. and Panzeri, S. (1995). The upward bias in measures of information derived from limited data samples. *Neural Comp.*, 7:399–407.