# An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes

**Felix Franke · Michal Natora · Clemens Boucsein · Matthias H. J. Munk · Klaus Obermayer**

**Abstract** For the analysis of neuronal cooperativity, simultaneously recorded extracellular signals from neighboring neurons need to be sorted reliably by a spike sorting method. Many algorithms have been developed to this end, however, to date, none of them manages to fulfill a set of demanding requirements. In particular, it is desirable to have an algorithm that operates online, detects and classifies overlapping spikes in real time, and that adapts to non-stationary data. Here, we present a combined spike detection and classification algorithm, which explicitly addresses these issues. Our approach makes use of linear filters to find a new representation of the data and to optimally enhance the signal-to-noise ratio. We introduce a method called "Deconfusion" which de-correlates the filter outputs and provides source separation. Finally, a set of well-defined thresholds is applied and leads to simultaneous spike detection and spike classification. By incorporating a direct feedback, the algorithm adapts to non-stationary data and is, therefore, well suited for acute recordings. We evaluate our method on simulated and experimental data, including simultaneous intra/extracellular recordings made in slices of a rat cortex and recordings from the prefrontal cortex of awake behaving macaques. We compare the results to existing spike detection as well as spike sorting methods. We conclude that our algorithm meets all of the mentioned requirements and outperforms other methods under realistic signal-to-noise ratios and in the presence of overlapping spikes.

**Keywords** Realtime spike sorting · Extracellular multi electrode recordings · Tetrode recordings · FIR filters · Deconfusion

F. Franke (✉) · K. Obermayer
Bernstein Center for Computational Neuroscience,
Berlin, 10099, Germany
e-mail: ff@cs.tu-berlin.de

K. Obermayer
e-mail: oby@cs.tu-berlin.de

M. Natora
Institute for Software Engineering and Theoretical
Computer Science, Berlin Institute of Technology,
Berlin, 10587, Germany
e-mail: natora@cs.tu-berlin.de

C. Boucsein
Bernstein Center for Computational Neuroscience,
Freiburg, 79104, Germany
e-mail: boucsein@biologie.uni-freiburg.de

M. H. J. Munk
Max Planck Institute for Biological Cybernetics,
Tübingen, 72076, Germany
e-mail: matthias.munk@tuebingen.mpg.de

## 1 Introduction

In order to understand higher brain functions and the interactions between single neurons, an analysis of the simultaneous activity of a large number of individual neurons is essential. One common way to acquire the necessary amount of neuronal activity data is to use simultaneous extracellular recordings, either with single electrodes or, more recently, with multi electrodes like tetrodes (O'Keefe and Recce 1993). However, the recorded data does not directly provide the isolated activity of single neurons, but a mixture of neuronal

activity from many neurons additionally corrupted by noise. The task of so called "spike sorting" algorithms is to reconstruct the single neuron signals (i.e. spike trains) from these recordings. Many approaches for analyzing the data after acquisition, i.e. *offline* spike sorting algorithms, have been developed in the last years; see for example Vargas-Irwin and Donoghue (2007), Delescluse and Pouzat (2006), Pouzat et al. (2004), Kim and Kim (2003), Takahashi et al. (2003), Shoham et al. (2003), Hulata et al. (2002), Lewicki (1998), Fee et al. (1996a). Although more methods are available in this category, there are several reasons to favor methods which provide results already during the recordings, termed *realtime online* sorting algorithms. For example, realtime online spike sorting techniques are indispensable for conducting "closed-loop" experiments and for brain-machine interfaces (Rutishauser et al. 2006; Obeid and Wolf 2004). The few existing approaches to realtime online sorting (Thakur et al. 2007; Rutishauser et al. 2006; Aksenova et al. 2003) are *clustering based* and have at least one of the following drawbacks: 1) They are not explicitly formulated for data acquired from multi electrodes, 2) they do not resolve overlapping spikes, 3) they do not perform well on data with a low signal-to-noise ratio 4) they are not able to adapt to non-stationarities of the data as caused by tissue drifts. We discuss the reasons and importance of these issues in the following:

1) Multi electrodes (e.g. tetrodes) provide significantly more information about the local neuronal population than single electrodes (Harris et al. 2000; Rebrik et al. 1999). Having several recording electrodes closely spaced instead of one, the same action potential is present on more than one recording channel. The so called stereo-effect—a neuron specific amplitude distribution among the recording channels—allows for a better discrimination between action potentials from different neurons (Gray et al. 1995). This allows also for more a reliable resolution of overlapping spikes.

2) With tetrodes recording an increased number of neurons compared to high impedance single electrodes, overlapping spikes are more likely to occur. Also, studies stress the relevance of ensemble coding, which translates into local synchronized firing and hence a raised occurrence frequency of overlapping spikes (Sakurai and Takahashi 2006). To identify such a code, the resolution of overlapping spikes is crucial and efforts have been made addressing this issue (Ding and Yuan 2008; Wang et al. 2006; Zhang et al. 2004; McGill 2002; Chandra

and Optican 1997). However, the cited approaches are all computationally very expensive, making a realtime online implementation difficult. One of the reasons for this computational complexity is the implementation of separate sub-routines for the processing of overlapping spikes, which, additionally, are more complex than the processing steps for non-overlapping spikes.

3) Most of the spike sorting approaches use a stand-alone standard spike detection technique (see for example Choi et al. 2006; Obeid and Wolf 2004; Rebrik et al. 1999 for commonly used spike detection techniques), and a separate classification procedure. Neither the shape of the waveforms nor their change over time or their amplitude distribution across the recording channels is taken into account by the spike detection method. This leads to a poor detection performance, in particular when the signal-to-noise ratio (SNR) is low. Further, the spikes are cut and aligned on some feature (e.g., peak position) as a preprocessing to the classification algorithm. However, overlapping spikes, which severely alter the spike waveform, are not identified as such. This leads to wrong alignments and false classifications by the sorting procedure.

4) There are two general approaches to extracellular recording with electrodes, namely acute and chronic recording methods. In acute recordings, individual electrodes are advanced into tissue at the beginning of each recording session anew, causing a compression of the tissue (Cham et al. 2005). During the experiment the tissue relaxes and the distances between the electrodes and neurons change; an effect called tissue drift (Branchaud et al. 2006). As a consequence, the shape of the measured waveforms and the characteristic of the background noise changes. Sorting algorithms which do not take into account such variations will perform poorly on data from acute recordings.

An approach based on *blind source separation* (*BSS*) *techniques* and addressing primarily problems 1) and 4) was presented in Takahashi et al. (2002), in which independent component analysis (ICA) was applied to multichannel data recorded by tetrodes (4 channels). Later, the method was adopted to data recorded by dodecatrodes (12 channels) (Takahashi and Sakurai 2005). However, both approaches had to deal with several new problems: Amongst others, time delays between the channels were not considered, biologically meaningless independent components had to be discarded manually, and different neuronal signals with

similar channel distributions could not be classified correctly. Furthermore, the methods can only be applied to data recorded with certain electrode types (i.e. tetrodes, dodecatrodes). The most severe problem, though, is the fact that the method cannot deal with data containing neuronal activity from a greater number of neurons than recording channels (over-completeness).

In this work, we present a realtime online spike sorting method based on the BSS idea, which explicitly addresses the four issues 1)–4), but also avoids the drawbacks of the method in Takahashi et al. (2002) and Takahashi and Sakurai (2005). In sum, a spike sorting algorithm for multi electrode data, which detects and resolves overlapping spikes with the same computational cost as non-overlapping spikes, is formulated. The method makes optimal use of an arbitrary number of simultaneously recorded channels and can even run on single channel data. Moreover, since spike detection, spike alignment, and spike classification are not separate parts, but are combined into a single algorithm, our method performs well on data with low SNR and containing many overlapping spikes. By incorporating a direct feedback, the algorithm adapts to varying spike shapes and to non-stationary noise characteristics. The algorithm is fully automatic and due to its linear and parallel computation steps it is ideally suited for realtime applications (see Fig. 4 for a summary of our method).

This paper is organized as follows: In Section 2 we present our method step by step. First, we briefly introduce linear filters. These filters were used in radar applications (Turin 1960), geophysics (Robinson and Treitel 1980) as well as for spike detection (Thakur et al. 2007; Vollgraf et al. 2005), but to our knowledge have not been applied to spike sorting yet. Moreover, in contrast to those studies, we do not directly apply a threshold to the filter outputs, but consider them as a new representation of the data. In this representation the spike sorting task can be handled as a well defined BSS problem, which we solve with a un-mixing technique we will refer to as "Deconfusion".

The evaluation of our method is done on two different datasets from real recordings and also on simulated data. The experimental setup, used equipment and the characteristic of recorded data are described in Section 3. The advantages and abilities of the method are demonstrated in Section 4. Evaluations of the spike detection performance are done using data from simultaneous intra- and extracellular recordings made in slices of rat visual cortex, and show that the proposed algorithm is superior to conventional spike detection methods. The noise robustness and the ability to successfully resolve overlapping spikes is evaluated systematically on synthetic data. Finally, the method is applied to data from extracellular recordings made in the prefrontal cortex of awake behaving macaques. This data is particularly challenging, because the tetrodes are not implanted chronically, but inserted before every experiment anew, leading to tissue drifts. We conclude that our method adopts to non-stationarities and also successfully resolves overlapping spikes in real data. A summary and a discussion of further improvements is given in Section 5.

## 2 Methods

### 2.1 Glossary of mathematical notation

We use a notation in which symbols for scalar quantities are represented by lower case letters, vectorial quantities are represented by bold lower case letters, and operators or matrices are represented by bold upper case letters. Matrices representing several vectorial quantities, but not linear transformations, are labeled with an additional bar. In Table 1 all important quantities are listed. The corresponding vectorial quantities are defined by concatenating all channel-wise defined vectors. As an example the vectorial template $\boldsymbol{\xi}^i$ of neuron $i$ is given by

$$\boldsymbol{\xi}^i := \left( \xi^i_{1,1} \cdots \xi^i_{1,T_f} \cdots \xi^i_{N,1} \cdots \xi^i_{N,T_f} \right)^\top$$

where the superscript $^\top$ means transpose. The vectors $\boldsymbol{v}^i$, $\boldsymbol{x}$, $\boldsymbol{f}^i$ are defined in the same way. Analogously,

**Table 1** Definitions of important quantities and their meaning

| | |
|---|---|
| $M$ | Number of neurons |
| $N$ | Number of recording channels ($N = 4$ for tetrodes) |
| $T$ | Period of time during which the templates are constant |
| $T_f$ | Length of the filters and templates in samples on one channel |
| $x_{k,t}$ | Measured signal on recording channel $k$ at time $t$ |
| $\bar{X}$ | Data matrix, $(\bar{X})_{k,t} = x_{k,t}$ |
| $v^i_t$ | Intrinsic signal of neuron $i$ at time $t$ |
| $\xi^i_{k,t}$ | Multichannel template (i.e. mean waveform) of neuron $i$ on channel $k$ at time $t$, $i = 1, ..., M$, $t = 1, ..., T_f$, $k = 1, ..., N$ |
| $s^i_{k,t}$ | $:= \sum_\tau v^i_{t-\tau} \xi^i_{k,\tau}$, noiseless, extracellular signal of neuron $i$ |
| $f^i_{k,t}$ | Filter (designed to detect template $i$) on channel $k$ at time $t$ |
| $y^i_t$ | $:= \bar{X} \star f^i := \sum_{k,\tau} x_{k,t+\tau} f^i_{k,\tau}$, filter output of filter $f^i$ |
| $\bar{Y}$ | Matrix whose $i$-th row is $\mathbf{y}_i^\top$ |
| $z^i$ | Output of filter $f^i$ after Deconfusion |
| $\bar{Z}$ | Matrix whose $i$-th row is $\mathbf{z}_i^\top$ |
| $H^i$ | Covariance matrix of $s^i$ |
| $\eta_{k,t}$ | Multivariate zero mean Gaussian noise |
| $C$ | Noise covariance matrix |
| $R$ | Data covariance matrix |

covariance matrices, e.g, the data covariance matrix $\boldsymbol{R}$, are defined as

$$\boldsymbol{R} := \begin{pmatrix} \boldsymbol{R}_{1,1} & \dots & \boldsymbol{R}_{1,N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{R}_{N,1} & \dots & \boldsymbol{R}_{N,N} \end{pmatrix}.$$

with $(\boldsymbol{R}_{k,l})_{t_1,t_2} := Cov(x_{k,t_1}, x_{l,t_2})$. $\boldsymbol{R}$ is a symmetric $N \cdot T_f$ by $N \cdot T_f$ Toeplitz matrix. Alternatively, it can be expressed as

$$\boldsymbol{R} = \sum_i \boldsymbol{H}^i + \boldsymbol{C}. \tag{1}$$

### 2.2 Generative model

We assume an explicit model for the neuronal data recorded extracellularly. The underlying assumptions are:

1. Each neuron generates a unique spike waveform $\boldsymbol{\xi}^i$ (called template), which is constant over a time period of length $T$.
2. All time series $\boldsymbol{v}^i$ of spike times of neuron $i$ (called spike trains) are statistically independent of the noise $\boldsymbol{\eta}$. Furthermore, these quantities sum up linearly.
3. The noise statistic is entirely captured by a covariance matrix $\boldsymbol{C}$.

As discussed extensively in Pouzat et al. (2002), these assumptions are reasonable and are used explicitly or implicitly in most spike sorting techniques. Consequently the measured data $\mathbf{x}$ can be expressed as

$$x_{k,t} = \sum_i \sum_\tau v_{t-\tau}^i \xi_{k,\tau}^i + \eta_{k,t} = \sum_i s_{k,t}^i + \eta_{k,t}. \tag{2}$$

The measured data are a convolution of the mean waveforms with the corresponding intrinsic spike trains corrupted by colored Gaussian noise (see also Fig. 1(a)–(c)).

### 2.3 Calculation of linear filters

Spike sorting is achieved when the intrinsic spike trains $\boldsymbol{v}^i$ are reconstructed from the measured data $\bar{\boldsymbol{X}}$. Since, according to the model assumptions, the data were generated by a convolution of intrinsic spike trains with fixed waveforms, the most straightforward procedure would be to apply a deconvolution on $\bar{\boldsymbol{X}}$ in order to retrieve $\boldsymbol{v}^i$. For an exact deconvolution a filter with an infinite impulse response is necessary. In general, such a filter is not stable and would amplify noise (Robinson and Treitel 1980). Nevertheless, a noise robust approximation for an exact deconvolution can be achieved with
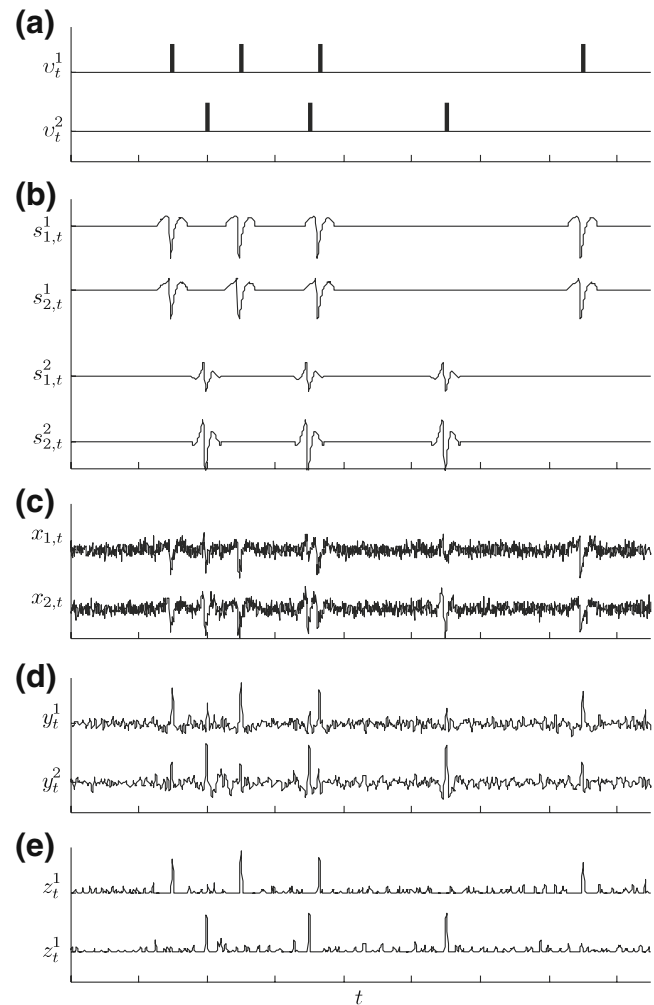


**Fig. 1** Sketch of the generative model (**a**–**c**) and the processing stages of the algorithm (**d**–**e**). (**a**) Spike trains of two neurons. (**b**) Simulated waveforms of each neuron on a hypothetical multi electrode (two recording channels, without noise). (**c**) Simulated data of a multi electrode recording. The signals of the two neurons and the noise are mixed linearly. (**d**) Filter output of two optimal linear filters. (**e**) Output after Deconfusion; for details see text

finite impulse response filters, to which we will refer as linear filter.

Let us briefly summarize the idea of these filters: The goal is to construct a set of filters $\{\boldsymbol{f}^1, \dots, \boldsymbol{f}^M\}$ such that each filter $\boldsymbol{f}^i$ has a well defined response of 1 to its matching template $\boldsymbol{\xi}^i$ at shift 0 (i.e. $\boldsymbol{\xi}^{i\top} \cdot \boldsymbol{f}^i = 1$), but minimal response to the rest of the data. This means that the spikes of neuron $i$ are the signal for filter $\boldsymbol{f}^i$ to detect but will be treated as noise by filter $\boldsymbol{f}^{j \neq i}$.

Incorporating these conditions leads to a constrained optimization problem

$$\boldsymbol{f}^i = \underset{\boldsymbol{f}^i}{\operatorname{argmin}} \, Var\left(\bar{\boldsymbol{X}} \star \boldsymbol{f}^i\right) \quad \text{subject to } \boldsymbol{\xi}^{i\top} \cdot \boldsymbol{f}^i = 1 \tag{3}$$

to which the solution are the desired filters (see Appendix A for a more detailed derivation). A major advantage is the fact that the mentioned optimization problem can be solved analytically. In particular, the filters are given by the following expression:

$$f^i = \frac{R^{-1}\xi^i}{\xi^{i\top} R^{-1}\xi^i} \qquad\qquad i = 1, ..., M \qquad (4)$$

where $R$ is the data covariance matrix defined in Section 2.1. Linear filters maximize the signal-to-noise ratio and minimize the sum of false negative and false positive detections, and are, therefore, optimal in this sense (Melvin 2004).

## 2.4 Filtering the data

Once the filters are calculated, they are cross-correlated with the measured signal, i.e. $\sum_{k,\tau} x_{k,\tau+t} f^i_{k,\tau} =: y^i_t$. Note that we do not have to pre-process the data with a whitening filter, but the filters can be applied directly to $\bar{X}$. This is because the noise statistics is already captured in the matrix $R$.

From a different point of view, the filtering just changes the representation of the templates. While in the original space the template $i$ was represented by $\xi^i$, its representation in the filter output space is given by the vectors $\xi^i \star f^j$, $j = 1, ..., M$, where $(\xi^i \star f^j)_t := \sum_{k,\tau} \xi^i_{k,t+\tau} f^j_{k,t}$, see also Fig. 2. This interpretation of filtering will be useful in the next section.

## 2.5 Deconfusion

The linear filters derived in Section 2.3 should suppress all signal components except their corresponding template with zero shift. Thus, the filter response to all templates (and their shifted variants) has to be minimal. This already leads to $(2T_f - 1) \cdot M$ minimization constraints; a number which is normally greater than the number of free variables of a filter which is $T_f \cdot N$. In addition, if the SNR is low, the noise covariance matrix $C$ dominates Eq. (1).

The lower the SNR, the less spikes from other neurons a filter will suppress. Thresholding of every filter output $y^i$ individually will, thus, lead to false positive detections. The idea is to de-correlated the filter output in order to achieve an improved spike detection and classification.

We have seen in the previous section that each template $\xi^i$ can be represented in the filter output by $M$ vectors $\xi^i \star f^j$, $j = 1, ..., M$. Since the detection and classification of the spikes is based on the detection of high positive peak values in the filter output (by

construction), all values below zero in the filter output are irrelevant, and thus, can be discarded. As a result, we ignore all values below zero by applying a half-wave rectification $I(x)$ to the filter output $\bar{Y}$, where

$$I(x) := \begin{cases} x, & x > 0 \\ 0, & x \le 0 \end{cases} \qquad\qquad (5)$$

The next step is to consider $I(\bar{Y})$ as a linear mixture of different sources, where every source is the intrinsic spike train $v^i$ of a neuron. Since there are as many filters as neurons, the dimension of the filter output space is equal to the number of neurons, and therefore, the detection and classification problem can be considered as a complete BSS problem. However, it is not guaranteed that the maximal response of filter $f^i$ to spikes from neuron $j$ will be at a shift of 0, i.e., when the filter and the template overlap entirely. This leads to the following model for the rectified filter output:

$$I(y^i_t) = \sum_j (A)_{i,j} \, v^j_{t+\tau_{i,j}} \qquad\qquad (6)$$

with $A$ being the mixture matrix, and $\tau_{i,j}$ being the shifts between the maximal response of filter $f^j$ to template $\xi^i$; i.e.,

$$(A)_{i,j} = \max_\tau \left\{ (\xi^i \star f^j)_\tau \right\}$$
$$\tau_{i,j} = \operatorname*{argmax}_\tau \left\{ (\xi^i \star f^j)_\tau \right\} \qquad\qquad (7)$$

where $(A)_{i,i} = 1$ and $\tau_{i,i} = 0 \ \forall i$ by construction. We want to reconstruct the sources $v^i$ by solving the corresponding inverse problem:

$$v^i_t \approx z^i_t = \sum_j (W)_{i,j} \, I(y^j_{t-\tau_{j,i}}) \qquad\qquad (8)$$

with $W = A^{-1}$. Here, the relation to ICA becomes clear, since this is a similar inverse problem ICA solves. In contrast to ICA, we do not have to estimate $W$ and $\tau_{i,j}$ from the data, but can calculate them directly from the responses (i.e. cross-correlation functions) of all filters to all templates, as illustrated in Fig. 2.

All steps of these procedure are summarized under the term "Deconfusion" (see also Fig. 1(d)–(e) for a schematic illustration). After Deconfusion the false responses of the filters to non-matching templates are suppressed (see Fig. 3). In principle, it is possible that the inverse problem in Eq. (8) is not exactly solvable, if the shifts are not consistent. Consistent shifts have to satisfy the following equation:

$$\tau_{j_1,k} - \tau_{j_1,i} = \tau_{j_2,k} - \tau_{j_2,i} \qquad\qquad \forall i, j_1, j_2, k \qquad (9)$$

A derivation is given in Appendix B. For arbitrary templates and data covariance structures, Eq. (9) can
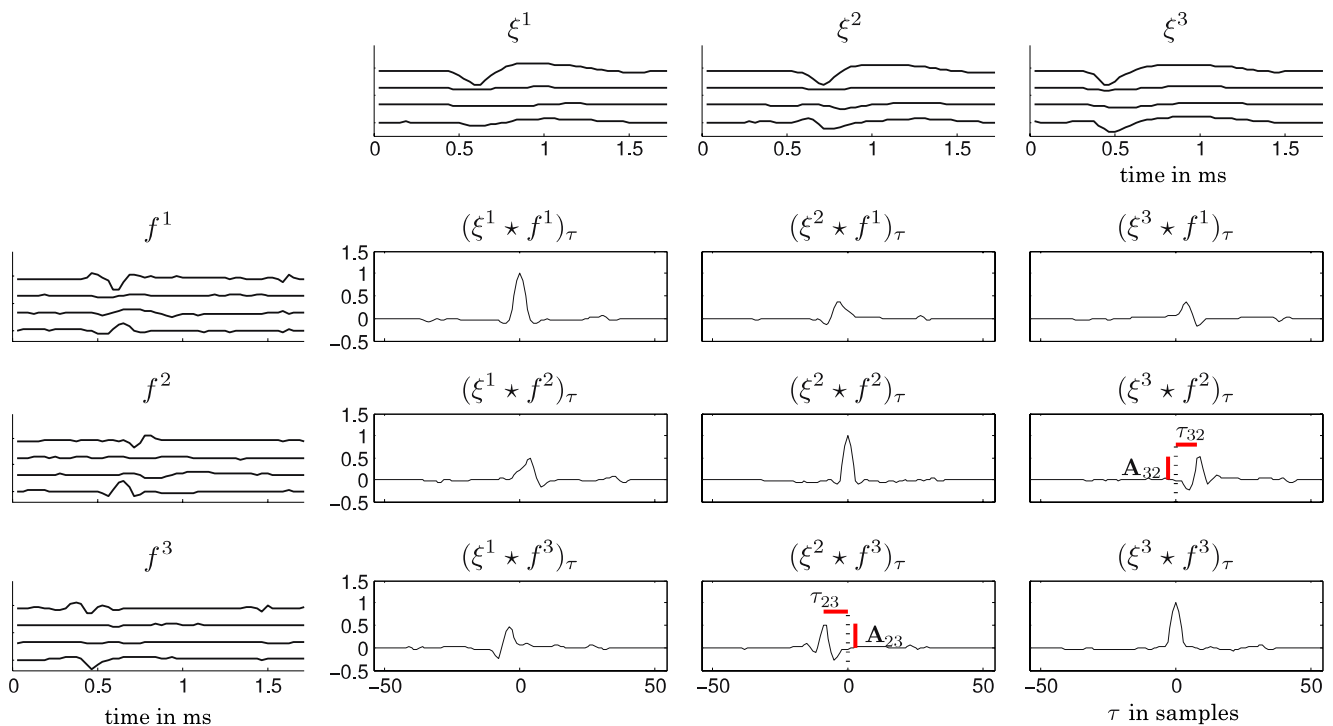
**Fig. 2** This figure exemplary illustrates the representation of the templates in the filter output space and the calculation of the Deconfusion parameters. In this example, three templates ($\xi^1$, $\xi^2$, $\xi^3$, *top row* of the figure) originating from tetrode recordings are used. The corresponding linear filters are calculated by Eq. (4) and are shown on the left. The 9 plots show the responses of the linear filters to the templates, i.e. the cross-correlations $\xi^i \star f^j$, $i, j = 1, 2, 3$. The template $\xi^i$ is now represented by the three vectors $\xi^i \star f^j$, $j = 1, 2, 3$. Although filter $f^i$ has a maximal response of 1 to template $\xi^i$, the filters do not provide an exact deconvolution, as the responses of filters $f^{j \neq i}$ to template $\xi^i$ are not equal to zero. However, since every template is represented on all filter output channels, the problem of extracting the signal from neuron $i$ can be viewed as a source separation problem. The entry at position $i, j$ of the mixing matrix $A$ is given by the maximal peak value of $\xi^i \star f^j$; exemplary $(A)_{2,3}$ and $(A)_{3,2}$ are shown. The shift indicates the position at which this maximal values occur; as an example the shifts $\tau_{2,3}$ and $\tau_{3,2}$ are shown

in principle be violated. However, with templates from real experiments we did not observe this to be a problem.

## 2.6 Spike detection and classification

In the final step, thresholding is applied to every row $i$ of $\bar{Z}$. Again, by construction we have only to consider positive peaks. All local maxima after a threshold crossing are identified as spiking times of neuron $i$. In this sense, spike detection and spike classification is performed simultaneously.

The threshold is set for each row of $\bar{Z}$ individually such that the total error of false negative and false positive detections is minimal. Amongst others, the threshold depends on the variance of the noise, on the Deconfusion output, and on the firing frequencies of the neurons. A detailed derivation is given in Appendix C.

## 2.7 Artifact detection

Artifacts were removed from our data in two ways. First, all periods during which the animal had to perform a physical task (e.g., pressing a button) were not considered for further analysis. Secondly, for each period of length 10 ms the number of zero-crossings on each data channel was counted and summed up. All periods, in which this number was below 10% of the maximal number of possible zero crossings, were not considered for further analysis. This second type of heuristic removal aims at eliminating artifacts caused by oscillations of the electrode shaft inside the guiding tube (e.g., caused by movement of the animal).

## 2.8 Noise estimation

The noise covariance matrix $C$ is determined by calculating the auto- and cross correlation functions of every channel. Only data points which were not part of any spike nor any artifact period, were used for the calcu-
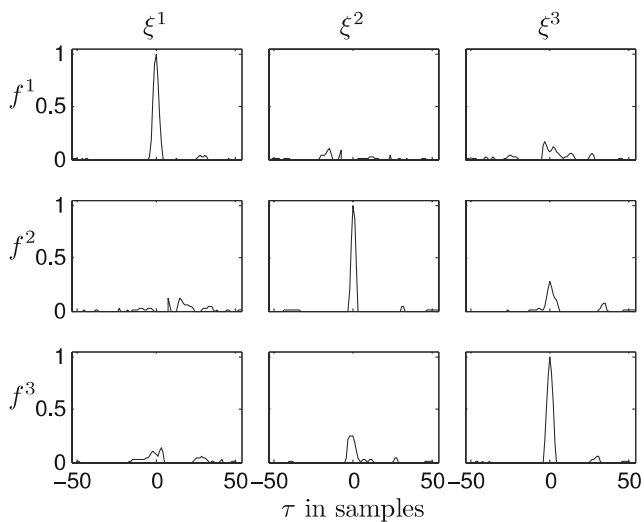
**Fig. 3** The figure shows the effect of Deconfusion on the filter outputs. The input for Deconfusion were the filter responses $\xi^i \star f^j$, $i, j = 1, 2, 3$ shown in Fig. 2. After Deconfusion the signal of neuron $i$ is mainly present on the output channel $i$

lation. The noise covariance matrix is needed for the initialization phase, see Section 2.10, and for evaluation of the sorting result on real data, see Section 4.2.3.

### 2.9 Adaptation

Due to tissue relaxations the measured waveforms change over time as the relative distance between the multi electrode and the neurons change. In order to track these changes we re-estimate the templates as well as the data covariance matrix after every time period of length $T$. Each template $\xi^i$ is re-estimated as the mean of the last 350 spikes (see Section 5 for a discussion of this value) detected from neuron $i$; whereas the spikes of neuron $i$ are aligned on the maximal peak of the response of filter $f^i$. For the re-estimation only spikes which were classified by our method as non-overlapping spikes are used. The data covariance matrix is re-estimated from the last 30 s of the recordings and the linear filters are re-calculated. Consequently, the Deconfusion and the thresholds are re-computed as well. In Section 4.2.3 we show that we can indeed track drifts with this approach.

Templates whose SNR decreases over time might be a concern. By constantly adapting the template, finally, there is a risk of getting a template which is very close to the noise signature, and the corresponding filter will detect pure noise. This can be prevented by removing filters at the appropriate moment. Consequently, we stop tracking templates whose SNR drops below 0.65. This value proved to be appropriate during simulations (see Section 4.2.2).

### 2.10 Initialization phase

Most of the analysis done in the precedent sections was based on the assumption of known initial templates. Hence, before applying our method, one needs an initialization phase during which the templates are found. In principle, any supervised or unsupervised learning method can be applied.

We want to emphasize that the initialization phase is only necessary at the beginning of a recording session (Fig. 4): Once the initial templates are estimated, the main algorithm runs online. Furthermore, because of the feedback described in Section 2.9, the initialization does not have to be very accurate, as the templates are re-estimated after every period of length $T$. Usually we used an initialization phase of about 30 s in our real recordings (Section 3.3). This time window is short enough so that the templates change only very slightly in time and can, therefore, be clustered reliably, but long enough to acquire enough spikes to estimate robustly the mean waveforms.

#### 2.10.1 Initial spike detection and initial spike alignment

During the initialization phase spike detection can be done with any conventional technique. We used an energy based approach, since it usually delivers a better performance than other methods (Mtetwa and Smith 2006; Obeid and Wolf 2004).

In particular, we applied the MTEO detector (see Section 4.1 for definition) with $k$-values [1, 3, 5] to each recording channel separately and set the threshold to 3.5 times the median of its output. Spike periods were defined as intervals of length 1.5 ms, in which the output of the MTEO detector exceeded the threshold value at least once.

Correct spike alignment is crucial for a good clustering result. While in many studies an alignment based on the maximal and/or minimal peak value of a spike is used, again, methods based on the energy of a spike usually yield better results (Fee et al. 1996a). After cutting out all spikes around the peak of the detector, we used the following algorithm for alignment:

1. Calculate the average template over *all* spikes
2. Minimize the energy difference between every spike and the template by shifting the spikes
3. Repeat until convergence or a maximal number of iterations is reached

In our experiments described in Section 3.3 the average number of spikes in the first 30 s of recordings is around 2500 and convergence is obtained after 15 to 20 iterations.
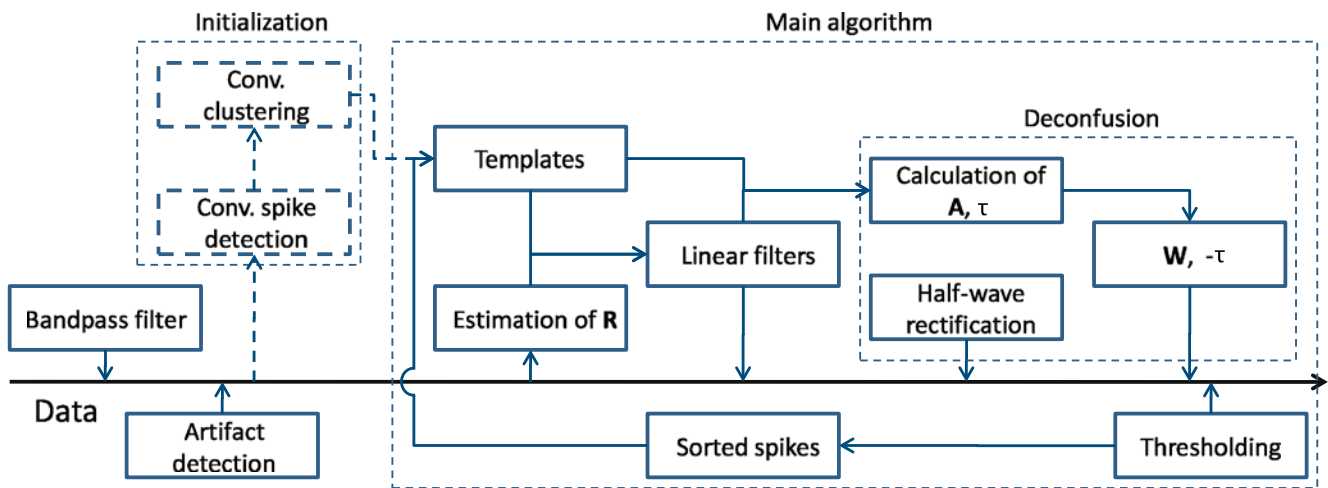
**Fig. 4** Schematic illustration of the way data is processed: The data is bandpass filtered and periods containing artifacts are excluded from further analysis (Section 2.7). During the initialization phase a conventional spike detection and clustering method is used to determine initial templates (Section 2.10). The data covariance matrix **R** is estimated and for every template the corresponding linear filter is calculated as described in Section 2.3. The data are filtered and all values in the filter output below zero are set to zero (half-wave rectification). From all filter responses to all templates the un-mixing transformation is determined and applied to the processed data (Section 2.5). A threshold is applied to the Deconfusion output resulting in simultaneous spike detection and classification. The newly found spikes are used to re-estimated the templates. Also the covariance matrix of the data is re-calculated after regular time intervals (Section 2.9)

### 2.10.2 Initial clustering

Although a broad range of sophisticated clustering algorithms is available, we used a standard approach, since a very accurate initialization is not crucial for our method. The aligned spikes are whitened (e.g., see Pouzat et al. 2002) and projected into the space of the first 6 principle components. The clustering consists of a Gaussian mixture model in combination with the Expectation-Maximization algorithm (Xu and Wunsch 2005). For every number of cluster means between 1 and 15 the clustering procedure is executed 3 times with random initial means. The covariance matrices are fixed to 2.5 times the identity matrix. The run and the number of means with the highest score according to the Bayesian inference criterion (Xu and Wunsch 2005) are selected as initialization for the main algorithm.

### 2.11 Signal-to-noise ratio (SNR)

The SNR is a scalar value which is an indicator for the difficulty of detecting a signal in noisy data. In this sense, the SNR definition should be dependent on the method used for signal detection. Several definitions of the SNR are used in the spike sorting literature. A very common one is to define the SNR by some maximal value, e.g., the maximal amplitude, the maximal difference in amplitudes (peak to peak distance), or

the maximum of the absolute value of the amplitude, divided by the variance of noise $\sigma^2$, i.e.,

$$\mathrm{SNR}_p(\boldsymbol{\xi}) := \sqrt{\frac{(\text{Extremum value of } \boldsymbol{\xi})^2}{\sigma^2}}$$

(e.g. see Choi et al. 2006). Another current definition for the SNR is based on the energy of a signal, i.e.,

$$\mathrm{SNR}_e(\boldsymbol{\xi}) := \sqrt{\frac{\boldsymbol{\xi}^2}{N \cdot T_f \cdot \sigma^2}}$$

(e.g. see Rutishauser et al. 2006). We introduce a definition of SNR which is based on the Mahalanobis distance of a template $\boldsymbol{\xi}$ to zero:

$$\mathrm{SNR}_m(\boldsymbol{\xi}) := \sqrt{\frac{\boldsymbol{\xi}^\top \boldsymbol{C}^{-1} \boldsymbol{\xi}}{N \cdot T_f}}. \tag{10}$$

In the special case of single electrode data and of 1-dimensional templates ($T_f = 1$), all SNR definitions are equivalent. To show that $\mathrm{SNR}_m$ is an appropriate SNR definition for the linear filters, while the other definitions are in contradiction with the meaning of signal-to-noise ratio, we simulated datasets containing a single neuron, which fired according to a Poisson statistic, and

a noise covariance matrix $C(\alpha) := (1 - \alpha) \cdot \mathbf{1} + \alpha \cdot \frac{C_{exp}}{\sigma^2}$, where $\mathbf{1}$ denotes the identity matrix, and $C_{exp}$ is a noise covariance matrix from one of the experiments described in Section 3.1, with $(C_{exp})_{i,i} = \sigma^2$ for all $i$. The used template was extracted from the same experiment. We simulated datasets for ten different $\alpha$ values between 0 and 1. The $SNR_m$ decreased with increasing $\alpha$, and consistently the detection performance of our method decreased, see Fig. 5. Note that $SNR_p = SNR_e = 1$ for all $\alpha$ values, which means that those definitions are inappropriate for the proposed method. Nevertheless, we always provide values for all three definitions of SNR in order to allow comparisons with other publications.

## 3 Experiments and datasets

For the performance evaluation of our method, three different datasets were used. All experiments were performed in accordance with German law for the protection of experimental animals, approved by the local authorities ("Regierungspräsidium"), and are in full compliance with the guidelines of the European Community (EUVD 86/609/EEC) for the care and use of laboratory animals.

### 3.1 Simultaneous intra/extra-cellular recordings

The experiments were done in acute brain slices from Long Evans rats (P17–P25). In every experiment a pyramidal cell from visual cortex, Layer 3 or 5 depending on the experiment, was simultaneously recorded intracellularly and extracellularly. Extracellular spike waveforms were recorded using a 4-core-Multifiber Electrode (Tetrode) from Thomas RECORDING GmbH, Germany. The cell was intracellularly stimulated by a current injection (varying from experiment to experiment between 80 pA and 350 pA). Extracellular recordings were sampled at 28 kHz and filtered with a bandpass FIR filter (300 Hz to 5000 Hz).

The intracellularly recorded spikes were detected using a manually set threshold on the membrane potential. The threshold crossings in the membrane potential were used as triggers to cut out periods from the extracellular recordings (2 ms before and 5 ms after the trigger). In total, data was recorded from 6 different cells, which resulted in 9957 intracellularly detected spikes. For analysis only the recording channel with the highest SNR was considered. The SNR of the different experiments varied from $SNR_m = 0.20$ ($SNR_p = 0.79$, $SNR_e = 0.39$) to $SNR_m = 2.37$ ($SNR_p = 7.09$, $SNR_e =$
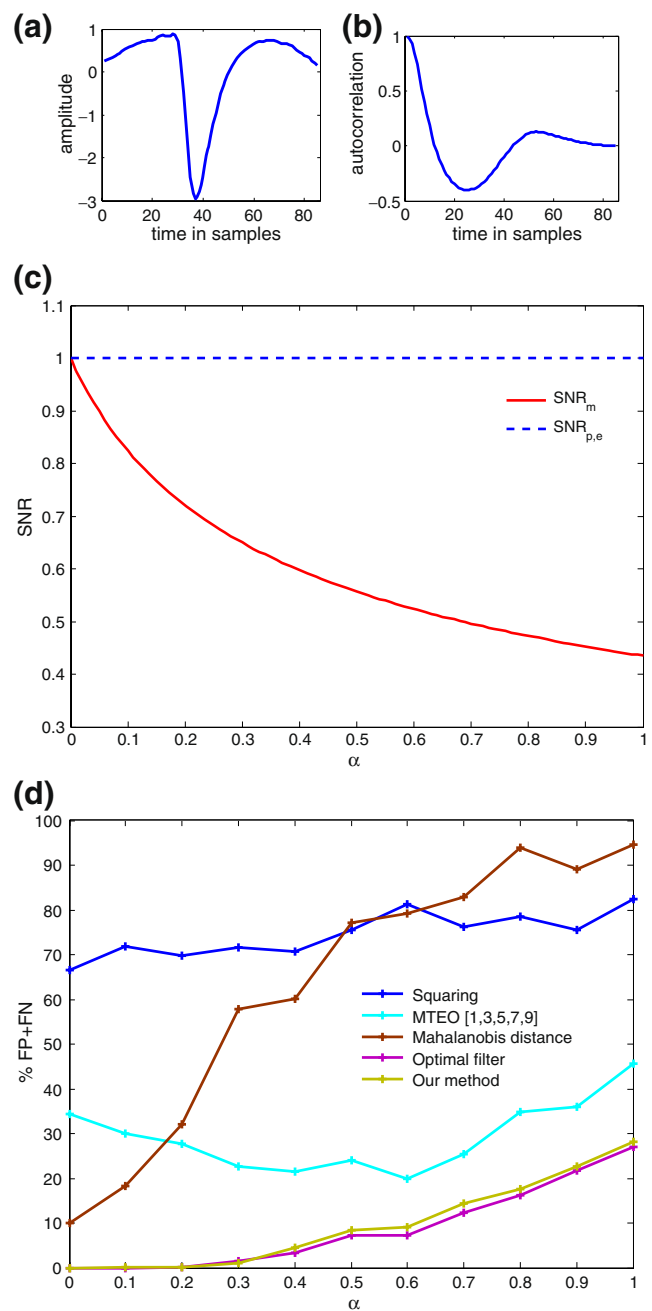


**Fig. 5** (**a**) Template $\xi$ (in arbitrary units) used for the simulations. (**b**) Noise autocorrelation function of the same experiment from which the template was extracted. This autocorrelation was used to calculate $C$. (**c**) Plot of $SNR_p(\xi)$, of $SNR_e(\xi)$ and of $SNR_m(\xi)$ in dependence of $\alpha$ (see text for definition). (**d**) Average detection performance of different spike detection methods (described in Section 4.1) for different values of $\alpha$. For each $\alpha$ value the average was done over 5 datasets, each with a noise covariance matrix $C(\alpha)$ (see text for definition)

3.64). A short period of recordings with a moderate SNR ($SNR_m = 1.16$, $SNR_p = 4.3$, $SNR_e = 1.97$) is shown in Fig. 6, top row.
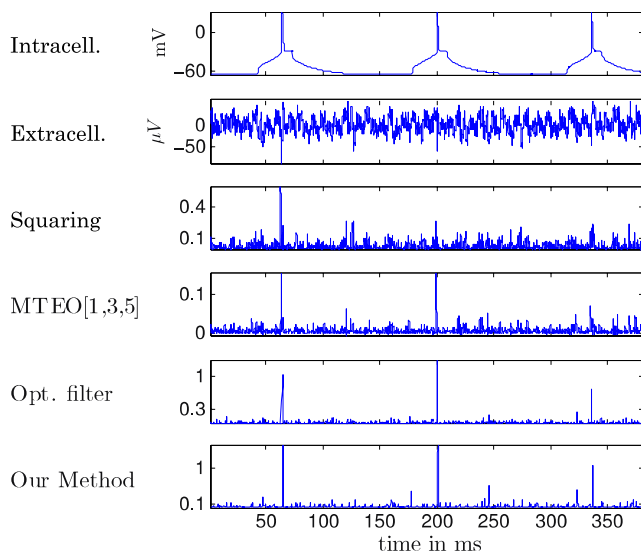
**Fig. 6** A short piece (approx. 400 ms) of extracellularly recorded data from slices of rat visual cortex is processed with different spike detection techniques (rows 3–6). Data were recorded simultaneously intracellularly (*first row*) and extracellularly with a tetrode. In this experiment the cell was repeatedly stimulated with 30 ms pulses of 350 pA current injection to elicit action potentials. Only the tetrode channel with the highest SNR was used for further analysis (*second row*)

## 3.2 Simulated data

The artificially generated data simulates a single channel recording of 15 s length at a sample frequency of 32 kHz containing activity from three neurons. Every dataset contained exactly 750 equidistantly distributed spikes of every neuron, which corresponds to a firing frequency of 50 Hz. The three used templates were extracted from the recordings described in Section 3.1 and had a length of 2.1 ms. The noise was generated by an ARMA model (Hayes 1996) approximating the noise characteristic shown in Fig. 5(b).

### 3.2.1 Dataset with overlapping spikes

The relative number of overlapping spikes was systematically varied from 1% up to 50%. 75% of all overlapping spikes consist of overlaps between two templates (25% for each combination), and 25% of all overlapping spikes consist of overlaps between all three templates. The amount of overlap, i.e., how much the templates overlap, is distributed according to a uniform distribution on the interval [1/3, 1]. The SNR was kept constant for all overlapping ratios, namely, all three templates were scaled to an equal SNR, which was $SNR_m = 1.2$. This corresponds to $SNR_p = 5.42$ and $SNR_e = 2.12$ (average values over the three templates).

### 3.2.2 Dataset with SNR variation

The $SNR_m$ was systematically varied from 0.6 to 1.4 (which is equivalent to 2.71 to 6.32 average $SNR_p$ and 1.06 to 2.48 average $SNR_e$). The amount of overlapping spikes was constant and set to 7%, which is approximatively the overlap ratio resulting by chance under the assumption of independent spike trains.

The over-completeness, the equal SNR of all three templates, and the presence of overlapping spikes make these datasets particularly challenging.

## 3.3 Acute recordings

Tetrodes were placed in ventral prefrontal cortex for individual recording sessions, sampling data from the same region across experiments. Recordings were performed simultaneously from up to 16 adjacent sites with an array of individually movable fiber micro-tetrodes (Eckhorn and Thomas 1993). Recording positions of individual tetrodes were manually chosen to maximize the recorded activity and the signal quality. Data were sampled at 32 kHz and bandpass filtered between 0.5 kHz and 10 kHz.

Neuronal activity was recorded while 2 macaque monkeys performed a visual short-term memory task. The task required the monkeys to compare a test stimulus to a sample stimulus presented after a 3 s long delay and to decide by differential button press whether both stimuli were the same or not. Stimuli consisted of 20 different pictures of fruits and vegetables which were presented for 0.5 s (test stimulus) or for 2 s (sample stimulus). Correct responses were rewarded. Match and non-match trials were randomly presented with an equal probability. This experimental setup was presented in Wu et al. (2008).

Approximately, the monkeys perform 2000 trials per session, which is equivalent to almost 4 h of recording time. For the evaluation of our algorithm only the first 5 s of every trial were processed, as the remaining data might contain severe artifacts caused by the monkey's movement.

## 4 Results and discussion

The performance of a spike sorting method depends on its capability to detect spikes and to assign every spike to a putative neuron. As described in Section 2.6, our method achieves both simultaneously. We evaluated the performance of our approach, first, as a pure detection method, and then, as a combined detection

and classification technique. In both categories we compared it against techniques commonly used.

### 4.1 Spike detection performance

The evaluation was done on the in-vitro dataset described in Section 3.1. Although the extracellular signal was recorded with a tetrode, we used only one recording channel for further analysis, since most conventional spike detection methods are only defined for single channel data. The detectors used are:

1. Mahalanobis distance: This method is described in Rebrik et al. (1999). In brief, periods having a greater Mahalanobis distance to zero than a certain threshold are identified as spikes. The noise covariance matrix was estimated from data pieces in which the neuron was not stimulated. The size of the matrix was chosen to match the observed length of spikes in the experiment and was then applied window-wise. Local maxima crossing the threshold are identified as spike times.
2. Squaring: The raw data is squared and normalized. Local maxima crossing the threshold are identified as spike times. In case of an one-dimensional noise covariance matrix, this method is equivalent to the method "Mahalanobis distance".
3. Squaring smoothed: A Savitzky-Golay filter of span 5 and order 2 is additionally applied to the output of the method "squaring". This method is very similar to the one used in Rutishauser et al. (2006).
4. MTEO: This method is described in Choi et al. (2006). In brief, the data is smoothed with a Hamming window and a quantity (which depends on parameters $k$) related to the energy of the signal is computed. We used two parameter sets for this method, one with $k$-values of $[1, 3, 5]$ and one with $k$-values of $[1, 3, 5, 7, 9]$.
5. Optimal filter: Since the occurrence of the spikes is known (due to the intra-cellular recording), the optimal filter is calculated using the average waveform of all spikes of the recorded neuron.
6. Our method: In the case of a single neuron, our spike sorting method corresponds to a single "estimated filter" detector, i.e., the initial filter is calculated using the average waveform of all spikes found by the MTEO [1, 3, 5] with a threshold set to 3.5 times the median of its output.

A short piece of the recordings and some of the corresponding detector outputs are shown in Fig. 6.

We compared the performance of the different spike detection methods using receiver operating characteristic (ROC) curves. For every detector the threshold
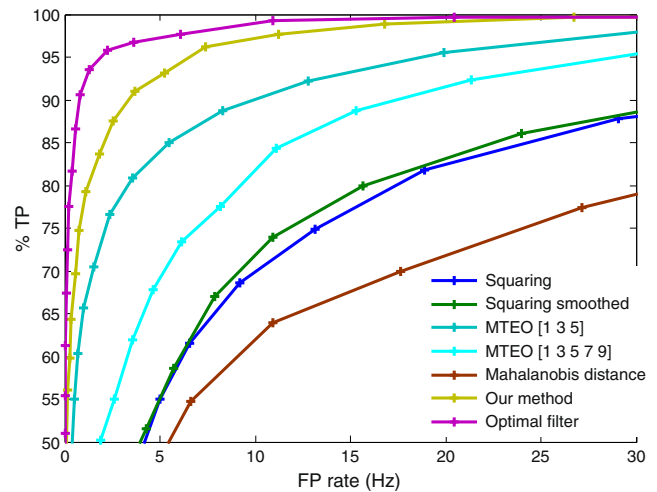


**Fig. 7** ROC curves for different spike detection methods. In this experiment $SNR_m = 0.81$, $SNR_p = 3.28$, $SNR_e = 1.63$

is systematically varied between 0, resulting in zero false negative detections (FN), and the minimal value which does not detect any spikes; i.e., zero true positive detections (TP). For every threshold the percentage of TP is plotted against the false positive (FP) rate. Such a curve is shown for one exemplary experiment in Fig. 7. The curve for the best possible detector (i.e. no FP, but 100% TP detections) would pass through the point (0, 100). The area under such a curve (AUC) is, thus, a measure for the performance of a detector. The normalized AUC values for the area up to 30 Hz of FPs of all detectors averaged over all available datasets are shown in Table 2. Although only the average performance is presented, our method and the optimal linear filter also achieved higher scores on every individual dataset described in Section 3.1. In all experiments the optimal filter was superior to the other detectors, while our method scored second with a very similar performance. This shows that taking into account the full

**Table 2** Average normalized area under the curve for each evaluated spike detection method

| Method | Average normalized AUC |
| --- | --- |
| Mahalanobis distance | 0.70827 |
| Squaring | 0.76371 |
| Squaring smoothed | 0.77073 |
| MTEO[1 3 5] | 0.78501 |
| MTEO[1 3 5 7 9] | 0.77255 |
| Optimal filter | 0.81957 |
| Our method | 0.80887 |

The averaging was done over all 6 datasets described in Section 3.1. The AUC values were computed up to a FP frequency of 30 Hz and normalized by 3000 which is the AUC of a perfect detector
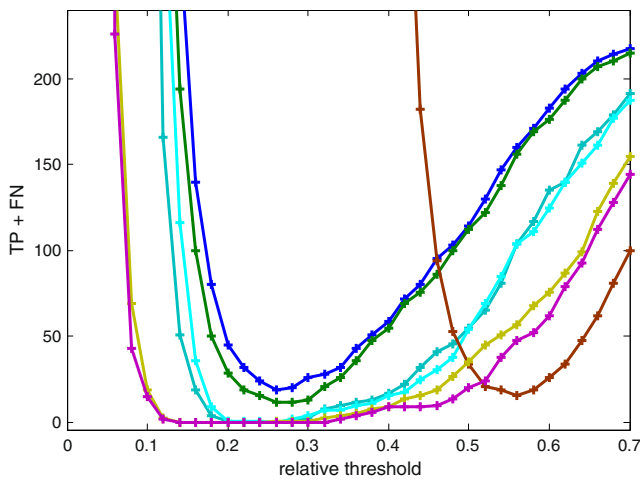
**Fig. 8** Number of errors for each spike detection method in respect to varying thresholds. The color coding is the same as in Fig. 7. For each detection method the maximal threshold was determined and normalized to 1. The maximal threshold is defined as the smallest threshold so that no spikes are detected. The total error is plotted in dependence of threshold values equally sampled from the interval [0, maximal threshold]. The total error for the linear filter increases slower than for the other methods, when the threshold deviates from the optimal threshold. For this evaluation a dataset from the recordings described in Section 3.1 was used with $SNR_m = 1.39$, $SNR_p = 6.31$, $SNR_e = 3.04$

**Table 3** Average performance of the proposed method for non overlapping and overlapping spikes

|   | N | A | B | C | AB | AC | BC | ABC |
|---|---|---|---|---|----|----|----|-----|
| A | 0.0 | **96.0** | 0.1 | 0.0 | **91.7** | **93.5** | 1.7 | **92.0** |
| B | 0.0 | 0.0 | **98.2** | 0.1 | 87.4 | 9.7 | **92.8** | 87.2 |
| C | 0.0 | 0.0 | 0.0 | **97.8** | 1.1 | **92.0** | **92.1** | 88.7 |

Each column represents the true category of events detected as spikes (e.g. "N" meaning "noise", "AB" meaning an overlapping spike of template A and template B, etc.), while each row represents the category to which they were assigned by our algorithm. Each total number of classifications was divided by the number of corresponding spike events, resulting in a percentage value. The bold numbers represent the percentage of correct classifications. The table shows the average performance over 10 datasets with an overlap ratio of 40% (see Section 3.2.1). For a systematic evaluation over different overlap ratios the absolute numbers of the correct classifications were added and divided by the total number of inserted spikes; see Fig. 9

waveforms as well as the data statistic always greatly improves the detection performance. The optimal linear filter was included into the evaluation to provide an upper bound on the performance one can achieve with our method. Our method offers another advantage for the detection of spikes, namely a bigger robustness to threshold variations, see Fig. 8. This means that a deviation from the optimal threshold has a less drastic impact on the total error (FP + FN) than for the other methods.

### 4.2 Spike sorting performance

#### 4.2.1 Resolution of overlapping spikes

We recall that the applied operations to the recorded data could be summarized in Eq. (8). The cross-correlation between the filters and the data is a linear operation. The following Deconfusion consists of a half-wave rectification, which is a non-linear operation, but affects only noise and not the action potentials (represented in the filter output), and the un-mixing, which is linear again. Hence, one can expect that if the superposition of spike waveforms is also linear, overlaps should be resolved successfully. We validated this assumption on the dataset described in Section 3.2.1. The algorithm was executed in the same

way as described in Section 2. In order to allow the method to adapt (Section 2.9), the method was iterated 5 times on the same dataset. We also compared the performance of our method to those of two popular clustering based offline methods, one of them being the method described in Section 2.10.2, which will be abbreviated as "GMM". Since this is also the method which is used for initialization of our algorithm, the comparison with GMM directly provides information about the improvements in sorting when our method is used.

The other algorithm, called "KlustaKwik", was explicitly developed for clustering neuronal data and was
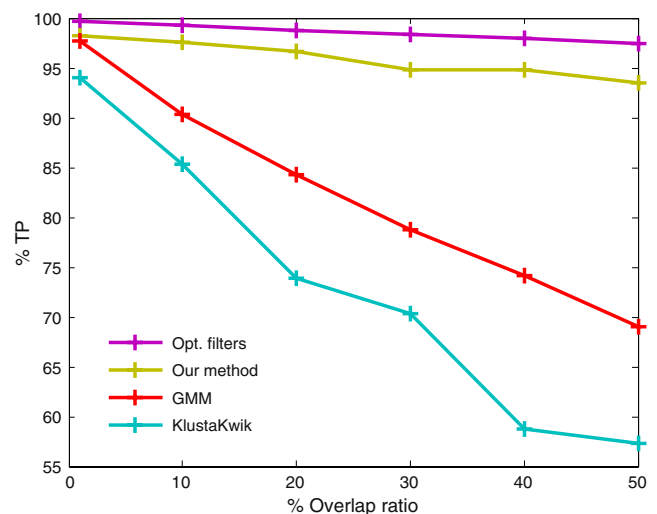


**Fig. 9** Average performance of the different spike sorting methods over 10 simulations. The x-axis indicates the overlap ratio, i.e. the relative number of overlapping spikes (see Section 3.2) while the y-axis represents the correct classifications in percentage (true positives divided by total number of spikes)

**Table 4** Same evaluation as in Table 3, but for the method "GMM" described in Section 2.10

|   | N | A | B | C | AB | AC | BC | ABC |
|---|---|---|---|---|----|----|----|-----|
| A | 0 | **81.0** | 0.0 | 0.1 | **27.8** | 27.3 | 0.4 | **21.5** |
| B | 0.2 | 14.5 | **100.0** | 0.6 | **68.0** | 4.2 | 45.0 | 42.7 |
| C | 0.1 | 4.4 | 0.1 | **99.4** | 4.7 | 69.0 | 53.2 | 41.7 |

The method sorts non overlapping spikes well, but has difficulties in resolving overlapping spikes

first introduced in Harris et al. (2000). The clustering parameters were set to their default values. Spike detection and alignment was done in the same way as described in Section 2.10.1. To provide an upper bound on the performance our approach could achieve, we included the evaluation with the optimal filters calculated directly from the real templates. Note that other existing, purely clustering-based sorting methods, either in the PCA space or in the original data space, would perform similarly to GMM and KlustaKwik.

For the evaluation the relative number of TP was counted (Tables 3 and 4).

The simulations show that our method indeed resolves overlapping spikes and outperforms the clustering based methods; see Fig. 9. Our method works even for datasets with a large amount of overlapping spikes, and the performance is close to the theoretical bound of this approach. On the other hand, the performance of the purely clustering based methods rapidly decreases with an increasing amount of overlapping spikes. Overlapping spikes are mostly detected as single events by conventional spike detection techniques, which leads to a high FN rate. Furthermore, since the waveforms of overlapping spikes are distorted, their distances to the corresponding cluster means are large, making it difficult to assign them to a neuron. This results in a low TP score for clustering based methods.

### 4.2.2 Performance for various SNR

The evaluation on the dataset with a varying SNR (see Section 3.2.2) was done in the same way as in the previous section. The results are shown in Fig. 10. The performance of the clustering based methods is severely affected by a low SNR. The performance of the proposed method follows the one of the GMM algorithm, since it relies on its output for initialization. Nevertheless, our method is always superior to it. Because of the rapid decrease in performance from a SNR level of 0.7 to an SNR level of 0.6, we stopped the algorithm from detecting spikes for templates with a lower SNR than 0.65 in real recordings by deleting the corresponding templates and filters. In contrast, the
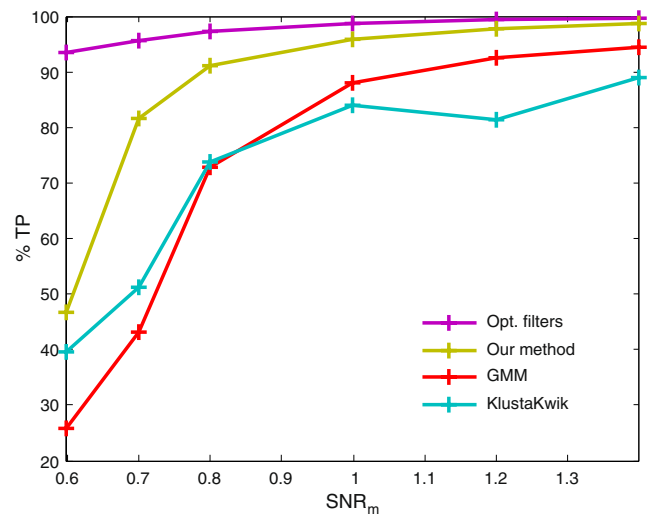


**Fig. 10** Average performance of the different spike sorting methods over 10 simulations in respect to various SNR levels. Note that the performance of the proposed method degrades with the performance of the GMM algorithm. This is because the output of the GMM is used as the initialization for our method. However, the our method is always superior to it. Low SNRs do not severely affect the performance of the optimal filter

optimal filter method is only slightly affected by a low SNR level, indicating that a more elaborate initialization would increase the performance of the proposed method on datasets with very low SNRs.

### 4.2.3 Performance on experimental data

We applied our method to data recorded in the prefrontal cortex of monkeys performing a short-term memory task as described in Section 3.3. For illustrative purposes, we show the results obtained by processing data from one tetrode, since the qualitative outcomes from processing other tetrodes and different recording sessions are similar.

For the initialization phase we used the first 7 trials of the recording. The initial spike detection and clustering was done as described in Section 2.10, resulting in a total of 3219 detected spikes, which were assigned to 8 clusters. This basic clustering was used as an initialization for the main algorithm, which was executed in the same way and with the same parameters as described in Section 2 (see also Fig. 4 for a summarization). The 7 trials used for initialization were also processed with the main method in order to improve the sorting quality.

The templates after the first 90 trials are shown in Fig. 11, and seem to be reasonable by visual inspection of an expert. In total, our method found almost 200000 spikes (57111, 18060, 50724, 51709, 3974, 7057, 444, 10915 for each template). Two well-established tests
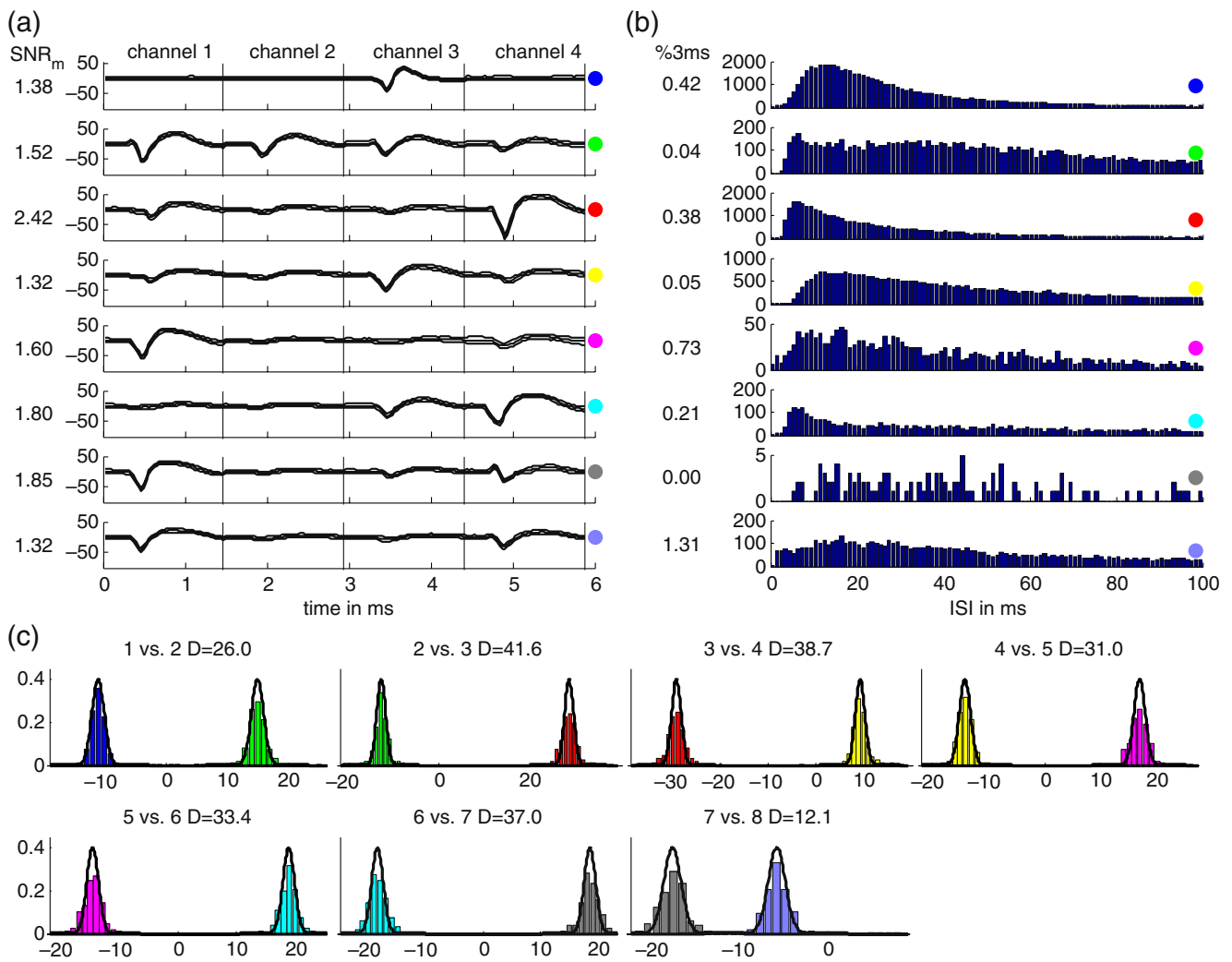
**Fig. 11** (**a**) Plot of the concatenated templates and their standard deviation. For the averaging all detected spikes from trial 50 to trial 90 were used. The vertical lines indicate the concatenation points, while the colored dots on the right serve as a label. On the left, the $SNR_m$ value is shown, the channel length of the template being $T_f = 47$ and $N = 4$. The corresponding $SNR_p$ are (10.06, 13.28, 21.82, 11.57, 13.12, 13.32, 14.27, 10.34), and the $SNR_e$ are (1.84, 3.73, 4.22, 2.91, 2.90, 3.45, 2.99, 2.53), respectively. (**b**) Histograms of the inter-spike interval distributions with a bin size of 1 ms. The numbers on the left represent the percentage of spikes with an inter-spike interval of less than 3 ms. (**c**) Projection test of the found clusters. The fit (solid line) represents a Gaussian distribution whose mean is the corresponding template and with variance 1. The D value indicates the distance in standard deviations between the means. Note that in case of acute recordings, the waveforms change over time and thus the projection test is only meaningful for short time intervals; see also Fig. 13. For the projection test the same spikes as in (**a**) were used

to quantitatively asses the sorting quality of a method performing on real data are the inter spike interval distribution and the projection test (Rutishauser et al. 2006; Pouzat et al. 2002); the evaluation of our sorting with both tests is shown in Fig. 11. The relative number of spikes during the first 3 ms is smaller than 1.5% for all neurons, implying that the refractory period is respected. On the other hand, the projection test verifies that the spikes of a single neuron have not been artificially split by the sorting algorithm into multiple clusters or that spikes from multiple neurons

are assigned to the same cluster. The sorting of our method also passes the projection test since the cluster distributions do not overlap and are close to the theoretical prediction of a normal distribution with a variance 1. In sum, the good results of these two tests imply that the found clusters are well separated and indeed correspond to single neurons, as well as that the assumptions made in Section 2.2 are justified.

Since we inserted the tetrodes before every experiment anew, our algorithm has to deal with the variability in the data caused by tissue drifts. The adaption
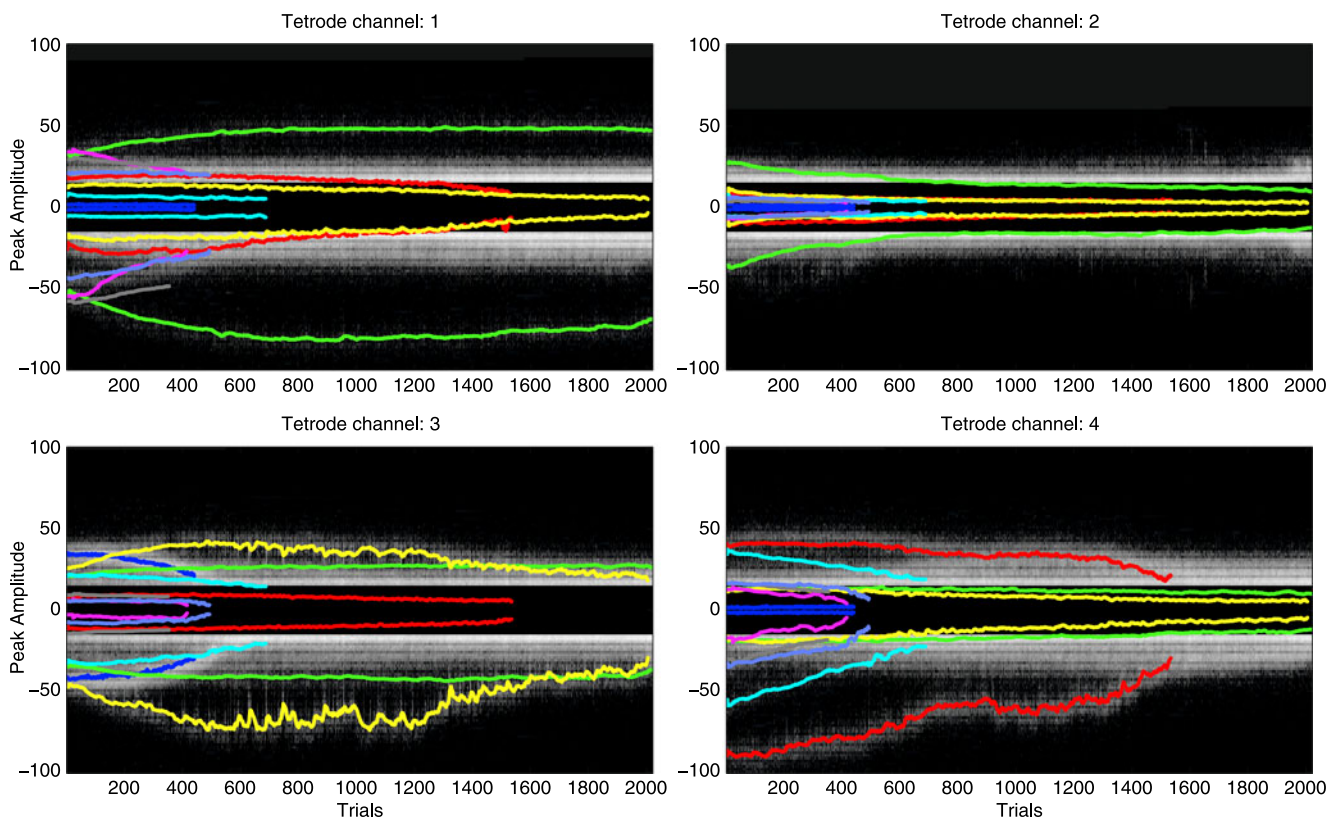
**Fig. 12** Effects of tissue drift on the amplitudes of all templates over the whole experiment. For every recording channel of the tetrode and for every trial a peak amplitude histogram was calculated. In every trial the number of local extrema of 50 samples width and a certain amplitude interval was counted for positive and negative peaks independently and normalized by the trial length, giving an estimate for the rate of spikes of that amplitude. Shown is the logarithm of that count, where light pixels correspond to high counts. Small amplitude peaks were ignored because these are strongly effected by noise. A neuron with a high SNR should be visible as two light horizontal bands, one with a high positive amplitude and one with a high negative amplitude, respectively. Superimposed are the minimal and maximal amplitudes of the found templates in every trial. The color code is the same as in Fig. 11. The plot reveals that the amplitudes of the spikes change drastically over time. Due to the feedback described in Section 2.9 the algorithm adapts to this changes and successfully tracks the neurons

procedure described in Section 2.9 was executed after every trial and adapted the algorithm correspondingly. The time period over which the templates were assumed to be constant was set to $T = 5$ s.[1] As a result, 2 neurons could be tracked from the beginning to the very end of the experiment, see Fig. 12. The other templates were deleted earlier, since their $\mathrm{SNR}_m$ dropped below 0.65. The importance of taking temporal variations for sorting into account is demonstrated in Fig. 13. If the drift is not accounted for, the clusters are elongated and their spread is larger, making any classification more difficult.

The disappearance of neurons from the recording volume is a common phenomenon in our recordings. However, the opposite, i.e., the appearance of new neurons during recordings, is rarely observed. This might be explained by the fact, that at the beginning of the experiments, the tetrodes are explicitly placed at a position where a lot of neuronal activity is measured. Therefore, it is more probable that during the tissue drifts the high activity population of neurons disappears than that new, highly active neurons appear. We discuss this problem also in Section 4.4.

In Section 4.2.1 we have already demonstrated on simulated data the ability of our method to resolve overlapping spikes instantaneously. This is also the case for real data, see Fig. 14. The same figure also shows, that it would be very difficult to classify correctly these overlapping spikes with a purely clustering based algorithm.

---

[1] The value of $T$ was set to 5 s just for convenience of implementation, since the first 5 s of each trial were processed.
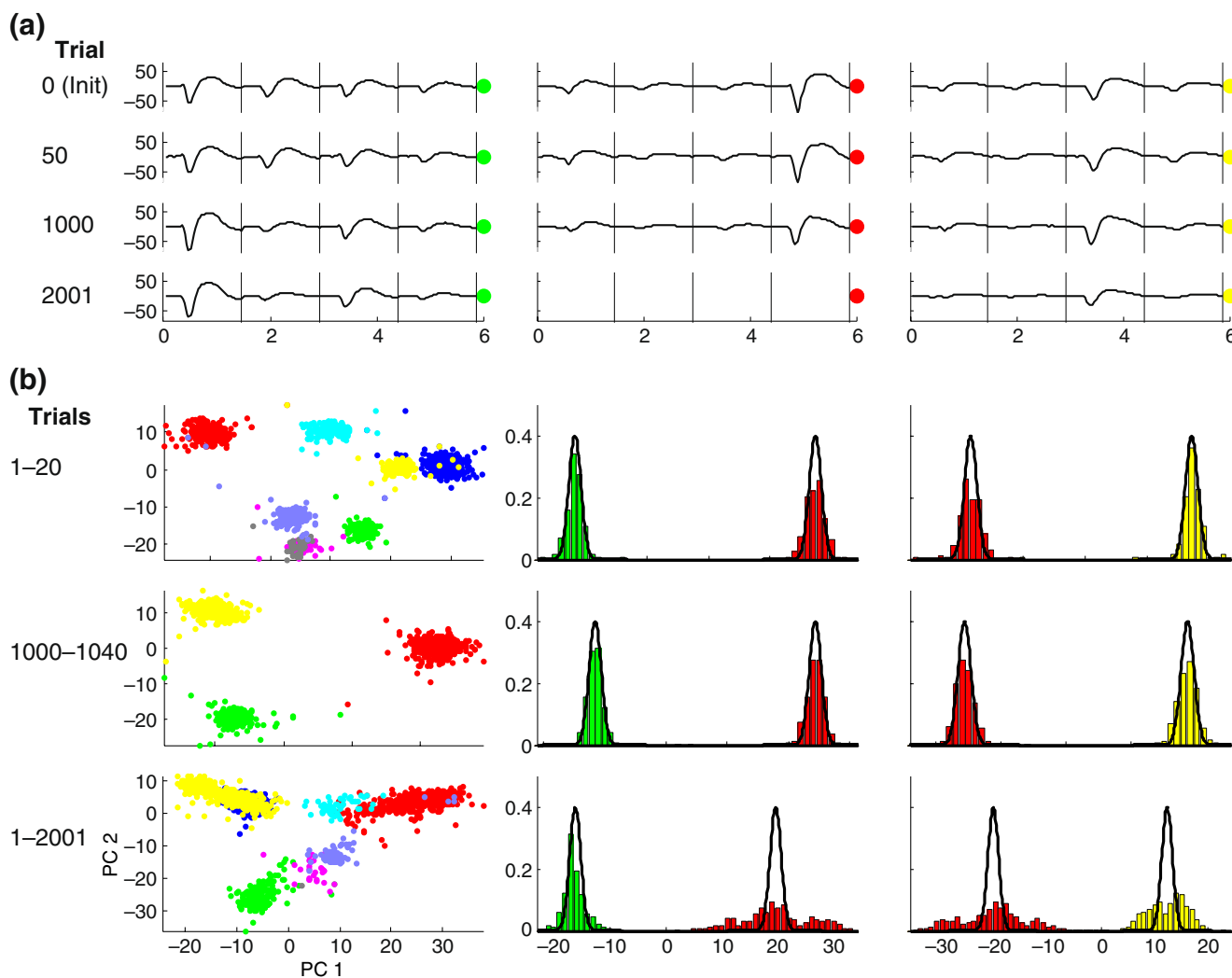
**Fig. 13** Effects of tissue drift on templates and cluster distributions in the PCA space. (**a**) For three filters which detected spikes nearly throughout the whole experiment, the corresponding templates of the initialization and at trials 50, 1000 and 2001 are shown. The color code is the same as in Fig. 11. Note that the middle template was deleted by the algorithm during trials 1000 and 2001 (compare Fig. 12). (**b**) The projections of the whitened spikes on the first two principle components (*left column*) and the projection test for three selected templates (*middle* and *right column*) are shown during three different periods. Note that during the two short periods (*upper two rows*) the whitened spikes of every neuron are nearly standard normal distributed. However, if the spikes are collected over a longer period (bottom row), the clusters are elongated and overlapping, making a clustering difficult. For sake of clarity only every 100th trial was plotted for the 1–2001 period

The evaluation in Fig. 11 and Fig. 13 shows that the clustered spikes, although whitened, are not perfectly Gaussian distributed. This deviation is caused by overlapping spikes, but it is also due to an intrinsic waveform variability, as it is observed for example during bursts (Fee et al. 1996b). In this sense, the generative model assumed in Section 2.2 is not strictly valid anymore. Nevertheless, our method achieves a good performance, even for datasets containing bursting neurons identified by visual inspection. This can be explained by the fact that the scaling of the waveform during burst is close to linear (Rutishauser et al. 2006).

Because of the linear character of our method (e.g. see Section 4.2.1), the response to a linearly scaled waveform will also only be scaled by the same factor. Hence, the algorithm classifies spikes from bursting neurons correctly as long as the amplitude degradation of the spikes is not too strong.

### 4.3 Limitations of our method

We have shown that our method is of great potential for spike detection and classification applications. However, there is a principle limitation: Since the fil-
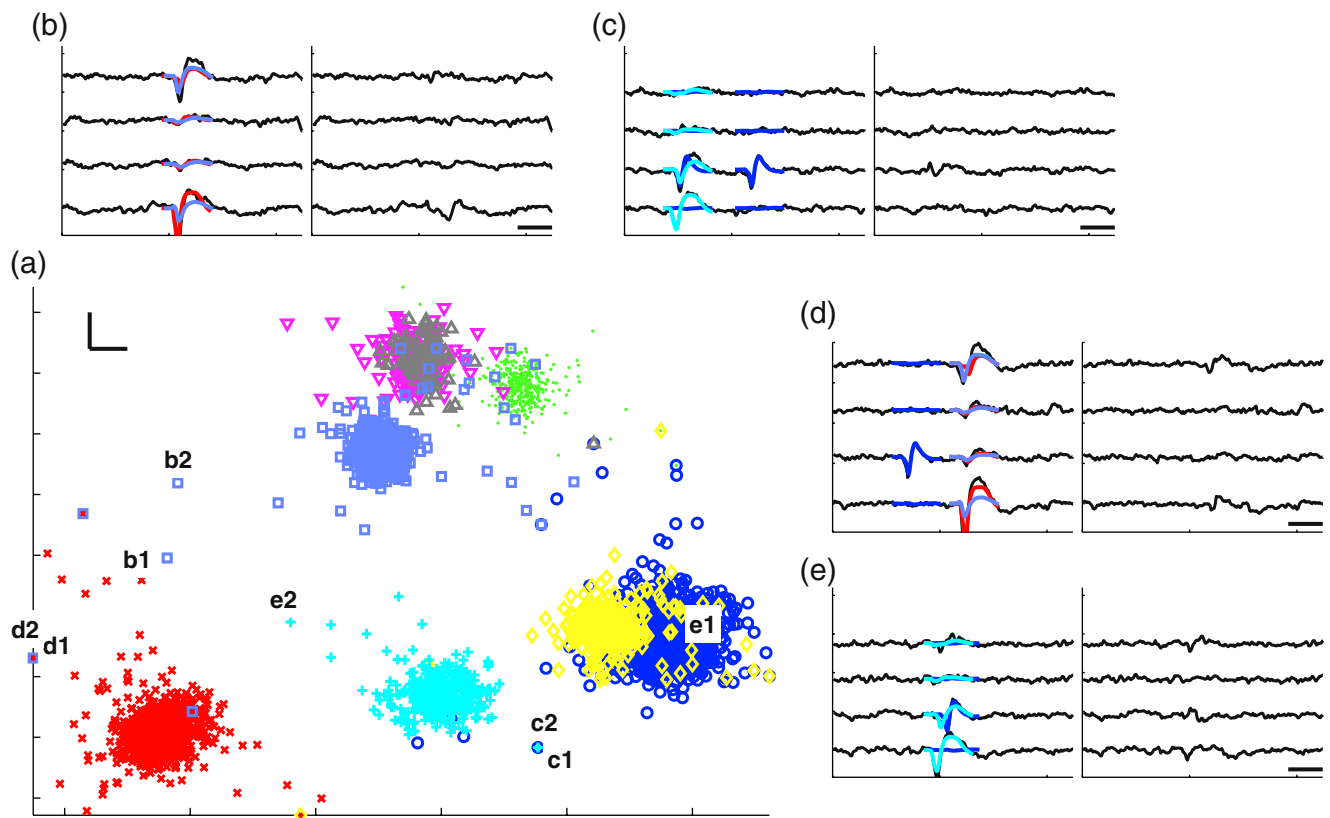
**Fig. 14** Ability of our algorithm to resolve overlapping spikes in real data. (**a**): Projection of all detected and whitened spikes from the trials 50–90 into the space of the first two principle components (the solid bars correspond to 3 standard deviations each). Spikes were detected and classified with the proposed method (color scheme is the same as in Fig. 11). For the additionally letter labeled spikes, the corresponding sections in the original recorded data are shown in (**b**)–(**e**), indicating that these spikes are overlapping spikes (the *solid bar* corresponds to 1 ms). In (**b**)–(**e**) all detected spikes are shown but only those participating in an overlap are labeled in (**a**). The plots right next to the original recordings show the same data after subtracting the templates of the neurons to which the spikes were assigned. The residual is similar to the noise signature, suggesting that our sorting was indeed correct. Note that the templates were *not* scaled to account for the amplitude variability of the spikes. This would reduce the residual. An example for a putative overlapping spike is labeled "d1" and would probably be detected as a single spike event by a standard spike detection. Furthermore its misclassification by a purely clustering based algorithm is likely, because its distance to the corresponding cluster means is large. Also spikes like "b1" or "b2" would have been probably classified as outliers by standard sorting methods

tering and the Deconfusion are linear operations, it is impossible to discriminate waveforms which are strictly linear dependent, i.e., when the spike waveform of one neuron is a multiple of the waveform of another neuron. A possible way to solve this problem is to sort the templates according to their SNR. Spikes with the highest SNR are detected first. Whenever a spike is found, the corresponding template is subtracted from the data and all other filter outputs are re-calculated for the affected period. This procedure is repeated for templates with a lower SNR. Further, if the sum of the waveforms of two different neurons with a certain shift is nearly identical to another neurons spike waveform, it is impossible to judge whether a spike is an overlap or not. Only probabilistic methods or soft clustering could give a hint at where the waveform came from.

### 4.4 Newly appearing neurons

We have not addressed the problem of neurons which are not detected during the initialization phase. As we observe spikes from neurons whose SNR decreases due to tissue drifts, and finally disappear completely from the recorded data, the opposite might also happen; i.e., neurons, previously undetected, slowly appear in the recording volume. A possible solution would be to run a conventional spike detection method in parallel to our method. All spikes detected by the

conventional spike detection technique, but not by our method, could be collected, aligned and clustered. Respecting the newly found clusters, corresponding filters could be initialized and the Deconfusion procedure adapted accordingly.

## 4.5 Implementation and computational complexity

Especially for a real-time implementation the runtime of an algorithm is crucial. After the initialization phase, the proposed method consists mainly of linear operations. The adaptation of the covariance matrix, of the templates and of the Deconfusion parameters need only to be computed every few seconds. Therefore, the computational burden lies in the application of the linear filters and the Deconfusion to a new sample of recorded (multichannel) data. The current implementation was done in Matlab, however the source code is not ready for publication yet. We will make the method available e.g. on ModelDB as soon as the implementation is finished.

### 4.5.1 Runtime analysis

If a new multichannel sample of data is recorded, first the cross-correlation between the filters and the data has to be calculated and afterwards Deconfusion is applied. The number of operations needed for the cross-correlation of a filter (the number of filters equals the number of neurons $M$) and the data is directly proportional to the product of the length of the filter $T_f$ and the number of recording channels $N$. The Deconfusion procedure consists of a half-wave rectification, which is just a sample wise trivial non-linearity, and a matrix-vector multiplication between the square matrix $\mathbf{W}$ of dimension $M \times M$ and the shifted and half-wave rectified filter outputs. To sum up, the computational complexity for a newly arriving data sample is $\mathcal{O}(MNT_f) + \mathcal{O}(M^2)$. Since we can assume the number of filters to be higher than the number of recording channels, the resulting complexity is $\mathcal{O}(M^2 T_f)$. This means the runtime complexity mainly depends on the number of filters and the filter length.[2]

---

[2]In principle, the cross-correlation can be calculated with the help of the fast Fourier transform more efficiently. However, this pays off for long data pieces only, and thus would require to buffer the data first, spoiling the real-time idea.

### 4.5.2 Parallel computing

It is important to note that the cross-correlation for every filter—even for every channel of every filter—are independent of each other and can, thus, be computed in parallel as simple vector-matrix multiplications. For a so called vector processor such a multiplication would be one single operation only. E.g this could be implemented on a modern consumer computer-graphics hardware or on programmable digital signal processors.

## 5 Conclusion and outlook

An automatic method for simultaneous spike detection and spike classification was presented, having several advantages which were demonstrated on various datasets. Explicitly, the method makes use of the additional information provided by multi electrodes and has no constraints concerning the number of recording channels or the number of neurons present in the data. It resolves overlapping spikes instantaneously, performs well on datasets with a low SNR, and it adapts to non-stationarities present in the data. Moreover, the method operates online and is well suited for a realtime implementation.

In the first step of our algorithm, optimal linear filters were used to enhance the SNR. Linear filters, being an approximation to an exact deconvolution, account for the noise statistics as well as for the full, multi-channel template, and are, therefore, superior to other methods in detecting spikes of a specific neuron. An evaluation on simultaneous intra/extra-cellular recordings in slices of rat visual cortex and on realistic synthetic data shows that the difference in performance is considerable.

Further, we used the output of the linear filters as a new representation of the data. The advantage of the filter output space is that its dimension is equal to the number of neurons, whereas this was not the case in the original data space. This allowed us to treat the spike sorting problem as a well defined source separation problem and solve it by Deconfusion.

In the final step, a channel specific threshold was applied providing simultaneous spike detection and classification. Unlike in many other methods, the thresholds need not to be set manually by a human supervisor but are determined automatically in an optimal way. The advantage of a combined spike detection and classification, in contrast to existing spike sorting methods, was demonstrated on simulated datasets. Especially in the presence of overlapping spike and low SNR, our method achieved better performances. We showed

that, in the case of linear filters, a proper definition of the signal-to-noise ratio is based on the Mahalanobis distance, whereas other commonly used definitions do not reflect the difficulty in detecting the signal.

By iteratively updating all quantities, namely the linear filters, the Deconfusion parameters, and the thresholds, the algorithm adopts to non-stationarities present in the data. As such, the method is also suitable for recordings made in acute experiments in which the multi electrodes are inserted each time anew. The number of spikes detected by a filter which were used for the calculation of the template, was set manually to a fixed value, equal for all filters. Instead, one could develop a model for the tissue drift and derive an optimal value which depends on the estimated drifting velocity, the firing rate of the neurons, on the SNR, and on the error tolerance. This is the aim of a future study.

Two drawbacks of the proposed method were discussed, namely the incapability to detect newly appearing neurons and the problem of strictly linear dependent templates. However, for both problems a possible solution was sketched. The detailed study and realization of these solutions will be the scope a future study.

By qualitative arguments, systematic runs on realistically simulated data and on real data from awake behaving macaques, we have shown that the algorithm is capable of resolving overlapping spikes; without additional computing time. However, for the acute recordings in awake behaving monkeys we cannot proof that the found solution is correct, since the ground truth is unknown. Only massive simultaneous intra- and extracellular recordings *in vivo* could be used to asses the quality of the sorting in real experiments. Due to technical limitations, such a dataset is currently not available.

The algorithm mainly consist of linear, independent operations, which can be executed in parallel and implemented in hardware. Therefore, the algorithm can be used for realtime implementations, making it an potential spike sorting method for brain-machine interfaces and for the execution of closed-loop experiments.

## Appendix A: Derivation of optimal linear filters

Filter $f^i$ should respond with a peak to its matching template $\xi^i$, but should have minimal response to the rest of the data. In particular, one demands that the response to the matching template is 1, i.e. $\xi^{i\top} \cdot f^i = 1$. The response of the filter to the data is $\bar{X} \star f^i$, where $\left(\bar{X} \star f^i\right)_t = \sum_{k,\tau} x_{k,\tau+t} \cdot f^i_{k,\tau}$. Using the third assumption of Section 2.2 the response of a filter to $\bar{X}$ will be small (and therefore well distinguishable from the peak response of 1 to the matching template) if the variance of the filter output is small, i.e., one has to minimize $Var\left(\bar{X} \star f^i\right)$. In summary, the constrained minimization problem is stated as

$$f^i = \underset{f^i}{\text{argmin}} \left\{ Var\left(\bar{X} \star f^i\right) \right\} \quad \text{subject to } \xi^{i\top} \cdot f^i = 1 \tag{11}$$

A short calculation shows that

$$Var\left(\bar{X} \star f^i\right) = \cdot f^{i\top} \cdot R \cdot f^i. \tag{12}$$

Thus, the Lagrangian $L$ of this minimization problem is given by

$$L = f^{i\top} \cdot R \cdot f^i + \lambda \left( \xi^{i\top} \cdot f^i - 1 \right) \tag{13}$$

where $\lambda$ is the Lagrange multiplier. Since the objective function is convex in $f^i$, there exists a single minimum, which can be found by solving $\nabla_{f^i, \lambda} L = 0$. In fact, the minimum is attained at

$$f^i = \frac{R^{-1} \xi^i}{\xi^{i\top} R^{-1} \xi^i}. \tag{14}$$

Often, linear filters are derived in the frequency domain instead, but linear filter defined in the time domain have several advantages, see Vollgraf and Obermayer (2006).

## Appendix B: Derivation of Deconfusion

$I(y^i_t)$ can be expressed as a linear combination of the sources $\upsilon^j$ at shifts $\tau_{i,j}$:

$$I(y^i_t) = \sum_j (A)_{i,j} \upsilon^j_{t+\tau_{i,j}} \tag{15}$$

We show that

$$z^i_t := \sum_j (W)_{i,j} I(y^j_{t-\tau_{j,i}}) \tag{16}$$

with $\boldsymbol{W} = \boldsymbol{A}^{-1}$ is the corresponding inverse problem. By inserting the expression in Eq. (15) into Eq. (16) one obtains

$$
\begin{aligned}
z_t^i &= \sum_j (\boldsymbol{W})_{i,j} \sum_k (\boldsymbol{A})_{j,k} \, \upsilon_{t+\tau_{j,k}-\tau_{j,i}}^k \\
&= \sum_{j,k} (\boldsymbol{W})_{i,j} (\boldsymbol{A})_{j,k} \, \upsilon_{t+\tau_{j,k}-\tau_{j,i}}^k \\
&= \sum_j (\boldsymbol{W})_{i,j} (\boldsymbol{A})_{j,i} \, \upsilon_{t+\tau_{j,i}-\tau_{j,i}}^i + \sum_{j,k\neq i} (\boldsymbol{W})_{i,j} (\boldsymbol{A})_{j,k} \, \upsilon_{t+\tau_{j,k}-\tau_{j,i}}^k \\
&= \upsilon_t^i + \sum_{j,k\neq i} (\boldsymbol{W})_{i,j} (\boldsymbol{A})_{j,k} \, \upsilon_{t+\tau_{j,k}-\tau_{j,i}}^k \qquad (17)
\end{aligned}
$$

Hence,

$$
z_t^i = \upsilon_t^i \iff \sum_{j,k\neq i} (\boldsymbol{W})_{i,j} (\boldsymbol{A})_{j,k} \, \upsilon_{t+\tau_{j,k}-\tau_{j,i}}^k = 0 \quad \forall j, k, i \neq k \tag{18}
$$

This is true, if

$$
\tau_{j_1,k} - \tau_{j_1,i} = \tau_{j_2,k} - \tau_{j_2,i} \qquad \forall j_1, j_2, i, k. \tag{19}
$$

Note that this condition is always satisfied for $k = i$.

### Appendix C: Derivation of the optimal threshold

If we assume that the noise in the Deconfusion output is still a mixture of Gaussians (as an approximation for a mixture of truncated Gaussians), it follows for its variance

$$
\begin{aligned}
\sigma_k^2 &:= Var\left(\boldsymbol{z}^k\right) \\
&= Var\left(\sum_i w_{k,i} \cdot y_{\tau_{k,i}}^i\right) \\
&= \sum_{i=1}^M \sum_{j=1}^M Cov\left(w_{k,i} y_{\tau_{k,i}}^i, w_{k,j} y_{\tau_{k,j}}^j\right) \\
&= \sum_{i=1}^M (w_{k,i})^2 \, Var\left(y_{\tau_{k,i}}^i\right) + 2 \sum_{i=1}^M \sum_{j>i}^M w_{k,i} w_{k,j} \\
&\quad \times Cov\left(y_{\tau_{k,i}}^i, y_{\tau_{k,j}}^j\right) \\
&= \sum_{i=1}^M (w_{k,i})^2 \, \boldsymbol{f}^{i\top} \boldsymbol{C} \boldsymbol{f}^i + 2 \sum_{i=1}^M \sum_{j>i}^M w_{k,i} w_{k,j} \boldsymbol{f}^{i\top} \\
&\quad \times \boldsymbol{C}_{|\tau_{k,j}-\tau_{k,i}|} \boldsymbol{f}^j \tag{20}
\end{aligned}
$$

where $\boldsymbol{C}_{|\tau_{k,j}-\tau_{k,i}|}$ are shifted covariance matrices, i.e. taking temporal correlations into account of order $T_f + |\tau_{k,j} - \tau_{k,i}|$.

The optimal threshold for the detection and classification of spikes from neuron $k$ is chosen such that the overlap between the distribution of the spikes from neuron $k$ and the distribution of the other spikes (from neurons $j$, $j = 1, ..., M$, $j \neq k$) is minimal. We assume the distributions to be Gaussian, with means $\mu_{k,j}$ and variance $\sigma_j^2$. The $\mu_{k,j}$ are given by the maximal response values of filter $j$ to template $k$ after Deconfusion, i.e.

$$
\mu_{k,j} = \max_\tau \left\{ \left(\boldsymbol{W} \cdot I\left(\left(\boldsymbol{\xi}^1 \star \boldsymbol{f}^j \ldots \boldsymbol{\xi}^M \star \boldsymbol{f}^j\right)^\top\right)\right)_{k,\tau} \right\} \tag{21}
$$

whereas the variance is given by Eq. (20). One has only to consider the maximal false response and not the whole response, because the refractory period is in general longer than the length of the template. Thus the optimal threshold $\epsilon_k$ is given by

$$
\begin{aligned}
\epsilon_k = \underset{\epsilon_k}{\arg\min} &\left\{ 1 - \frac{\beta_k}{2} erfc\left(\frac{\epsilon_k - 1}{\sqrt{2}\sigma_k}\right) \right. \\
&\left. + \sum_{j\neq k} \frac{\beta_j}{2} erfc\left(\frac{\epsilon_k - \mu_{k,j}}{\sqrt{2}\sigma_j}\right) \right\}
\end{aligned}
$$

where $erfc$ denotes the complementary error function, and $\beta_j$ is a normalized weight proportional to the firing frequency of neuron $j$ in order to minimize the total error. Note that the threshold must lie in the interval $[0, 1]$, hence this minimization problem can be solved numerically with a line search algorithm, for example using the "fminbnd" command of MATLAB.

### References

Aksenova, T. I., Chibirova, O. K., Dryga, O. A., Tetko, I. V., Benabid, A. L., & Villa, A. E. P. (2003). An unsupervised automatic method for sorting neuronal spike waveforms in awake and freely moving animals. *Methods, 30*(2), 178–187.

Branchaud, E., Burdick, J., & Andersen, R. (2006). An algorithm for autonomous isolation of neurons in extracellular recordings. In *Proc. first IEEE/RAS-EMBS international conference on biomedical robotics and biomechatronics BioRob 2006* (pp. 939–945). doi:10.1109/BIOROB.2006.1639212.

Cham, J. G., Branchaud, E. A., Nenadic, Z., Greger, B., Andersen, R. A., & Burdick, J. W. (2005). Semi-chronic motorized microdrive and control algorithm for autonomously isolating and maintaining optimal extracellular action potentials. *Journal of Neurophysiology, 93*(1), 570–579. doi:10.1152/jn.00369.2004.

Chandra, R., & Optican, L. M. (1997). Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural net-

work. *IEEE Transactions on Biomedical Engineering, 44*(5), 403–412. doi:10.1109/10.568916.

Choi, J. H., Jung, H. K., & Kim, T. (2006). A new action potential detector using the mteo and its effects on spike sorting systems at low signal-to-noise ratios. *IEEE Transactions on Biomedical Engineering, 53*(4), 738–746. doi:10.1109/TBME.2006.870239.

Delescluse, M., & Pouzat, C. (2006). Efficient spike-sorting of multi-state neurons using inter-spike intervals information. *Journal of Neuroscience Methods, 150*(1), 16–29. doi:10.1016/j.jneumeth.2005.05.023.

Ding, W., & Yuan, J. (2008). Spike sorting based on multi-class support vector machine with superposition resolution. *Medical and Biological Engineering and Computing, 46*(2), 139–145. doi:10.1007/s11517-007-0248-0.

Eckhorn, R., & Thomas, U. (1993). A new method for the insertion of multiple microprobes into neural and muscular tissue, including fiber electrodes, fine wires, needles and microsensors. *Journal of Neuroscience Methods, 49*(3), 175–179.

Fee, M. S., Mitra. P. P., & Kleinfeld, D. (1996a). Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-gaussian variability. *Journal of Neuroscience Methods, 69*(2), 175–188. doi:10.1016/S0165-0270(96)00050-7.

Fee, M. S., Mitra, P. P., & Kleinfeld, D. (1996b). Variability of extracellular spike waveforms of cortical neurons. *Journal of Neurophysiology, 76*(6), 3823–3833.

Gray, C. M., Maldonado, P. E., Wilson, M., & McNaughton, B. (1995). Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *Journal of Neuroscience Methods, 63*(1–2), 43–54.

Harris, K. D., Henze, D. A., Csicsvari, J., Hirase, H., & Buzsáki, G. (2000). Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *Journal of Neurophysiology, 84*(1), 401–414.

Hayes, M. H. (1996). *Statistical digital signal processing and modeling*. New York: Wiley.

Hulata, E., Segev, R., Ben-Jacob, E. (2002). A method for spike sorting and detection based on wavelet packets and shannon's mutual information. *Journal of Neuroscience Methods, 117*(1), 1–12.

Kim, K. H., & Kim, S. J. (2003). Method for unsupervised classification of multiunit neural signal recording under low signal-to-noise ratio. *IEEE Transactions on Biomedical Engineering, 50*(4), 421–431. doi:10.1109/TBME.2003.809503.

Lewicki, M. (1998). A review of methods for spike sorting: The detection and classification of neural action potentials. *Network: Computation in Neural Systems, 9*(4), 53–78.

McGill, K. C. (2002). Optimal resolution of superimposed action potentials. *IEEE Transactions on Biomedical Engineering, 49*(7), 640–650. doi:10.1109/TBME.2002.1010847.

Melvin, W. (2004). A stap overview. *IEEE Aerospace and Electronic Systems Magazine, 19*(1), 19–35. doi:10.1109/MAES.2004.1263229.

Mtetwa, N., & Smith, L. (2006). Smoothing and thresholding in neuronal spike detection. *Neurocomputing, 69*(10–12), 1366–1370.

Obeid, I., & Wolf, P. D. (2004). Evaluation of spike-detection algorithms for a brain-machine interface application. *IEEE Transactions on Biomedical Engineering, 51*(6), 905–911. doi:10.1109/TBME.2004.826683.

O'Keefe, J., & Recce, M. L. (1993). Phase relationship between hippocampal place units and the eeg theta

rhythm. *Hippocampus, 3*(3), 317–330. doi:10.1002/hipo.450030307.

Pouzat, C., Mazor, O., & Laurent, G. (2002). Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *Journal of Neuroscience, 122*(1), 43–57.

Pouzat, C., Delescluse, M., Viot, P., & Diebolt, J. (2004). Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: A markov chain monte carlo approach. *Journal of Neurophysiology, 91*(6), 2910–2928. doi:10.1152/jn.00227.2003.

Rebrik, S., Wright, B., Emondi, A., & Miller, K. D. (1999) Cross-channel correlations in tetrode recordings: Implications for spike-sorting. *Neurocomputing, 2627*, 1033–1038.

Robinson, E. A., & Treitel, S. (1980). *Geophysical signal analysis*. Englewood Cliffs: Prentice Hall

Rutishauser, U., Schuman, E. M., & Mamelak, A. N. (2006). Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, *in vivo*. *Journal of Neuroscience Methods, 154*(1–2), 204–224. doi:10.1016/j.jneumeth.2005.12.033.

Sakurai, Y., & Takahashi, S. (2006). Dynamic synchrony of firing in the monkey prefrontal cortex during working-memory tasks. *Journal of Neuroscience, 26*(40), 10141–10153. doi:10.1523/JNEUROSCI.2423-06.2006.

Shoham, S., Fellows, M. R., & Normann, R. A. (2003). Robust, automatic spike sorting using mixtures of multivariate t-distributions. *Journal of Neuroscience Methods, 127*(2), 111–122.

Takahashi, S., Anzai, Y., & Sakurai, Y. (2003). Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes. *Journal of Neurophysiology, 89*(4), 2245–2258. doi:10.1152/jn.00827.2002.

Takahashi, S., & Sakurai, Y. (2005). Real-time and automatic sorting of multi-neuronal activity for sub-millisecond interactions *in vivo*. *Neuroscience, 134*(1), 301–315. doi:10.1016/j.neuroscience.2005.03.031.

Takahashi, S., Sakurai, Y., Tsukuda, M., & Anzai, Y. (2002). Classification of neural activities using independent component analysis. *Neurocomputing, 49*, 289–298.

Thakur, P. H., Lu, H., Hsiao, S. S., & Johnson, K. O. (2007). Automated optimal detection and classification of neural action potentials in extra-cellular recordings. *Journal of Neuroscience Methods, 162*(1–2), 364–376. doi:10.1016/j.jneumeth.2007.01.023.

Turin, G. (1960). An introduction to matched filters. *IRE Transactions on Information Theory, 6*(3), 311–329. doi:10.1109/TIT.1960.1057571.

Vargas-Irwin, C., & Donoghue, J. P. (2007). Automated spike sorting using density grid contour clustering and subtractive waveform decomposition. *Journal of Neuroscience Methods, 164*(1), 1–18. doi:10.1016/j.jneumeth.2007.03.025.

Vollgraf, R., Munk, M., Obermayer, K. (2005). Optimal filtering for spike sorting of multi-site electrode recordings. *Network, 16*(1), 85–113.

Vollgraf, R., & Obermayer, K. (2006). Improved optimal linear filters for the discrimination of multichannel waveform templates for spike-sorting applications. *IEEE Signal Processing Letters, 13*(3), 121–124. doi:10.1109/LSP.2005.862621.

Wang, G. L., Zhou, Y., Chen, A. H., Zhang, P. M., & Liang, P. J. (2006). A robust method for spike sorting with automatic overlap decomposition. *IEEE Transactions on Biomedical Engineering, 53*(6), 1195–1198. doi:10.1109/TBME.2006.873397.

Wu, W., Wheeler, D. W., Staedtler, E. S., Munk, M. H. J., & Pipa, G. (2008). Behavioral performance modulates spike

field coherence in monkey prefrontal cortex. *Neuroreport, 19*(2), 235–238. doi:10.1097/WNR.0b013e3282f49b29.

Xu, R., Wunsch, I. D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*(3), 645–678. doi:10.1109/TNN.2005.845141.

Zhang, P. M., Wu, J. Y., Zhou, Y., Liang, P. J., Yuan, J. Q. (2004). Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. *Journal of Neuroscience Methods, 135*(1–2), 55–65. doi:10.1016/j.jneumeth.2003.12.001.