Towards online spike sorting for high-density neural probes using discriminative template matching with suppression of interfering spikes

Jasper Wouters[1], Fabian Kloosterman[2,3,4], and Alexander Bertrand[1]

[1]KU Leuven, Electrical Engineering Dept. (ESAT), Stadius Center for Dynamical Systems, Signal Processing, and Data Analytics, Belgium

[2]Neuro-Electronics Research Flanders (NERF), Leuven, Belgium

[3]KU Leuven, Brain & Cognition Research Unit, Belgium

[4]VIB, Leuven, Belgium

***Abstract*—*Objective*: The process of grouping neuronal spikes in an extracellular recording according to their neuronal sources, is generally referred to as spike sorting. Currently, the use of spike sorting is mainly limited to an offline usage, where spikes are sorted after the data acquisition has been completed. In this paper, we propose a threshold-based spike sorting algorithm with a low and deterministic algorithmic delay, allowing for fast online spike sorting.**

***Approach*: At its core, threshold-based spike sorting is driven by linear filters. We propose a discriminative template matching filter design algorithm, which optimizes the output signal-to-peak-interference ratio in a data-driven fashion. The latter allows the filter to suppress the spikes of interfering neurons and to resolve spike overlap.**

***Main results*: The proposed discriminative template matching filters are validated on in-vivo ground truth data and are shown to provide a similar spike sorting performance as a state-of-the-art algorithm for offline spike sorting.**

***Significance*: The low algorithmic complexity allows for computationally cheap and fast spike sorting. Also the proposed filters are guaranteed to be stable and have a deterministic delay. These characteristics make the proposed spike sorting algorithm a valuable tool for online spike sorting, thereby enabling unit activity-based real-time and closed-loop experiments.**

This work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven Special Research Fund projects C14/16/057 and CoE PFV/10/002 (OPTEC), and the Research Foundation Flanders (FWO) project FWO G0D7516N (Distributed signal processing algorithms for spike sorting in next-generation high-density neuroprobes). The scientific responsibility is assumed by its authors. A conference precursor of this research has been published in [Wouters et al., 2018].

I. Introduction

Neurons communicate with each other through all-or-nothing action potentials, also known as spikes, which can be recorded with an extracellular electrode placed in the vicinity of the neurons. An electrode will generally pick up spikes from several neurons near the electrode. Therefore, to decode brain processes, neuroscientists have to identify and sort these spikes according to their underlying neuronal origins, a source separation process which is often referred to as spike sorting [Lewicki, 1998].

Currently, spike sorting is performed as a post-processing step once a complete extracellular recording is available. This allows only correlative conclusions to be drawn from the spike sorting results and behavioural data. When performing online spike sorting as part of a closed-loop experiment in which neural processes can be altered through stimulation, causal relationships could be discovered between spiking patterns and

behavioural outcomes. The availability of such an algorithm would be a game changer for experimental neuroscientists.

Recently, on the hardware side, there has been a shift from conventional extracellular recording devices with a few electrodes to high-density (HD) neuroprobes with more than thousand electrodes and electrode pitches of around $20\mu m$ [Lopez et al., 2016] [Raducanu et al., 2016] [Jun et al., 2017b]. Such HD probes enable recording from the entire depth of a rodent brain. For these new probes, conventional spike sorting algorithms [Shoham et al., 2003] [Quiroga et al., 2004] [Kadir et al., 2014] [Rossant et al., 2015] fail, because the computational complexity scales supralinear with the number of electrodes. On top of that, the cluster analysis will suffer from the "*curse of dimensionality*" [Steinbach et al., 2004], a general phenomenon that occurs when clustering in high dimensional spaces.

To overcome these computational problems, a number of new spike sorting algorithms have been developed [Pachitariu et al., 2016] [Yger et al., 2016] [Jun et al., 2017a] [Chung et al., 2017] for sorting HD probe data. All of these algorithms are developed to perform real-time spike sorting on HD extracellular data. Important to note however, is that they are real-time with respect to the offline processing time, meaning that they can process a finished recording in approximately the same time as the duration of the recording. For closed-loop stimulation experiments in which the stimulation depends on the outcome of the spike sorting process, an online real-time deadline has to be met which is dependent on the specific process under investigation [Ciliberti and Kloosterman, 2017]. For instance an experimental study of synaptic plasticity requires stimulation to occur within $2ms$, the synaptic integration time scale, after a target neuron generated an action potential [Jun et al., 2017a].

Many conventional spike sorting algorithms also struggle with handling overlapping spikes. The reason for their failure is that spikes are classified based on features used during the clustering analysis [Gibson et al., 2012]. When features are extracted from a segment containing, e.g., two overlapping spikes from different neurons, the features are unlikely to be related to the features extracted from single spike segments from either neuron, leading to spurious clusters. To solve this problem, recent algorithm pipelines have been augmented with a final template matching [Marre et al., 2012] [Ekanadham et al., 2014] [Franke et al., 2015] phase. Prototypical spike templates are in practice extracted from an initial spike sorting, where only non-overlapping spikes are taken as the input. Each extracted spike template is then applied to the data as a linear matched filter. The outputs of these matched filters are then used to decide on the final cluster assignment for spikes in the recording.

Such linear filtering-based template matching is a particularly interesting approach for online spike sorting. For each sortable unit in the extracellular recording, a linear filter is trained which responds to the spatio-temporal spike footprint of that unit. Such filters can be applied to the data in a streaming fashion. This implies that a spike can be sorted before its contribution in the extracellular recording has faded away. Classification is then done by thresholding the different filter outputs. Another advantage of using linear filters is their low computational complexity. The only computations involved in applying such filters are scalar multiplications of recording samples in a finite shift register and a summation over the different products. This makes linear filters extremely fit for fast (hardware) implementation and online real-time application.

Recent work on template matching-based spike sorting [Franke et al., 2015] [Yger et al., 2016] uses only information from target spikes in training the template matching filters. This leads to the problem of multiple filters responding to a single spike, when their target spike pattern shows some similarity with the spike pattern of the active neuron. This makes it difficult to determine whether these simultaneous responses are caused by a single spike triggering multiple filter responses, or by multiple overlapping spikes each triggering a single filter response. To solve this problem, both methods apply an iterative scheme where they assign the spike to the neuron of which the corresponding filter response is maximal and then subtract the spatio-temporal template of this neuron

from the data. The data from which the template is subtracted is iteratively filtered again (each time followed by another spike template subtraction) until all filter responses on this part of the data are below the detection threshold. A consequence of such an iterative scheme is that it takes multiple iterations over the same data. This causes an unpredictable delay in the processing, which might lead to missing real-time deadlines. Even if there is just one spike active, a second pass over the data is necessary to confirm that the exceeded thresholds in other filter outputs are not caused by overlapping spikes.

Another potential issue arises directly from relying on the identification of the maximal filter response for assigning a spike to its putative neuron. Since the filters are subjected to noise, the filter outputs might be distorted and as such the filter with the maximal response is not always the one that corresponds to the firing neuron, leading to a misclassification of that spike.

To overcome the aforementioned issues, the method proposed in this paper designs linear filters which simultaneously maximize the response to the target neuron, while suppressing all other sources causing spurious peaks in the filter output, including interfering spikes from neighboring neuron cells, as well as general background noise. From an optimization point of view, our filter design method is based on maximizing the *signal-to-peak-interference ratio* (SPIR) objective function. So we propose to use template matching filters for HD extracellular recordings, which are more discriminative and stronger tuned towards a single unit, such that a significant filter response is generated only when the corresponding neuron is active. Such an approach removes the necessity of iterative filtering schemes, it improves the threshold-based spike sorting accuracy, and it allows for a more controllable processing delay.

The optimal filter design will be based on the Generalized Eigenvalue Decomposition [Golub and Van Loan, 1996] of pairs of estimated signal and interference covariance matrices. Furthermore a practical offline scheme for estimating these covariance matrices will also be proposed, such that the optimality of the filters is aligned with the ultimate goal: maximizing spike sorting accuracy. Basically this comes down to an implicit optimal weighting of the different interfering sources in the interference covariance matrix.

The remainder of this paper is built up as follows. In Section II a model for online spike sorting, namely *threshold-based spike sorting*, used throughout this paper is presented and matched filtering-based template matching is reviewed. In section III our *discriminative template matching* filter design method is derived. Next the proposed method is validated on HD in-vivo recorded ground truth data in Section IV. Finally conclusions are drawn in Section V.

II. Threshold-based spike sorting pipeline

This section will first conceptually introduce the threshold-based spike sorting pipeline, followed by the derivation of a matched filter that can be used as the pipeline's core template matching filter. **Mathematical notation:** Scalars are represented by non-bold letters a or A, vectors by bold lowercase letters $\mathbf{b}$ and matrices by bold uppercase letters $\mathbf{C}$. $\mathbf{C}^T$ denotes the matrix transpose of $\mathbf{C}$ and $\mathbf{C}^{-1}$ denotes the matrix inverse of $\mathbf{C}$.

A. Threshold-based spike sorting

Consider a HD extracellular neural probe with N electrodes, or channels, as shown in Figure 1. Such a probe is lowered into the brain so that it can record spikes from neurons in the vicinity of the electrodes. Spikes from a specific neuron will typically be picked up by multiple electrodes close to that neuron.

Consider $\mathbf{y}[t]$ to be an N-dimensional vector containing the observations from the N-channel neural probe at sample time t. To extract the activity of a neuron of interest n, usually only a subset of the channels in $\mathbf{y}[t]$ is useful. Therefore, we introduce for each neuron n the $K^{(n)}$-channel signal $\mathbf{y}^{(n)}[t]$ consisting of $K^{(n)}$ channels of $\mathbf{y}[t]$ that contain useful information for isolating the activity of neuron n. Selecting a subset of channels is represented in Figure 1 by the *channel selector* block. Some practical guidelines on how this subset of channels can be selected will be given in Section IV.

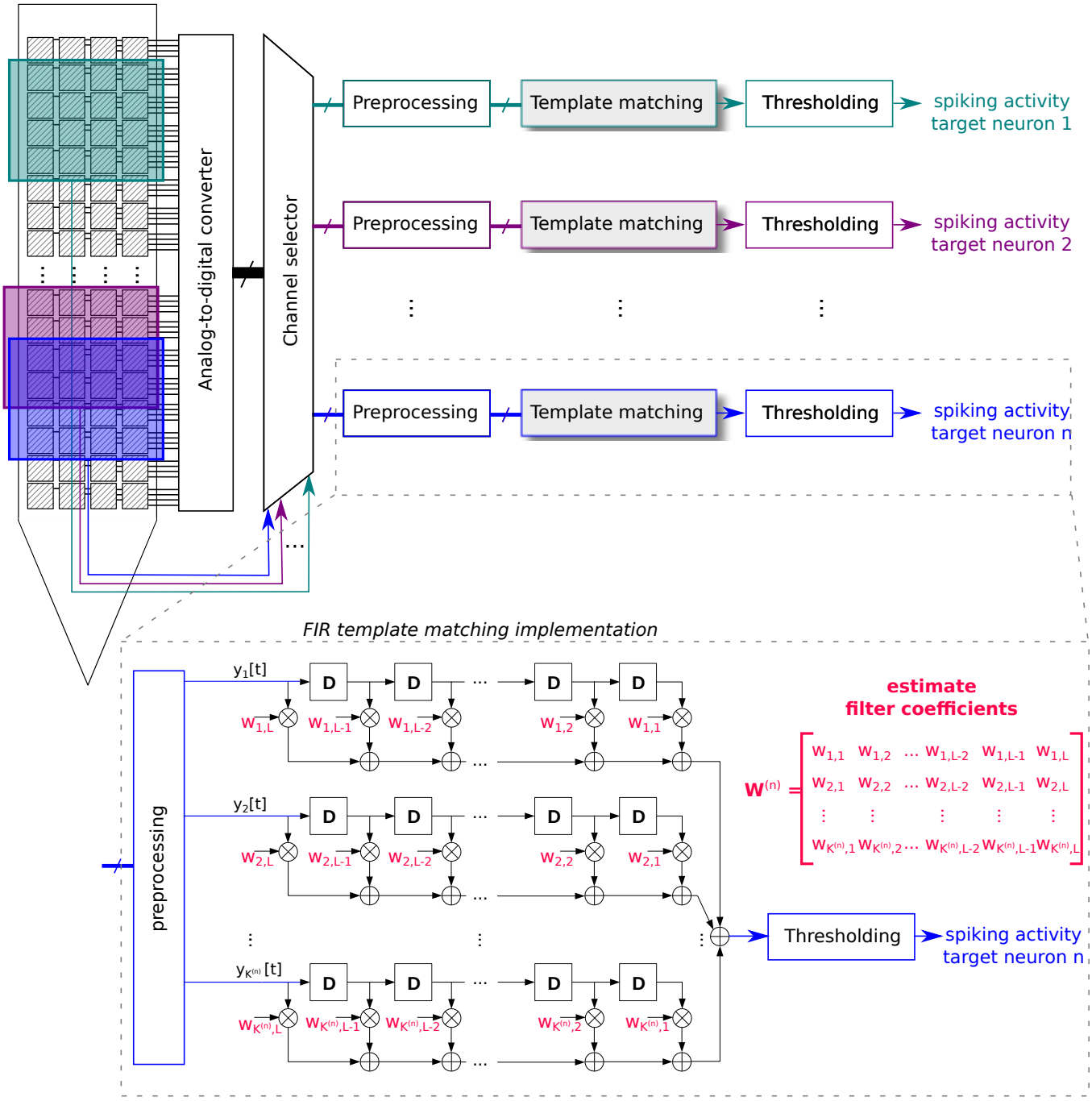

Figure 1: A schematic representation of the *threshold-based spike sorting* paradigm. For every sortable neuron, a separate processing pipeline needs to be implemented. Every pipeline includes preprocessing, template matching and thresholding, and generates single-unit activity at its output. A detailed view of such a pipeline (bottom) reveals the finite impulse response (FIR) template matching filters that are considered in this paper.

Consider $y_k^{(n)}[t]$ to be the observation of the kth channel of $\mathbf{y}^{(n)}[t]$. Without loss of generality, we assume $y_k^{(n)}[t]$ to be zero-mean for each k, which is generally achieved through a prior high-pass filtering on the data. This high-pass filtering is schematically represented by the *preprocessing* blocks in Figure 1. Define $\overline{\mathbf{y}}_k^{(n)}[t] = \left[y_k^{(n)}[t] \quad y_k^{(n)}[t-1] \quad \dots \quad y_k^{(n)}[t-L+1]\right]^T$ as the L-dimensional time-lagged vector describing a delay line containing the current and $L-1$ previous samples of $y_k^{(n)}[t]$ at time t. By stacking all these delay lines for every channel of $\mathbf{y}^{(n)}[t]$, we obtain the vector $\overline{\mathbf{y}}^{(n)}[t] = \left[\overline{\mathbf{y}}_1^{(n)}[t]^T \quad \dots \quad \overline{\mathbf{y}}_{K^{(n)}}^{(n)}[t]^T\right]^T \in \mathbb{R}^{LK^{(n)}}$.

The goal of this paper is to design a linear spatio-temporal filter $\mathbf{w}^{(n)} \in \mathbb{R}^{LK^{(n)}}$ for each detectable neuron n in the recording, such that spike sorting can be done via a simple thresholding operation on the individual filter outputs $z^{(n)}[t] = \mathbf{w}^{(n)^T}\overline{\mathbf{y}}^{(n)}[t]$, without using iterative schemes for handling overlapping spikes. Note that $\mathbf{w}^{(n)^T}\overline{\mathbf{y}}^{(n)}[t]$ corresponds to a filter-and-sum operation containing $K^{(n)}$ finite impulse response (FIR) filters, each with L taps. Spike sorting based on simple thresholding of $z^{(n)}[t]$ will be referred to as *threshold-based spike sorting*. Each filter will work independent of one another and all filters can be executed in parallel. The neuron corresponding to a filter is also referred to as the *target neuron* of that filter.

An FIR filter can be implemented with delay elements, adders and multipliers as shown in the schematically detailed bottom part of Figure 1. Such filters are guaranteed to be stable, they have a predictable processing delay, and have a low computational complexity allowing for fast online processing. The actual filter design boils down to choosing the involved channels (electrodes) and the number of delay elements, and estimating the filter coefficients such that the filter has the desired behaviour. Section II-B and III will focus on designing template matching filters for threshold-based spike sorting. The presented filter design algorithms are meant for offline use, meaning that these algorithms estimate the filter coefficients from initially available recording data, after which the online filter can be applied to streaming data. However, these algorithms are candidates for adaptive implementations, such that the filter coefficients can be corrected for varying signal statistics in an online fashion.

B. Matched filtering-based template matching

In the remainder of this paper it is assumed that the number of reliably detectable neurons in a recording are known and a representative set of example spikes is available for each neuron. This assumption is common to all template matching approaches. In practice this prior knowledge can be extracted directly from the recording using a clustering-based initial spike sorter to perform an initial coarse clustering, possibly on only a subset of data.

Because all filters operate in parallel and independently from one another, as can be seen from Figure 1, the mathematics will be derived for a single neuron of interest n.

As the signal-to-noise ratio (SNR) at the output of the filter has a direct impact on the classification accuracy in a threshold-based spike sorting paradigm, it is important to properly design the filter in order to maximize its overall output SNR. In standard matched filtering-based template matching [Kaneko et al., 1999], the filter coefficients $\mathbf{w}^{(n)}$ are chosen to be the spatio-temporal template of neuron n, yielding a so-called matched filter design. This template $\boldsymbol{\tau}^{(n)}$ is a $\left(K^{(n)}L\right)$-dimensional vector and can be estimated by averaging over multiple aligned example spikes generated by neuron n:

$$\mathbf{w}_{match}^{(n)} = \boldsymbol{\tau}^{(n)} = \frac{1}{S^{(n)}} \sum_{j=1}^{S^{(n)}} \overline{\mathbf{y}}^{(n)}[t_j^{(n)}], \tag{1}$$

where $S^{(n)}$ is the number of spike waveforms used for estimating the template of neuron n and all $t_j^{(n)}$ are chosen such that all example spikes of neuron n are mutually temporally aligned. Alternatively, a median operator can be used to estimate a template, which is less sensitive to outliers compared to averaging. The output $z_{match}^{(n)}[t]$ of the matched filter $\mathbf{w}_{match}^{(n)}$ will have a large response when a spike of neuron n is presented at its input at time t.

It is noted that a matched filter only optimizes the output SNR if the background noise is uncorrelated both in space and time. Therefore, a pre-whitening operation [Pouzat et al., 2002] [Franke et al., 2015] is performed on the signals in order to obtain a white background noise without spatio-temporal structure. The zero-phase component analysis (ZCA) whitening matrix [Kessy et al., 2017] is given by the square root of the inverse of the spatio-temporal covariance matrix of the background noise:

$$\mathbf{W}^{(n)} = \mathbf{R}_{\overline{\mathbf{b}}\overline{\mathbf{b}}}^{(n)^{-\frac{1}{2}}} = E\left\{\overline{\mathbf{b}}^{(n)}[t] \cdot \overline{\mathbf{b}}^{(n)}[t]^T\right\}^{-\frac{1}{2}}, \qquad (2)$$

with $\overline{\mathbf{b}}^{(n)}[t]$ the additive background noise. Since $\overline{\mathbf{b}}^{(n)}[t]$ is unknown, in practice $\mathbf{R}_{\overline{\mathbf{b}}\overline{\mathbf{b}}}^{(n)}$ is estimated from $\overline{\mathbf{y}}^{(n)}[t]$ at times t when no spikes are detected, i.e., where only baseline noise is observed. The square root can be computed using, e.g., a Cholesky factorization or an eigenvalue decomposition on $\mathbf{R}_{\overline{\mathbf{b}}\overline{\mathbf{b}}}^{(n)^{-1}}$.

Applying the whitening matrix to the data will reduce the spatio-temporal structure of the background noise, but it will also alter the spike waveform, and as such the template. Therefore matched filtering-based template matching using whitened data can be written as:

$$\begin{aligned} z^{(n)}[t] &= \left(\mathbf{W}^{(n)}\boldsymbol{\tau}^{(n)}\right)^T \left(\mathbf{W}^{(n)}\overline{\mathbf{y}}^{(n)}[t]\right) \\ &= \left(\mathbf{R}_{\overline{\mathbf{b}}\overline{\mathbf{b}}}^{(n)^{-1}}\boldsymbol{\tau}^{(n)}\right)^T \overline{\mathbf{y}}^{(n)}[t]. \end{aligned} \qquad (3)$$

From the above equation it can be seen that the whitening can be combined with template matching using a single FIR filter $\mathbf{w}_{match}^{(n)} = \mathbf{R}_{\overline{\mathbf{b}}\overline{\mathbf{b}}}^{(n)^{-1}}\boldsymbol{\tau}^{(n)}$.

The noise covariance matrix will likely be dominated by the baseline noise from far-away neurons in which case a pre-whitening in combination with a matched filter will obtain its optimal SNR by reducing the baseline noise. This improvement in SNR is unlikely to influence the threshold-based spike sorting accuracy, because false detections mainly arise from sparse transient noise events and spikes from close-by neurons. Therefore, the matched filter as described in this section, and as used in many spike sorting algorithms, is often not optimal for threshold-based spike sorting. The problem with the matched filter used for threshold-based spike sorting is that the spatio-temporal structure of a spike template might not be sufficiently discriminative to generate a high output variance *only* when neuron n is firing.

III. Discriminative template matching

A. Filter design for SPIR maximization

Consider again the goal of this paper: the design of a linear FIR template matching filter for each detectable neuron in a HD probe recording, such that single-unit activity can be obtained by a simple thresholding operation on the individual template matching filter outputs.

Conceptually the optimal filter for threshold-based spike sorting, would be the filter where spurious input peaks are suppressed towards the same level as the filter's response to background noise, while peaks resulting from activity of the target neuron are still significantly higher. Such a filter design should not focus on optimizing overall output SNR, but on maximizing *signal-to-peak-interference ratio* (SPIR) instead. A filter optimizing the SPIR can be thought of as a pre-whitening with a matched filter, which is designed using a weighted noise covariance estimate.

To increase the threshold-based spike sorting accuracy we have to extend the definition of noise: *noise should also include spikes from other neurons*. Furthermore, some noise sources are more severely affecting spike sorting accuracy than others, and as such should be treated differently. The optimal filter for threshold-based spike sorting should have a maximal response to the target neuron, while having a minimal response to the most important interfering sources (e.g. spikes from nearby neurons). Mathematically speaking, this means that the ratio between the output signal powers in both of these cases should be maximized, i.e.,

$$\mathbf{w}_{disc}^{(n)} = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\left(\mathbf{w}^T\boldsymbol{\tau}^{(n)}\right)^2}{\sum_k p_k E\left\{\left(\mathbf{w}^T\bar{\mathbf{i}}_k^{(n)}[t]\right)^2\right\}}, \qquad (4)$$

where $p_k \in \mathbb{R}$ are weights that denote the importance of each interfering source k, and where $\bar{\mathbf{i}}_k^{(n)}[t]$ is the signal generated

by the interfering source k (e.g., sharing significant spatio-temporal structure with the spikes generated by the target neuron), also referred to as peak interferer k. As such these peak interferers might influence the threshold-based spike sorting accuracy. The filter given by (4) thus maximizes the filter response to the spike template, while maintaining a minimal response to the interfering sources.

We define the weighted interference covariance matrix as $\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} = \sum_k p_k E\left\{\bar{\mathbf{i}}_k^{(n)}[t]\bar{\mathbf{i}}_k^{(n)}[t]^T\right\}$, describing the spatio-temporal structure of the interfering sources. (4) can then be rewritten as:

$$\mathbf{w}_{disc}^{(n)} = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^T \boldsymbol{\tau}^{(n)} \boldsymbol{\tau}^{(n)^T} \mathbf{w}}{\mathbf{w}^T \mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} \mathbf{w}}. \tag{5}$$

It is noted that the solution of (4) and (5) is only defined up to a scaling, which can be resolved by adding a unit norm constraint (see Appendix).

Assume for now that we have such an interference covariance estimate $\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)}$ available which leads to a filter output with optimal SPIR. A direct-form solution for the optimization problem in (5) is:

$$\mathbf{w}_{disc}^{(n)} = \mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)^{-1}} \boldsymbol{\tau}^{(n)}. \tag{6}$$

For a proof of (6) see Appendix A. A key ingredient in the above filter design is the interference covariance matrix. The estimation of the interference covariance matrix is the subject of Section III-C.

The type of filter discussed in this section will be referred to as a *discriminative template matching* filter. The filter design method allows for optimally discriminating between two different spatio-temporal structures as described by a pair of covariance matrices. In the context of this work the pair of covariance matrices will consist of the target spike covariance matrix (here replaced by a rank-1 matrix based on the template $\boldsymbol{\tau}^{(n)}$) and the estimated covariance matrix of signal segments who otherwise cause spurious peaks on the filter output.

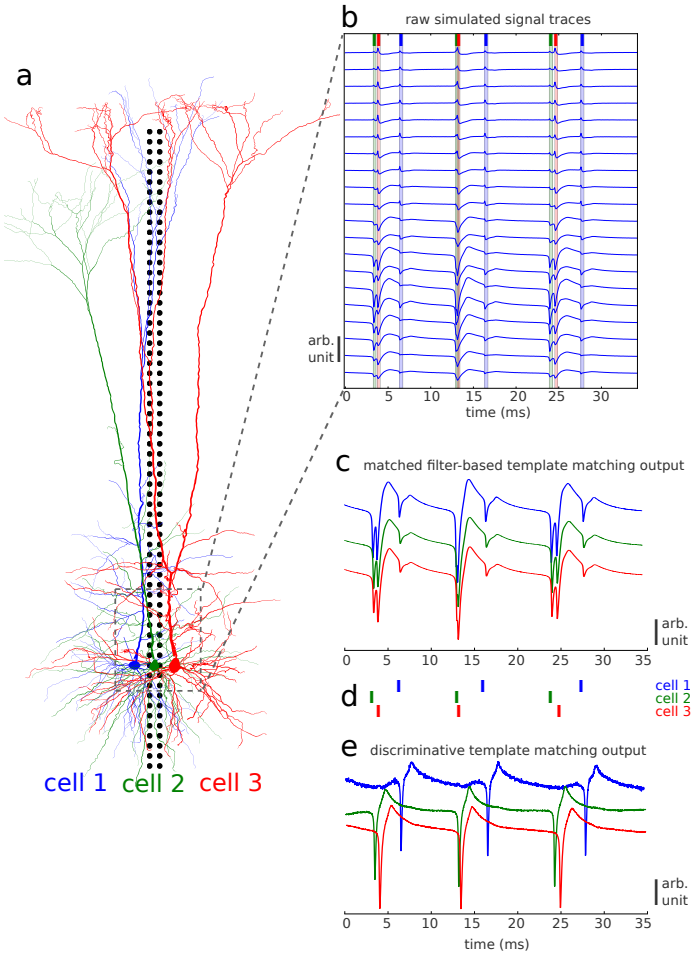

Figure 2: **a** Drawing of the simulated setup. The setup contains three neocortical layer 5b morphologically detailed pyramidal cells (cell 1: blue, cell 2: green, cell 3: red) and a 128-channel high density probe of which only the electrode locations are shown by black dots. **b** 35ms of simulated data for 20 out of 128 channels. Times at which the simulated neurons generate spikes are highlighted by the active cell's corresponding color. From this it can be seen that each individual neuron is active three times in the given data slice. **c** Matched filter-based template matching output of the three filters trained on the individual templates which were extracted using the ground truth spike times. **d** Ground truth spike times for each cell. **e** Discriminative template matching output of the three filters each trained using all three templates.

B. Illustrative example

For illustrative purposes, a small neural ensemble containing three cells [Hay et al., 2011] was simulated using LFPy [Lindén et al., 2014]. For each neuron n the matched filter coefficients $\mathbf{w}_{match}$ are calculated according to (1). The matched filter outputs $\mathbf{z}^{(n)}_{match}[t] = \mathbf{w}^T_{match}\overline{\mathbf{y}}^{(n)}[t]$ are shown in Figure 2**c** and are color coded according to the template's putative neuron.

One can observe from Figure 2**c** that the filter outputs of all three matched filtering-based template matching filters are very similar. This is due to the fact that all three templates have a similar spatial distribution. These filter outputs are not suitable for spike sorting. Even a greedy iterative approach will fail, because cell 1 has a significantly lower spike energy in the extracellular recording. In iterative schemes for resolving overlap each filter output is normalized with respect to its corresponding template energy. This will relatively boost the response of cell 2 and cell 3 on the filter output of cell 1, leading to all spikes being classified as belonging to cell 1.

In Figure 2**e** the color coded filter outputs of the proposed discriminative template matching filters are shown. The filter coefficients $\mathbf{w}^{(n)}_{disc}$ are calculated by setting $\hat{\mathbf{R}}^{(n)}_{\overline{\mathbf{i}}\overline{\mathbf{i}}} = \sum_{j\neq n}\hat{\mathbf{R}}^{(j)}_{\boldsymbol{\tau}\boldsymbol{\tau}} + \alpha\mathbf{I}$ in (6) with $j \in \{1,2,3\}$, i.e., the interference covariance matrix is built from the templates of the two non-target spikes. A scaled version of the identity matrix $\mathbf{I}$ is added to each interference covariance matrix to make sure all interference covariance matrices are invertible. Thresholding the discriminative template matching outputs directly solves the spike sorting problem.

The threshold-based spike sorting capability of the discriminative template matching might seem to come at a cost. Comparing the filter outputs in Figure 2**c** with the filter outputs in Figure 2**e**, it can be seen that the filter outputs in Figure 2**e** are more noisy. Because background baseline noise is generally not causing the detection of false positives, the increased background noise is acceptable. We argue that the optimal spike sorting filter must focus on attenuating non-target spikes and other interfering transients while maintaining a decent response to the target spike. A quantitative validation of the proposed filter design method will be given in Section IV.

C. Estimation of the interference covariance matrix

In this section a method will be proposed for estimating an interference covariance matrix $\mathbf{R}^{(n)}_{\overline{\mathbf{i}}\overline{\mathbf{i}}}$, which will capture the most harmful interfering sources, and hence put a larger emphasis on these in the optimization of (5). This matrix can then be used to design the optimal filter coefficients given by (6) to isolate the spikes of a single neuron while suppressing interfering spikes of other neurons or interfering transients which might cause false positives otherwise.

The interference covariance matrix estimation algorithm consists of six steps, as visualized in Figure 3, which have to be performed for each detectable neuron separately. This algorithm is operated a-priori on a representative subset of the data, which we refer to as the training set (it is noted that – for offline spike sorting – the training set can in principle contain the complete data set). It uses classic template matching to compute reasonable estimates of the interference covariance matrices, which at their turn will then be used to design (better) discriminative template matching filters:

1) **Required prior knowledge**: every template matching algorithm assumes that a representative spike template for the target neuron is available. In practice this template is estimated by using a clustering based spike sorting algorithm which can identify a subset of the spikes of the target neuron contained in the training set. The template is then estimated by averaging over this subset of target spikes (see (1)). This interference covariance estimation algorithm will also use the a priori known spike times of the target neuron.
2) **Matched filtering**: The available target spike template is then used for a matched filtering-based template matching pass over the training set data.
3) **Target safe zone**: We assume that it is possible that several of the target spikes in the training set have not been identified in the prior clustering phase of step 1.

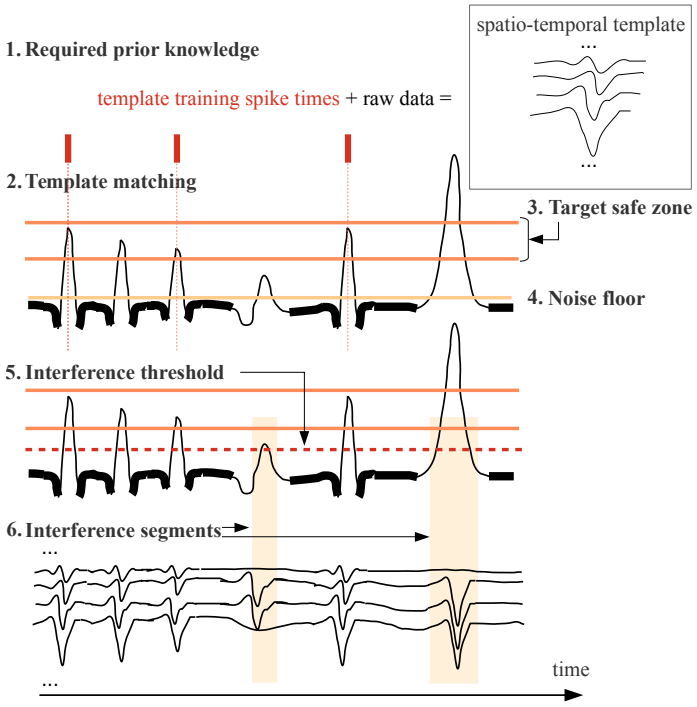

Figure 3: Graphical representation of the proposed algorithm for estimating the interference covariance matrix needed for the discriminative template matching filter design. 1. The prior knowledge required for the algorithm exists of a set of example spike times for the target neuron. 2. A matched filter is computed and applied to the recording, using the spike template as filter coefficients. The effect of background noise on the filter output is depicted by bold signal segments. 3. In order to prevent leakage of spikes from the target neuron into the interference covariance matrix estimate, a target safe zone is determined. Each peak that falls within this target safe zone is excluded for the estimation of the interference covariance matrix. 4. Based on the filter output statistics the noise floor is estimated. 5. By combining both the target safe zone and the noise floor, the interference threshold is calculated. 6. All peaks that have their maximum above the interference threshold and outside the target safe zone are used for estimating the interference covariance matrix. The signal segments involved in the estimation of the interference covariance matrix are shaded.

Therefore, to avoid a too strong leakage of the spatio-temporal structure of the target spike covariance into the interference covariance estimate, a *target safe zone* is defined. This target safe zone determines the amplitude range within which the output signal of the matched filter in step 2 can be considered to be a target spike with sufficient confidence. The target safe zone is calculated from the known spike times in step 1. Using the available target neuron spike times, the maximum output response of the matched filter is determined in a finite window around each available spike time. Next the maximum p_{max} and minimum p_{min} over all these output response maxima are computed. The target safe zone is defined as the interval $((1-\alpha)p_{min}, (1+\alpha)p_{max})$ with e.g. $\alpha = 0.1$.

4) **Noise floor**: The following step consists of estimating the amplitude of the baseline noise n_{est} in the output response of the matched filter over the training set data. This is done by calculating the median of the absolute value of the output response where all $2ms$ windows around known spike times are ignored in the calculation to further reduce the estimation bias.

5) **Interference threshold**: The interference threshold T_i is then calculated as $T_i = \beta p_{min} + (1-\beta)n_{est}$ with e.g. $\beta = 0.5$. Data segments corresponding to time instants t_j where the matched filter response crosses the interference threshold and where the maximum filter response of the crossing event does not fall into the target safe zone, are classified as interference data segments. Cases exist where no interference is detected and the matched filter-based template matching is sufficiently discriminative already.

6) **Interference covariance**: With the interference data segments $\bar{\mathbf{i}}^{(n)}[t_j] = \bar{\mathbf{y}}^{(n)}[t_j]$ at time t_j available, one can estimate the covariance matrix $\hat{\mathbf{R}}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)}$ characterizing the spatiotemporal coherence of the interfering events:

$$\hat{\mathbf{R}}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} = \sum_j \bar{\mathbf{i}}^{(n)}[t_j]\bar{\mathbf{i}}^{(n)}[t_j]^T. \quad (7)$$

This interference covariance matrix can together with the

target spike template be used in (6) to find the filter coefficients for discriminative template matching. Due to the use of an interference threshold, only data segments with the most relevant interference activity are included, allowing the filter design to mainly focus on the cancellation of these harmful sources. This interference threshold thus implicitly determines the weights p_k in (4). If all data segments without target spike would be used to estimate the interference covariance matrix, the filter design would 'waste' its degrees of freedom on the cancellation of noise that is not harmful anyway. In that case, it would optimize the average SNR, but not the SPIR.

We assume that the interference covariance estimation algorithm is not perfect, and that some target spike information might be included in the interference covariance matrix estimate. However, in Appendix B it is shown that the filter design method can cope with such target spike leakage, and as such that imperfect interference covariance estimates are not necessarily a big concern.

If interference is sufficiently attenuated in the discriminative template matching filter output, one can recalculate the interference threshold as explained in step 5 (possibly with a different value for β) on the discriminative matching output and use that threshold as a spike detection threshold.

IV. Validation

A. Description of validation data set

Discriminative template matching with an interference covariance matrix computed using the algorithm from Section III-C is compared to matched filtering (with pre-whitening) as in Section II-B. The comparison will focus on the capability of both filter types to perform threshold-based spike sorting using a simple thresholding operation. The threshold-based spike sorting capability is of interest for online real-time spike sorting, because this will allow a fast and cheap spike sorting implementation.

Both types of template matching are assessed on in-vivo extracellular ground truth data. Two different types of ground truth data are used for validation:

1) **Paired recordings** [Neto et al., 2016] are obtained by closely positioning an extracellular silicon probe to a neuron which is targeted by a juxtacellular micro-pipette. The signal measured by the micro-pipette can be used to extract ground truth spike times, because it contains high SNR peaks at times when the targeted neuron fires an action potential. For a spike of the targeted neuron to be visible on the extracellular probe, the distance between the micro-pipette and the silicon probe should be less than $50\mu m$.
2) **Hybrid data** [Rossant et al., 2015] is obtained from in-vivo HD recordings. Each recording is spike sorted using some available spike sorting algorithm. An operator then manually selects the spike clusters which consist of single unit activity only. The spikes belonging to some verified single unit cluster are then subtracted from the original recording and migrated to another spatial region resulting in a so-called hybrid recording. Since the timestamps of the newly injected spikes are known in the hybrid recording, these recordings can be used as ground truth data sets.

The specific datasets used for this work are available online [1,2].

Fifteen neurons were selected from this data for further analysis. The selection criteria used are: the existence of interfering sources for the considered neuron and a lower bound on the number of spikes (> 50 spikes) for that neuron. The spike count lower bound was chosen such that sufficient spikes are available for the estimation of templates. From all recordings the first five minutes of data are analyzed.

The channel selector for each neuron passes only data from electrodes within a circular region with a radius of $100\mu m$ around the electrode which shows the highest voltage deflection for spikes of that neuron. The exact number of electrodes involved in calculating a covariance matrix is ultimately determined by the probe layout used to make the recording containing the neuron under investigation. The delay

[1]Paired recording data: http://www.kampff-lab.org/validating-electrodes/
[2]Hybrid data: http://phy.cortexlab.net/data/sortingComparison/datasets/

line length for all filters is chosen such that samples from the last $1ms$ can be stored, which is usually sufficient to capture a spike waveform.

Prior to the filter design as well as the actual filtering, the data is minimally preprocessed. Preprocessing of the raw extracellular data is kept limited because online algorithms require a limited processing delay. Each channel is separately bandpass filtered between 300 and 10000Hz using a linear phase finite impulse response filter. This filter operation extracts the multi-unit activity from the raw data and suppresses low-frequency signal content and high-frequency measurement noise.

B. Validation parameters

We examined spike sorting sensitivity and precision for each combination of validation neuron and spike sorting method. *Sensitivity* is defined as the number of recovered target spikes (true positives) over the total number of ground truth spikes for that target neuron. A related measure that is even more important in spike sorting is precision. *Precision* is defined as the number of recovered target spikes (true positives) over all recovered events, including false detection. The importance of precision comes directly from the definition of spike sorting: unraveling multi-unit spike activity into single unit spike activity. A precision lower than 1 indicates the presence of non-target spikes/events in the spike sorting output.

For threshold-based spike sorting, sensitivity and precision depend on the value of the threshold applied at the filter output. To make it possible to fairly compare different methods, an optimal threshold is defined for each filter output. The optimal threshold is chosen by sweeping the threshold and calculating the sensitivity and precision for each position of the threshold. Because precision is more valuable for spike sorting than sensitivity, the optimal threshold is chosen from all thresholds with a precision > 0.9 that maximizes the sum of the sensitivity and precision. When no thresholds are available that give a precision > 0.9, the optimal threshold is chosen as the threshold that maximizes only the precision.

C. Signal-to-peak-interference maximization

Before showing quantitative results, the key mechanism of the discriminative template matching filter, namely maximizing the SPIR at its output, is graphically validated for in-vivo recorded data. For this purpose a segment of both a matched filter output and a discriminative template matching output is shown in Figure 4. The target neuron spiked once over the duration of the depicted segment, this spike time is indicated by the red asterisk.

For this segment the matched filter output (blue) has a clear peak when the target neuron is active. Besides the peak coinciding with the neuron being active, two more minor peaks can be found in the matched filter output later in time. The discriminative template matching filter output (green) also has a significant peak when the target neuron spikes. Apart from this activity-related peak, there are no other clear peaks in the filter output. This observation is also reflected by the increase in SPIR for the depicted segment between the matched filtering output and the discriminative template matching output. This increase in SPIR is beneficial for the accuracy and robustness of the proposed threshold-based spike sorting scheme.

From Figure 4 it can also be seen that the increase in SPIR comes at a cost. This cost is an increase of the baseline noise (as was already pointed out in Section III-B). Up to some extent this increase of the baseline noise will not harm the threshold-based spike sorting performance.

D. Spike sorting performance

Figure 5**a** shows the spike sorting sensitivity and accuracy for each validation neuron for three different spike sorting methods. First threshold-based spike sorting using matched filters (blue) is compared to threshold-based spike sorting using discriminative template matching (green). It can be observed that the use of discriminative template matching filters generally outperforms the use of matched filter in terms of the sum of sensitivity and precision. In terms of sensitivity and precision separately the use of discriminative template matching outperforms template matching, with the exception of neuron 12 which has a notably higher precision. However

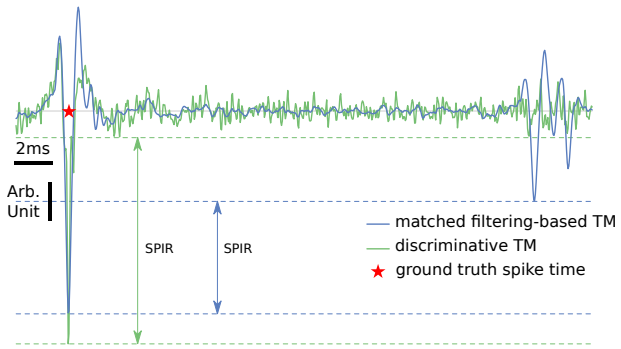

Figure 4: The output of both the matched-filtering based template matching (blue) and the discriminative template matching with interference suppression (green) filter applied to in-vivo recorded data is shown. Both filter outputs are normalized with respect to their standard deviation. The ground truth spike time of the target neuron in this segment, is indicated by a red asterisk. The double arrows indicate the signal-to-peak-interference ratio for both outputs.

this precision is only reached at an almost zero sensitivity, making this matched filter practically useless.

Next, threshold-based spike sorting using discriminative template matching is compared to spyking circus (red) [Yger et al., 2016]. Spyking circus is a state-of-the-art spike sorting pipeline, which employs a greedy template matching algorithm (with iterative template subtraction) for resolving overlapping spikes. To allow for a fair comparison between both methods, the manual curation phase of spyking circus is replaced by ground truth assisted cluster merging. From Figure 5**a** it can be seen that both methods perform very similar in terms of both sensitivity and precision, except for neuron 11.

The reason for the failure of threshold-based spike sorting using discriminative template matching on neuron 11 comes from the fact that the response of the interfering neurons to the initial matched filter (see step 2 of the interference covariance matrix estimation algorithm in Subsection III-C) by accident falls within the target spike safe zone. This leads to a discriminative template matching filter which does not put any effort in minimizing this specific peak interferer. This problem can be alleviated during the filter design phase by re-estimating the interference covariance matrix on the intermediate discriminative template matching outputs in the training set. As the initial discriminative template matching filter alters the response to the interfering neuron compared to the original matching filter (as both are based on different filter design paradigms), it will not fall in the spike safe zone anymore with a high probability. As a result, the interfering neuron will now be included in the interference covariance matrix for the re-estimation of the discriminative template matching filter. Such an approach leads for neuron 11 to a spike sorting sensitivity and precision above 95 percent. For the other neurons, applying the interference covariance estimation on the initial matched filter output proved sufficient.

Figure 5**b** summarizes the precision and sensitivity for all three methods. From this figure it can be seen that both threshold-based spike sorting using discriminative template matching and spyking circus perform equally well.

V. Conclusion

This work introduced the threshold-based spike sorting pipeline, which is promising for online spike sorting and closed-loop neuroscience. The idea behind threshold-based spike sorting is that single-unit activity can be obtained from a simple thresholding operation on the output of each template matching filter. Such a pipeline has a low computational complexity and a predictable delay. However, the template matching filters have to sufficiently discriminate between the target neuron and interfering neurons for the threshold-based spike sorting to work.

It was shown that a matched filtering-based template matching approach is not suitable for use in threshold-based spike sorting. Matched filtering methods have to rely on iterative schemes to extract single-unit activity from the template matching filter outputs. Therefore, the discriminative template matching method was proposed which was shown to outperform the matched filtering approach in terms of threshold-based spike sorting performance. When comparing the single-shot threshold-based spike sorting performance with the performance of a widespread spike sorting package like spyking circus [Yger et al., 2016], it was shown that both

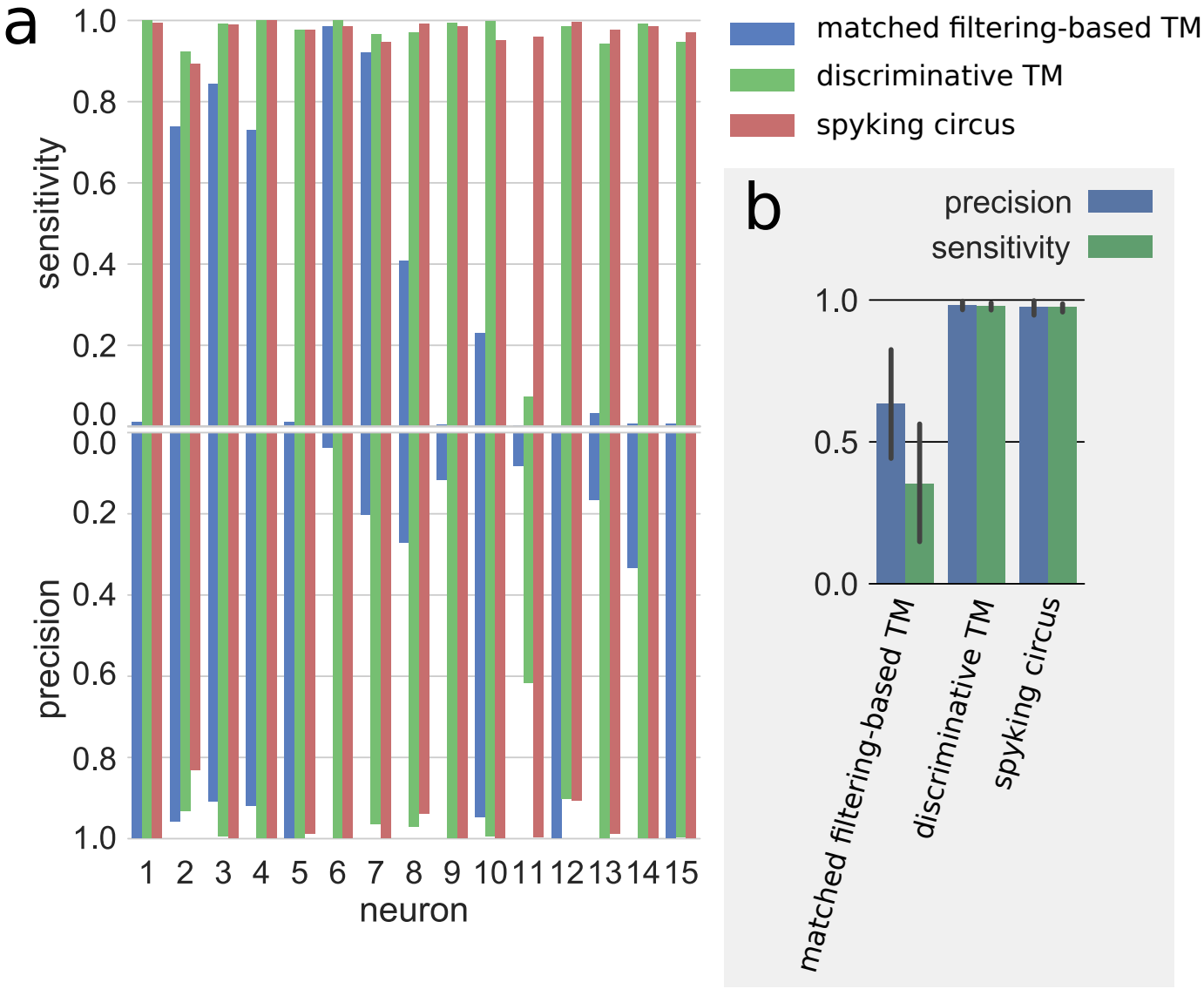

Figure 5: **a** The spike sorting sensitivity and precision for all analyzed ground truth neurons are shown. For each neuron three different algorithms were tested: template matching threshold-based (blue), discriminative template matching with interference suppression threshold-based (green), and spyking circus [Yger et al., 2016] (red). **b** Summary of precision (blue) and sensitivity (green) over all neurons for the three methods. The error bars indicate a 95 percent confidence interval.

approaches have a similar spike sorting performance. These results indicate that threshold-based spike sorting using discriminative template matching is a useful alternative for recent spike sorting packages, where threshold-based spike sorting has the advantage of algorithmic simplicity and a low (and deterministic) input-output delay, making it suitable for online closed-loop neuroscience experiments.

The power of discriminative template matching lies in the way of estimating the so-called interference covariance matrix, which is an important ingredient of the filter design method. The interference covariance matrix estimation algorithm implicitly weights the different interfering sources, taking into account whether or not the segments cause spurious peaks on a matched filtering template matching output, and as such are classified as peak interferers. The discriminative template matching filter can be thought of as optimizing the SPIR,

instead of the SNR as is the case for a matched filter.

An approach also trying to suppress interfering spikes was already proposed in [Roberts and Hartline, 1975]. However, this template matching technique did not make it into today's spike sorting packages. One possible reason for this is given by [Franke et al., 2012], where the filters are argued to be less robust to noise, because they are designed under stronger constraints. We believe that the advent of HD probes results in increased spatial information and as such brings new opportunities for template matching filters with an extra constraint on suppression of interference. Discriminative template matching differs from approaches like [Roberts and Hartline, 1975] mainly in the way the interference is estimated. Instead of a model-based approach, the interference in our filter design method is estimated in a data-driven way. Such a data-driven approach removes the necessity of a complete prior spike sorting, i.e., only example spike times of target neurons of interest are required as an input. Another advantage of our approach over model-based approaches is that different interfering sources are automatically weighted based on their harmfulness.

Another advantage of threshold-based spike sorting is that it could be a candidate for on-probe spike sorting. The need for on-probe sorting might arise from an ever increasing channel count on HD probes together with a limited data transfer bandwidth, making it a challenging task to transfer raw data from a probe to, e.g., a computer. The simple architecture of a finite impulse response filter and the low complexity of a single thresholding operation, can be cheaply implemented in hardware. Recently, active CMOS probes [Raducanu et al., 2016] have been developed, already having integrated electronics on probe. On-probe sorting also eliminates the transfer delay, which is beneficial for use in closed-loop experiments.

APPENDIX

A. Derivation of the optimal filter

The proposed filter $\mathbf{w}_{disc}^{(n)}$ for threshold-based single-unit activity detection of neuron n is the solution to the optimization problem defined below:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau\tau}}^{(n)} \mathbf{w}}{\mathbf{w}^T \mathbf{R}_{\mathbf{\bar{i}\bar{i}}}^{(n)} \mathbf{w}} \quad \text{s.t.} \quad \|\mathbf{w}\| = 1, \tag{8}$$

where

$$\mathbf{R}_{\boldsymbol{\tau\tau}}^{(n)} = \boldsymbol{\tau}^{(n)} \boldsymbol{\tau}^{(n)^T} \tag{9}$$

is the positive semidefinite spatio-temporal covariance matrix of the deterministic spike template waveform and $\mathbf{R}_{\mathbf{\bar{i}\bar{i}}}^{(n)}$ is the positive definite spatio-temporal peak interference covariance matrix for neuron n. The filter satisfying (8) thus optimizes the signal-to-peak-interference ratio at its output. To derive a closed-form solution, (8) is rewritten as:

$$\max_{\mathbf{w}} \mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau\tau}}^{(n)} \mathbf{w} \quad \text{s.t.} \quad \begin{matrix} \|\mathbf{w}\| = 1 \\ \mathbf{w}^T \mathbf{R}_{\mathbf{\bar{i}\bar{i}}}^{(n)} \mathbf{w} = C \end{matrix}, \tag{10}$$

with $C \in \mathbb{R}$ a constant. One can easily see that maximizing a fraction is equal to maximizing the numerator while keeping the denominator constant. To solve this constrained optimization problem, the method of Lagrange multipliers [Boyd and Vandenberghe, 2004] is used. The Lagrangian of this optimization problem (ignoring the norm constraint) is given by:

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau\tau}}^{(n)} \mathbf{w} - \lambda \left(\mathbf{w}^T \mathbf{R}_{\mathbf{\bar{i}\bar{i}}}^{(n)} \mathbf{w} - C \right). \tag{11}$$

The solution to the optimization problem is then found by setting the partial derivative w.r.t. $\mathbf{w}$ equal to zero and solving it for $\mathbf{w}$.

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 2 \left(\mathbf{R}_{\boldsymbol{\tau\tau}}^{(n)} - \lambda \mathbf{R}_{\mathbf{\bar{i}\bar{i}}}^{(n)} \right) \mathbf{w} = \mathbf{0}. \tag{12}$$

Rearranging (12) reveals that (8) reduces to a generalized eigenvalue problem [Golub and Van Loan, 1996], where the optimal filter coefficients correspond to a generalized eigenvector:

$$\mathbf{R}_{\boldsymbol{\tau\tau}}^{(n)} \mathbf{w} = \lambda \mathbf{R}_{\mathbf{\bar{i}\bar{i}}}^{(n)} \mathbf{w}. \tag{13}$$

By combining (10) and (13), we get

$$\mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau\tau}}^{(n)} \mathbf{w} = \lambda \mathbf{w}^T \mathbf{R}_{\mathbf{\bar{i}\bar{i}}}^{(n)} \mathbf{w} = \lambda C. \tag{14}$$

This shows that (10) is maximized if λ is maximal, i.e., if $\mathbf{w}$ is chosen equal to the generalized eigenvector corresponding to the maximal generalized eigenvalue λ, i.e., the solution of (13) for maximal lambda.

In this work the template covariance matrix is a rank-1 matrix by definition as shown in (9). This rank-1 structure makes that there is only one generalized eigenvector which has a non-zero eigenvalue, which then also leads to a closed-form solution. This can be seen from substituting (9) in the generalized eigenvalue problem (13):

$$\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)^{-1}} \boldsymbol{\tau}^{(n)} \boldsymbol{\tau}^{(n)^T} \mathbf{w} = \lambda \mathbf{w}. \tag{15}$$

It is noted that the left-hand side of (15) consists of a scaled version of the vector $\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)^{-1}} \boldsymbol{\tau}^{(n)}$ scaled by the scalar $\boldsymbol{\tau}^{(n)^T} \mathbf{w}$. As a result, the closed-form solution of (8) is:

$$\mathbf{w}_{disc}^{(n)} = \frac{\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)^{-1}} \boldsymbol{\tau}^{(n)}}{\left\| \mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)^{-1}} \boldsymbol{\tau}^{(n)} \right\|}, \tag{16}$$

where the norm division is added to satisfy the norm constraint that was ignored before.

It is noted that the GEVD-based approach also allows for the use of higher-rank template covariance matrices $\mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)}$. The optimal filter can then be found by solving (13) using a generic generalized eigenvalue problem solver [Golub and Van Loan, 1996]. This might be beneficial in the case of neurons with non-stationary spike waveforms, e.g. bursting neurons.

B. Robustness against target leakage

Assuming that the interference covariance estimation algorithm is imperfect, at some point, target spike segments will leak into the interference covariance matrix estimate. Such a faulty interference covariance matrix $\hat{\mathbf{R}}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)}$ can be modeled as:

$$\hat{\mathbf{R}}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} = \mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} + c\ \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)}, \tag{17}$$

with $c \in \mathbb{R}$.

The discriminative template matching filter for the pair of matrices $\left(\mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)}, \hat{\mathbf{R}}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)}\right)$ is the solution to the following optimization problem:

$$\mathbf{w}_{disc}^{(n)} = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \mathbf{w}}{\mathbf{w}^T \left(\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} + c\ \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \right) \mathbf{w}}. \tag{18}$$

Through algebraic manipulation of (18), it is shown that the discriminative template matching filter for $\left(\mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)}, \hat{\mathbf{R}}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)}\right)$ is equal to the discriminative template matching filter for $\left(\mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)}, \mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)}\right)$:

$$\begin{aligned}
&\underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \mathbf{w}}{\mathbf{w}^T \left(\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} + c\ \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \right) \mathbf{w}} \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\mathbf{w}^T \left(\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} + c\ \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \right) \mathbf{w}}{\mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \mathbf{w}} \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \left(\frac{\mathbf{w}^T \mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} \mathbf{w}}{\mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \mathbf{w}} + c \right) \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\mathbf{w}^T \mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} \mathbf{w}}{\mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \mathbf{w}} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^T \mathbf{R}_{\boldsymbol{\tau}\boldsymbol{\tau}}^{(n)} \mathbf{w}}{\mathbf{w}^T \mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)} \mathbf{w}}.
\end{aligned} \tag{19}$$

We can conclude that target spike leakage into the interference covariance matrix, as modeled by (17), has no influence on the calculation of discriminative template matching filter coefficients. In practice, the filter coefficients will slightly differ because the leakage covariance will be calculated on a per spike basis including the background noise. This additional background noise will slightly change the relative weights between the interfering spike covariance and the background noise within $\mathbf{R}_{\bar{\mathbf{i}}\bar{\mathbf{i}}}^{(n)}$.

REFERENCES

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[Chung et al., 2017] Chung, J. E., Magland, J. F., Barnett, A. H., Tolosa, V. M., Tooker, A. C., Lee, K. Y., Shah, K. G., Felix, S. H., Frank, L. M., and Greengard, L. F. (2017). A fully automated approach to spike sorting. *Neuron*, 95(6):1381–1394.

[Ciliberti and Kloosterman, 2017] Ciliberti, D. and Kloosterman, F. (2017). Falcon: a highly flexible open-source software for closed-loop neuroscience. *Journal of Neural Engineering*, 14(4):045004.

[Ekanadham et al., 2014] Ekanadham, C., Tranchina, D., and Simoncelli, E. P. (2014). A unified framework and method for automatic neural spike identification. *Journal of Neuroscience Methods*, 222:47–55.

[Franke et al., 2012] Franke, F., Jäckel, D., Dragas, J., Müller, J., Radivojevic, M., Bakkum, D., and Hierlemann, A. (2012). High-density microelectrode array recordings and real-time spike sorting for closed-loop experiments: an emerging technology to study neural plasticity. *Frontiers in Neural Circuits*, 6(December):1–7.

[Franke et al., 2015] Franke, F., Quian Quiroga, R., Hierlemann, A., and Obermayer, K. (2015). Bayes optimal template matching for spike sorting – combining fisher discriminant analysis with optimal filtering. *Journal of Computational Neuroscience*, 38(3):439–459.

[Gibson et al., 2012] Gibson, S., Judy, J. W., and Marković, D. (2012). Spike sorting: The first step in decoding the brain. *IEEE Signal processing magazine*, 29(1):124–143.

[Golub and Van Loan, 1996] Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

[Hay et al., 2011] Hay, E., Hill, S., Schürmann, F., Markram, H., and Segev, I. (2011). Models of Neocortical Layer 5b Pyramidal Cells Capturing a Wide Range of Dendritic and Perisomatic Active Properties. *PLoS Computational Biology*, 7(7):e1002107.

[Jun et al., 2017a] Jun, J. J., Mitelut, C., Lai, C., Gratiy, S., Anastassiou, C., and Harris, T. D. (2017a). Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction. *bioRxiv*.

[Jun et al., 2017b] Jun, J. J., Steinmetz, N. A., Siegle, J. H., Denman, D. J., Bauza, M., Barbarits, B., Lee, A. K., Anastassiou, C. A., Andrei, A., Aydın, Ç., et al. (2017b). Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232.

[Kadir et al., 2014] Kadir, S. N., Goodman, D. F., and Harris, K. D. (2014). High-dimensional cluster analysis with the masked em algorithm. *Neural computation*.

[Kaneko et al., 1999] Kaneko, H., Suzuki, S., Okada, J., and Akamatsu, M. (1999). Multineuronal spike classification based on multisite electrode recording, whole-waveform analysis, and hierarchical clustering. *IEEE Transactions on Biomedical Engineering*, 46(3):280–290.

[Kessy et al., 2017] Kessy, A., Lewin, A., and Strimmer, K. (2017). Optimal Whitening and Decorrelation. *The American Statistician*, 1305(September):0–0.

[Lewicki, 1998] 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):R53–R78.

[Lindén et al., 2014] Lindén, H., Hagen, E., Łęski, S., Norheim, E. S., Pettersen, K. H., and Einevoll, G. T. (2014). LFPy: a tool for biophysical simulation of extracellular potentials generated by detailed model neurons. *Frontiers in Neuroinformatics*, 7(January):41.

[Lopez et al., 2016] Lopez, C. M., Mitra, S., Putzeys, J., Raducanu, B., Ballini, M., Andrei, A., Severi, S., Welkenhuysen, M., Van Hoof, C., Musa, S., and Yazicioglu, R. F. (2016). A 966-electrode neural probe with 384 configurable channels in 0.13m SOI CMOS. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 59, pages 392–393. IEEE.

[Marre et al., 2012] Marre, O., Amodei, D., Deshmukh, N., Sadeghi, K., Soo, F., Holy, T. E., and Berry, M. J. (2012). Mapping a Complete Neural Population in the Retina. *Journal of Neuroscience*, 32(43):14859–14873.

[Neto et al., 2016] Neto, J. P., Lopes, G., Frazão, J., Nogueira, J., Lacerda, P., Baião, P., Aarts, A., Andrei, A., Musa, S., Fortunato, E., Barquinha, P., and Kampff, A. R. (2016). Validating silicon polytrodes with paired juxtacellular recordings: method and dataset. *Journal of Neurophysiology*, 116(2):892–903.

[Pachitariu et al., 2016] Pachitariu, M., Steinmetz, N., Kadir, S., Carandini, M., and Harris, K. D. (2016). Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *bioRxiv*.

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

[Quiroga et al., 2004] Quiroga, R. Q., Nadasdy, Z., and Ben-Shaul, Y. (2004). Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering. *Neural Computation*, 16(8):1661–1687.

[Raducanu et al., 2016] Raducanu, B. C., Yazicioglu, R. F., Lopez, C. M., Ballini, M., Putzeys, J., Wang, S., Andrei, A., Welkenhuysen, M., van Helleputte, N., Musa, S., Puers, R., Kloosterman, F., van Hoof, C., and Mitra, S. (2016). Time multiplexed active neural probe with 678 parallel recording sites. In *2016 46th European Solid-State Device Research Conference (ESSDERC)*, volume 2016-Octob, pages 385–388. IEEE.

[Roberts and Hartline, 1975] Roberts, W. M. and Hartline, D. K. (1975). Separation of multi-unit nerve impulse trains by a multi-channel linear filter algorithm. *Brain Research*, 94(1):141–149.

[Rossant et al., 2015] Rossant, C., Kadir, S. N., Goodman, D. F. M., Schulman, J., Belluscio, M., Buzsaki, G., and Harris, K. D. (2015). Spike sorting for large, dense electrode arrays. *bioRxiv*, 19(4):015198.

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

[Steinbach et al., 2004] Steinbach, M., Ertöz, L., and Kumar, V. (2004). The challenges of clustering high dimensional data. In *New directions in statistical physics*, pages 273–309. Springer.

[Wouters et al., 2018] Wouters, J., Kloosterman, F., and Bertrand, A. (2018). Data-driven multi-channel filter design with peak-interference suppression for threshold-based spike sorting in high-density neural probes. In *Proc. IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP), Calgary, Alberta, Canada, Apr. 2018*.

[Yger et al., 2016] Yger, P., Spampinato, G. L. B., Esposito, E., Lefebvre, B., Deny, S., Gardella, C., Stimberg, M., Jetter, F., Zeck, G., Picaud, S., Duebel, J., and Marre, O. (2016). Fast and accurate spike sorting in vitro and in vivo for up to thousands of electrodes. *bioRxiv*.