

Automated detection of latency tracks in microneurography recordings using track correlation

Brian Turnquist^{a,*}, Brandon RichardWebster^{a,b}, Barbara Namer^c

^a*Department of Mathematics and Computer Science, Bethel University, Arden Hills, MN 55112, USA*

^b*Department of Computer Science and Engineering, University of Notre Dame, IN 46556, USA*

^c*Department of Physiology and Pathophysiology, University of Erlangen, Universitätsstr. 17, 91054 Erlangen, Germany*

Abstract

Background: The marking technique in microneurography uses stimulus-induced changes in neural conduction velocity to characterize human C-fibers. Changes in conduction velocity are manifested as variations in the temporal latency between periodic electrical stimuli and the resulting APs. When successive recorded sweeps are displayed vertically in a “waterfall” format, APs correlated with the stimulus form visible vertical tracks. Automated detection of these latency tracks is made difficult by sometimes poor signal-to-noise ratio in recordings, spontaneous neural firings uncorrelated with the stimuli, and multi-unit recordings with crossing or closely parallel tracks.

New Method: We developed an automated track-detection technique based on a local linearization of the latency tracks of stimulus-correlated APs. This technique enhances latency tracks, eliminates transient noise spikes and spontaneous neural activity not correlated with the stimulus, and automatically detects latency tracks across successive sweeps in a recording.

Results: We evaluated our method on microneurography recordings showing varying signal quality, spontaneous firing, and multiple tracks that run closely parallel and cross. The method showed excellent detection of latency tracks in all of our recordings.

Comparison with existing method(s): We compare our method to the commonly used track detection method of Hammarberg as implemented in the Drever program.

Conclusions: Our method is a robust means of automatically detecting latency tracks in typical microneurography recordings.

Keywords: microneurography, latency tracking, marking, track correlation

1. Introduction

Various methods have been used to characterize C-fiber types. For nociceptive and single skin sympathetic C-fibers, microneurography has been used for decades and has been well described [1] [2]. For the study of sympathetic muscle and skin C-fibers, compound action potentials are typically used. In microneurography, a recording electrode is inserted into a peripheral nerve of an awake, human volunteer. When the needle is close to a C-fiber bundle, neuronal activity characteristic of unmyelinated afferents can be induced by scratch stimuli

applied to the dorsum of the foot. Innervation territories of individual C-fibers are then located through transcutaneous electrical stimulation with a pointed electrode. C-fiber action potentials (APs) are identified by their low conduction velocity (< 2 m/s). A pair of thin needles is then inserted intracutaneously into the innervation territory and used to stimulate the C-fibers under observation. It is typical at each recording site for several C-fibers to be activated by the electrical stimulus and the APs of up to twenty C-fibers to be recorded simultaneously. There are several challenges in discriminating the APs of these C-fibers. In contrast to single-fiber recordings in animals, the shape and size of successive APs of an individual C-fiber in microneurography may vary considerably during a recording due to noise and additional overlapping APs. Furthermore, the shape and size of APs of different C-fibers can sometimes be

*Corresponding author at: Department of Mathematics and Computer Science, Bethel University, 3900 Bethel Drive, Arden Hills, MN 55112, USA.

Email addresses: turnquist@bethel.edu (Brian Turnquist), bjrichardwebster@gmail.com (Brandon RichardWebster), namer@physiologie1.uni-erlangen.de (Barbara Namer)

very similar, posing a problem for template-based discrimination. For instance, APs of sympathetic fibers, which are driven centrally, are also recorded and cannot be easily differentiated based on shape from those evoked by the electrical stimuli in the periphery. Thus automatic spike sorting (the assigning of APs to fibers) based only on spike shape has a high error rate in many recordings.

However, a second fiber characteristic can aid in classification. Following repetitive electrical stimulation at a fixed frequency from the skin, APs of individual C-fibers can be detected with the recording electrode at stable latencies from the stimulus. Variations in conduction velocities of distinct C-fibers produce distinct latencies from the stimulus. However, the AP response to a natural stimulus (e.g. mechanical stimulation) cannot be reliably assigned to an individual C-fiber, since they have varying shapes and sizes and other APs will be present as well. To assess the responsiveness of individual C-nociceptive fibers and skin sympathetic fibers to different natural stimuli the “marking method” was developed. A sudden increase in latency is observed when the stimulation frequency at the skin is increased or after the afferent fiber has been otherwise activated, e.g., by natural stimuli (“marking”) [2]. This increased latency in the marking technique can be seen by displaying successive sweeps recorded after the repeated electrical stimuli in a “waterfall” or “falling-leaf” display (Figure 1). Note the typical pattern of a stable latency, followed by a significant increase in latency in response to the additional stimulus, and then followed by a gradual recovery period back to the stable latency. Marking is due to activity-dependent slowing of conduction in C-fibers. Conduction of an AP renders the axonal membrane of some afferent C-fibers less excitable for tens of seconds and thus reduces conduction velocity of subsequent APs [3].

It has been proposed that the slowing mechanism is dependent on sodium channel kinetics and intracellular sodium accumulation [4] [5]. The marking method has also been used to examine the biophysical properties of C-fibers. Two main afferent C-fiber types have been described by their biophysical and sensory properties. Mechano-sensitive C-fibers, which resemble the common polymodal nociceptor, show only moderate slowing in response to repetitive electrical stimulation with rising frequencies, whereas mechano-insensitive C-fibers, which are also termed “silent-nociceptors,” show a considerable amount of slowing. A limitation of the marking technique is that A-fibers and mechano-sensitive C-fibers show very little slowing with single extra action potentials, so the marking is not evident

in noisy recordings. Further, this method is only applicable in nociceptive C-fibers and skin sympathetic C-fibers showing sufficient activity-dependent slowing. Although the marking correlates well with the number of action potentials fired beforehand, a computation of the firing rate is not possible. [3]

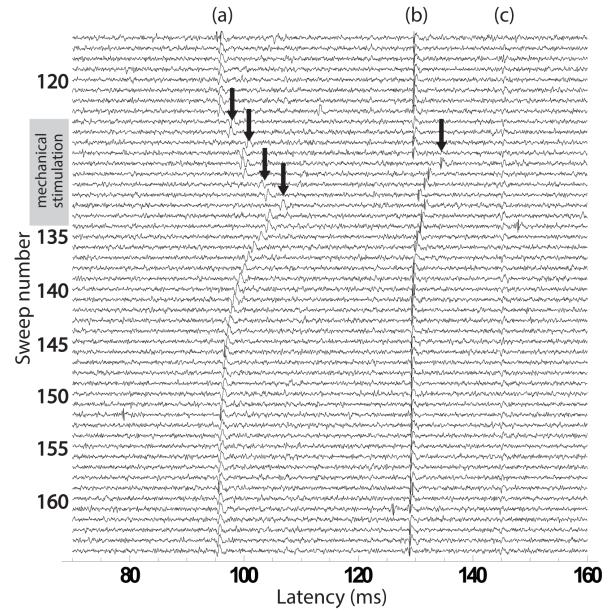


Figure 1: Latency tracks with markings in three C-fibers. Each horizontal sweep (116 through 165) shows the signal recorded 70–160 ms after an electrical stimulus delivered every 4 seconds. The electrical, stimulus-evoked action potentials in three individual C-fibers are shown at stable latencies (a), (b), (c). The arrows point to “markings”, which are sudden latency increases caused by activity-dependent slowing of conduction due to additional action potentials evoked by mechanical stimulation (in this case, poking the skin using a stiff 75g Von Frey hair during sweeps 123–134). Fibers (a) and (b) were mechano-sensitive as they showed marking during mechanical stimulation followed by a gradual recovery to their original stable latency. Fiber (c) showed no marking, so it is classified as a mechano-insensitive C-fiber or silent nociceptor.

As can be seen in Figure 1, APs that are correlated with the electrical stimulus can be observed as vertical “tracks” on a waterfall plot. Hammarberg developed a technique for automatically tracking changes in AP latency in waterfall displays based on three steps: AP detection, AP tracking, and parameter estimation [6]. In the detection phase, each sweep is scanned to identify all subsignals that correlate with a fixed AP template. The correlation threshold is set by the user with lower thresholds leading to more detected APs in this phase. In the tracking phase, Multiple Hypothesis Tracking links the APs identified in the detection phase into coherent tracks across successive sweeps in the waterfall. Multiple Hypothesis Tracking is a Bayesian probabilis-

tic technique based on a solution to the classic multiple-target radar detection problem. It is assumed in this phase that changes in AP latency during the recovery phase will follow an exponential decay model (which generates the predicted latencies) and a Kalman filter tunes the parameters of the model. Once all tracks have been identified, they are characterized in the parameter estimation phase by a least-squares fit to an exponential function $y_0 + Ae^{-a_0(t-t_0)}$ where y_0 is the stable latency of the AP track and t_0 is the time of the additional stimulus.

The AP template in Hammarberg's method is always the same and was created "once for all" by averaging a set of manually selected APs from previous recordings. The correlation of each possible subsignal of the sweep and its latency from the electrical stimulus are features used for AP discrimination. However, as noted above, experience in microneurography over the past decade has shown that several factors hinder accurate automated classification of AP tracks: poor signal-to-noise ratio (SNR), crossing tracks, and parallel tracks. As Hammarberg's approach begins with a threshold-based discrimination of APs to use in the track-detection phase, it is dependent on good SNR and the correlation threshold chosen. Recordings such as that shown in the raw signal in Figure 3A produce an "all or nothing" problem for the experimenter attempting to set the threshold for the template matching filter to discriminate noise spikes from AP signals. A second problem is in the tracking phase when two or more AP tracks are running in parallel or cross each other (e.g. Figure 5). First-hand experience has shown that the Drever implementation of Hammarberg's technique has difficulty tracking APs in these scenarios.

In contrast to Hammarberg, our technique begins with latency track detection and makes no assumptions about AP shape. It assumes only that AP tracks will show greater power along a series of linearly aligned subsignals of neighboring sweeps. After detecting tracks in the recording, we use those tracks to detect and discriminate APs. We make no assumption of any particular AP shape during the track detection phase, but in the subsequent phase we build a template along the detected tracks to discriminate APs. We make no assumption of any particular model for the slowing or speeding or AP latencies, only that changes in latency are locally smooth except for when marking occurs. As with Hammarberg, once a track has been identified as a series of latencies, the parameter estimation phase is straightforward via a least-squares fit to an exponential (or any other) model.

2. Materials and methods

2.1. The Track-Correlation Filter

The basis of our track-correlation filter is that the most important signals studied in microneurographic recordings are temporally correlated with the stimulus and their latencies change smoothly in time. As shown in Figure 1, for each fiber in a recording, there is an implicit function mapping the sweep number of the electrical stimulus to the latency of the resulting AP. This function may show significant variation over the entire course of an experiment. However, when viewed over a small number of neighboring sweeps the *local* change in latency measured in milliseconds per sweep is approximately linear (except in the case of the jump that occurs with marking). Let $\text{RMS}(k, t)$ be the root mean square power measurement for a small time window of samples surrounding a latency t on sweep S_k (Figure 2). The time window width is fixed and should encompass the most significant power in a typical AP (1 to 3 ms). We let R denote a radius of sweeps above and below S_k to be included in our calculation. For instance, if $R = 2$, then there will be 5 sweeps included in the calculation. Given a latency shift m (ms/sweep), we compute $\overline{\text{RMS}}_m(k, t)$, the median RMS power R sweeps above and below t lying along a line with slope m , as

$$\overline{\text{RMS}}_m(k, t) = \text{median of } \{\text{RMS}(k+r, t+rm) : -R \leq r \leq R\}$$

The value of $\overline{\text{RMS}}_m(k, t)$ changes with the slope m as the band of RMS windows above and below t better aligns with any local tracks near t . We define $\text{TC}(k, t)$, the *track correlation* on sweep S_k at latency t as the maximum value of $\overline{\text{RMS}}_m(k, t)$ among all latency shifts m , and we define the *track correlation slope* $M(t, k)$ to be latency shift m where that maximum occurred (Figure 2, band (b)). So,

$$\text{TC}(k, t) = \overline{\text{RMS}}_{M(t, k)}(k, t)$$

Note that except for the first and last R sweeps in the recording and for latencies near the edges of each sweep, the track correlation is defined everywhere regardless of whether the time point lies along a track of APs. The largest track correlations on a sweep will coincide with AP latency tracks crossing that sweep. By computing the track correlation of each latency on each sweep, we can create a surface plot of track correlation (Figure 3B) clearly showing the locations of tracks in the recording. Even with very noisy recordings, tracks will be visible on this plot.

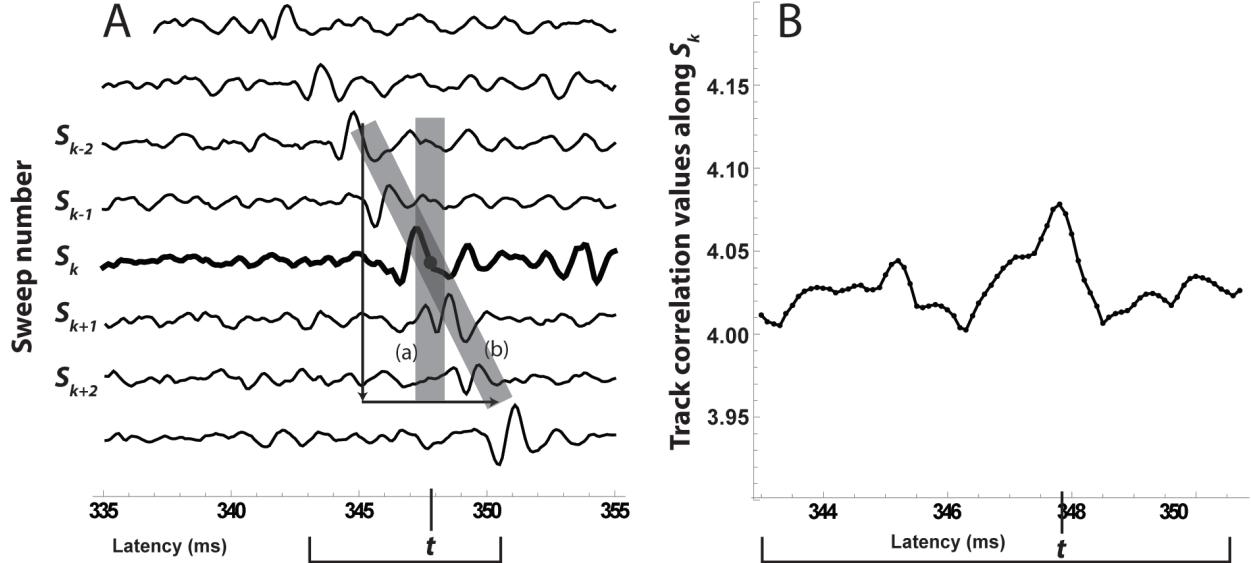


Figure 2: Panel A: Two bands of RMS power windows for different latency shifts at latency t on sweep S_k . The radius $R = 2$ means we include two sweeps above and two sweeps below S_k . The RMS power is measured on each sweep within a 1.0 ms window. The latency shift (slope) for the vertical band (a) is 0.0 ms/sweep and for band (b) is 1.0 ms/sweep. The median of the five RMS windows for latency shift (a) will be close to the background noise level as the large RMS value at t will be an outlier. For band (b) all RMS values are measured near APs of the slowing latency track so the median will be detectably distinct from background noise. In this example, the latency shift (b) will produce the maximum median value at t so this median will be defined as $\text{TC}(k, t)$ the track correlation at t on sweep k . The associated slope of (b) (1.0 ms/sweep) is defined as $M(k, t)$. Panel B: Track correlations computed along S_k at each latency t from 343.0 to 350.5 ms in this 10 kHz recording.

2.2. Iterative Track Extension

Given a starting sweep and latency, we describe an iterative process for creating a track by upward and downward extensions of a local linearization of the track to the likely location of the track on neighboring sweeps.

2.2.1. Measuring Background Noise

To determine whether an arbitrarily chosen latency along a sweep is part of a track of APs, we need a measurement of the background noise. The overall distribution of track correlations is right-skewed with a very small group of outliers at the upper end of the distribution which are the track correlation values lying along AP tracks. There are significant variations in this distribution from one recording to the next. To analyze a given experimental recording, we first randomly sample 1000 points on the waterfall and define Noise_{TC} to be the median of their track correlations. Since very few track correlations in the sample will be from AP latency tracks (typically less than 5%), the median gives a stable measurement of track correlations where there is no AP track.

2.2.2. Track Initialization

Given any latency t_k on a sweep S_k , we shift it slightly to the left and right in a window $t_k \pm \varepsilon$ to find the latency \hat{t}_k having maximal track correlation. This relocates the starting latency t_k to the center of the nearest likely track. If $\text{TC}(k, \hat{t}_k) > \text{Noise}_{TC}$, then \hat{t}_k becomes the first latency on the track. Otherwise, t_k is assumed to not lie near an AP track and the procedure stops with no track detected at that latency.

2.2.3. Upward Track Extension

Given a point \hat{t}_k ($k > 1$) already on a track, we define \hat{m}_k , the slope (ms/sweep) of the local linearization of the track at \hat{t}_k , as follows. If the current track has length at least R , then \hat{m}_k is defined to be the slope of the linear least-squares fit of the latencies $\hat{t}_k, \hat{t}_{k+1}, \hat{t}_{k+2}, \dots, \hat{t}_{k+R-1}$ on their respective sweeps. If the track has length less than R , we define $\hat{m}_k = M(k, \hat{t}_k)$ as a likely approximation of the slope of the local linearization of the track.

We next project \hat{t}_k upward to the previous sweep S_{k-1} at latency $t_{k-1} = \hat{t}_k - \hat{m}_k$, which is the latency lying above \hat{t}_k along a slope \hat{m}_k . Although this latency t_{k-1} should lie very close to the desired track, we maximize its track correlation to a nearby \hat{t}_{k-1} within the window $t_{k-1} \pm \varepsilon$ as in the initialization step above. We must also ensure that

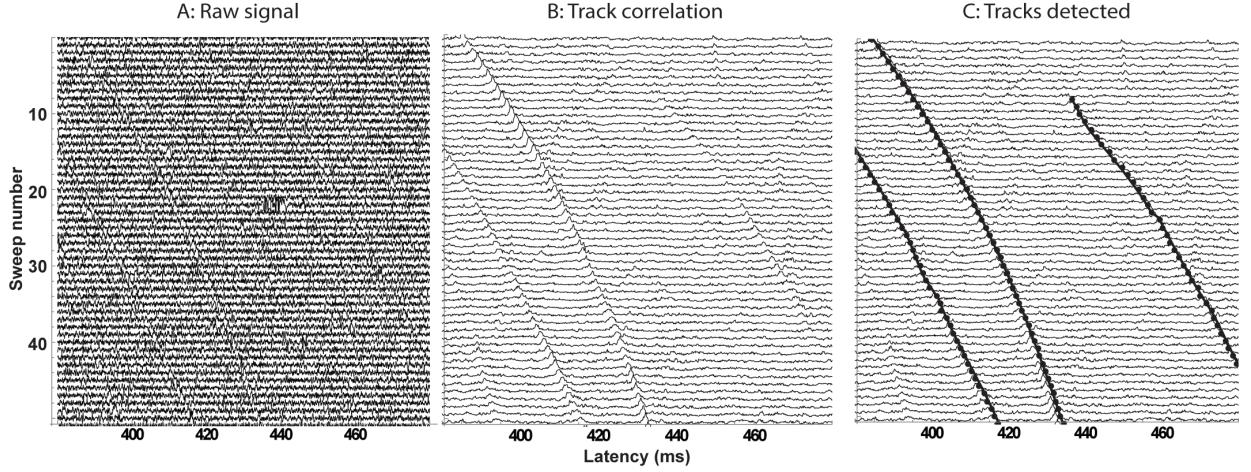


Figure 3: Panel A shows a recording with significant noise. B shows the same portion of the recording with the track-correlation filter applied at 0.1 ms increments along each sweep. A larger radius (in this case, $R = 5$) increases the contrast between tracks and background noise. Note that the surge at 435 ms in A is eliminated by the track-correlation filter. Panel C shows the three latency tracks detected along the track correlations visible in B.

the track correlation slope of the selected latency \hat{t}_{k-1} closely aligns with the established slope \hat{m}_k of the existing track. This requirement allows a track of low amplitude APs to successfully cross a track of higher amplitude APs without gravitating toward the higher RMS power of the intersecting track. Specifically, assuming ε is the maximal track latency shift allowed from one sweep to the next, we define \hat{t}_{k-1} to be the latency t in the window $t_{k-1} \pm \varepsilon$ satisfying the following conditions

- (a) The value of $\cos(\frac{\pi}{\varepsilon}(\hat{m}_k - M(k-1, t))) \cdot \text{TC}(k-1, t)$ is maximal in the window $t_{k-1} \pm \varepsilon$
- (b) \hat{t}_{k-1} lies near a peak of the track correlations in the window $t_{k-1} \pm \varepsilon$

The cosine term in condition (a) creates a scaling factor on $\text{TC}(k-1, t)$ that favors latencies whose track correlation slopes $M(k-1, t)$ are close to the established direction \hat{m}_k . When $M(k-1, t)$ and \hat{m}_k are identical, then the scaling factor is one. As $M(k-1, t)$ deviates from \hat{m}_k , the scaling factor gradually falls off toward 0.

When a small amplitude track runs closely parallel to a large amplitude track, the actual maximum of the track correlations in a window for the smaller track may lie at the edge of the window close to the larger track (Figure 5B). The condition in (b) forces \hat{t}_{k-1} to be flanked by points with lower track correlations. Specifically, we require that

$$\min(\text{TC}(k-1, \hat{t}_{k-1}) - \text{TC}(k-1, \hat{t}_{k-1} - \frac{\varepsilon}{2}), \text{TC}(k-1, \hat{t}_{k-1}) - \text{TC}(k-1, \hat{t}_{k-1} + \frac{\varepsilon}{2})) > 0$$

This need not be a strict maximum of the track correlations in the window, but (b) ensures that \hat{t}_{k-1} is close to a local maximum.

2.2.4. Edge Detection

Since the median is an edge-preserving filter, then the upper and lower ends of a track are relatively easy to detect using only the current value \hat{t}_{k-1} and the previous value \hat{t}_k . In the case of upward track extension, the iterative track extension process will stop when it either reaches the first recorded sweep in the experiment or it reaches the upper end of the track of APs. In the case where \hat{t}_{k-1} has been computed as in the previous step, we append it to the front of the existing track $\hat{t}_k, \hat{t}_{k+1}, \hat{t}_{k+2}, \dots$ if and only if $\text{TC}(k-1, \hat{t}_{k-1}) \geq (\text{TC}(k, \hat{t}_k) - \text{Noise}_{\text{TC}})/2$. In other words, we require that the track correlation at \hat{t}_{k-1} has dropped less than 50% since the previous latency \hat{t}_k relative to the background noise level.

2.2.5. Downward Track Extension and Edge Detection

The downward track extension process is analogous to upward track extension where we have an already established track $\dots, \hat{t}_{k-2}, \hat{t}_{k-1}, \hat{t}_k$ and wish to add a new latency \hat{t}_{k+1} to the track. The process of projection to the sweep S_{k+1} followed by scaled maximization of track correlation will produce \hat{t}_{k+1} , which we use to extend the track $\dots, \hat{t}_{k-2}, \hat{t}_{k-1}, \hat{t}_k$ provided it satisfies the edge detection criteria relative to \hat{t}_k .

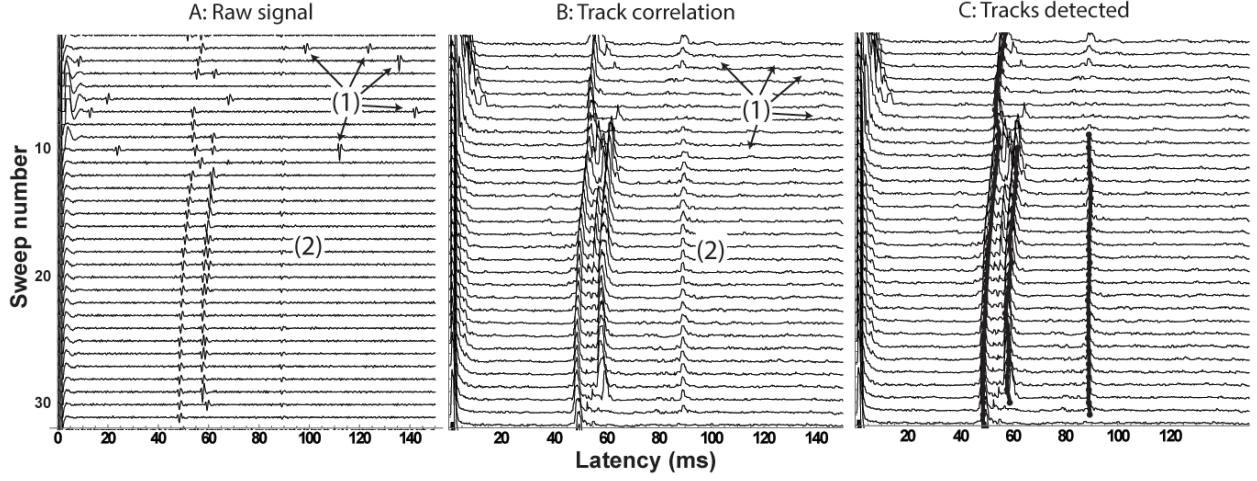


Figure 4: Panel A shows an unfiltered waterfall plot having unusually good signal-to-noise ratio. The electrical stimulus artifact is visible at the far left. Panel B shows the corresponding track correlations ($R = 2$). Note that the filter eliminates APs not lying on a track (1) and enhances the faint track at (2). C shows three latency tracks detected using the track correlation. The electrical artifact can also produce a latency track, but this is not useful.

2.3. Automated Track Detection

The method in 2.2 above describes creating a track from any starting latency t on a sweep. Although these starting points can be selected manually by the experimenter (for example, through a mouse click), we describe here a method for automatically detecting those points that will produce good AP tracks. The common sense approach would begin with computing all track correlations at all points in the recording (Figures 3 and 4) and generating tracks at the points with the highest track correlations. However, a typical recording could require 10 million track correlation computations so this is inefficient and unnecessarily slow considering that the vast majority of the track correlations calculated will not lie along AP tracks. Further, even if all track correlations are calculated, the largest values will lie along the same few tracks in the recording yet only one representation of each track is needed. Different choices of starting points on the same AP track may produce slightly different results. For instance, if a shorter track is contained within a longer track then we should only retain the longer track. If $T = \hat{t}_k, \hat{t}_{k+1}, \hat{t}_{k+2}, \dots, \hat{t}_{k+n}$ is a track then we define its *track quality* as

$$Q(T) = \sum_{i=k}^{k+n} \text{TC}(i, \hat{t}_i)$$

so that longer tracks and tracks that run along larger APs will have the highest track quality. We also say that a track T_1 *subsumes* a track T_2 if T_1 contains all of the track latencies of T_2 , allowing for very small variations in the exact latencies on each included sweep. We note

from the definition that if T_1 subsumes T_2 , then $Q(T_1) \geq Q(T_2)$.

To efficiently find the tracks with the highest quality in the recording, we begin by randomly sampling 1000 RMS values from the overall recording and setting a threshold θ of two standard deviations above the mean. We then build a list of all local maxima above this threshold on each sweep, so that there is one peak per threshold crossing. This restricted list will typically contain less than 0.1% of the total possible points so calculating these track correlations is computationally tractable (Figure 5). To generate the entire AP track, we only need one AP from the track to cross the RMS threshold θ and serve as the starting point for the track extension process described above. Assume that the list $p_1, p_1, p_2, p_3, \dots, p_n$ enumerates the latencies and the associated sweeps where these peaks occurred and that P_i is the track generated at p_i . We start by letting $T_1 = P_1$ and inductively suppose that $T_1, T_2, T_3, \dots, T_m$ is already sorted in descending order by track quality where no track in the list subsumes any other track. Using the value of $Q(P_i)$ we assume that P_i is to be inserted at position k . Before inserting it we check to see whether P_i is subsumed by any tracks in the list $T_1, T_2, T_3, \dots, T_{k-1}$. If so, then P_i is redundant and not inserted into the list. Otherwise, we insert P_i at position k , shifting $T_k, T_{k+1}, T_{k+2}, \dots, T_m$ to the right and also removing any tracks subsumed by P_i . Once this process has been applied to the entire list $P_1, P_2, P_3, \dots, P_n$, the list $T_1, T_2, T_3, \dots, T_s$ has the following properties: (1) $Q(T_1) \geq Q(T_2) \geq Q(T_3) \geq \dots \geq Q(T_s)$ and (2) ev-

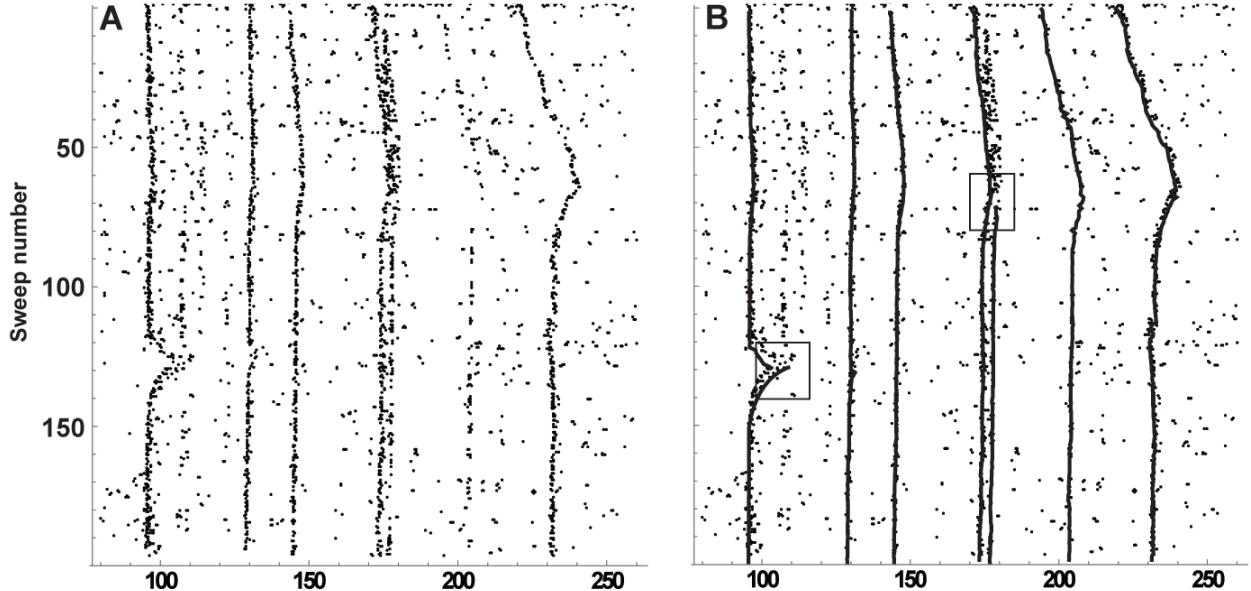


Figure 5: Panel A shows the local peak RMS values at least two standard deviations above the mean, which form a starting set for generating latency tracks. This restricted set contains less than 0.1% of the total possible starting points. Panel B shows the latency tracks detected. The insets in B are shown in Figure 6.

every track in the original list $P_1, P_2, P_3, \dots, P_n$ is either included in the list $T_1, T_2, T_3, \dots, T_s$ or subsumed by one of these tracks. So $T_1, T_2, T_3, \dots, T_s$ is a ranked list of the highest quality tracks in the recording and it is of a computationally manageable size.

2.4. Automated AP Discrimination along Tracks

In contrast to Hammarberg's approach [6], we discriminate APs only after tracks have been created. For any latency track $T = \hat{t}_k, \hat{t}_{k+1}, \hat{t}_{k+2}, \dots, \hat{t}_{k+n}$, we create a template of likely APs along that track by first measuring the maximum RMS power near each \hat{t}_i . We then take the second and third quartiles of those power measurements as being typical of the APs that lie along this track, and then form a template as the average waveform of those subsignals. Using this template we either retain or discard each \hat{t}_i along the track as follows. For each \hat{t}_i , all subsignals in a small neighborhood around it are compared to the template. If there is a subsignal having at least a least 50% of the RMS power (a configurable setting) of the template and also sufficiently matching the template, then \hat{t}_i is retained on the track. Otherwise, it is discarded (Figure 9). The similarity of each subsignal to the template is measured with cross-correlation using a user-configurable threshold (for instance, 0.75). Note that APs detected along the track will not typically lie right on the track (see Discussion section below).

While AP shape associated with a C-fiber may vary during the course of an experiment and the shape of APs associated with different fibers may be quite similar, it is typical to see differences in shape between APs from different fibers and those shapes will frequently remain stable for much of a recording (Figure 1). By building a new template along each detected track prior to AP detection, our approach can take advantage of the typical situation when an AP track shows a fairly consistent waveform shape, and it is also better suited to discriminating APs having an atypical shape.

3. Results

3.1. Track Correlation Filter

One goal of our technique was to increase the contrast between signal and noise for action potentials that are correlated with the stimulus. It is clear from Figure 3 that signals correlated with the electrical stimuli are enhanced relative to background noise. To measure this effect we selected a variety of latency tracks ($N = 30$) from different recordings and with different signal levels relative to the background noise in the recording and then compared the SNR in the unfiltered waterfall to the SNR in the track-correlated waterfall. Using the mean and standard deviation in the overall recording as a reference ($N = 10,000$), we computed

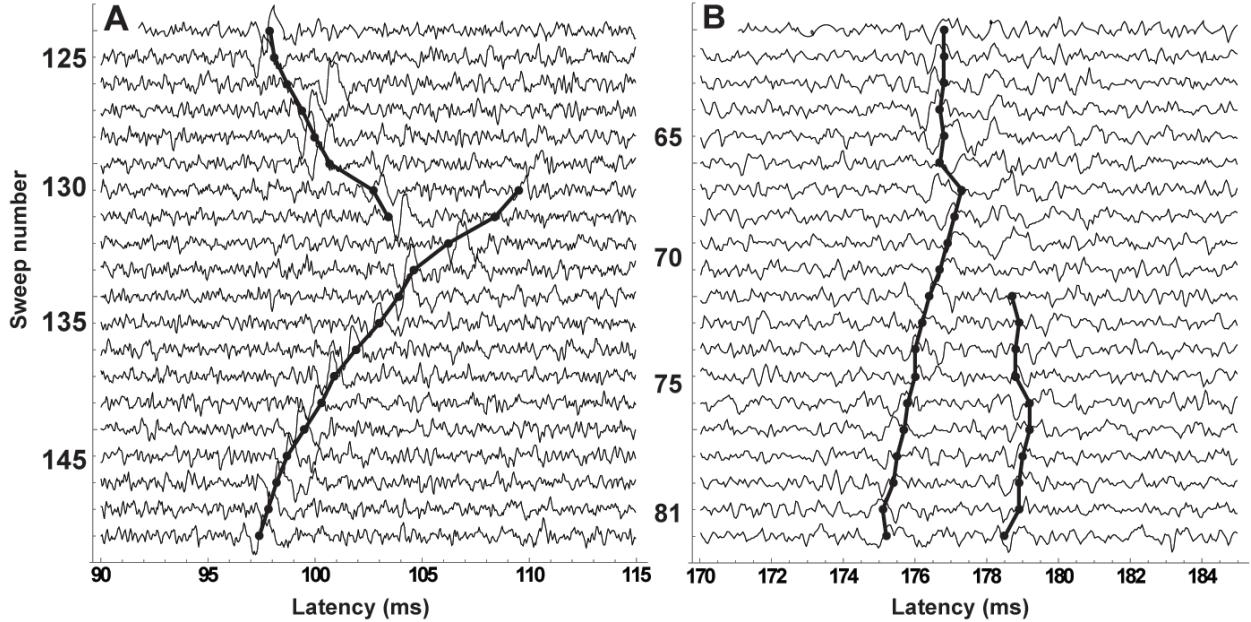


Figure 6: Panels A and B show the insets from Figure 5. In A, the lower track was detected first and then the upper track. In B, the longer track was detected first and the other track second. The track extension process stopped when it reached and coincided with the existing track.

the mean z-value along each latency track as a measurement of how detectable that track is relative to background noise. For each track we then compared its SNR measurement in the unfiltered waterfall to its SNR measurement in the track-correlation filtered waterfall (Figure 7). For recordings with poor signal quality relative to background noise (Figure 3), the track correlation SNR was 75 to 80% greater than the unfiltered SNR. For recordings where the signal was already good, the effect of the track-correlation filter was negligible. Figure 7 shows the smoothing effect of the track correlation filter and why it enhances the accuracy of track detection. The track correlation along the latency track stays well above the background noise level in the filtered signal in comparison to the RMS values in the unfiltered signal which often dip below the background noise level. As would be expected from its definition, the mean of track correlations in the background noise in a recording will be higher than in the unfiltered recording, but importantly, the standard deviation will be smaller. Finally, since the median is used to compute the track correlation, we also see that the track correlation falls off quickly at the edges of the latency track. In addition to its smoothing effect, track correlation also eliminates solitary spikes arising from electrical disturbances and spontaneous neural activity (Figures 3 and 4). Since these types of non-stationary signals will be outliers of the median along the optimal latency shift (Figure 2)

they are completely rejected by the filter. Specifically, if the radius for the filter is R , then the track correlation suppresses spikes not correlated with the electrical stimulus provided they form a track of length R or less. Conversely, in most experiments, there will be one or more sweeps where an expected AP on a latency track is missing. The track correlation filter will tolerate up to R missing spikes along a track (Figure 4) since the noise-level RMS values of the missing APs will be outliers in the median. Thus, track correlation is tolerant of up to R missing spikes along a track.

3.2. Automated Track Detection

For isolated tracks, the detection algorithm works extremely well, typically tracking exactly to the upper and lower edge of each latency track. However, in a multi-unit recording where some units are activated by the marking stimulus and others are not, there can be many AP tracks, some of which may run closely parallel to each other and even cross. As these types of responses occur commonly in recordings, our automated track detection method was tested on a variety of recordings with these features and varying signal-to-noise values. We also developed an artificial set of recordings to test the limits of the algorithm.

Crossing tracks pose a potential problem for track extension in determining which track should be chosen beyond the sweep where they cross. Our technique will

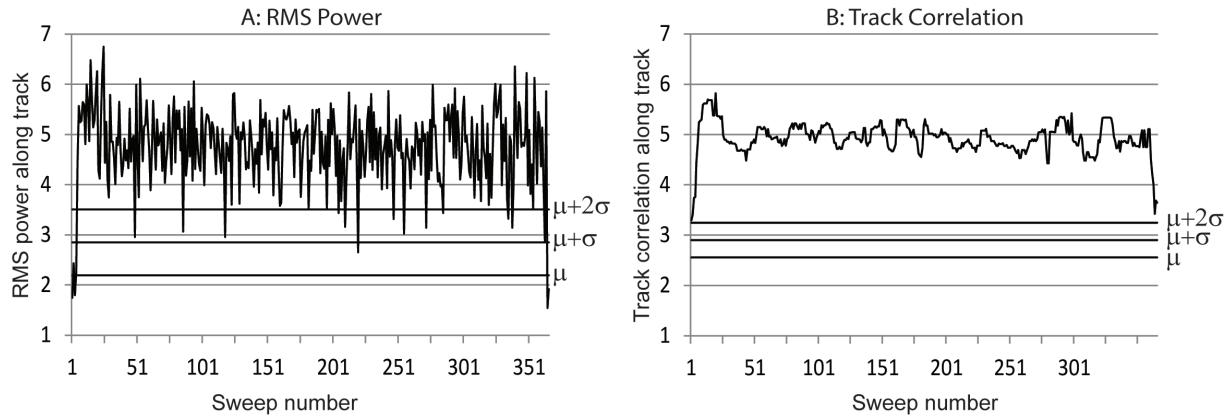


Figure 7: Panel A shows the RMS power values within 1 ms windows along a latency track spanning 366 sweeps. The horizontal lines show the mean and two standard deviations of the RMS values at 10,000 randomly sampled points in the recording. Panel B shows the track correlation values along the same latency track in the filtered waterfall. The horizontal lines show the mean and two standard deviations of track correlations at 10,000 randomly sampled points in the waterfall. Note that the track correlation values along the latency track remain well above the background noise level (in contrast to the corresponding RMS values).

generally choose to extend along the track with local latency shift most like the track being extended. Better SNR in the recording and using a larger radius leads to better performance in crossing scenarios. Figure 8 shows a signal with poor SNR and crossing tracks with latency shifts of -1.3 and 1.3 ms per sweep. In this example the 2.6 ms per sweep difference was artificially reduced to 2.0 ms per sweep (more nearly parallel) with identical results. However, if in addition the radius is reduced from 5 to 3 sweeps, then the second-ranked track (in terms of track quality described above) will stop at the crossing point. The next track automatically added will be a third track beginning just beyond the crossing point.

Parallel tracks are even more common in microneurographic recordings. A larger signal running parallel to a smaller signal poses a potential problem for the algorithm as the RMS values of the larger signal could draw the track extension from the smaller signal to the larger. However, the median rejects the single large RMS value from the larger signal and furthermore the cosine scaling factor prevents sharp changes in track extensions. In practice, even very small signals running closely parallel to larger signals can be effectively tracked (Figure 5).

3.3. Automated AP Discrimination along Tracks

Once the track has been detected, it is straightforward to detect sweeps along that track where an AP is missing (Figure 9). As described above, a template is created along the latency track by finding the latency for

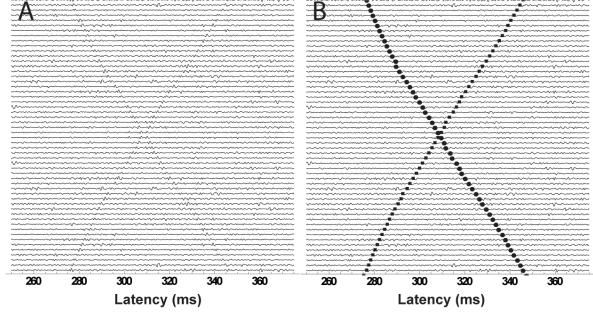


Figure 8: Occasionally latency tracks may cross along successive sweeps. Panel A shows an artificial file used for testing the algorithm's behavior when latency tracks cross at different angles. In this case the tracks cross with one showing slowing of conduction velocity of 1.3 ms per trace and the other showing speeding at -1.3 ms per trace. Panel B shows the two tracks detected using a radius of 5 traces. If two tracks cross at a very small angle, then the second track detected will stop where they join (cf. Figure 6B).

each sweep that maximizes power near the track. If the RMS value is less than 50% of the template power or the cross-correlation with the template is less than the desired amount (e.g. 75%), then the latency value on the track for that sweep is dropped and it is assumed that the unit did not respond to that particular stimulus or at least that its AP does not lie along that track.

4. Discussion

The threshold correlation filter and the track extension method for finding tracks can be considered inde-

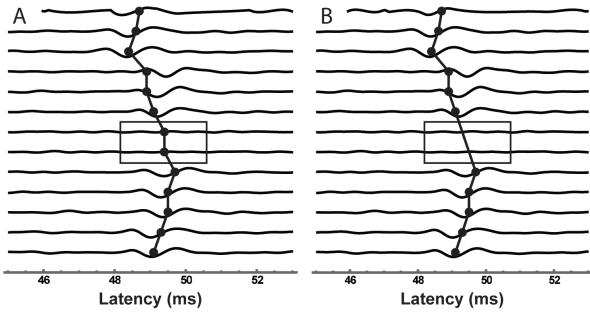


Figure 9: Panel A shows a portion of a latency track ($R = 5$). Note that the track is able to extend across the two missing APs at the center since R was at least 2. In Panel B, a template is created along the latency track to determine the average spike shape along that track. Using a requirement of at least 50% power and a 75% cross-correlation match, the latencies at the two missing APs were automatically dropped.

pendently. The track correlation filter on its own will smooth the background noise of the waterfall while preserving the signal amplitude in the original recording. It will also reject signals not correlated with the stimulus across multiple sweeps and is tolerant of missing APs along tracks. The track correlation filter could be used to preprocess a recording prior to application of other analysis techniques, such as Hammarberg's tracking method (where track correlation threshold discrimination would be used instead of a raw template-based discrimination).

In a microneurographic recording, the very small changes in latency that occur from one sweep to the next (called "jitter") have been used to characterize neural firing in various neuropathic pain models [7]. By measuring the skewness of the jitter distribution, the presence of spontaneous activity can be detected and differentiated from stimulus-evoked multiple firing. The tracks created by our algorithm are inherently smooth and the latency of the track itself will typically differ slightly from the latency of the nearby AP. Separate options in the software in which our algorithm is implemented (Dapsys 8, www.dapsys.net) allow each latency in the track to be aligned or "snapped" to various features of the nearby AP such as minimum peak value, maximum peak value, maximum power, and template match. As a result, the latency of the raw track could be compared to the latency of the AP near that track to provide an alternative measurement of jitter. In this case, the AP whose deviation from the track is being measured should be omitted from the slope calculation of the latency track to retain independence of the sample variables.

The detection algorithm can be used on any type of recording where the slowing or speeding of AP conduction is a characteristic of the neuron. For instance, in single-fiber electrophysiology recordings, the algorithm finds tracks easily. In such recordings, the generally good SNR also allows latencies of very small units to be tracked in post hoc analysis.

Performance of the overall algorithm is fast and runs in linear time on all types of recordings. If a latency track is created by double-clicking directly on the desired location in the waterfall, the creation of the track through the track extension algorithm is essentially instantaneous. If the option is used to automatically select the starting location for track extension, then indexing the tracks the first time takes several minutes. For instance, on a recording with 984 sweeps of 10,000 samples each, the indexing required four minutes. After this indexing of optimal starting locations was complete, each new track was detected and created instantaneously. For the purpose of creating Figures 3 and 4, we computed the track correlation at every point in the recording. This is time-consuming and unnecessary for the algorithm as mentioned earlier. However, if for some reason, there were a need to apply the filter to an entire recording, then near-perfect linear speed-up could be obtained through a parallel implementation on any number of cores. This would be a straightforward modification of the serial algorithm since track correlation is a localized computation.

5. Acknowledgements

Partial funding for the project was from the Bethel University REU and the first author receives remuneration for development of the software Dapsys. Thanks to Ryan Morgan for assistance with the early investigations into track correlation and to Martin Schmelz for helpful conversations about jitter.

- [1] A. B. Vallbo, K. E. Hagbarth, Activity from skin mechanoreceptors recorded percutaneously in awake human subjects., *Exp Neurol* 21 (1968) 270–289.
- [2] H. E. Torebjörk, R. G. Hallin, Responses in human a and c fibres to repeated electrical intradermal stimulation, *J. Neurol. Neurosurg. Psy.* 37 (1974) 653–664.
- [3] M. Schmelz, C. Forster, R. Schmidt, M. Ringkamp, H. O. Handwerker, H. E. Torebjörk, Delayed responses to electrical stimuli reflect c-fiber responsiveness in human microneurography, *Exp Brain Res* 104 (1995) 331–336.
- [4] R. De Col, K. Messlinger, R. W. Carr, Conduction velocity is regulated by sodium channel inactivation in unmyelinated axons innervating the rat cranial meninges, *The Journal of Physiology* 586 (4) (2008) 1089–1103.
- [5] J. Tigerholm, M. E. Petersson, O. Obreja, A. Lampert, R. Carr, M. Schmelz, E. Fransen, Modeling activity-dependent changes of

- axonal spike conduction in primary afferent c-nociceptors, Journal of Neurophysiology 111 (9) (2014) 1721–1735.
- [6] B. Hammarberg (Hansson), C. Forster, E. Torebjörk, Parameter estimation of human nerve c-fibers using matched filtering and multiple hypothesis tracking, IEEE Transactions on Biomedical Engineering 49 (4) (2002) 329–336.
- [7] J. Serra, H. Bostock, J. Aleu, E. Garcia, B. Cokic, X. Navarro, C. Quiles, Microneurographic identification of spontaneous activity in c-nociceptors in neuropathic pain states in humans and rats, Pain (2012) 42–55.