MANUSCRIPT ACCEPTED FOR PUBLICATION IN PSYCHOPHYSIOLOGY
[APPEARED SEPTEMBER 2015].
A published version of the article can be found here:

http://onlinelibrary.wiley.com/doi/10.1111/psyp.12521/abstract

Latency as a Region Contrast: Measuring ERP Latency Differences with Dynamic Time Warping

A.Zoumpoulaki[a], A.Alsufyani[a], M.Filetti[b], M.Brammer[c], H.Bowman[a,d]

[a] Center for Cognitive Neuroscience and Cognitive Systems, School of Computing, University of Kent, CT2 7NF, U.K.

[b] Helsinki Institute for Information Technology HIIT, Aalto Univeristy, Laskut, 01051, Finland.

[c] Institute of Psychiatry/ Centre for Neuroimaging Science, Kings College London, De Crespigny Park, London SE5 8AF, U.K.

[d] School of Psychology, University of Birmingham, Edgbaston, Birmingham, B15 2TT, U.K.

Corresponding Author:

Alexia Zoumpoulaki

Room SW12, School of Computing

University of Kent, Canterbury, Kent, CT2 7NF, U.K.

Tel: +44 (0)1227 823823, Fax: +44 (0)1227 762811

Email: az61@kent.ac.uk

Abstract

Methods for measuring latency contrasts are evaluated against a new method utilizing the Dynamic Time Warping algorithm. They are applied on simulated data, for different signal to noise ratios and two sizes of window (broad vs narrow). The results are subjected to statistical and ROC analysis. The analysis suggests that *DTW* performs better than the other methods, being less sensitive to noise as well as to placement and width of the window selected.

Keywords: ERP latency, *DTW, fractional area, peak,* P300

1.Introduction

Latency contrasts are central to Event Related Potential (ERP) research. For example, determining how an experimental manipulation changes the latency of a particular component can be extremely revealing of the cognitive process that it modulates. In particular, measurement of latency is key to determining the order in which cognitive processes are performed, and plays a pivotal role in mental chronometry.

In general, ERP latencies are considered very hard to measure, with the available methods being characterized by several weaknesses and the obtained results being difficult to interpret (Luck, 2005 p. 242). The most common methods used to determine latency differences between components are *peak* latency, fractional *peak* latency and *fractional area* (Luck, 2005; Handy, 2005; Kiesel et al., 2008). The following are three of the issues with these methods:

1) Point measure: the measures used to quantify the latency of a component all identify a single point in the time series, e.g. the *peak* or the point representing 50 percent of the area of a deflection. Then the temporal offset of that point relative to stimulus presentation is taken as a proxy for the component's latency. Identification of such a single point, especially in *peak* measures, is potentially highly sensitive to noise in the ERP, which may, for example, elevate a point far from the "true" *peak* of an ERP component (Figure 1). This difficulty can generate both type I errors (i.e. false positives in which the null hypothesis is inappropriately rejected) and type II errors (i.e. misses, in which the null hypothesis is inappropriately *not* rejected). This problem of sensitivity to noise, is probably particularly pronounced with the *peak* latency measure, and different approaches have been proposed to overcome it, i.e. using more robust methods such as the maximal average *peak*, local vs absolute *peak* etc. However, it can also arise under percentage area measures, e.g. see how the *50% fractional area* moves in Figure 1. At the same

time, although *fractional area* takes into consideration the shape of the waveform, it still returns a single point – the point where the preset fraction is met – for each of the conditions under investigation.

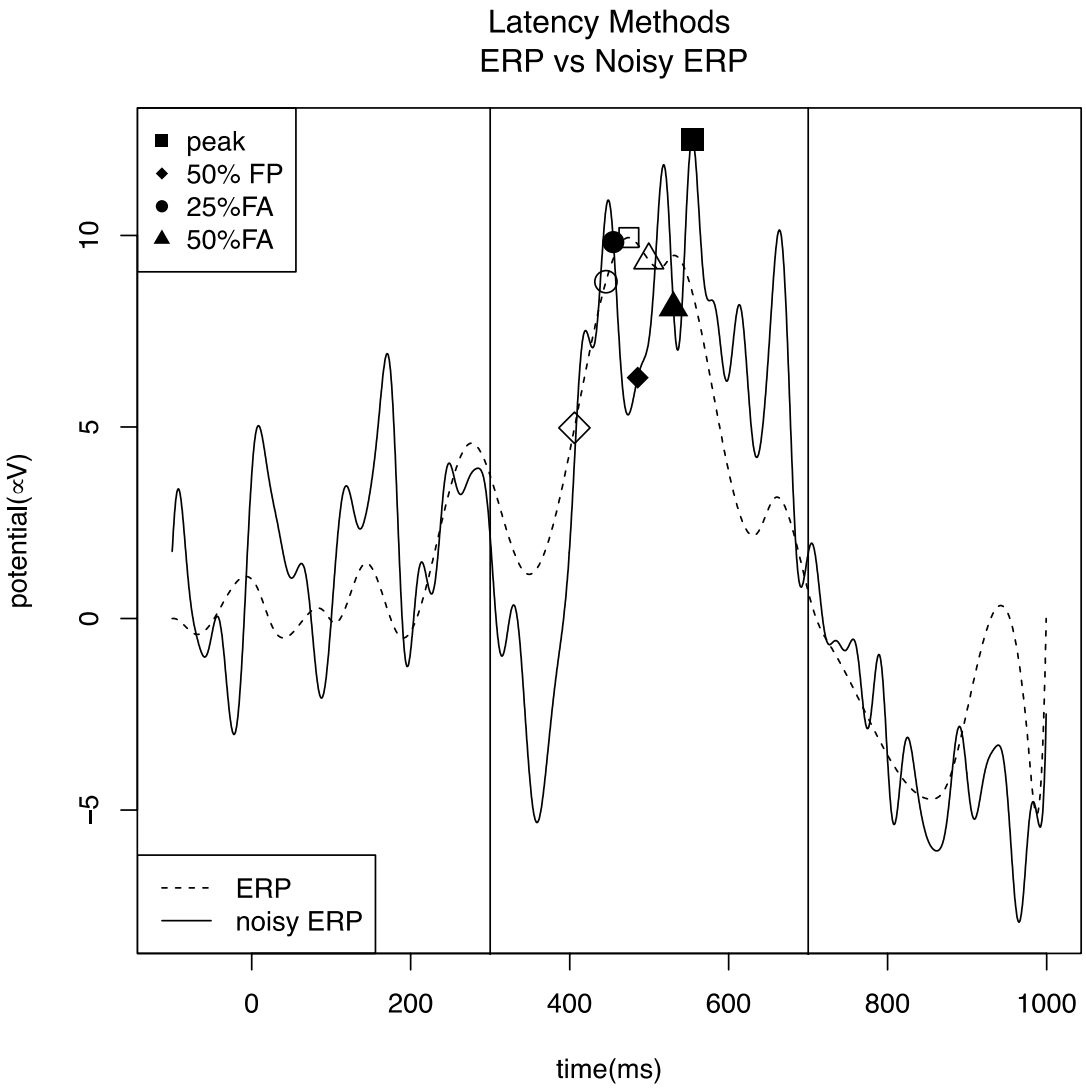

Figure 1: Example of how noise can affect fractional area and peak methods. The window for the analysis is placed at 300-700ms after stimulus presentation. The noisy ERP (solid line) is produced by generating 50 trials composed of the ERP signal (dashed line) with random noise and then averaging point wise, across trials. One can easily notice the latencies of all measures change, and in different directions. Note also that although *25%* F.A. does not move much, in this example, for other noise additions it would.

2) Choice of measure: latency contrasts are also extremely sensitive to the choice of measure used. In particular, the alternatives: *peak* latency, percent area, can produce quite different results. Furthermore, parameter settings associated with a measure can change the outcome of a latency contrast, e.g. using *25%* area latency versus *50%*. Also *fractional area* requires further parameter setting associated with the method for calculating the area under a curve, i.e. rectified area, numeric integration, area for positive regions and area for negative regions. These decisions are often hard to justify with objective, a priori, case independent criteria.

3) Window placement: *fractional area* based latency contrasts are acutely sensitive to choice of windows. The latency obtained is not only relative to the window, i.e. shorter windows can result in earlier latencies (Luck, 2005 p.p. 240), but also the shape of the signal inside the window and the existence of other components can change the results drastically. This issue can also arise in the *peak* latency methods but in a different manner. Firstly, if the *peak* is located at the end/border of the window then there is uncertainty over the validity of the measurement and secondly, when the fractional *peak* is used, researchers need to decide what latency should be returned if the fraction chosen is not met inside the window (Luck, 2005, p.237-242). This is likely to cause researchers to search for the best window placements in which the hypothesized latency difference is present, in order to facilitate the method employed. Such “fishing” amounts to performing multiple comparisons, which are not typically corrected for, e.g. through Bonferroni correction. Indeed, it is usually unclear how many comparisons have been made when an optimal window is searched for by eye, especially how many independent comparisons have been performed, making an attempt to correct for multiple comparisons effectively impossible.

All of these three problems are, we argue, at least to some extent, ameliorated by using a new method, which rather than comparing the latency of a single point in the ERPs of each condition, compares the temporal offset between ERPs across an entire region. This involves using Dynamic Time Warping (*DTW*) (Meyers et al., 1980; Keog & Pazzani, 2001; Muller, 2007; Senin, 2008) to (literally) warp the ERP from one condition into the ERP of the other, where the algorithm is seeking to make the warped ERP as similar to the other (the reference) ERP as possible. This process yields a warping path that indicates how points in the two time series are associated under this warping. For example, it might show that the time points in ERP1 are generally mapped to later time points in ERP2. This would indicate that ERP1 has a shorter latency than ERP2.

The fact that latency is measured across a region, which we argue should typically be selected to be broad (Kilner, 2013), means that the *DTW* should be more robust against noise. That is, the effect of noise may be particularly large on any individual data point, however determined, but should "wash out" across many time points. Sensitivity to activations throughout a region has also been seen as an advantage of *fractional area* against *peak* methods, but at the same time *DTW* measures latency differences as the temporal relationship of the two time series and for that reason is less affected by the size of the window and the presence of overlapping components. These will be illustrated in the simulations we present later.

DTW allows changes in latency to be quantified across all the data points in a region rather than with respect to just one. This then also raises the possibility to quantify the trade-off in temporal offset between two ERPs across a region, where perhaps one is earliest in one sub-region and the other is earliest in another. A further advantage of *DTW* is that it may be used in an exploratory fashion. This could be a systematic and principled alternative to 'eye balling' the

time series, which is effectively the method currently employed to explore whether latency differences are present and where they might be in the time series. The relationship between two ERPs based on their warping path is discussed in the *DTW* section.

This paper, then, presents a statistical analysis for assessing latency contrasts using Dynamic Time Warping (*DTW*). We formulate the statistical inference using a Monte Carlo resampling permutation test (Manly, 1997). This has the advantage of being nonparametric, freeing us from normality assumptions. Using simulated noise, we investigate the sensitivity (statistical power) and specificity (type I error rate) of *DTW* to the classic methods. The tests are applied to two windows, a broad and a narrow so as to investigate how the method's discriminability changes with window width. The results are then subjected to a ROC analysis so as to examine how discriminability varies with the significance threshold. At the end, *fractional area* is compared to *DTW* in regards to detection of latency differences in consecutive windows. This investigation demonstrates the effectiveness of the method.

2.Methods

We compared the performance of Dynamic Time Warping (described in detail in the next two sections) in detecting latency differences against four methods commonly used in the literature: 25% and 50% *fractional area*, *peak* latency, and 50% fractional *peak*. We assessed the methods on a number of artificially generated EEG datasets, each containing a predetermined amount of noise. This is described in the 'General Simulation Protocol' section. We performed 100 permutation tests (obtaining 100 p-values) for *each* noise level. The procedure was applied to EEG datasets that were time-shifted by 50ms (to calculate hit rates) and was repeated on EEG datasets that contained no latency difference (to calculate false positive rates). This is explained in the 'Statistical analysis' section.

2.1.Dynamic Time Warping

DTW made its appearance in the 60s and since then, it has gained popularity in the analysis of time series data. It has been widely used in automatic speech and handwriting recognition, but it has also been applied in other areas, such as: bioinformatics, computer vision, music and signal processing (Meyers et al., 1980; Senin, 2008). Although not very popular yet in the analysis of EEG data, there are a number of works in this area, which mainly use it as a method to align single trials (Wang, 2001; Casarotto, 2005; Liang & Bougrain, 2008), before averaging to produce ERPs. However, it has not previously been used to perform latency contrasts.

Dynamic Time warping is a technique for comparing time series data, by minimizing distortions in time. It, for example, allows the detection of similar components that differ in phase and/or distortions through time, and produces three outputs. The first of these is a distance measure (*DTW (X, Y)*) between the two time series (*X, Y*) that is not sensitive to local time stretches and compressions, the second is the warping results ($X_{wx}(k), Y_{wy}(k)$), the time series' deformed into each other, and the third is the warping path, which contains the information on how to manipulate the time series in order to align them.

The main idea behind *DTW* is relatively simple (Senin, 2008). Given two discrete time series $X = x_1, x_2, \dots, x_i, \dots, x_N$ and $Y = y_1, y_2, \dots, y_j, \dots, y_M$ of size N and M respectively (that have been sampled at equidistant points in time), the first step is to create a cost matrix (Figure 2). The cost/distance matrix contains all pairwise distances $d(i,j) = d(x_i, y_j)$. The most popular distance measure that is used is the Euclidean distance, but other approaches can also be used, i.e. Manhattan, Mahalanobis e.t.c.

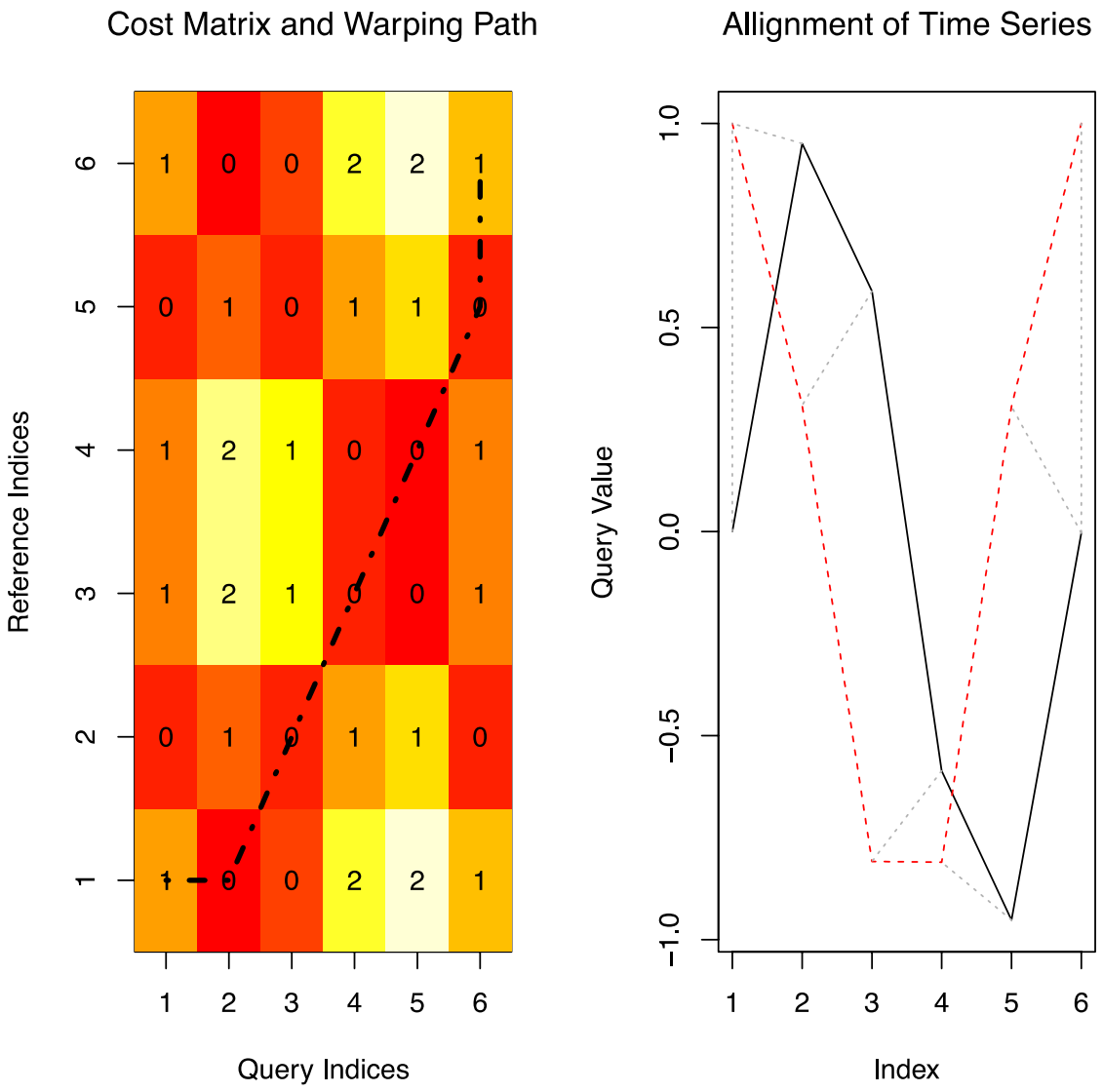

Figure 2: Example of *DTW*. On the left is the generated local cost matrix, and the selected warping path. Each cell shows the distance between a pair of elements, one from each time series (shown on the right), while the path consists of the matched indices for the optimal path. In this example, *wp={(1,1), (2,1), (3,2), (4,3), (5,4), (6,5), (6,6)}*. On the right, the signals used and the matched indices are depicted.

Then a warping path, $WP = wp_1, wp_2, \ldots, wp_c, \ldots, wp_K$ of size K, $\max(M, N) \leq K < M + N + 1$ with $wp_c = (i, j)$, $1 \leq i \leq N, 1 \leq j \leq M$, the mapping between the i-th element of X and the j-th element of Y, is constructed (Keogh & Pazzani, 2001). In simpler terms, each

element of the warping path is a pair indicating the matched indices between the two time series. This path traverses the low cost areas of the matrix that satisfy certain conditions. In particular, an optimal warping path between *X* and *Y* is one with minimal total cost among all possible warping paths. Since traversing all the possible paths can be computationally very expensive, *DTW* uses dynamic programming to find the lowest cost path. At each step, the cumulative distance is calculated as the distance in the current cell of the cost matrix and the minimum of the cumulative distances of the adjacent cells, starting from point $(M, N) = wp_K$ until $(1,1) = wp_1$ (traversing the cost matrix backwards).

Which (adjacent) cells are considered on each step, is determined by the step pattern of the algorithm, which may also weight certain cells differently to others. The cell with the minimum contribution to the path's cumulative distance is added to the warping path (Figure 2). The default step pattern (symmetric2) of the *DTW* package (which is used in this paper), *computes a global alignment, with no windowing, a symmetric local continuity constraint, and the Euclidean local distance. In particular, the symmetric continuity constraint implies that arbitrary time compressions and expansions are allowed, and that all elements must be matched* (Georgino, 2009). So, at each step of the algorithm, one of the three lower left immediate adjacent cells (Figure 3) from the cost matrix is selected, with the overall distance being calculated as:

$$dist_{cum(i,j)} = d(x_i, y_j) + \min\{2 * dist_{cum(i-1,j-1)}, dist_{cum(i-1,j)}, dist_{cum(i,j-1)}\}$$ [1]

If there is no difference between the time series, then the warping path coincides with the main diagonal of the cost matrix.

[1] Traversing the cost matrix backward, each cell of the cost matrix is selected based on the following pattern: $wp_c = (i, j)$, where $wp_{c-1} = (i - r, j - s)$ and $(r, s) = argmin[(y + z) * dist_{cum(i-y,j-z)}], (y, z) \in \{(1,1), (1,0), (0,1)\}$, where i is the index of the i[th] point from one time series and j the index of j[th] point of the second time series.

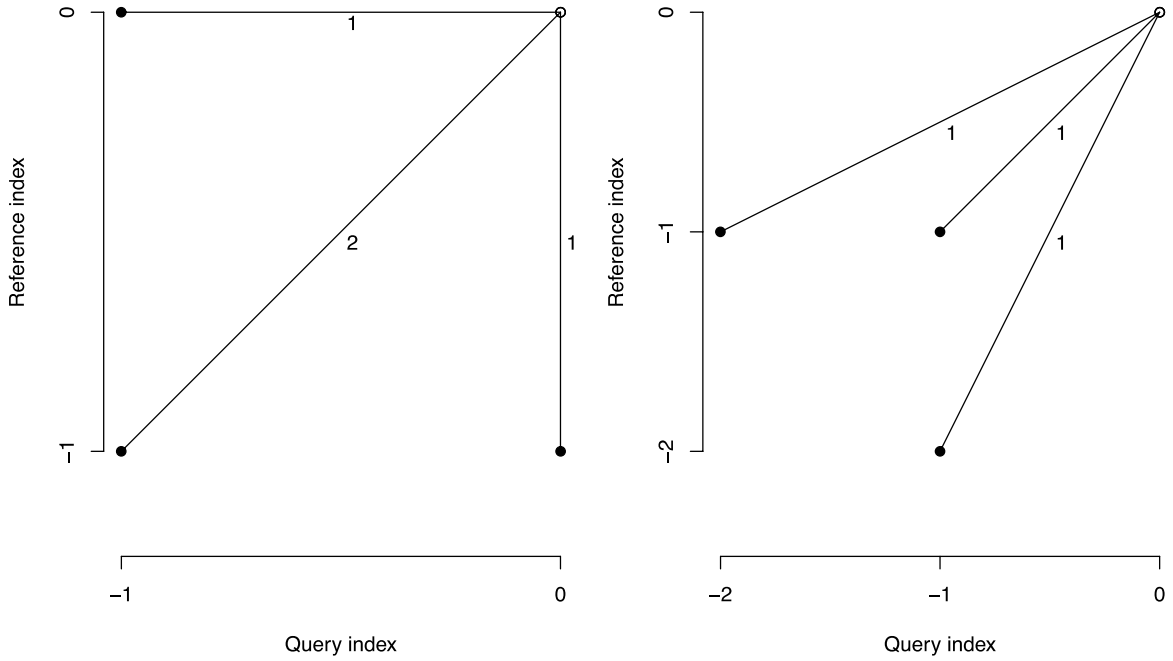

Figure 3: The two step patterns that were used in this analysis. On the left, the default step pattern, known as symmetric2 and on the right, the typeIIa step pattern (the graphic representation of the step patterns is provided by the dtw package in R). On the default, the diagonal is weighted by 2, so movements on the axes are more likely to be chosen, while on the typeIIa, for each step parallel to the axes, the path also moves one step diagonally.

The distance measure that is returned (which is a similarity measure) consists of the summation of all the distances between the matching pairs of the warping path (Senin, 2008) and can be normalized by the number of points in the path.

$$DTW_{distance}(X,Y) = \frac{1}{K}\sum_{c=1}^{K} d(wp_c)$$

The warping path *WP* is described by a number of properties. These are:

- Boundary: This condition enforces that the starting and ending points of each sequence are aligned to each other. This means that the first point of X is aligned with the first point of Y and the same for the ending points: $wp_1 = (1,1),\ wp_K = (M, N)$.
- Monotonicity: The path cannot go backwards. $If\ wp_c = (i, j), wp_{c+1} = (i', j'), then\ i \leq i', j \leq j'$. This way needless loops are also avoided (Giorgino, 2009).
- Continuity constraints: The continuity constraints define the number of time compressions and expansions allowed (i.e. number of time points that can be repeated and skipped) and the size of the neighborhood of each cell. The default continuity constraint of *DTW* allows *arbitrary* time compressions and expansions. But there are numerous constraints that are applied. For example, only considering neighborhood points: $If\ wp_c = (i, j), wp_{c+1} = (i', j'), then\ \ i - i' \leq 1\ and\ j - j' \leq 1$ (Berndt, Clifford, 1994).

Besides the above properties, there are other conditions and constraints that can be applied in order to produce warpings that are consistent with the particular domain of application of *DTW*. For example, if the two time series represent Event Related Potentials then limiting extreme mappings or mappings of one time point with several from the other time series might be appropriate. There are a number of global path constraints\windowing parameterizations, where certain regions of the cost matrix such as the upper left and bottom right corners are not considered. Two of the most popular implementations are, the Itakura parallelogram and the Sakoe - Chiba band (Figure 4). Another way to constrain the warping path is to add extra weights to the distances in the cost matrix (Figure 3), which results in penalizing or favoring certain steps, e.g. creating a bias towards the diagonal. Finally, constraints can be applied to the size of

the step, which can include limiting the size of the change allowed from one point to the next or the number of consecutive steps in one direction (Meyers, 1980; Müller, 2007).

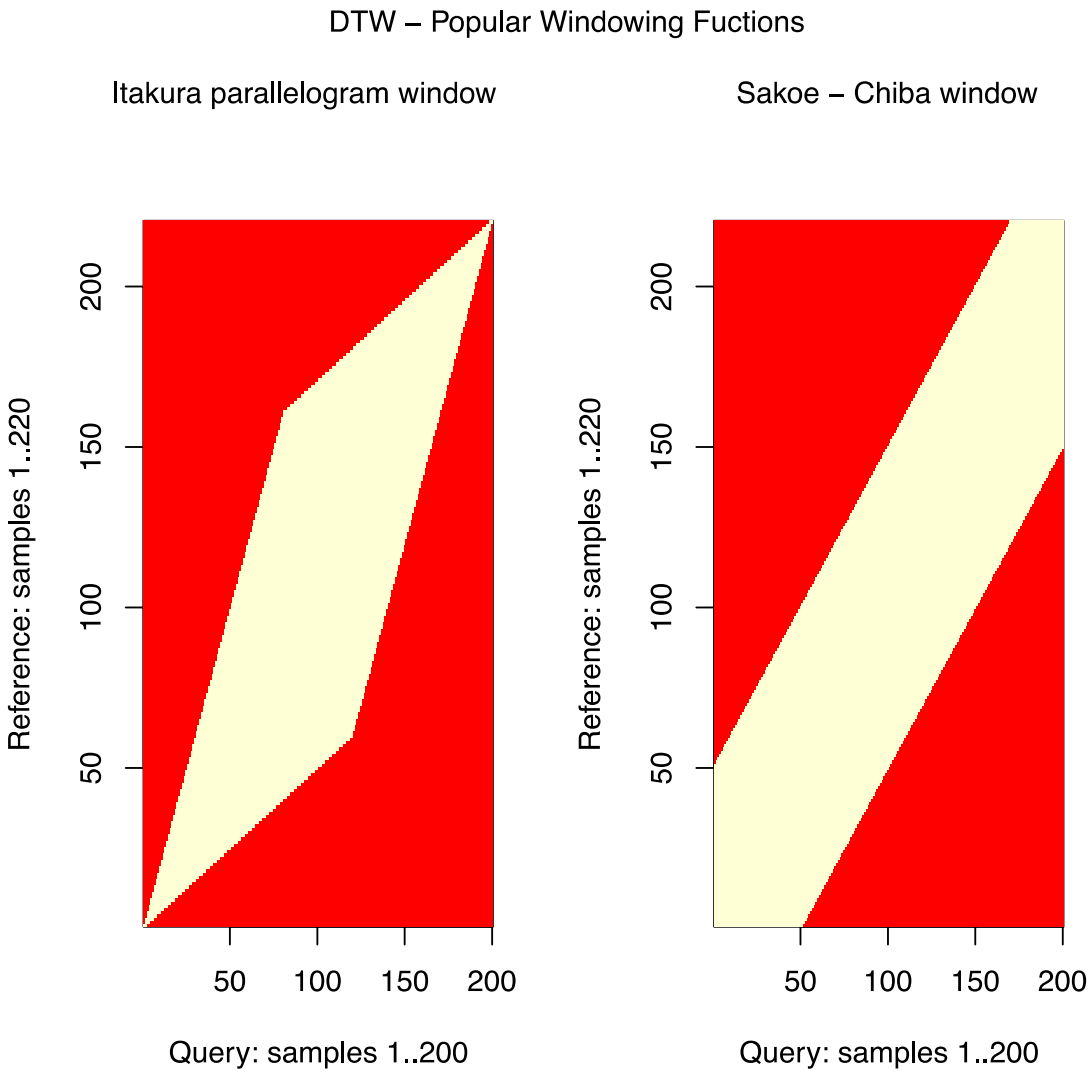

Figure 4: Two popular global region constraints: Ikatura parallelogram (left) and Sakoe-Chiba band (right).

2.2.Dynamic Time Warping as a Method for Measuring Latency Contrasts

DTW's power lays in its ability to allow comparison of time series data by eliminating differences in the time axis. The shifts and compressions in the time axis are obtained by warping the sequences according to the warping path. This results in the warping path between two EEG time series being inseparably linked with the latency differences that are present. As mentioned earlier, if two sequences are identical then the warping path will coincide with the main diagonal. Based on the process that produces the warping path, we suggest that its distance from the diagonal provides a reliable measure of latency differences between two time series. A positive distance, which results from the warping path being below the diagonal, indicates that the reference time series, used for alignment, precedes the query time series, while a negative indicates that the reference time series follows the query. In other terms, i$f\ DTW_{area_diff} > 0$, then reference is earlier than query.

Some examples of different signals and the resulting warping paths can be seen in Figures 5,6,7. The measurement proposed and tested in this paper, is the area between the warping path and the main diagonal of the cost matrix.

$$DTW_{area\_diff} = A_{diagonal} - A_{WP}$$

where $\mathrm{X} = \mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{N}$ is the query ERP signal; $\mathrm{Y} = \mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_\mathrm{M}$ is the reference ERP signal; $\mathrm{A}_{\mathrm{diagonal}}$ is the area under the diagonal line between points $\mathrm{wp}_1 = (1,1)$ and $\mathrm{wp}_\mathrm{K} = (\mathrm{N}, \mathrm{M})$; the starting and ending points of the warping path; and A_{wp} is the area under the warping path constructed by the *DTW* for the two time sequences. As the size of the diagonal and the warping path are not constant, but depend on the size of the time series to be aligned (for ERP analysis N=M, since the same window, time range, is used for all conditions under examination), the above measurement could be normalized by dividing by the area under the diagonal, becoming:

$$DTW_{diff} = \frac{A_{diagonal} - A_{WP}}{A_{diagonal}}$$

As mentioned earlier, there are a number of parameterizations that can be applied to the *DTW*, which influence the warping path generated. As an illustration, in this paper, we consider the typeIIa step pattern (Figure 3) where the cost function for selecting the cells of the warping path becomes:

$dist_{cum(i,j)} = d(x_i, y_j) + \min\{dist_{cum(i-1,j-1)}, dist_{cum(i-1,j-2)}, dist_{cum(i-2,j-1)}\}$.

This particular step pattern was chosen because it constrains the deviation from the diagonal, forcing the path for each step parallel to the axis to also move one step diagonally. This step pattern produces more realistic warping between two ERPs, since it limits extreme mappings and multiple matching's, i.e. repeated movements on one axis, while the other remains unchanged.

DTW, as mentioned earlier, could also be used in an exploratory fashion allowing researchers to visualize the relationship between two ERPs in a manner that reveals latency differences as deviations above and below the main diagonal. In figures 5,6,7, we present three simple examples. First we show the warping path between an ERP signal and the same signal offset by 100 time points. The warping path is all below the diagonal, with a constant offset from the diagonal of 100 time points, indicating the latency difference between the two time series. In the second example, the same scenario is presented, only the second ERP is 80% of the amplitude of the first. This, firstly, shows how *DTW* deals with amplitude differences, but also how latency differences can be visualized using the warping path. It allows one to detect that between 1200:1300 time points, the relationship of the two time series could be perceived as the opposite of what was introduced (an example where windowing could generate false conclusions). In the last figure, the first half of the dashed ERP was produced by offsetting the original ERP (solid line) 100 time points to the left and the second half by offsetting the original

ERP 100 time points to the right. The warping path clearly indicates this relationship, with the first half of the path being above the diagonal and the second half below. These are relatively simple examples to show how *DTW* could be used in an exploratory fashion to identify regions of latency difference between two ERP signals.

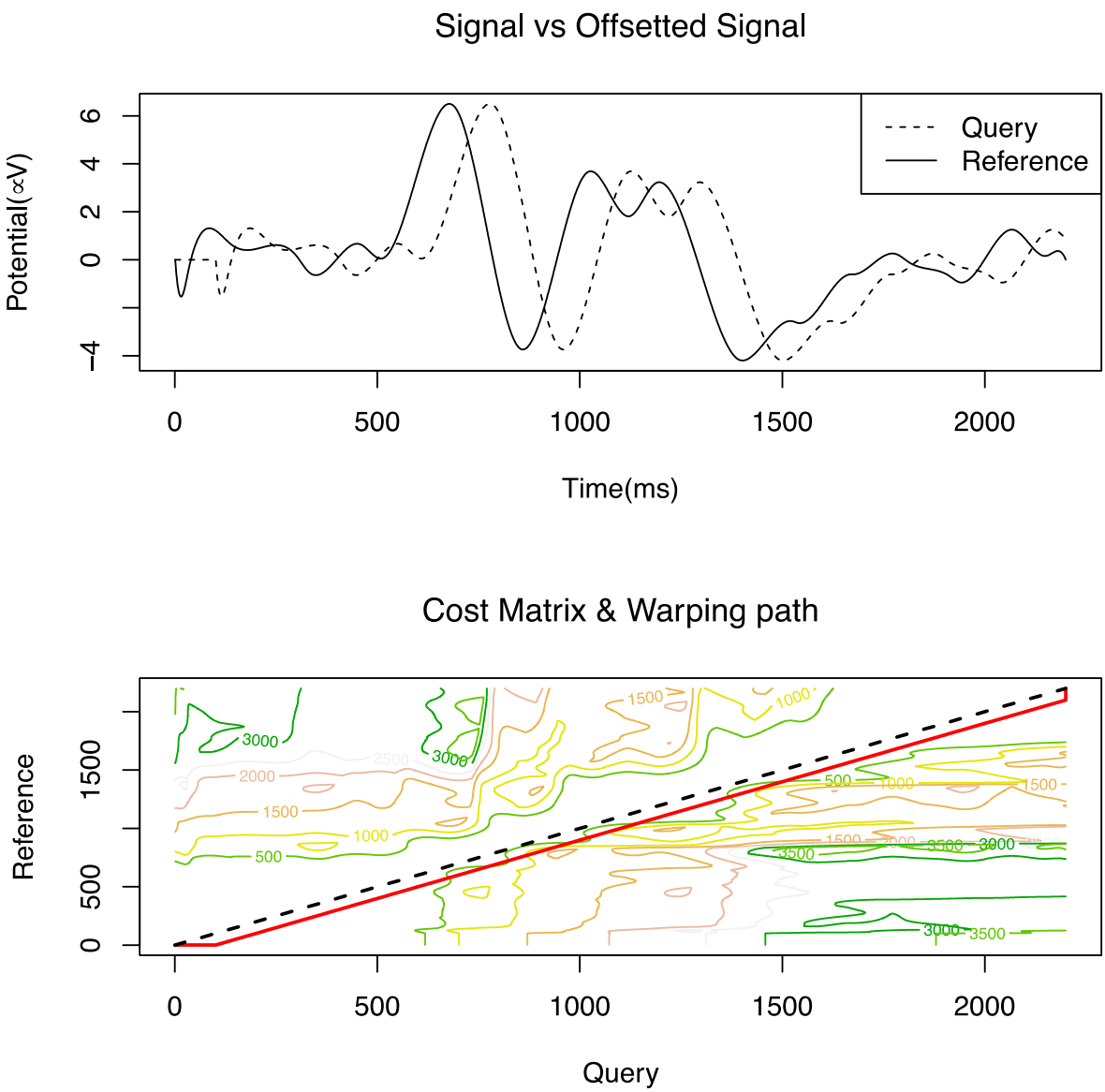

Figure 5: Example of how the warping path can be used to explore latency differences between two time series. Upper plot shows the two signals used. The second signal was produced by offsetting the first by 100 time points. The second plot shows the warping path produced between the two signals. The 100 time points offset can be clearly seen as a deviation from the main diagonal (black dotted line). The whole warping path (red line) is under the diagonal indicating that the second signal (query) is later that the first (reference).

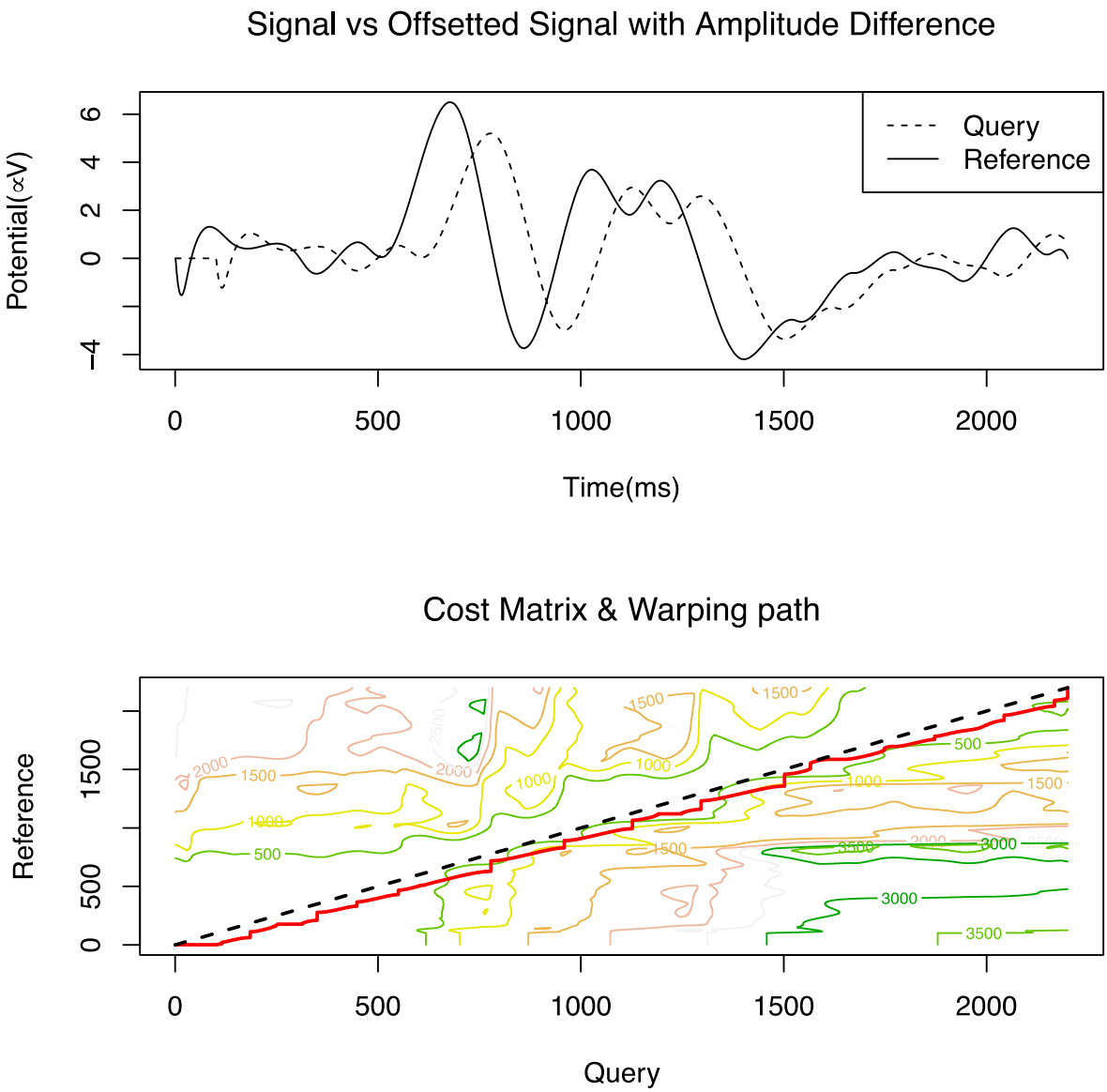

Figure 6: The upper plot shows the two signals that were warped. The second signal (solid line) was produced with 80% amplitude of the first and 100 time points offset. The lower plot shows the warping path between the two signals. It shows how DTW handles amplitude differences. It is interesting to observe that if the window was placed at 1200:1300 time points, the opposite latency could be detected. This demonstrates the dangers of selecting small windows and how window placement could lead to dubious conclusions.

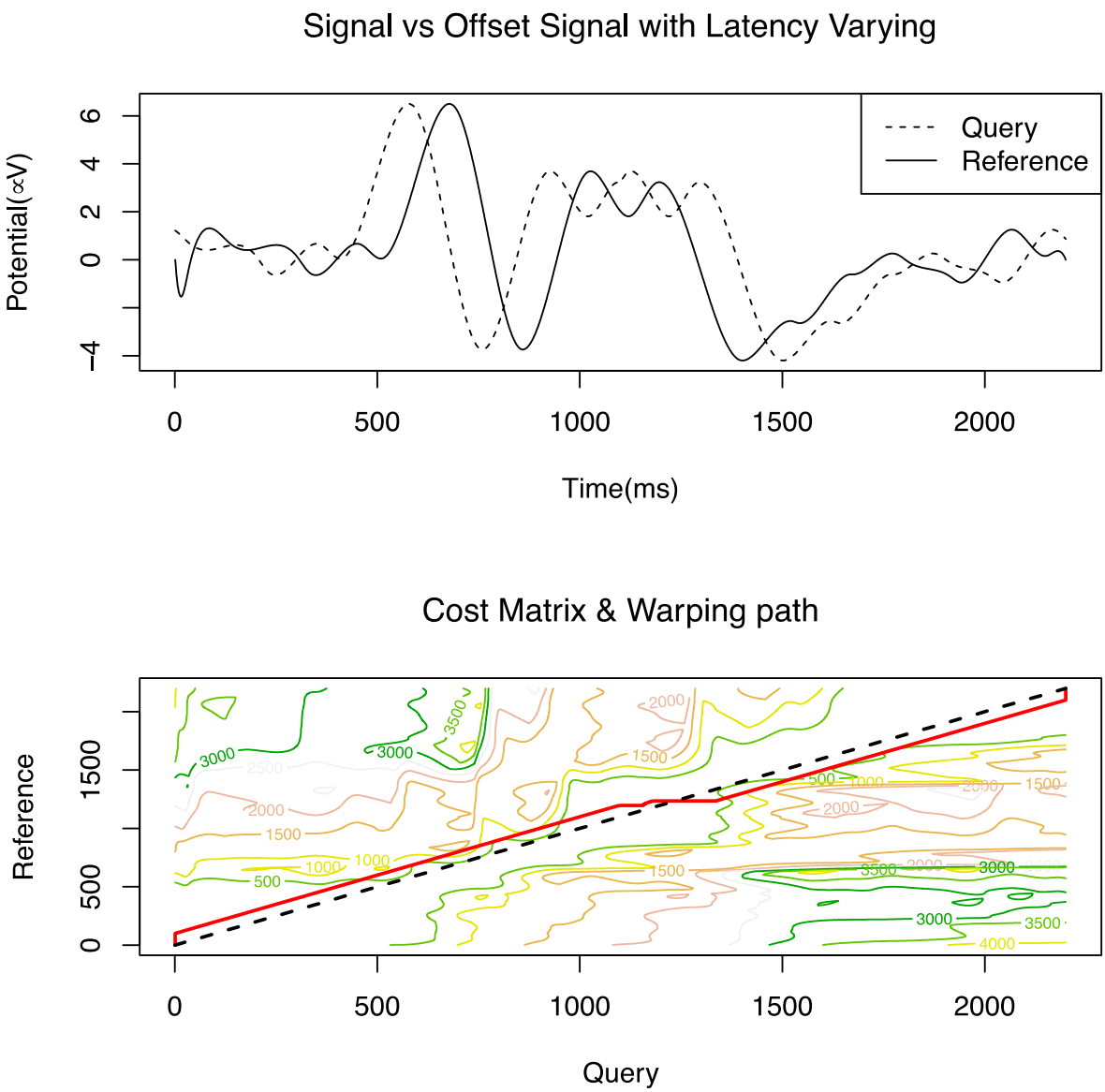

Figure 7: The first plot show the two signals that were warped. The second signal was produced with it's first half being offset 100 time points on the left and the second half (1100 tp onwards) on the right. The warping path reveals this relationship, with the first half being above the diagonal and the second half below.

2.3.Application of methods for determining ERP Latency Differences

The methods evaluated in this paper are *DTW*, *DTW* typeIIa, *peak*, *fractional area* and Fractional *peak*. *Peak* latency was measured as the time point where each condition reached its maximum voltage. *Fractional area* was calculated for two different fractions, 25%, which in the literature is most commonly used to identify onset latency differences and 50% (Handy, 2005; Luck, 2005; Kiesel, 2008). In order to calculate the area at each time point, the absolute value of the voltage was taken and then the time point that separated the overall area to the prespecified fraction was determined. Fractional *peak* was calculated by determining the time point where the ERP reached 50% of the maximum voltage. If such a point was not present in the window, then the last point was returned. All measurements were calculated separately for each condition and then subtracted in order to determine latency direction (i.e. negative difference indicating subtrahend earlier than minuend). All methods were implemented in R (Ihaka & Gentleman, 1996), and specifically for *DTW*, the *DTW* package was used (Giorgino, 2009; Tormene et al., 2009).

The methods were applied to two different windows for each channel. For the Fz channel, where the analysis was focused on the P3a component, the two windows where: 175-325ms from stimulus presentation, which from now on we will call the narrow window and 100-400ms, the broad window. The first window was placed after inspecting the data, which is likely to inflate the type I error rate, and trying to include as much of the effect as possible for both conditions but at the same time not so much of the subsequent negativity. The second window, broad, was centered at 250ms with 150ms width on each side, as this is the expected latency for the P3a based on the literature (Luck & Kappenman, 2012), without taking into consideration the shape of the specific signals. For Pz, the placement was performed using the same process; the narrow window was 350-650ms (from stimulus presentation), while the broad window was 250-750ms

(centered at 500ms with a width of 500ms). The signals and the two types of windows used are shown in Figure 8. We performed the analysis in two separate windows in order to determine how the performance of each method is affected by the size and placement of windows, and whether the analysis of latency differences can be applied successfully in predefined windows based on the time that a component theoretically is expected to appear and its size.

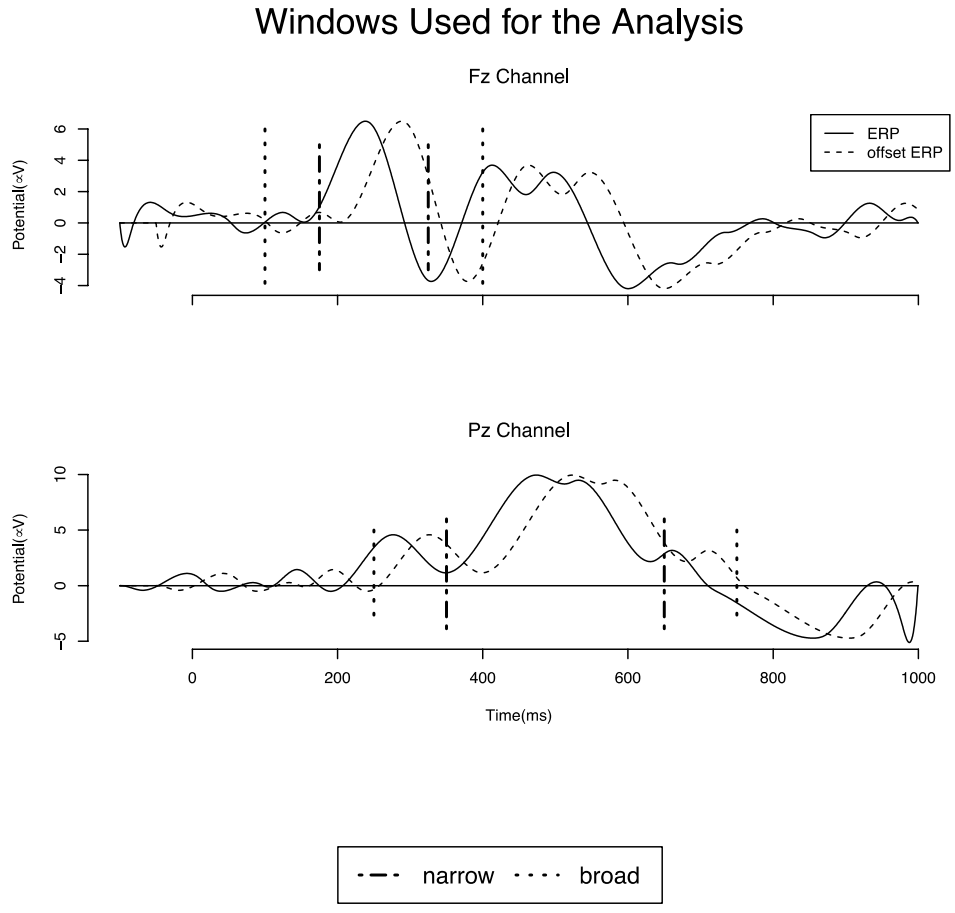

Figure 8: The signals that were used in the first case study (latency contrasts only present). Upper plot shows the signals and the two windows used for channel Fz, while the lower shows them for channel Pz.

2.4.General Simulation Protocol

The performance of the methods was evaluated on data produced using a simulation protocol where the latency difference effect was artificially introduced. In order for our analysis to be reliable we used signals, to represent the first condition, that were obtained from a real EEG experiment (Bowman et al, 2013). In particular, we generated ERP signals from a deception detection experiment, where EEG data were recorded from 15 participants. The participants were presented RSVP streams, which contained their name (Probe), an assumed name (Fake) or one of two preselected names. The methods were examined on two different channels Fz and Pz and the analysis was focused on the P3a and P3b components respectively, as in the original analysis. The steps followed in each channel were the same and consisted of firstly generating the ERP signal following all the preprocessing steps described in Bowman et al (2013). The signal $ERP_{s1Channel}$ (ERP_{s1Fz}, ERP_{s1Pz}, for channels Fz and Pz respectively), which was the ERP for the probe condition (generated from the grand average across participants), was then smoothed by interpolating between a set of *peak*s. This signal was used as the first condition. In order to test the methods' robustness to noise, we generated new EEG data of 50 trials, each trial (of 2200 time points) consisting of the obtained signal, point wise added to the noise, which was generated at the spectrum of human EEG[2]. The amplitude of the noise was increased as a factor of 5, from 5 to 35, generating SNR in the range of [0.7:0.012].

$ERP_{cond_1}(t) = \frac{1}{50}\sum_{i=1}^{50} EEG_{cond1}(t)$, $i \in \{1,2,\dots,50\}$ is the number of trials and at each trial: $EEG_{cond1}(t) = ERP_{s1Channel}(t) + noise(t)$, where t is the time and

$noise(t) = k * EEG_{noise}(t),\ k \in \{\chi \in 5{:}35 \mid \frac{\chi}{5} \in \mathbb{N}\}$

[2] The code used to generate noise can be found at http://www.cs.bris.ac.uk/~rafal/phasereset/.

We applied the methods and tested their performance for each step of k. The SNR provided is an approximation. The main term of it was calculated at the trial level as the mean square amplitude (ms) of the points making up the signal time series divided by the mean square amplitude of the points making up the noise time series. The main term was then turned into an average of the number of contributing trials and multiplied by the square root of that same number of trials ($\sqrt{50}$) in order to estimate the SNR at the ERP level (Luck, 2005; Van Drongelen, 2006).

$$SNR_{ERP} = \sqrt{50} * \frac{1}{50}\sum_{i=1}^{50}\frac{ms(ERP_{sChannel})}{ms(noise(i))}$$

The second condition's ERP signal ($ERP_{s2Channel}$) was generated by introducing a latency difference to the first condition ($ERP_{s1Channel}$).

The performance of the methods was evaluated based on their power to detect the latency difference. We didn't measure the size of the effect but instead we focused on detecting the correct direction of the latency (early vs late). Then we measured the false positive rate by applying the methods when the conditions do not differ in latency and then using these two measurements we performed ROC analysis.

More specifically, for the methods' power to detect an effect, the ERP of the second condition was produced by generating 50 trials each of which consisted of the first condition's signal offset by 100 time points (50ms), and then adding noise.

$$EEG_{cond2}(t) = ERP_{s1Channel}(t + 100) + noise(t)$$

The offset was 50ms, and it was based on the latency difference measured between the Fake and Probe conditions from the experimental data (Bowman et al., 2013). The analysis is described in more details below.

2.5.Statistical analysis

The performance of the methods was evaluated using Monte Carlo resampling permutation tests (also called randomization procedures), which consisted of swapping the labels of the conditions in the set of trials (Blair, 1993; Manly, 1997). We tested for a true null hypothesis, meaning that there is no latency difference between the two conditions. We examined whether the methods can identify the underlying effect, and thus reject the null hypothesis ($p < \alpha$ (=0.05)) under different SNRs and window sizes. We obtained 100 p-values from the same number of randomization tests. Each test consisted of generating a new EEG data set (addition of new set of random noise) and performing 1000 randomizations (swapping single trials and generating randomized ERPs). The p-values were calculated as the proportion of the generated latency differences that were greater than the observed ones. These p-values represent the evaluation of the methods' results under the randomization distribution.

When comparing the performance of methods it is important to directly compare their statistical power, which can be measured as the probability to obtain statistically significant results (Kiesel et al., 2008) when a difference is present, or in other words the true positive rate. In order to provide a measure for statistical power, we present the percentage of p-values from the 100 permutation tests that were below the critical alpha level for each SNR and window size.

Then another 100 permutation tests, each consisting of 1000 permutations, were performed where there was no latency difference present, for each SNR and broad/narrow window in order to measure the false positive rate.

2.6.Receiver Operating Characteristics (ROC analysis)

ROC curves are widely used for visualizing and evaluating the accuracy of predictors. They are two-dimensional graphs, where the x-axis represents the false positive rate of a classifier/predictor and the y-axis represents the true positive rate. The (bottom left to top right)

diagonal line represents the result of randomly assigning a class to an instance (i.e. pure guessing). For a quantitative measure, the area under the curve (AUC) statistic is used. The value of the AUC ranges between 0 and 1 ($0 \leq AUC \leq 1$), with 0.5 being the performance of a random classifier. When different classes are not present, instead of a criterion a varied threshold is used, which in this analysis would be the alpha level (Fawcett, 2006; Flach, 2010).

ROC curves were used to assess the rate at which each method produces p-values below the critical alpha level when applied to noisy signals containing latency contrasts (i.e. true positives). They allow the examination of how that rate varies with the rate at which lower than alpha p-values were obtained for signals when no latency difference was present (i.e. false positives). The resulting AUC is a measurement of the accuracy with which a method can correctly identify the presence of latency differences.

In order to generate the AUC from the collected data a gradually increasing threshold was used. The threshold was varied from 0 to 1 with a step of 0.001. Each point in the AUC represents the proportion of p-values below the threshold. A step of 0.001 was selected for high accuracy (the closer the points of the ROC curve, the more accurate the measure of the AUC). Since the p-values were obtained from 1000 permutations, accuracy to the third decimal place was available.

3. Results

3.1.Statistical Power

The results from the experiments are presented in tables 1-6. For channel Fz, in regards to the methods' power to detect the latency difference when applied at the narrow window *DTW*, *DTW*_typeIIa and *50% fractional area* have similar performance for most SNRs. *DTW* and *DTW*_typeIIa start to have higher power than *50% fractional area* for SNRs below 0.020 (produced by a k>30). All three methods clearly perform better than *25% fractional area*, *peak*

and fractional *peak*. When they are applied to the broad window, *DTW*_typeIIa's power improves while for all other methods it drops, which results in *DTW*_typeIIa having the highest power in all SNRs with *DTW* following. *Fractional area*'s power (for both *50%* and *25%*) drops greatly when applied to the broad window. *Peak* and fractional *peak* have higher performance for the narrow window but in general have similar behavior across SNRs (Table 1; Figure 9).

Table 1

Power to detect latency difference for Channel Fz when conditions differ only in latency, for different SNRs. (Probability that the method will return p-values below the alpha level)

SNR	Methods' Power at Channel Fz											
	DTW		DTW typeIIa		25% F.A.		50% F.A.		Peak		50% F.P.	
	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad
0.525	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.147	1.00	0.98	1.00	1.00	0.93	0.84	1.00	0.99	0.98	0.97	0.99	0.99
0.062	0.93	0.89	0.92	0.98	0.75	0.40	0.97	0.64	0.87	0.89	0.85	0.76
0.036	0.76	0.84	0.78	0.94	0.37	0.30	0.78	0.47	0.60	0.60	0.55	0.42
0.022	0.63	0.57	0.62	0.77	0.29	0.15	0.62	0.37	0.46	0.38	0.50	0.29
0.015	0.48	0.42	0.50	0.65	0.15	0.10	0.35	0.19	0.29	0.18	0.33	0.12
0.012	0.39	0.21	0.44	0.54	0.13	0.09	0.23	0.28	0.27	0.17	0.20	0.13
Mean	0.74	0.70	0.75	0.84	0.52	0.41	0.71	0.56	0.64	0.60	0.63	0.53

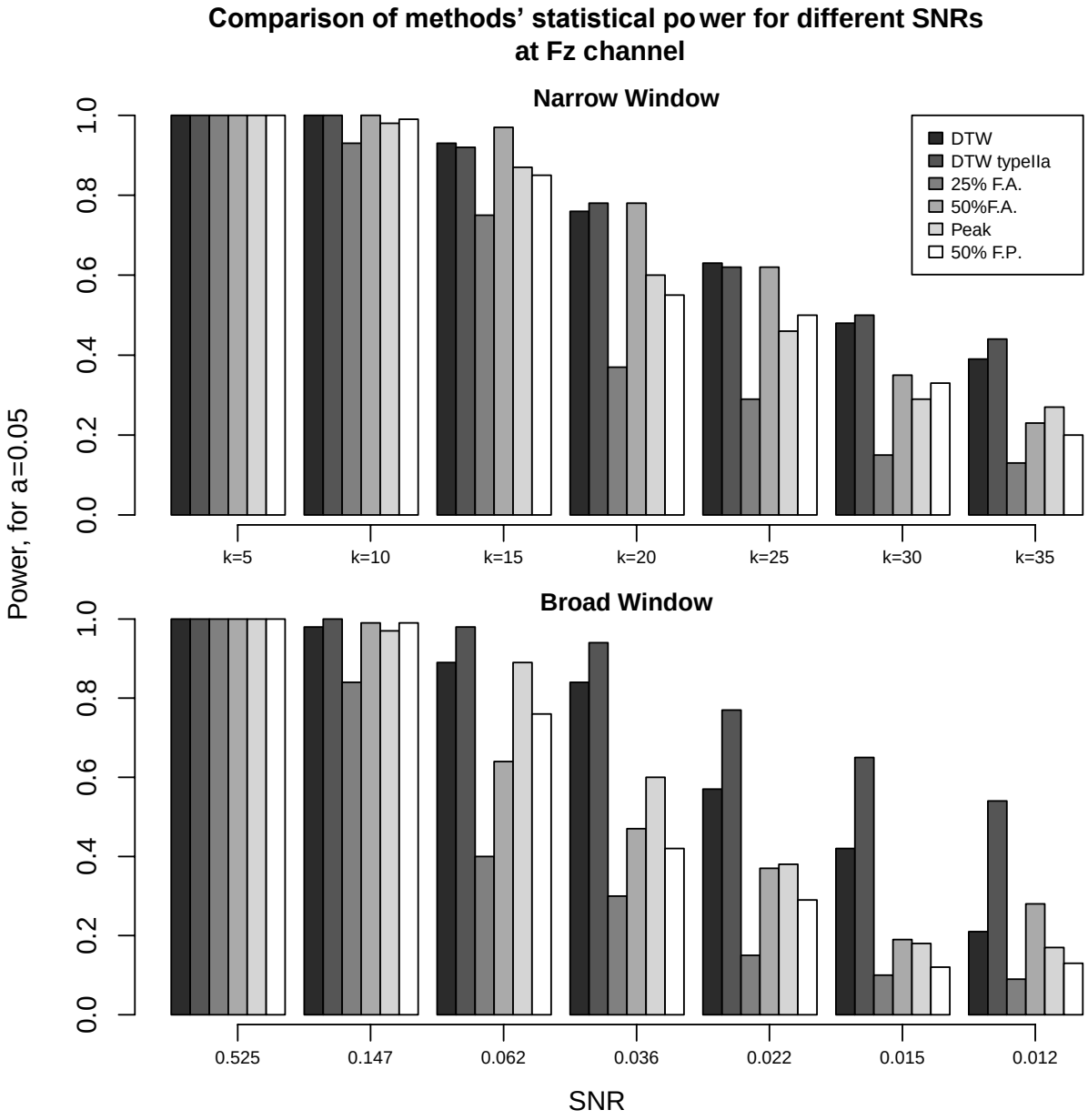

Figure 9: Comparison of the methods' statistical power at Fz channel. For the narrow window, *DTW* and *DTW* typeIIa generally do the best, with 50% fractional also performing well at high SNRs; *DTW* and *DTW* typeIIa show less sensitivity to noise. For the broad window, *DTW* typeIIa clearly outperforms all the methods and is the most resilient to noise. 25% fractional area has the lowest power independently of the window used. The different SNRs are presented in two ways, as a factor of k, but also as an approximation of the SNR at the ERP level. We present both values as, although we used a step of 5 for each noise level (as described in the general simulation protocol section) the approximated SNR varied at a different rate, e.g. at channel Fz, for k = 5 the SNR was 0.525, for k=10 the SNR was 0.147 and so on.

For channel Pz and a narrow window, *DTW* and *DTW*_typeIIa perform best at most SNRs (k=10 is the only exception were *25% fractional area* performs better), while *25%* and *50%* have similar power, with fractional *peak* following. The *peak* method has the worst performance with a power below *50%* for almost all SNRs. As with channel Fz, *DTW*_typeIIa's power improves for the broad window, while for the *fractional area* measurements the power drops significantly. *25% fractional area* fails to detect the correct latency difference for most SNRs with a power below 10% for SNRs lower than 0.05. The rest of the methods have similar power (*DTW*, *peak*, fractional *peak*) for both windows (Table 2; Figure 10).

3.2.False Positive Rate

The false positive analysis shows no inflation of type I errors for any of the methods under all SNRs, and whether window is narrow or broad. Although there are some cases where the false positive rate (FPR) is above 0.1 (i.e. at Fz channel *50%* fractional *peak* for SNR 0.5 has an FPR of 0.14) a closer inspection of the distributions does not reveal a clear bias and the inflation is not persistent (nor systematic) across different SNRs (Tables 3,4; Figures 11,12).

3.3.ROC analysis

For channel Fz and the narrow window, the ROC analysis shows that *25% fractional area* has the lowest AUC and is the most susceptible to noise. The rest of the methods have similar performance with *DTW* being the most resilient to noise. When applied to the broad window, *DTW* and *DTW*_typeIIa have greater AUC than for the broad window while the rest of the methods perform worse. Overall *DTW*_typeIIa has the best performance and is the most resilient to noise (Table 5, Figure 13).

At channel Pz, when the narrow window is used, *peak* has the worst performance, while *25% fractional area* is more affected by the increase of noise. *DTW*, *DTW*_typeIIa and *50%*

fractional area have similar AUCs for high SNRs but *fractional area* deteriorates at a larger rate with the addition of noise. When the broader window is used, *DTW*_typeIIa outperforms all methods for all SNRs and exhibits a great resilience to noise (Table 6, Figure 14).

3.4.Comparing method's sensitivity to window placement

Based on the presented analysis, *DTW*, especially when using the typeIIa step pattern, appears to have a greater power to detect latency differences and is more resilient to noise. From the rest of the methods, *50% fractional area* appeared to have the most competitive results. Since its performance has been demonstrated in other work as well (Kiesel et al., 2008), an extra comparison between the two methods was attempted. As *fractional area* is sensitive to the presence of overlapping components and thus the size and placement of the window, we wanted to test *DTW*'s behavior regarding these parameters. In order to quantify the methods' sensitivity, we performed the following analysis. A fixed point was selected as the starting point for the window and then different windows were sequentially created based on a step value. For example, the first experiment for channel Fz was conducted by placing the starting point at 50 ms (from stimulus presentation), using a step of 10 time points (5ms): the methods were applied to all the generated windows with an ending point from 60ms up to 1000ms (the end of the ERP), i.e. w_1 = 50-55ms, w_2 = 50-60ms, …w_{169} = 50-1000ms. The proportion of windows where a method failed to recognize the correct latency difference (i.e. that the early/first condition was earlier than the late/second condition) against the total number of windows was used as a measurement of sensitivity to window placement.

As the starting point of the window and the size of the step can play a significant factor, in order for the results to be more reliable, a set of different starting points (50ms, 100ms, 150ms, 300ms) and two steps 5 – 15ms, were used for the analysis. Although the same latency difference is present in the entire time period, for the Fz channel, windows starting at 50,100 and

150ms are of the most interest (as they represent realistic options for studying the P3a component), while for Pz, 150 and 300ms are the most appropriate.

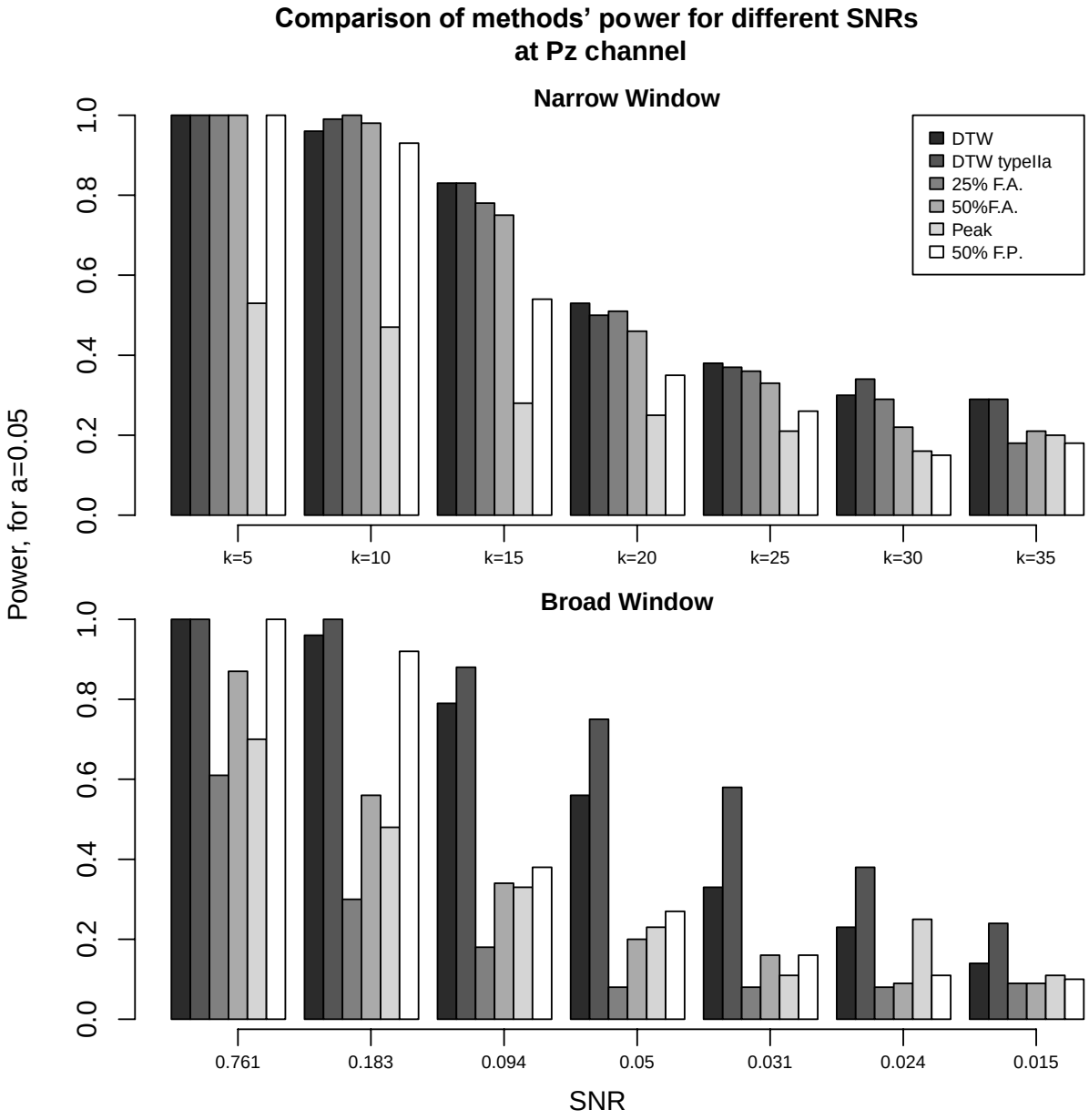

Figure 10: Comparison of the methods' statistical power at channel Pz. For the narrow window, *50% fractional area* has the best performance for high SNRs, while *DTW* and *DTW* typeIIa have consistently good power. When the broad window is used, *50% fractional area*'s power significantly reduces, while *DTW* typeIIa outperforms all methods for all SNRs.

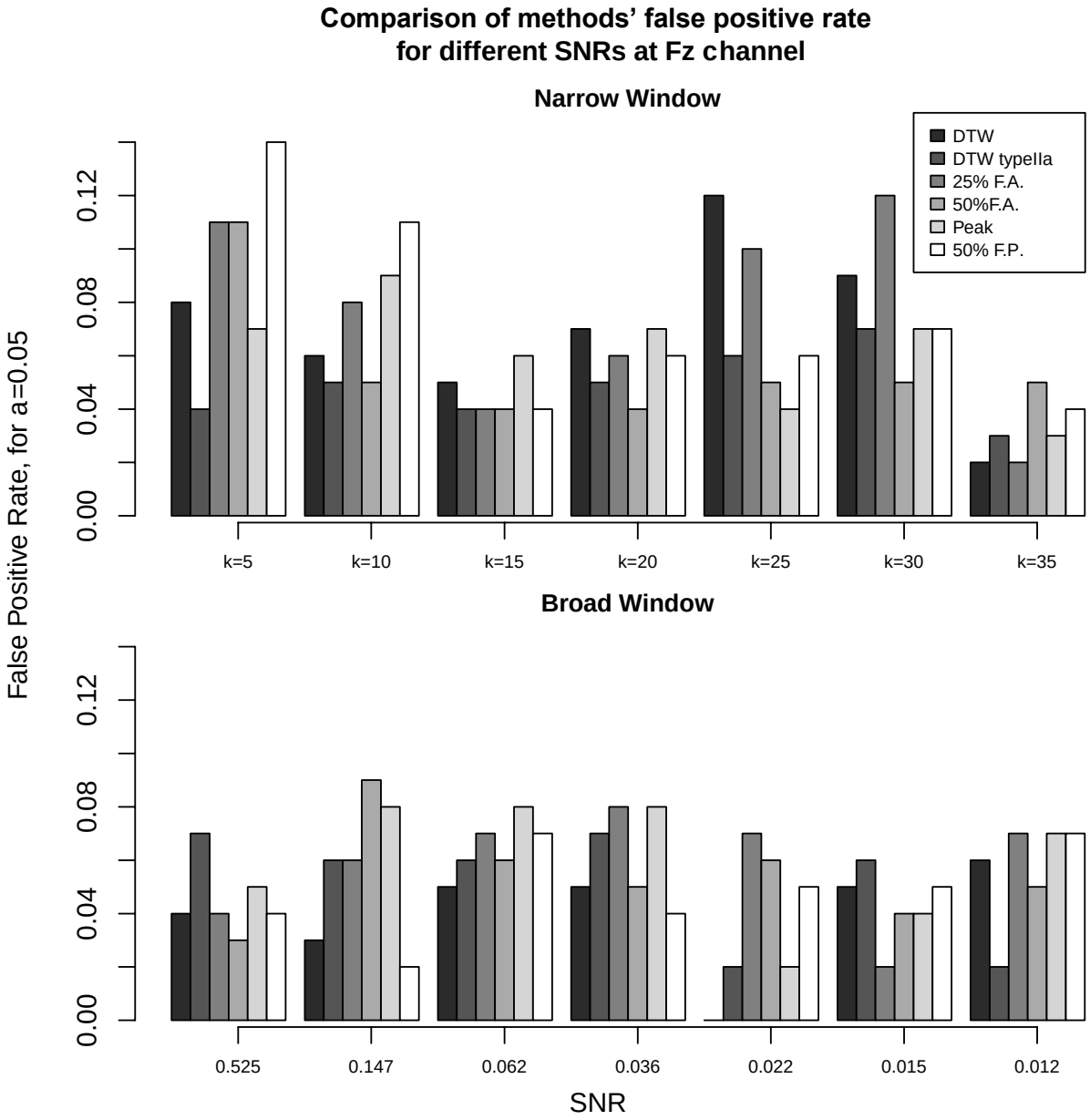

Figure 11: Comparison of the methods' false positive rate for different SNRs at channel Fz. The methods' were applied on conditions consisting of the same signal (no latency difference). The analysis does not show any systematic inflation of the false positive rate for any of the methods.

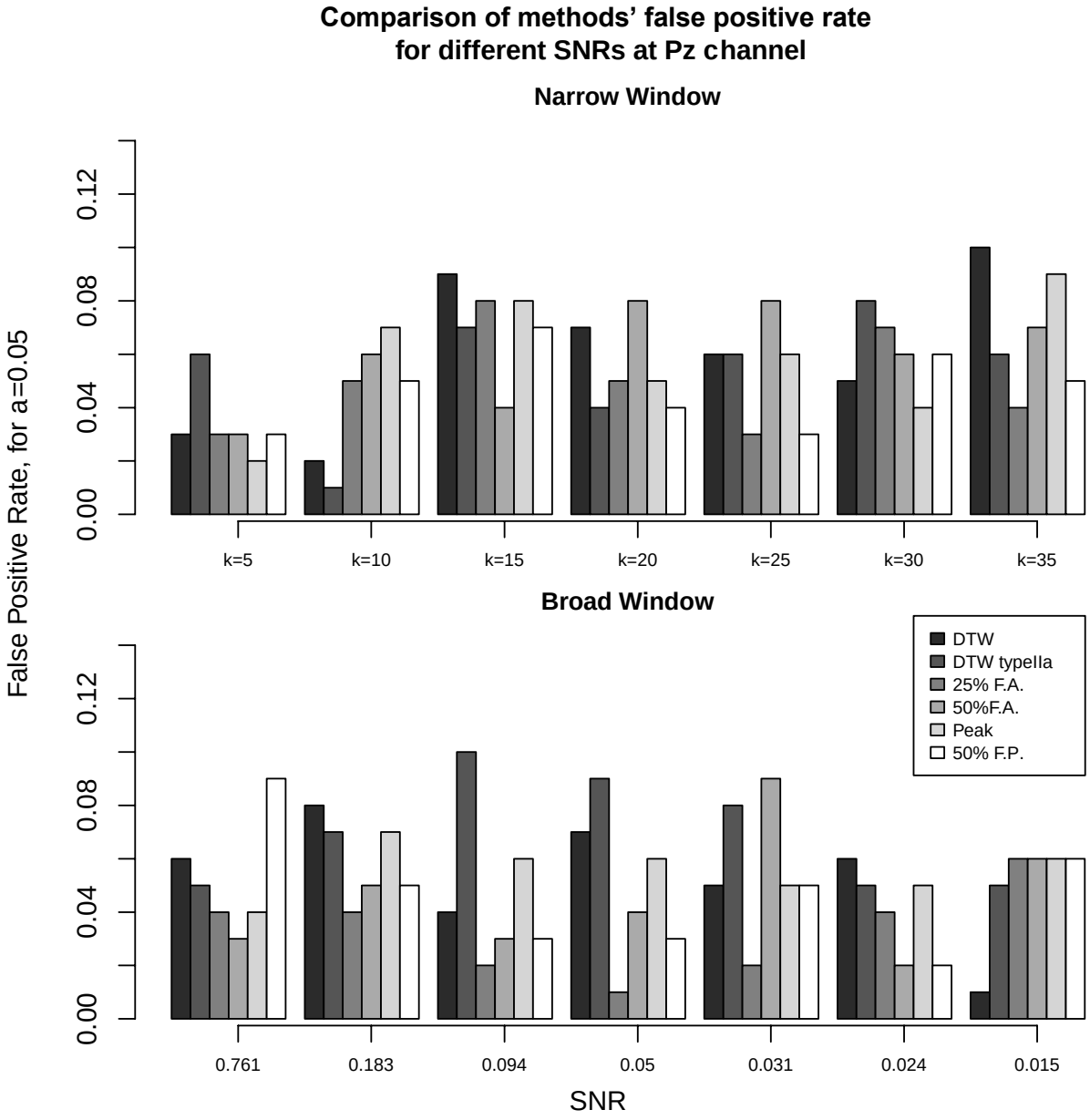

Figure 12: Comparison of the methods' false positive rate for different SNRs at channel Pz.

Table 2

Power to detect latency difference for Channel Pz when conditions differ only in latency. Percentage of p-values below alpha level for different SNRs.

SNR	Methods' Power at Channel Pz											
	DTW		DTW typeIIa		25% F.A.		50% F.A.		Peak		50% F.P.	
	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad
0.761	1.00	1.00	1.00	1.00	1.00	0.61	1.00	0.87	0.53	0.70	1.00	1.00
0.183	0.96	0.96	0.99	1.00	1.00	0.30	0.98	0.56	0.47	0.48	0.93	0.92
0.094	0.83	0.79	0.83	0.88	0.78	0.18	0.75	0.34	0.28	0.33	0.54	0.38
0.05	0.53	0.56	0.50	0.75	0.51	0.08	0.46	0.20	0.25	0.23	0.35	0.27
0.031	0.38	0.33	0.37	0.58	0.36	0.08	0.33	0.16	0.21	0.11	0.26	0.16
0.024	0.30	0.23	0.34	0.38	0.29	0.08	0.22	0.09	0.16	0.25	0.15	0.11
0.015	0.29	0.14	0.29	0.24	0.18	0.09	0.21	0.09	0.20	0.11	0.18	0.10
Mean	0.61	0.57	0.62	0.69	0.59	0.20	0.56	0.33	0.30	0.32	0.49	0.42

Table 3

False positive rate for channel Fz.

SNR	False Positive Rate at Channel Fz											
	DTW		DTW typeIIa		25% F.A.		50% F.A.		Peak		50% F.P.	
	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad
0.525	0.08	0.04	0.04	0.07	0.11	0.04	0.11	0.03	0.07	0.05	0.14	0.04
0.147	0.06	0.03	0.05	0.06	0.08	0.06	0.05	0.09	0.09	0.08	0.11	0.02
0.062	0.05	0.05	0.04	0.06	0.04	0.07	0.04	0.06	0.06	0.08	0.04	0.07
0.036	0.07	0.05	0.05	0.07	0.06	0.08	0.04	0.05	0.07	0.08	0.06	0.04
0.022	0.12	0.00	0.06	0.02	0.10	0.07	0.05	0.06	0.04	0.02	0.06	0.05
0.015	0.09	0.05	0.07	0.06	0.12	0.02	0.05	0.04	0.07	0.04	0.07	0.05
0.012	0.02	0.06	0.03	0.02	0.02	0.07	0.05	0.05	0.03	0.07	0.04	0.07
Mean	0.07	0.04	0.05	0.05	0.08	0.06	0.06	0.05	0.06	0.06	0.07	0.05

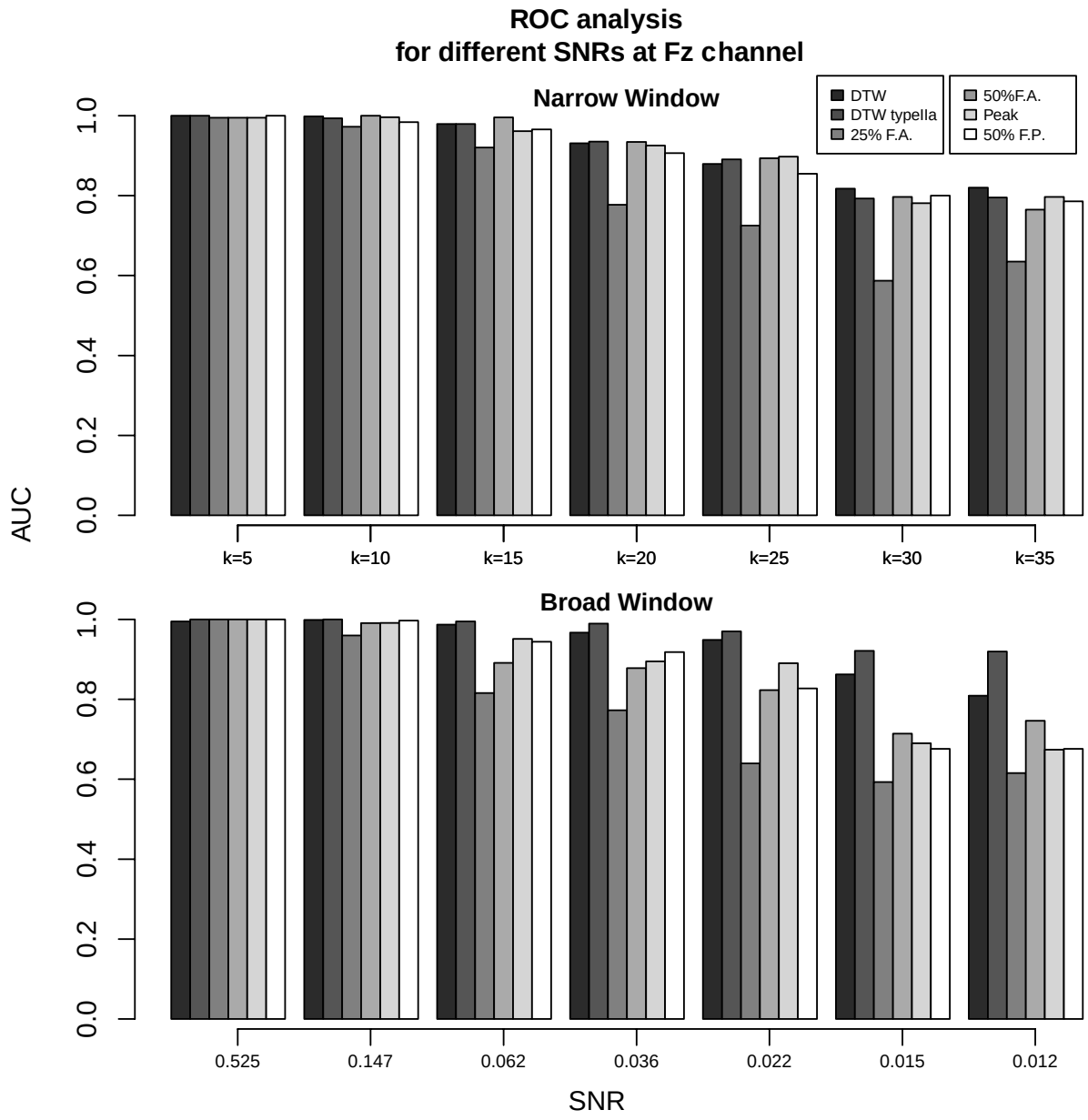

Figure 13: ROC analysis for the methods' at channel Fz. For the narrow window the ROC analysis shows that the methods have competitive performance with 25% fractional area having the smallest AUC across all SNRs. When the broad window is used, DTW typeIIa outperforms the rest of the methods.

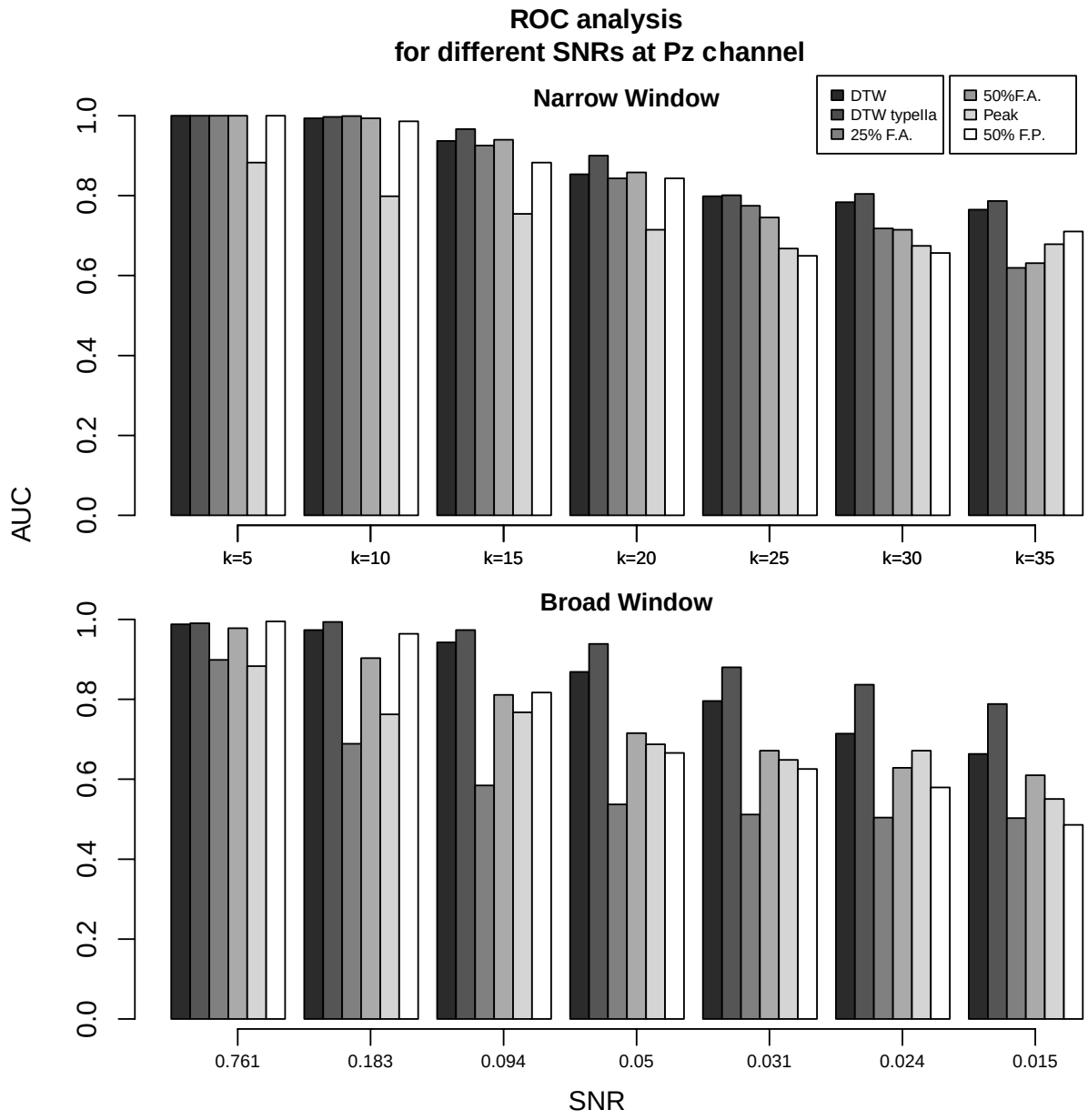

Figure 14: ROC analysis for the methods' at channel Pz. For both windows DTW typeIIa has the highest AUC and is the most resilient to noise.

Table 4

False positive rate for channel Pz..

SNR	Power											
	DTW		DTW typeIIa		25% F.A.		50% F.A.		Peak		50% F.P.	
	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad
0.761	0.03	0.06	0.06	0.05	0.03	0.04	0.03	0.03	0.02	0.04	0.03	0.09
0.183	0.02	0.08	0.01	0.07	0.05	0.04	0.06	0.05	0.07	0.07	0.05	0.05
0.094	0.09	0.04	0.07	0.10	0.08	0.02	0.04	0.03	0.08	0.06	0.07	0.03
0.05	0.07	0.07	0.04	0.09	0.05	0.01	0.08	0.04	0.05	0.06	0.04	0.03
0.031	0.06	0.05	0.06	0.08	0.03	0.02	0.08	0.09	0.06	0.05	0.03	0.05
0.024	0.05	0.06	0.08	0.05	0.07	0.04	0.06	0.02	0.04	0.05	0.06	0.02
0.015	0.10	0.01	0.06	0.05	0.04	0.06	0.07	0.06	0.09	0.06	0.05	0.06
Mean	0.06	0.05	0.05	0.07	0.05	0.03	0.06	0.05	0.06	0.06	0.05	0.05

Table 5

ROC Analysis: Area under the ROC curve for Channel Fz

SNR	Area Under ROC curve Channel Fz											
	DTW		DTW typeIIa		25% F.A.		50% F.A.		Peak		50% F.P.	
	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad
0.525	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
0.147	1.00	1.00	0.99	1.00	0.97	0.96	1.00	0.99	1.00	0.99	0.98	1.00
0.062	0.98	0.99	0.98	0.99	0.92	0.82	1.00	0.89	0.96	0.95	0.97	0.94
0.036	0.93	0.97	0.94	0.99	0.78	0.77	0.93	0.88	0.93	0.89	0.91	0.92
0.022	0.88	0.95	0.89	0.97	0.73	0.64	0.89	0.82	0.90	0.89	0.85	0.83
0.015	0.82	0.86	0.79	0.92	0.59	0.59	0.80	0.71	0.78	0.69	0.80	0.68
0.012	0.82	0.81	0.80	0.92	0.63	0.62	0.76	0.75	0.80	0.67	0.79	0.68
Mean	0.92	0.94	0.91	0.97	0.80	0.77	0.91	0.86	0.91	0.87	0.90	0.86

Table 6

Area under the ROC curve for Channel Pz when conditions differ in latency.

SNR	Area Under ROC curve for Channel Pz											
	DTW		DTW typeIIa		25% F.A.		50% F.A.		Peak		50% F.P.	
	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad	narrow	broad
0.761	1.00	0.99	1.00	0.99	1.00	0.90	1.00	0.98	0.88	0.88	1.00	1.00
0.183	0.99	0.97	1.00	0.99	1.00	0.69	0.99	0.90	0.80	0.76	0.99	0.96
0.094	0.94	0.94	0.97	0.97	0.93	0.58	0.94	0.81	0.75	0.77	0.88	0.82
0.05	0.85	0.87	0.90	0.94	0.84	0.54	0.86	0.72	0.71	0.69	0.84	0.67
0.031	0.80	0.80	0.80	0.88	0.77	0.51	0.75	0.67	0.67	0.65	0.65	0.63
0.024	0.78	0.71	0.80	0.84	0.72	0.50	0.71	0.63	0.67	0.67	0.66	0.58
0.015	0.76	0.66	0.79	0.79	0.62	0.50	0.63	0.61	0.68	0.55	0.71	0.49
Mean	0.88	0.85	0.89	0.91	0.84	0.60	0.84	0.76	0.74	0.71	0.82	0.73

3.4.1.Window sensitivity – Results

For channel Fz and a step of 5ms, *DTW* failed on average (across all starting points) 2.3%, while *DTW*_typeIIa, *25%* and *50% fractional area*, failed 6.23%, 32.61% and 24.62% respectively. If we restrict the results to only the first three starting points of interest, the percentages are adjusted to 3.07%, 8.30%, 14.21% and 16.29%. In all cases, *DTW* fails for the smallest number of windows. The average rate of windows where the methods fail remains the same when a step of 15ms is used. The only case that *fractional area* outperformed *DTW*_typeIIa was for Fz with the starting point at 100ms. That is because at 100ms the two conditions appear as if one is the mirror of the other until 150ms. For that period, 100ms-150ms, both *DTW* and *DTW*_typeIIa fail to recognize the correct latency, *DTW*_typeIIa being more sensitive, exhibits the same behavior until 215ms.

For the Pz channel with the 5 ms step, the methods average rates are 0.3%, 0.7%, 34.9% and 27.2% for *DTW*, *DTW*_typeIIa, *25%* and *50% fractional area* respectively (average rates across different starting points, for a 5ms step). If we isolate the analysis only for the starting points of 150ms and 300ms, which are more relevant for the P3b signal, then the previous rates

change to 0.6%, 1.35%, 30.8% and 28.35%. Clearly the *fractional area* methods have very poor performance when compared with the *DTW* showing the importance of 'fitting' the window to the shape of the underlying signal. The difference between the methods' performance remains the same when the 15ms step is used. The rates of windows that the methods fail for starting points of 150ms and 300ms is now 0% (*DTW*), 0.9% (*DTW*_typeIIa), 31.90% (*25%* f.a.) and 28.7% (*50%* f.a.). A breakdown of the results of the analysis is presented in Table 7.

Table 7

Comparing the sensitivity to window placement for DTW and Fractional area

			DTW	DTW typeIIa	25% F.A.	50% F.A.
Channel	Starting point	Step	% windows failed to detect direction of effect			
Fz	50ms	5ms	0.0%	7.4%	23.0%	24.1%
Fz	100ms	5ms	5.6%	12.8%	9.5%	11.2%
Fz	150ms	5ms	3.6%	4.7%	10.1%	13.6%
Fz	300ms	5ms	0.0%	0.0%	87.8%	49.6%
Fz	50ms	15ms	0.0%	6.5%	24.2%	25.8%
Fz	100ms	15ms	5.1%	11.9%	8.5%	11.9%
Fz	150ms	15ms	3.6%	3.6%	10.9%	14.5%
Fz	300ms	15ms	0.0%	0.0%	88.9%	48.9%
Pz	50ms	5ms	0.0%	0.0%	57.1%	30.7%
Pz	100ms	5ms	0.0%	0.0%	20.7%	21.2%
Pz	150ms	5ms	1.2%	2.7%	31.4%	20.7%
Pz	300ms	5ms	0.0%	0.0%	30.2%	36.0%
Pz	50ms	15ms	0.0%	0.0%	56.5%	30.6%
Pz	100ms	15ms	0.0%	0.0%	18.6%	22.0%
Pz	150ms	15ms	0.0%	1.8%	32.7%	21.8%
Pz	300ms	15ms	0.0%	0.0%	31.1%	35.6%

DTW's success over *fractional area* in terms of size and placement of the window is due to the fact that *DTW* examines the relationship between the two waveforms in order to determine their temporal differences. *Fractional area* however returns a single measure for each signal and difference between these two measures is used to calculate the latency contrast.

In this analysis, the same latency difference is present in the entire ERP, which means that even if other components are present in a particular window, they still have the same latency difference. This means that for most windows (there can always be a very small area where the opposite effect is present, e.g. see Figure 6), the latency contrast is always the same. The results obtained demonstrate that *fractional area* needs very careful placement of the window, based on inspection of the shape of the waveform, in order to detect the correct latency contrast. Such placement runs the risk of post hoc "fishing", inflating the type I error rate.

5.Discussion

The analysis presented in this paper shows that *DTW* is a promising technique for estimating latency differences between different experimental conditions. Previous approaches have shown that *fractional area*, specifically when the *50%* fraction is used, performs consistently well in detecting ERP latency differences (Kiesel et al., 2008). The cases examined here are consistent with these results. They also show that *DTW* generally outperforms *50% fractional area* in terms of power to detect latency differences, as well as outperforming all the other methods, especially as SNR drops and/or the size of the window used, broadens.

Although *DTW* has been successfully used in many domains, one issue is that variability in the y-axis (in our case: potential) can in some circumstances affect the warping on the x-axis (in our case: time). One of the side effects is that a time point from one time series could be mapped to a very large number of points from the other time series (Keogh, 2001). This is a

well-known issue and there are a few approaches to overcome it. In this analysis, we used a common step pattern (the typeIIa), as a means to obtain more realistic warpings. When *DTW* with step typeIIa is used to detect latency contrasts, performance improves, with greater power to detect latency in comparison to any of the other methods across most SNRs and/or types of window. This makes *DTW* typeIIa a very promising technique especially when experimental data are noisy, and/or not many trials are available. In addition, one can normalize (y-axis) amplitude values, by mapping points to z-scores. This transformation, in a sense equalizes overall amplitude differences across conditions. The resulting time series of z-values can be compared using DTW, providing a "purer" latency comparison.

Some might argue over the choice of windows used in this study, especially that a window, such as the broad one, is too broad, meaning that one might consider that it includes more than one component. In Kiesel et al, the authors advise that researchers should visually inspect their data so as to make sure that overlapping components are excluded from the analyzed data (determining appropriate time windows) but also in order to select the most appropriate method. However, it is widely accepted that placing time windows after inspection of the data, where the effect appears greatest can be dangerous as this can result in an inflation of false positives (Kriegeskorte et al, 2009; Kilner, 2013). Therefore, we argue that although visually our time window might seem broad, it is actually placed with regards to the a priori assumption that our component of interest arises during this window of time. Such window placement reduces the risk of type I errors. The analysis showed that *DTW*'s power increased for the broad window. This can be attributed to the fact that *DTW* examines latency contrasts based on the temporal relationship of two conditions (time series), instead of returning one value for

each method and then using the difference between them as a measurement of latency difference; this would also explain why it is more resilient to noise compared to *fractional area*.

DTW is a promising technique, not only because of its power to detect latency contrasts, and its robustness to noise and window placement, but also because it does not require parameterization based on the shape of the components under investigation. We propose that selecting the parameters for *fractional area*, fractional *peak* (i.e. the percentage, whether absolute values are used for amplitude, etc.), are decisions that will greatly affect the result, and are consequently difficult to select a priori, based on the nature of the experiment. Our analysis showed that *25% fractional area* had very poor performance, which indicates that the fraction chosen is of substantial importance when correctly identifying the underlying latency contrast. *DTW* does not require close post hoc inspection of the data in order to select the best parameters, which could allow better comparison across experiments, and help to produce more generalized conclusions about the latency differences across different conditions.

Although it was not strictly presented in this analysis, *DTW* allows visualizations of the temporal relationship between two ERPs, enabling depiction of how latency contrasts change across the entire time course. This relationship can be inspected by the deviations of the warping path from the main diagonal, which could provide an insight into how conditions differ in their manifestation through time. In this way, *DTW* can be used as an exploratory tool in ERP research.

Further investigation into different components, exploration and comparison with other methods regarding the power to detect different size of effects, could increase the confidence in *DTW*'s ability to correctly identify and measure latency contrasts. The work presented here is done on simulated data, as we wanted to have absolute control over the latency effect that was

present and the signal to noise ratio. Application of the method on experimental data is going to be part of our future research. Nonetheless, we propose that the results presented here support our central assertion: latency is best viewed as a region rather than a point measure, i.e. as a temporal relationship between two conditions across a whole segment of a time series, and that *DTW* is a promising method for this purpose.

Citations and References

Bowman, H., Filetti, M., Janssen, D., Su, L., Alsufyani, A., & Wyble, B. (2013). Subliminal salience search illustrated: EEG identity and deception detection on the fringe of awareness. *PloS One, 8*(1), e54258, doi: 10.1371/journal.pone.0054258.

Berndt, D. J., & Clifford, J. (1994, July). Using Dynamic Time Warping to Find Patterns in Time Series. In KDD workshop (Vol. 10, No. 16, pp. 359-370).

Casarotto, S., Bianchi, A. M., Cerutti, S., & Chiarenza, G. A. (2005). Dynamic time warping in the analysis of event-related potentials. *Engineering in Medicine and Biology Magazine, IEEE, 24*(1), 68-77, doi: 10.1109/MEMB.2005.1384103.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861-874, doi: 10.1016/j.patrec.2005.10.010.

Flach, P. (2010). ROC analysis. *Encyclopedia of Machine Learning*, 869-874.

Giorgino, T. (2009). Computing and visualizing dynamic time warping alignments in R: The *DTW* package. *Journal of Statistical Software, 31*(7), 1-24.

Handy, T. C. (2005). 3 basic principles of ERP quantification. *Event-Related Potentials: A Methods Handbook*, 33, ISBN: 0262083337.

Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics, 5*(3), 299-314, doi: 10.1080/10618600.1996.10474713.

Keogh, E. J., & Pazzani, M. J. (2001). Derivative dynamic time warping. *The 1st SIAM Int. Conf. on Data Mining (SDM-2001), Chicago, IL, USA*,

Kiesel, A., Miller, J., Jolicœur, P., & Brisson, B. (2008). Measurement of ERP latency differences: A comparison of single- participant and jackknife- based scoring methods. *Psychophysiology, 45*(2), 250-274.

Kilner, J. M. (2013). Bias in a common EEG and MEG statistical analysis and how to avoid it. *Clinical Neurophysiology : Official Journal of the International Federation of Clinical Neurophysiology*, doi:10.1016/j.clinph.2013.03.024

Kriegeskorte, N., Simmons, W. K., Bellgowan, P. S., & Baker, C. I. (2009). Circular analysis in systems neuroscience: The dangers of double dipping. *Nature Neuroscience, 12*(5), 535-540, doi: 10.1038/nn.2303.

Luck, S. J. (2005). An Introduction to the Event-Related Potential Technique. The MIT Press.

Luck, S. J., & Kappenman, E. S. (2012). *The oxford handbook of event-related potential components*, OUP USA.

Manly, B.J. (1997). Randomization, Bootstrap and Monte Carlo methods in biology. *New York: Chapman & Hall, 2002*, ISBN: 1584885416.

Müller, M. (2007). Dynamic time warping. *Information Retrieval for Music and Motion*, 69-84, ISBN: 3540740481.

Myers, C., Rabiner, L., & Rosenberg, A. (1980). Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on, 28*(6), 623-635, doi: 10.1109/TASSP.1980.

Senin, P. (2008). Dynamic time warping algorithm review. *Honolulu, USA.*

Tormene, P., Giorgino, T., Quaglini, S., & Stefanelli, M. (2009). Matching incomplete time series with dynamic time warping: An algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine, 45*(1), 11-34.

Drongelen, W. van. (2006). Signal Processing for Neuroscientists: An Introduction to the Analysis of Physiological Signals (p. 320), Academic Press.

Wang, K., Begleiter, H., & Porjesz, B. (2001). Warp-averaging event-related potentials. *Clinical Neurophysiology*, *112*(10), 1917-1924.