Neurocomputing 92 (2012) 156–169

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Sharing and integration of cognitive neuroscience data: Metric and pattern matching across heterogeneous ERP datasets

Haishan Liu [a,*], Gwen Frishkoff [b], Robert Frank [c], Dejing Dou [a]

[a] Department of Computer & Information Science, University of Oregon, Eugene, OR 97403, United States
[b] Department of Psychology & Neuroscience Institute, Georgia State University, Atlanta, GA 30303, United States
[c] NeuroInformatics Center, University of Oregon, Eugene, OR 97403, United States

ARTICLE INFO

Available online 13 March 2012

*Keywords:*
Schema matching
Cluster comparison
Assignment problem
Hungarian method
Neuroscience
Electrophysiology

ABSTRACT

In the present paper, we use data mining methods to address two challenges in the sharing and integration of data from electrophysiological (ERP) studies of human brain function. The first challenge, *ERP metric matching*, is to identify correspondences among distinct summary features (“metrics”) in ERP datasets from different research labs. The second challenge, *ERP pattern matching*, is to align the ERP patterns or “components” in these datasets. We address both challenges within a unified framework. The utility of this framework is illustrated in a series of experiments using ERP datasets that are designed to simulate heterogeneities from three sources: (a) *different groups of subjects* with distinct simulated patterns of brain activity, (b) *different measurement methods*, i.e, alternative spatial and temporal metrics, and (c) *different patterns*, reflecting the use of alternative pattern analysis techniques. Unlike real ERP data, the simulated data are derived from known source patterns, providing a gold standard for evaluation of the proposed matching methods. Using this approach, we demonstrate that the proposed method outperforms well-known existing methods, because it utilizes cluster-based structure and thus achieves finer-grained representation of the multidimensional (spatial and temporal) attributes of ERP data.

## 1. Introduction

Over the last few decades, neuroscience has witnessed an explosion of methods for measurement of human brain function, including high-density (multi-sensor) electroencephalography (EEG) and event-related potentials (ERP), or so-called “brainwave” methods. In comparison with other neuroimaging techniques, the ERP method has several advantages: it is completely noninvasive and, unlike fMRI (which measures blood flow), it is an inexpensive, portable, and direct measure of neuronal activity. Moreover, it has excellent (millisecond) temporal resolution, which is critical for accurate representation of neural activity. Furthermore, ERP studies have given rise to many complex neural patterns that can be used to predict human behavior and to detect clinically relevant deviations in behavior, cognition, and neural function [1,2]. Dozens of these patterns have been proposed over the past several decades. Yet there is remarkably little agreement in how these patterns should be identified and described. Further, tens of thousands of large and information-rich datasets have been collected and analyzed. Yet there are few (arguably, no) quantitative comparisons (“meta-analyses”) of ERP data from different studies. Given the unique importance of ERP research in human neuroscience, this lack of integration may be the central obstacle to a robust science of human behavior and brain function.

To address these challenges, we have designed a system called Neural ElectroMagnetic Ontologies, or NEMO [3–6]. NEMO includes a suite of computational methods and workflows that are built around formal ontologies (description logic representations for the ERP domain) and can be used to facilitate ERP data sharing, analysis, and integration.

In the present paper, we introduce a new component of the NEMO workflow, which uses data mining methods to address two key problems—what we term the *ERP metric matching* and *ERP pattern matching* problems. In both cases, our goal is to align variables across multiple, heterogeneous ERP datasets. This would provide a data-driven alternative to top-down (knowledge-driven) methods, such as advocating the use of restricted methods for analysis, or a controlled vocabulary for data annotation. While these top-down approaches are of considerable value [5,6], we believe that data-driven approaches may provide a complementary approach that can lead to new insights into complex ERP data structures.

The remainder of Section 1 describes the ERP metric and pattern matching problems and summarizes our approach to these two problems. Section 2 presents a theoretic framework, along with a

[*] Corresponding author.
*E-mail address:* ahoyleo@cs.uoregon.edu (H. Liu).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2012.01.028

description of our approach to the two matching problems. In Section 3, we describe two case studies in which we used our approach to align different variables (metrics and patterns) across simulated ERP data. These data were designed to mimic three sources of variability that are present in real ERP data. Section 4 presents results from the two case studies. Section 5 compares the proposed method with existing methods, summarizes some assumptions and limitations of our study, and suggests some directions for future work. Finally, Section 6 summarizes the contributions of the present work.

### 1.1. ERP metric matching problem

ERP patterns are characterized by three features: time course (e.g., early or late), polarity (positive or negative), and scalp distribution, or topography [4,7,8]. For example, the visual-evoked "P100 pattern" (Fig. 1A) has a peak latency of approximately 100 ms (Fig. 1B) and is positive over occipital areas of the scalp (Fig. 1C), reflecting generators in visual regions of the cerebral cortex.

Researchers use a variety of metrics to describe the three features. These metrics reflect different ways that temporal and spatial features of ERPs can be operationally defined. For example, one research group might use a measure of peak latency (time of maximum amplitude) to summarize the timecourse of the "P100" pattern in a visual object recognition experiment [9,10], while another group might use measures of pattern onset and offset to operationalize the concept of latency for the same dataset.

The use of different metrics has a long history in the ERP research. While these diverse practices present a nuisance for data sharing and integration, there are reasons to embrace this heterogeneity, since different metrics may yield distinct and complementary insights [11]. The challenge then becomes how to find valid correspondences between these metrics. In the previous work, we have described top-down (knowledge-driven) methods, that is, annotation of data using a formal ontology [4–7]. This approach minimizes heterogeneity that arises from the use of different labels (e.g., "latency" vs. "peak latency" for time of maximal amplitude). It does not, however, address heterogeneities that reflect different operational definitions of time (e.g., peak latency vs. duration of a pattern), as described above. For this reason, we have also explored the use of bottom-up (data-driven) methods [11] to align different metrics across ERP datasets. In the present paper, we extend our bottom-up approach by developing and testing a more general formulation of the metric matching problem. Specifically, we view metric matching as an assignment problem and articulate a more general solution that can also be used to address a second problem—that of ERP pattern matching.

### 1.2. ERP pattern matching problem

The ERP pattern matching problem is the problem of finding correspondences among ERP patterns from different datasets. This problem is challenging for several reasons. The most trivial reason is that authors use a variety of labels to refer to the same pattern [4,7], just as they use different names for the same or similar metrics. This issue is readily addressed by the use of a standard ontology (or controlled vocabulary), although there is no guarantee that such a would be adopted by all research labs. The second reason is related to the metric matching problem (Section 1.1): when two different measures are used to characterize the timecourse of a pattern, they may capture subtly different views of the same data. Accordingly, they may introduce additional variability into the pattern matching equation. Finally, the most profound challenge is a consequence of the physics and physiology of signal generation: scalp-measured ERPs reflect a complex and unknown mixture of latent patterns. The reason is that neuroelectric signals are generated in cortex and are propagated to the scalp surface. Moreover, at each moment, multiple regions of cortex are co-active. Thus, at every timepoint and at every point in the measurement (i.e., scalp) space, a pattern in the measured data actually corresponds to multiple overlapping patterns, that is, different underlying sources. This overlap or "superposition" is exacerbated by volume conduction of these signals through the resistive skull.

Given these complexities, ERP researchers have adopted a variety of solutions for identification and extraction of ERP patterns (e.g., [1,2]). It can therefore be hard to compare results from different studies, even when the same experimental stimuli and task are used. Nonetheless, alternative analysis methods, like alternative metrics, may provide different, and equally informative views, of the "same" data. Thus, we propose to embrace this heterogeneity, rather than forcing researchers to use a restricted set of solutions for data analysis. As a consequence, our approach to data integration will require pattern matching, as well as metric matching, across different ERP datasets. Moreover, this matching should ideally be robust to differences among patterns that arise from the use of alternative pattern analysis methods.

### 1.3. Study goals and hypotheses

In this paper, we address the ERP metric and pattern matching problems by transforming them into two more general problems,

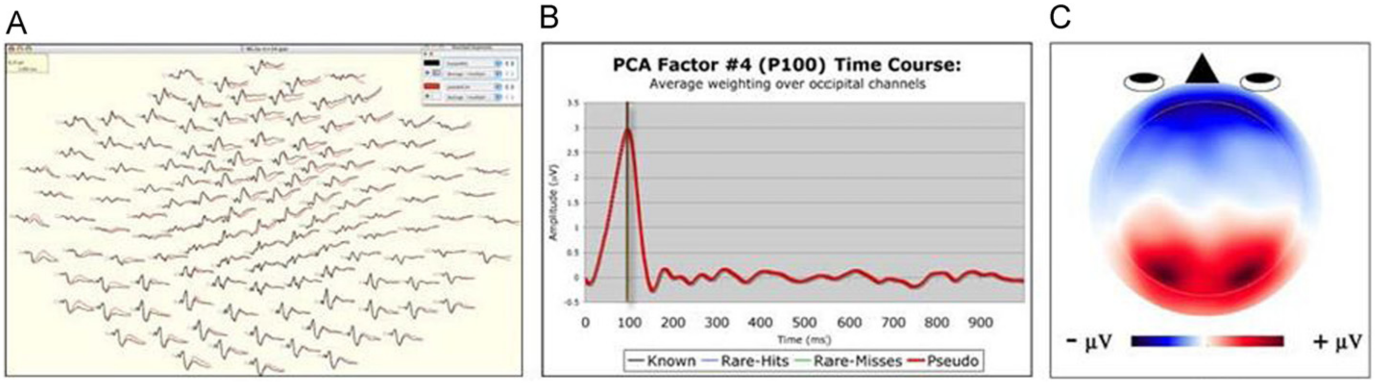


**Fig. 1.** (A) 128-Channel EEG waveplot; positive voltage plotted up. Black, response to words; red, response to non-words. (B) Time course of P100 pattern for same dataset, extracted using Principal Components Analysis. (C) Topography of P100 factor. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

both of which have well-known solutions. The metric matching problem is viewed as a schema matching problem, where the alternative sets of metrics are treated as attributes (column names) in the schemas that correspond to different ERP datasets. Similarly, the pattern matching problem is characterized as a cluster comparison problem, where patterns are extracted from different ERP datasets using cluster analysis. At an even more general level, the two problems are characterized as a type of assignment problem. In this way, they can be solved using a common framework, as described in the following section. We test this framework in a series of case studies, using ERP datasets that were designed to simulate heterogeneous data from different groups of subjects under different conditions (via distinct simulated brain activity profiles), as well as distinct measurement methods (spatial and temporal metrics) and distinct patterns (reflecting different pattern decomposition techniques).

To evaluate the performance of our proposed metric matching method, we use two other methods for comparison—SemInt [12] and cen-com, a method that relies on segmented central tendency. SemInt calculates statistics, such as maximum, minimum, mean, and variance, of the data. The statistics are extracted by running aggregation queries against the whole set of attribute values. However, it is possible that two different attributes could have similar mean values; thus, SemInt statistics may be too coarse-grained to represent different ERP attributes. In contrast with SemInt, our proposed metric matching method examines the grouping structure of attribute values through cluster analysis. It therefore supports fine-grained comparisons between attributes. In this way, it is conceptually similar to methods that rely on segmented central tendency—for example, the use of multiple, combined measures, such as ⟨Mean, StdDev⟩, to characterize substructures within a data distribution. Segmented central tendency (cen-com) is straightforward and suitable for applications where the data instances are unordered or randomly ordered. In the present case, the metric values are randomly ordered across instances (i.e., subjects). We thus apply smoothing to our point sequence curves as a post-processing step (i.e., after subsequence reordering), to reduce the within-cluster variance due to instance order. *In the present study*, *we hypothesize that our metric matching method will outperform SemInt. We further hypothesize that our method will perform at least as well as a method based on segmented central tendency*.

Our proposed pattern matching method uses cluster comparison methods. These methods are closely related to methods for determining cluster validity, such as the technique of external, or relative, indexing, which is used to compare different clustering results. Cluster validity methods include the Rand index [13], Jaccard index [14], normalized mutual information [15]. These methods aim at evaluating cluster membership, that is, the assignment of points to clusters. However, an important limitation of these methods is that they can only be reliably used to compare different cluster results on the same dataset. By contrast, the aim of our study is to align patterns across datasets that contain non-overlapping observations. Thus, membership-based cluster validity criteria are unsuitable. Furthermore, our method should be able to use information about the distribution of different ERP pattern attributes, i.e., spatial and temporal features of the data. To this end, we have represented clusters as density profiles, as proposed by Bae et al. [16], and have selected a cluster similarity index known as ADCO (Attribute Distribution Clustering Orthogonality). ADCO determines the similarity between two clusters based on their density profiles, which capture the distribution of values along each attribute. *We hypothesize that ADCO scores will reflect high* (80%) *precision in matching ERP patterns across datasets. We further expect that the more domain knowledge we leverage during the cluster analysis process*, *the more accurate the resulted clusters will be. To test this second hypothesis*, *we apply the ADCO method under three scenarios*, *where we leverage varying amounts of domain knowledge*.

## 2. Theoretic framework

### 2.1. Overview

In the present section, we present our theoretic framework for finding the general correspondence function $M(S,S')$ between two sets $S$ and $S'$. We discuss application-specific parameters in Sections 2.1.2 and 2.1.3.

#### 2.1.1. Transformation of matching problem to assignment problem

The goal of metric matching is to find correspondence between alternative sets of metrics $M$ and $M'$; and the goal for pattern matching is to find correspondences between different sets of patterns ($P$ and $P'$). These two problems can be formulated in a very general way, as illustrated in Fig. 2. Fig. 2A shows a sample ERP experiment data from one lab and Fig. 2B from another. Two sets of column headers, $M=\{M_1,M_2,\ldots,M_n\}$ and $M'=\{M'_1,M'_2,\ldots,M'_n\}$, denote alternative sets of metrics used by the two labs. The task of metric matching is to infer a function $M(M,M')$ that establishes a correspondence between the two metric sets. On the other hand, rows in the data represent individual experiment observations and can be grouped to clusters to capture ERP spatiotemporal patterns. Fig. 2 shows two sets of patterns, $P=\{P_1,P_2,\ldots,P_n\}$ and $P'=\{P'_1,P'_2,\ldots,P'_n\}$, extracted by different pattern separation and analysis methods from two different datasets. The task of pattern matching is then to infer a function $\mathcal{M}(P,P')$ that establishes a correspondence between the two sets of patterns.

Following Bae et al. [16], we propose a theoretic framework that encompasses the solution to both the metric matching and pattern matching problems by transforming them to the assignment problem. The assignment problem is formally posed as follows. Let us

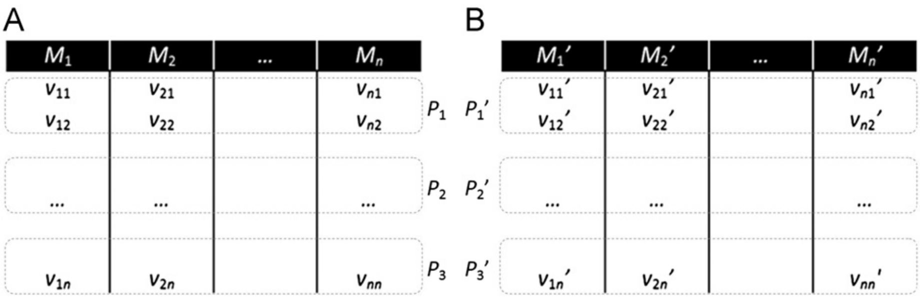


**Fig. 2.** Sample ERP data A and B from two laboratories. The goal of metric matching is to find correspondence between alternative sets of metrics $M$ and $M'$; and the goal for pattern matching is to find correspondences between different sets of patterns ($P$ and $P'$).

**Table 1**
A sample similarity table between two sets of ERP metrics $M$ and $M'$.

| | $M'_1$ | $M'_2$ | … | $M'_n$ |
|---|---|---|---|---|
| $M_1$ | 16 415 | 11 438 | | 9443 |
| $M_2$ | 11 395 | 12 394 | | 6317 |
| $\vdots$ | | | $\ddots$ | |
| $M_n$ | 9132 | 6384 | | 8376 |

first define $\rho$ as the stochastic permutation of a subscript index set. Without causing confusion, we also define function $\rho(k)$ that returns the value at index $k$ in the permutation set (i.e., $\rho = \{\rho(1), \rho(2), \ldots, \rho(n)\}$ is the permutation of $\{0, 1, \ldots, n\}$). We can view $\rho$ as a correspondence function between set $S_k$ and $S'_{\rho(k)}$, assuming correspondent positions in each set are considered to map to each other. For instance, if two sets both have three elements, then the subscript index set ranges over $\{0, 1, 2\}$. Supposing $\rho$ returns $\{2, 0, 1\}$, we can then obtain the following matchings: $S_0 \leftrightarrow S'_2$, $S_1 \leftrightarrow S'_0$, and $S_2 \leftrightarrow S'_1$.

With the notation defined, we can now formalize the problem of matching two sets of elements in general as an optimization problem, in which the aim is to select the best $\rho$ from the set of all possible permutations of the elements in one set, under the condition that some distance function characterizing the similarity between two sets in the matching is minimized. Eq. (1) expresses this idea

$$\mathcal{M}(S,S') = \arg\min_{\rho} \left( \sum_{k=1}^{K_{min}} \mathcal{L}(S_k, S'_{\rho(k)}) \right), \tag{1}$$

where $S$ and $S'$ are two sets under consideration, $K_{min}$ denotes the minimum cardinality of them, and function $\mathcal{L}$ denotes the distance function that we try to minimize. The form of $\mathcal{L}$ is subject to different applications.

To calculate $\rho$ that satisfies Eq. (1), rather than iterating through all possible permutations of elements in $S'$, we can consider the equation as a minimum-cost assignment problem between the sets $S$ and $S'$. Table 1, for example, illustrates a distance table between two metric sets $M$ and $M'$. Matching of the two sets can be considered as an assignment problem where the goal is to find an assignment of elements $\{M_k\}$ to $\{M'_k\}$ that yields the minimum total distance without assigning each $M_k$ more than once. This problem can be efficiently solved by the Hungarian method in polynomial time of $O(K^3_{min})$ [17]. It is worth noting that by formulating the problem as the assignment problem, we assume the matching between two sets to be a one-to-one function. We will discuss the limitation and implication of this assumption in Section 5.

### 2.1.2. Data representation and transformation

To apply Eq. (1) to solve metric matching and pattern matching problems, a central problem remains: what are the appropriate methods to model data, so that the distance function $\mathcal{L}$ can be calculated in a meaningful way? To this end, we develop the following approaches: in the schema matching problem, we represent ERP metrics (attributes) as either *point-sequence curves* or *segmented statistical characterization*, and in the cluster comparison problem, we model clusters by using *density profiles* [16].

*Schema matching problem*: The central objects of interest in our study of the schema matching problem are the numeric-typed attributes. In order to represent them in ways that meaningful quantities can be defined to measure the "goodness" of a matching decision, we propose to use two methods, namely, the point-sequence curve and the segmented statistical characterization to represent attributes.

The point-sequence curve is a curve plotted as attribute value against instance number. The distance between point-sequence curves characterizes the similarity between respective attributes. However, since instances are randomly ordered within a dataset and the instance numbering is inconsistent across datasets, the resulting original graph of point-sequence curves will show no clear pattern and thus no visible correspondence between alternative metrics (see Fig. 6, left frame). The key problem then becomes to align subsequences of curves in a principled way to enable valid comparisons among curves across datasets.

Our solution is to introduce structure into these (initially randomly ordered) point-sequence curves by applying clustering to extract similar subsequences, which are further labeled using pattern classes defined in the NEMO ontologies. These labeled subsequences can then be aligned across datasets (which implies that pattern matching must be carried out beforehand), and correspondences between curves established using standard techniques for sequence similarity search in time-series (see Fig. 1, right frame). This approach exploits prior knowledge of the patterns that are commonly seen in ERP experiments of a particular type (e.g., visual perception) while asserting no prior knowledge about the measures.

Numeric-typed attributes can be represented by the segmented statistical characterization, in which data instances are first partitioned into groups (e.g., through unsupervised clustering) and then characterized by a vector of indicators, each denoting a statistical characterization of the corresponding group. For example, if values of an attribute A are clustered into $n$ groups, then it can be represented by a vector of segmented statistical characterization as follows:

$$V_A = [\mu_1, \mu_2, \ldots, \mu_n],$$

where we choose the mean value $\mu_i$ for cluster $i$ as the statistical indicator. The indicators can be also defined as a combination of several statistical quantities such as a weighted sum of mean and standard deviation. In other words, an attribute can be represented as $a = [t_1, \ldots, t_n]$, where $t_i = \alpha c_{i_1} + \beta c_{i_2}$, $c_{i_1}$ and $c_{i_2}$ are the mean and standard deviation, respectively, $\alpha$ and $\beta$ are the weights, and $n$ is the number of clusters.

*Cluster comparison problem*: The *density profile* is a method of cluster representation that aims at capturing cluster structural information. By representing clusterings as density profiles, a novel clustering similarity measure, known as ADCO (Attribute Distribution Clustering Orthogonality) [16], has been designed which is able to take into account attribute distribution information, as well as point membership information.

To induce density profiles, we first conduct EM clustering on different datasets, and then label clusters with respect to the simulated ERP patterns or "components". The attribute's range in each cluster is discretized into a number of bins, and the similarity between two clusters corresponds to the number of points of each cluster falling within these bins (details in Section 2.3).

### 2.1.3. Choice of distance function

With the data modeling approaches presented in the previous section, we adopt two distance functions for the metric matching and pattern matching problems, respectively, described as follows.

(1) *Distance function in metric* (*schema*) *matching*: Representing attributes (metrics) using point-sequence curves enable us to utilize a range of sequence similarity search techniques. Keogh et al. [18] surveyed a wide range of sequence similarity measures and subjected these measures to extensive performance tests. They concluded that Euclidean distance outperformed other measures. Therefore, we used Euclidean distance to quantify degree of similarity among alternative metrics $\mathcal{L}$ in our study. We applied sub-sequence reordering and post-processing steps

(details in Section 2.2) prior to calculating the distance so that the resulting sequences could be compared in a principled way. Euclidean distance can also be used to calculate the similarity between attributes under segmented statistical characterization as well. In this case, the vector representations for two sets of attributes that derive from two different datasets need to be aligned, e.g., with reference to the cluster alignment. This alignment step corresponds to the sub-sequence reordering step in the point sequence-based method.

To apply Eq. (1) for matching two sets of ERP metrics (attributes) $A$ and $A'$ characterized by respective sets of point-sequence curves, we substitute the variables in Eq. (1) and derive Eq. (2), and plugging in the Euclidean distance for function $\mathcal{L}$ results in Eq. (3)

$$\mathcal{M}(A,A') = \arg\min_{\rho}\left(\sum_{k=1}^{K_{min}} \mathcal{L}(A_k, A'_{\rho(k)})\right), \tag{2}$$

$$\mathcal{M}(A,A') = \arg\min_{\rho}\left(\sum_{k=1}^{K_{min}} \sqrt{\sum_{i=1}^{N}(A_{ki} - A'_{\rho(k)i})^2}\right). \tag{3}$$

(2) *Distance function in pattern matching* (*cluster comparison*): After the density profile vectors of two clusterings $C$ and $C'$ are obtained, the degree of similarity between $C$ and $C'$ can be determined by calculating the dot product of the density profile vectors:

$$\mathrm{sim}(C,C') = V_C \cdot V_{C'}.$$

To derive correspondences for two sets of clusterings $C$ and $C'$ representing two sets of ERP patterns, we substitute the variables in Eq. (1), resulting in Eq. (4). Since the similarity function is inversely related to the distance function, in order to calculate Eq. (4), it suffices to use the similarity function sim($C$,$C'$) and change the operator to arg max. This is shown in Eq. (5). Details of the process for this calculation are described in Section 2.3

$$\mathcal{M}(C,C') = \arg\min_{\rho}\left(\sum_{k=1}^{K_{min}} \mathcal{L}(C_k, C'_{\rho(k)})\right), \tag{4}$$

$$\mathcal{M}(C,C') = \arg\max_{\rho}\left(\sum_{k=1}^{K_{min}} V_{C_k} \cdot V_{C'_{\rho(k)}}\right). \tag{5}$$

### 2.2. Metric matching method

We propose to view the attribute value vector of each ERP summary metric as forming a point-sequence curve. The similarity between metrics can then be addressed by calculating the Euclidean distance between the respective point-sequence curves. Fig. 3 (left frame) illustrates the flowchart for the metric matching process, which consists of the following steps:

1. To identify structured subsequences in each attribute vector, we use clustering and label discovered clusters with respect to the simulated ERP patterns or “components”, e.g., P100, N100, N3, MFN, and P300, as defined in our NEMO ontology [4].
2. By labeling the attribute instances in this way, we can group them in each dataset based on their pattern labels and then align the instance groups across datasets accordingly. This step can be viewed as a subsequence reordering process.
3. We then apply a sequence post-processing step to achieve better performance in the similarity search, leveraging the rich collection of sequence similarity search algorithms presently available.
4. The final step is to evaluate the similarity of the structured point-sequence curves that represent our two simulated ERP datasets as quantified by their respective measures. Instead of calculating the distance between all pairs of sequences from the two datasets, this evaluation is achieved by transforming the problem to an assignment problem that can be solved by the Hungarian method [17].

Next, we describe details of the subsequence reordering and sequence post-processing steps.

(1) *Data partitioning and reordering*: In the present study, we perform clustering on the spatial and temporal values of the two alternative sets of measures using the Expectation Maximization (EM) algorithm. The resulting clusters represent candidate ERP patterns, characterized by the central tendencies of their cluster attributes (i.e., mean values for the spatial and temporal metrics).

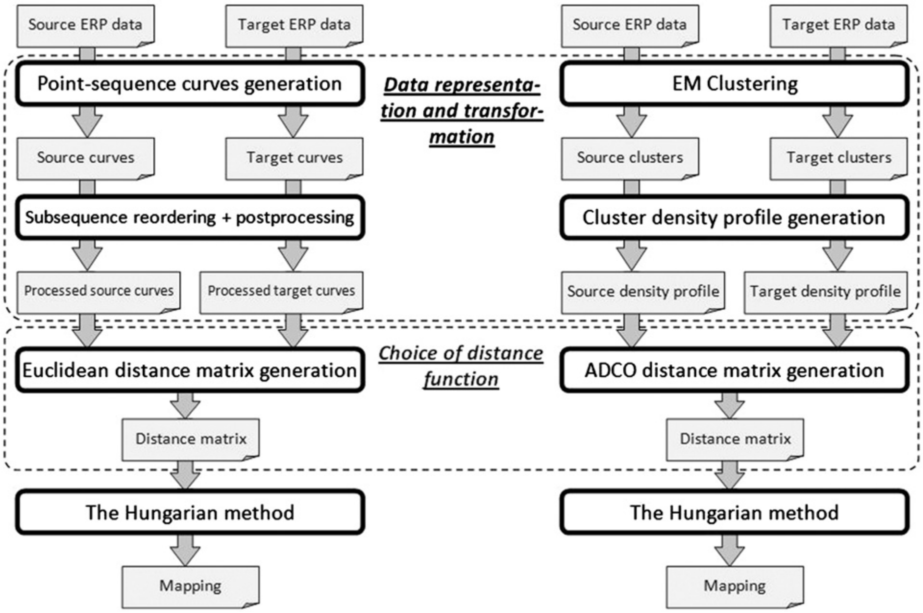


**Fig. 3.** Flowchart of the proposed approach (left—metric matching, right—pattern matching).

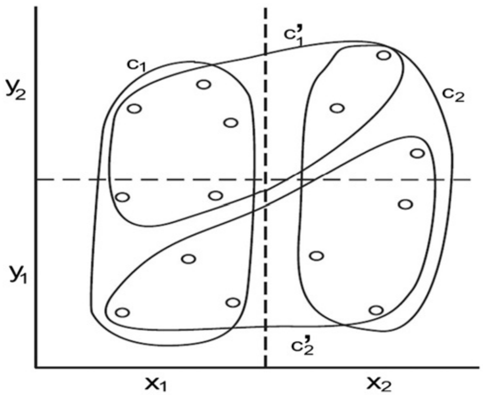


**Fig. 4.** Two clusterings $C=\{c_1,c_2\}$ and $C'=\{c'_1,c'_2\}$. Two attributes $X$ (attribute 1) and $Y$ (attribute 2) are discretized into two bins each. See [16] for details.

We label the resulting clusters with pattern labels defined in the NEMO ontologies (P100, N100, etc.) using rules specified by domain experts.

Following clustering and labeling, the pattern labels are used to align groups of instances across datasets, resulting in subsequence reordering. As illustrated on the right-hand graphs of Fig. 6, for example, the point-sequence curves for metrics IN-O1 and IN-LOCC (plotted using their original orderings prior to grouping/reordering on the left-hand side) are manifestly more similar after reordering subsequences in the two curves by aligning instances that belong to the same (or similar) patterns.

(2) *Sequence post-processing*: after alignment of the subsequences according to pattern labels defined in the NEMO ontology, we carry out three post-processing steps: (1) normalization, i.e., scaling all the sequence values to unit range; (2) smoothing, using a moving average method to reduce within cluster variance; and (3) interpolation of curves, if the number of points in two point-sequence curves is different. Fig. 7 illustrates the results of normalization, smoothing and interpolation to the point-sequence curves of IN-O1 and IN-LOCC in Fig. 6.

### 2.3. Pattern matching methods

As explained in Section 1, the problem of pattern matching can be framed as a cluster comparison task. Fig. 3 (right) illustrates the flowchart of the proposed pattern matching method comprising the following steps:

1. We first conduct EM clustering then label clusters with respect to the simulated ERP patterns or "components" (e.g., P100, N100, N3, MFN, and P300).
2. We then employ density profile to model clustering result and derive a matching between two clusterings by finding the best alignment of their respective density profiles that yields the highest similarity measure.
3. In the final step, we generate a distance matrix (e.g., Table 1) and then employ the Hungarian method to solve the alignment problem. The alignment induces matching between clusters, with the confidence of the matching expressed numerically by the ADCO score.

By representing clustering using a density profile, we are able not only to compare clusters derived from non-overlapping datasets, but also to take into account attribute distribution information in the matching process. We describe below how to represent clusters using density profiles and the calculation of the ADCO score. More details and explanations can be found in [16].

(1) *Density profile*: To represent clusters using density profiles, the attribute's range in each cluster is first discretized into a number of bins, and the similarity between two clusters corresponds to the number of points of each cluster falling within these bins. The formal definition for this number of points is the *density* of an attribute-bin region for cluster $c_k$ in clustering $C$, denoted as $dens_C(k,i,j)$. It refers to the number of points in the region $(i, j)$ – the $j$-th bin of the $i$-th attribute – that belongs to the cluster $c_k$ of clustering $C$. For example, for clustering $C$ in Fig. 4, $dens_C(1,1,1)=8$, because there are eight data points in region (1, 1) – the first bin of the first attribute $x$ – that belongs to the first cluster $c_1$.

The values of $dens_C(k,i,j)$ for all possible $k$, $i$, $j$ are then listed in a certain ordering to form a clustering's *density profile vector* (defined below). This ordering is imposed on all attribute-bin regions and must be applied to the two datasets in which the clusterings were generated. It is necessary, then, that both datasets must have the same attribute set. If this requirement does not stand, the matching between the sets must be specified in advance. Therefore, in order to apply the density profile method in the ERP pattern matching problem, we must first carry out measure matching. We further discuss the interdependence between pattern matching and metric matching in Section 5.

The density profile vector $V_C$ for a clustering $C$ is formally defined as an ordered tuple

$$V_C=(dens_C(1,1,1),dens_C(1,1,2),\ldots,dens_C(1,1,Q),dens_C(1,2,1),\\ \ldots,dens_C(1,R,Q),dens_C(2,1,1),\ldots,dens_C(K,R,Q)),$$

where $Q$ is the number of bins in each of the $R$ attributes, and $K$ is the number of clusters in $C$.

(2) *The ADCO measure*: After the density profile vectors of two clusterings $C$ and $C'$ are obtained, the degree of similarity between $C$ and $C'$ can be determined by calculating the dot product of the density profile vectors:

$$\mathrm{sim}(C,C')=V_C\cdot V_{C'}.$$

Given a permutation function $\rho$ under which the similarity function $\mathrm{sim}(C,\rho(C'))$ is maximized, an ADCO measure is calculated using a normalization factor ($NF$) corresponding to the maximum achievable similarity of the clusterings: $NF(C,C')=\max[\mathrm{sim}(C,C),\ \mathrm{sim}(C',C')]$. The ADCO$(C,C')$ measure is defined as follows:

$$\mathrm{ADCO}(C,C')=\frac{\mathrm{sim}(C,C')}{NF(C,C')}.$$

The ADCO measure can be transformed to a distance function (we refer interested readers to [16]). However, since the similarity function is inversely related to the distance function, in order to calculate matching in Eq. (1), it suffices to plug in the similarity function $\mathrm{sim}(C,C')$ and change the operator to arg max. This is shown in Eq. (7)

$$\mathcal{M}(C,C')=\arg\min_{\rho}\left(\sum_{k=1}^{K_{min}}\mathcal{L}(C_k,C'_{\rho(k)})\right),\tag{6}$$

$$\mathcal{M}(C,C')=\arg\max_{\rho}\left(\sum_{k=1}^{K_{min}}\sum_{i=1}^{R}\sum_{j=1}^{Q}dens(k,i,j)\times dens(\rho(k),i,j)\right).\tag{7}$$

## 3. Experiment

In the present section, we describe an application of our theoretic framework, described in Section 2, to the ERP metric matching and pattern matching problems. For this application, we used a set of simulated ERP datasets, which were designed to support evaluation of ERP analysis methods [7]. As previously noted, real ERP data arise from superposition of latent scalp-surface electrophysiological patterns, each reflecting the activity of a distinct cortical network that cannot be reconstructed from the scalp-measured data with any certainty. Thus, real ERP data are not appropriate for evaluation of ERP pattern matching. By contrast, simulated ERP data are derived from known source patterns and therefore provide the necessary gold standard for evaluation of our proposed methods.

The datasets for the present study were specifically designed to simulate heterogeneous data from different groups of subjects under different conditions (via distinct simulated brain activity profiles), as well as distinct measurement methods (spatial and temporal metrics) and distinct patterns (reflecting two different pattern decomposition techniques).

### 3.1. Simulated ERP datasets

The raw data for this study consist of 80 simulated event-related potentials (ERPs), in which each ERP comprises simulated measurement data at 150 time samples and 129 electrode channels for a particular subject ($n=40$) and experiment condition ($n=2$). The 40 simulated subjects are randomly divided into two 20-subject groups, SG1 and SG2, each containing 40 ERPs (20 subjects in 2 experimental conditions). Each ERP consists of a superposition of five latent varying spatiotemporal patterns that represent the scalp projections of distinct neuronal groups (dipoles). The varying spatiotemporal patterns modeled inter-subject and inter-conditional differences amongst the ERPs, and consisted of changes to the latency and standard deviation of their dipole activation curves (described below). This ensured that the simulated ERPs would be complex, realistic and yet tractable to the PCA and ICA decompositions.

To create a set of scalp-referenced patterns of neural activity, nine dipoles were located and oriented within a three-shell spherical head model (Fig. 5A–C). The dipole locations and orientations were designed to simulate the topographies of five ERP components (Fig. 5D) commonly seen in studies of visual word recognition. Each dipole was assigned a 600 ms activation consistent with the temporal characteristics of its corresponding ERP. Simulated "scalp-surface" electrode locations were then specified with a 129-channel montage, and a complex matrix of simulated noise was added to mimic known properties of human EEG. Because of volume conduction (overlap in spatial or scalp-topographic activity), as well as overlap in temporal activity, the dipole activations combine to yield a complex spatial and temporal superposition of the five modeled ERP patterns.

Spatiotemporal patterns were extracted from the two datasets, SG1 and SG2, using two techniques: temporal Principal Components Analysis (tPCA) and spatial Independent Components Analysis (sICA), two data decomposition techniques widely used in the ERP research [19]. tPCA decomposes each ERP into a sequence of spatiotemporal patterns, based upon the distribution of the ERP's temporal variance. The superposition of these patterns, in which each captures a topography with a distinct temporal evolution, reconstruct the original ERP. sICA is an analogous ERP decomposition, but is based upon the statistical independence of the ERP's spatiotemporal patterns in the decomposition sequence [20,21]. To quantify the spatiotemporal characteristics of the extracted patterns, two alternative metric sets, m1 and m2, were applied to the two

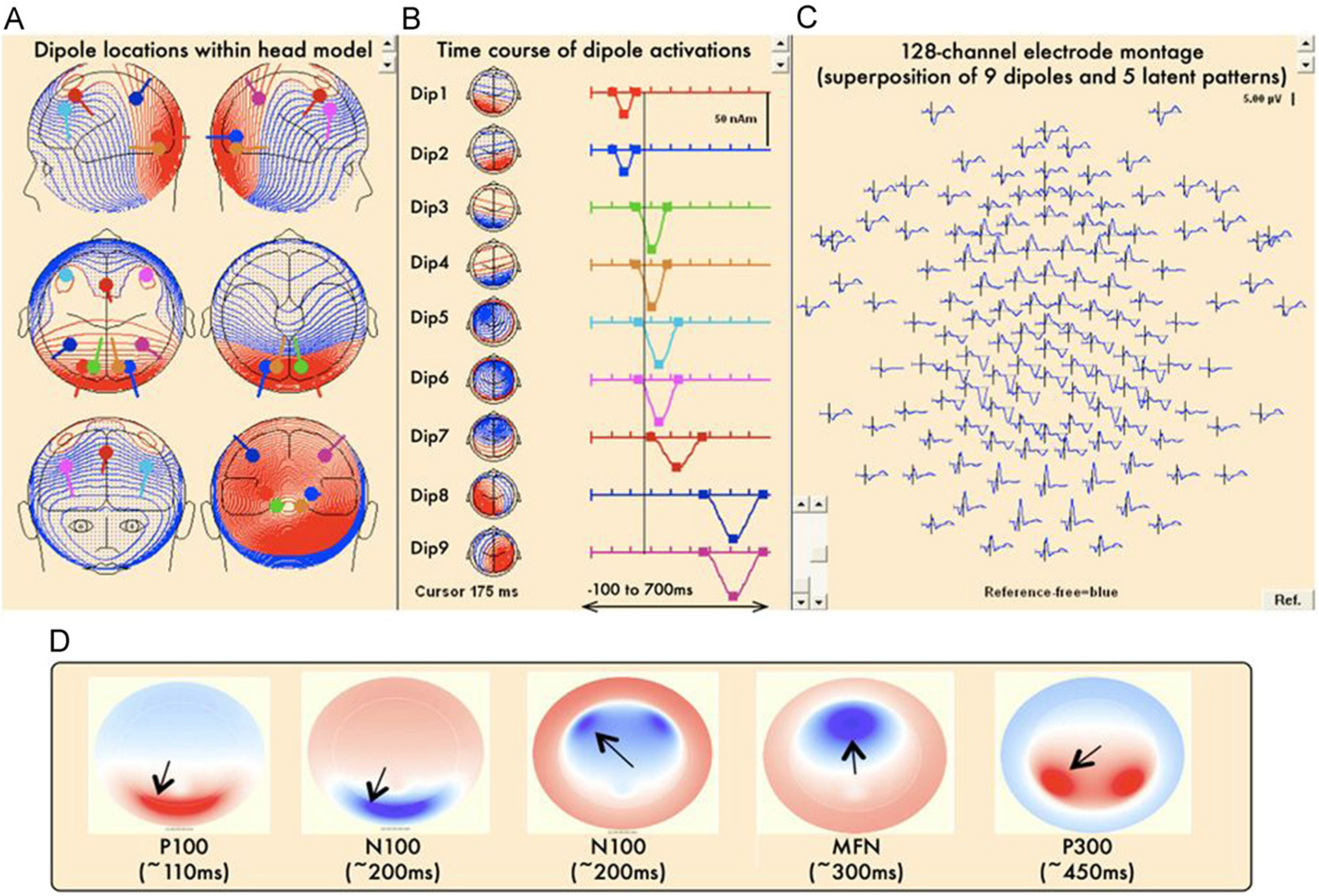


**Fig. 5.** Top (A)–(C), base model for generation of simulated ERP data. Bottom (D), representation of the scalp distribution of the five latent patterns. Approximate peak latency given in parentheses.

tPCA and the two sICA derived datasets. For a complete explanation of these alternative metrics, see Appendix in [7].

In summary, the simulated ERP data generation process yielded a total of eight test datasets, reflecting a 2 (subject groups) ×2 (metric sets) ×2 (decomposition methods) factorial design. The reason for including these three factors (simulated subjects, alternative metrics, and alternative decomposition methods) was to test the robustness of our matching method to different sources of heterogeneity across the different datasets.

### 3.2. Metric matching experiments

The experiment is conducted on the simulated datasets described in Section 3.1. The test cases for the metric matching discovery experiment are derived as follows: each test case contains a source and target dataset. The source dataset is derived from one subject group (SG1 or SG2) characterized with one metric set (m1 or m2) and formulated under one decomposition method (sICA or tPCA). The target dataset is derived from the other subject group, using the alternative metric set and decomposition method. This yields 2 (subject groups) ×2 (metric sets) ×2 (decomposition method)=8 test cases, each of which includes two different datasets, two alternative metric sets and two decomposition methods. In order to test the robustness of the proposed methods, we replicate the datasets for each test case into five copies with different random ordering of the instances, resulting in a total of 40 enriched test cases.

We test our method on each of these test cases. The overall result is presented in Section 4.1. Table 2, for example, shows the detail of one individual result calculated from tPCA-derived data SG1-m1 and SG2-m2. Cell values indicate the Euclidean distance between two point-sequence curves representing two ERP metrics (row header and column header that meet at the cell). Bold cells are the minimum distance assignments found by the Hungarian method. The assignments indicate matchings discovered by our methods. For example, from this table we can derive the following matchings: IN-O1↔IN-LOCC, IN-O2↔IN-ROCC, IN-C3↔IN-LPAR, etc. Note that the orders of the row and column header labels are such that the golden standard matching falls along the diagonal cells. Therefore, we can easily conclude that the precision of matching in this test case is 9/13=69.2% since 4 out of 13 cells are shifted off from the diagonal.

### 3.3. Pattern matching experiments

We design three test cases for pattern matching using different cluster analysis schemes to extract patterns from ERP data. The goal is to simulate different ERP analysis scenarios that leverage varying amounts of domain knowledge. The three cases are as follows:

1. Constrained EM clustering (clustering with known number of clusters).
2. Unconstrained clustering with postprocessing (clustering without a priori specification of the number of clusters, but with expert post-processing).
3. Unconstrained clustering (clustering without specifying number of clusters).

In the first case, the EM algorithm is performed and number of clusters is set to 5, conforming to the number of latent patterns that we modeled when generating the simulated dataset. The cluster labels are conveniently labeled according to the corresponding pattern labels determined by the EM algorithm.

In the second case, our domain expert examines the EM clustering results and collapsed similar clusters or drop outliers based on the distribution of spatial and temporal metrics. Note that this process could be partly or fully automated in principle, but was carried out by manual inspection of the clustering results in this case.

In the third case, we perform the EM algorithm without setting the number of clusters. The clustering result is then aligned to a priori pattern class labels and outliers are dropped.

Experiment results for the three test cases are shown in Section 4.2. We also keep a record of all detailed individual matching results. Table 3, for example, shows an individual matching result calculated from two clusterings derived from datasets SG01_tPCA_m1 and SG02_sICA_m1 in case study 1. Each cell value represents the density profile-based similarity measure. The column and row header elements in the table are cluster labels (aligned to pattern class labels). They are listed in the same order so that the golden standard matching falls along the diagonal cell. The bold cells represent the maximum similarity

**Table 3**
Sample similarity table for two clusterings from dataset *SG01_tPCA_m*1 and *SG02_sICA_m*1. Bold cells represent maximum similarity assignment found by the Hungarian method.

| | MFN | N1 | N3 | P1 | P3 |
|---|---|---|---|---|---|
| MFN | **23892** | 17 874 | 14 419 | 19 338 | 17 008 |
| N1 | 9228 | **10581** | 4590 | 8050 | 6945 |
| N3 | 7642 | 5796 | **6531** | 6336 | 5530 |
| P1 | 13 644 | 10 808 | 7835 | **13120** | 10 379 |
| P3 | 12 961 | 9200 | 7012 | 11 088 | **14196** |

*ADCO*: 0.8751232883731058.

**Table 2**
Sample distance table for two metric sets derived from dataset *SG1_tPCA_m*1 and *SG2_tPCA_m*2. Bold cells represent minimum distance assignment found by the Hungarian method.

| | IN-O1 | IN-O2 | IN-C3 | IN-C4 | IN-T7 | IN-T8 | IN-F7 | IN-F8 | IN-Fp1 | IN-Fp2 | IN-F3 | IN-F4 | TI-max2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IN-LOCC | **2.76** | 2.76 | 8.59 | 8.52 | 9.68 | 10.44 | 11.52 | 11.61 | 11.56 | 11.56 | 7.92 | 7.90 | 12.93 |
| IN-ROCC | 2.75 | **2.75** | 8.58 | 8.47 | 9.69 | 10.47 | 11.55 | 11.64 | 11.60 | 11.60 | 7.91 | 7.86 | 12.95 |
| IN-LPAR | 8.57 | 8.58 | **4.13** | 5.12 | 9.29 | 8.91 | 9.24 | 9.07 | 8.98 | 8.97 | 5.58 | 6.07 | 9.39 |
| IN-RPAR | 7.97 | 7.97 | 3.55 | **4.38** | 8.97 | 8.66 | 9.10 | 8.93 | 8.88 | 8.85 | 4.99 | 5.39 | 9.43 |
| IN-LPTEM | 9.32 | 9.34 | 8.54 | 9.23 | 5.00 | **4.26** | 5.62 | 5.34 | 5.73 | 5.72 | 7.37 | 7.88 | 11.42 |
| IN-RPTEM | 7.81 | 7.81 | 7.66 | 8.05 | **4.18** | 3.84 | 5.61 | 5.39 | 5.85 | 5.78 | 6.24 | 6.56 | 11.28 |
| IN-LATEM | 11.00 | 11.00 | 8.40 | 8.96 | 3.20 | 2.74 | **2.30** | 2.09 | 2.52 | 2.43 | 6.89 | 7.35 | 10.95 |
| IN-RATEM | 11.19 | 11.19 | 8.53 | 9.03 | 3.33 | 2.45 | 2.51 | **2.08** | 2.80 | 2.64 | 6.99 | 7.41 | 11.30 |
| IN-LORB | 9.58 | 9.58 | 6.00 | 6.48 | 4.23 | 4.50 | 3.58 | 3.63 | 3.35 | **3.26** | 4.36 | 4.83 | 10.31 |
| IN-RORB | 11.19 | 11.20 | 8.36 | 8.93 | 3.44 | 3.33 | 2.15 | 2.12 | **2.21** | 2.16 | 6.85 | 7.33 | 10.83 |
| IN-LFRON | 6.72 | 6.71 | 4.05 | 4.01 | 6.30 | 7.10 | 6.91 | 7.06 | 6.76 | 6.71 | **2.74** | 2.20 | 9.99 |
| IN-RFRON | 6.36 | 6.33 | 4.58 | 4.03 | 7.09 | 7.94 | 8.01 | 8.15 | 7.96 | 7.88 | 3.42 | **3.06** | 10.67 |
| TI-max1 | 11.72 | 11.71 | 7.18 | 7.74 | 12.12 | 11.74 | 12.02 | 11.88 | 11.89 | 11.87 | 9.36 | 9.61 | **8.58** |

assignment. The accuracy of matching induced from this assignment is 100% since all highlighted cells fall along the diagonal. We also calculate the ADCO measure (Section 2.3 for this assignment), which can be viewed as a measure that quantifies the degree of confidence of the matching result.

By inspecting the matching results, domain experts can get a sense of the quantitative relationship between two clusters in terms of density profile-based similarity measure. For example, in Table 3, although the assignment selected by the Hungarian method suggests a correct matching between two clusters labeled P1, one can also find that the P1 pattern in one dataset is close to the MFN pattern in another dataset. Such information can be very valuable to domain experts as it provides clues to the nature of the relationships between different ERP patterns.

## 4. Results

### 4.1. Metric matching results

Figs. 6 and 7 illustrate the effect of subsequence reordering and post-processing on point-sequence curves IN-LOCC and IN-O1, which are derived from two tPCA-derived datasets *SG1_tPCA_m*1 and *SG2_tPCA_m*2 with a random ordering of data points. IN-LOCC and IN-O1 are a matching according to the gold standard. As shown in the figure, the two point-sequence curves are manifestly more similar after subsequence reordering and post-processing.

The robustness of the methods is assessed by evaluating the overall performance over the 40 test cases (described in Section 3.2). Table 4 summarizes the result in terms of precision for each test case. The table consists of eight divisions, each of which illustrates the precision measures for the datasets generated by five samples of replication to one of the original eight test schemes with random instance ordering. Since the fact that the precision of matching by making a random guess is almost zero and that the results demonstrate consistent performance on randomly ordered data, the precision of our method appears markedly robust. Combining the matching results in the 40 test cases into an ensemble model by a majority vote of each individual matching, we obtain the ensemble matching result. The overall precision is $11/13 = 84.6$. We also carry out experiments to test the performance of attribute matching based on the segmented statistical representation. The experiment is conducted on the same eight test cases together with a baseline method based on the SemInt [12].

In the segmented statistical representation, an attribute can be represented as $a := \langle t_1, \ldots, t_n \rangle$, where $t_i = \alpha \cdot c_{i1} + \beta \cdot c_{i2}$, $c_{i1}$ and $c_{i2}$ are the mean and standard deviation, respectively, $\alpha$ and $\beta$ are weights, and $n$ is the number of clusters. In the experiment, $\alpha$ and $\beta$ are empirically set to 0.5. In the SemInt method, we configure the algorithm it to extract from each feature value vector two discriminators, i.e., mean and standard deviation. The feature value vector is then projected to a match signature characterized by these discriminators. A neural network is trained based on datasets from the eight test cases with one metric set and tested

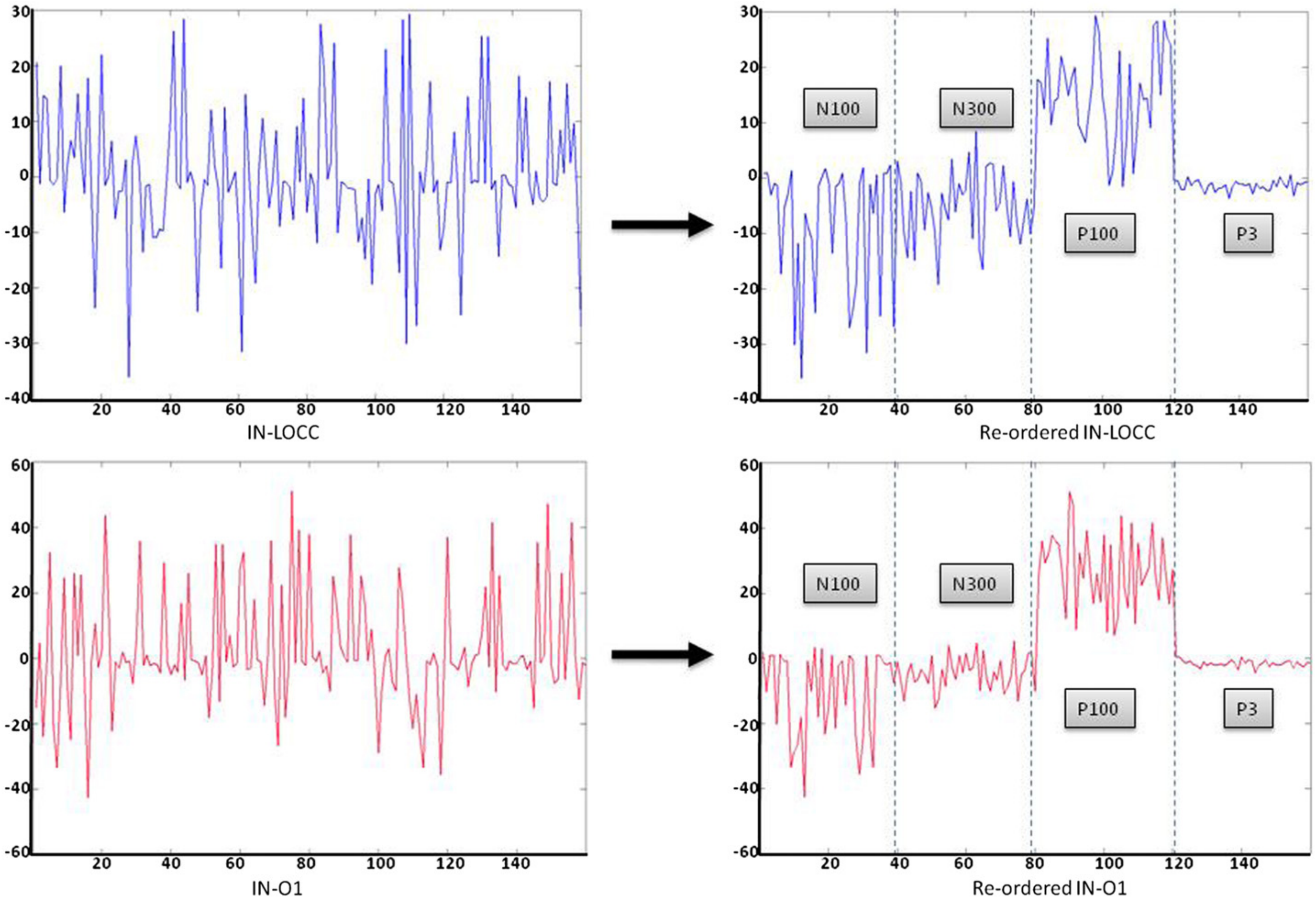


**Fig. 6.** (Left) IN-LOCC and IN-O1 point-sequence curves prior to grouping and reordering. (Right) Labeled curves for metrics IN-O1 and IN-LOCC after grouping/reordering.

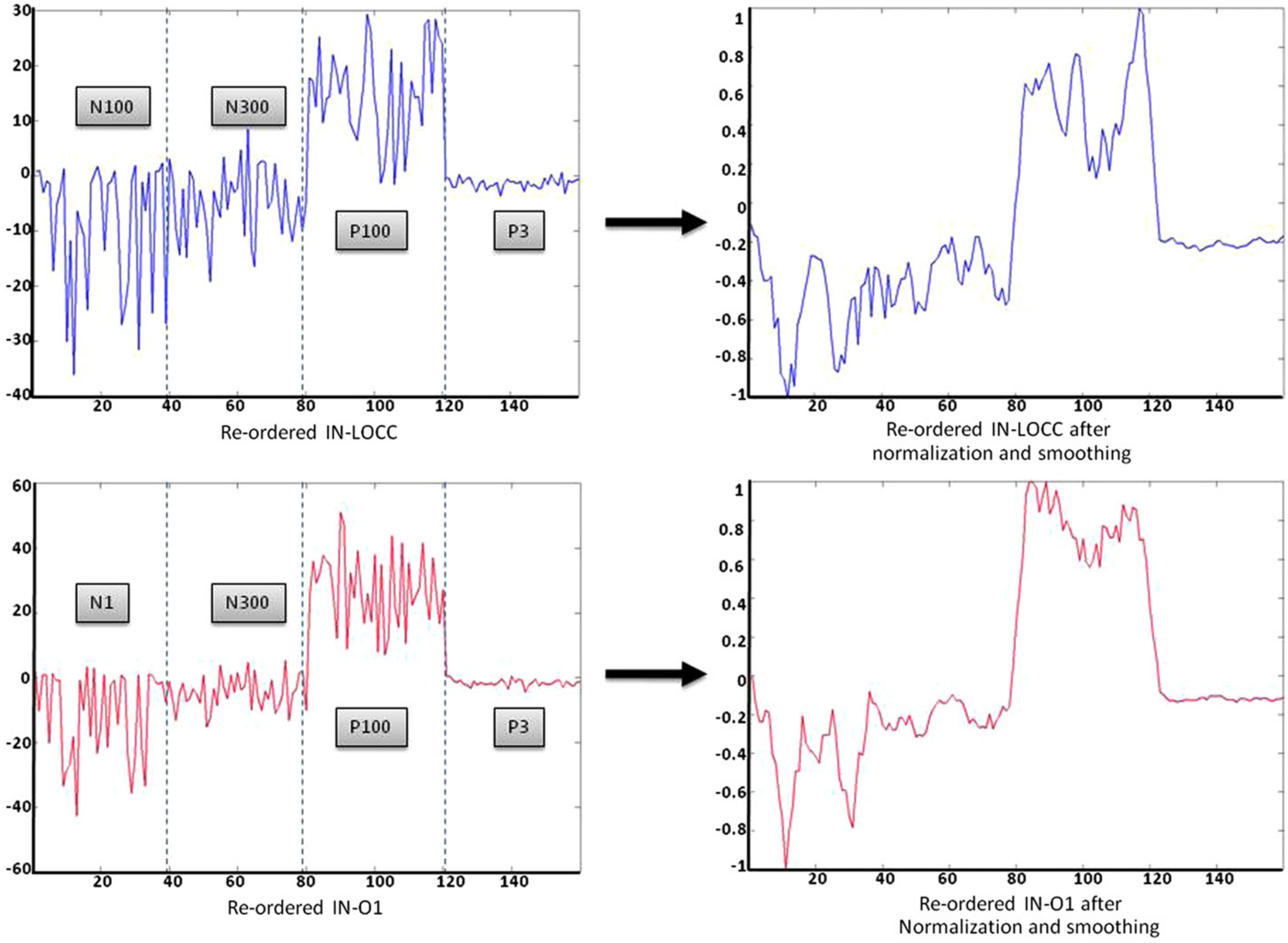


**Fig. 7.** After normalization, smoothing, and interpolation of point-sequence curves in Fig. 6.

**Table 4**
Precision results for 40 test cases.

| (SG1, sICA, m1) vs. (SG2, sICA, m2) | | (SG1, tPCA, m1) vs. (SG2, tPCA, m2) | | (SG1, sICA, m1) vs. (SG2, tPCA, m2) | | (SG1, tPCA, m1) vs. (SG2, sICA, m2) | |
|---|---|---|---|---|---|---|---|
| Input | Precision | Input | Precision | Input | Precision | Input | Precision |
| Sample 1 | 13/13 | Sample 1 | 9/13 | Sample 1 | 13/13 | Sample 1 | 5/13 |
| Sample 2 | 13/13 | Sample 1 | 9/13 | Sample 2 | 13/13 | Sample 1 | 5/13 |
| Sample 3 | 13/13 | Sample 1 | 9/13 | Sample 3 | 13/13 | Sample 1 | 5/13 |
| Sample 4 | 13/13 | Sample 1 | 9/13 | Sample 4 | 13/13 | Sample 1 | 5/13 |
| Sample 5 | 13/13 | Sample 1 | 9/13 | Sample 5 | 13/13 | Sample 1 | 5/13 |
| (SG2, sICA, m1) vs. (SG1, sICA, m2) | | (SG2, tPCA, m1) vs. (SG1, tPCA, m2) | | (SG2, sICA, m1) vs. (SG1, tPCA, m2) | | (SG2, tPCA, m1) vs. (SG1, sICA, m2) | |
| Input | Precision | Input | Precision | Input | Precision | Input | Precision |
| Sample 1 | 9/13 | Sample 1 | 9/13 | Sample 1 | 5/13 | Sample 1 | 7/13 |
| Sample 2 | 9/13 | Sample 1 | 9/13 | Sample 2 | 8/13 | Sample 1 | 7/13 |
| Sample 3 | 9/13 | Sample 1 | 9/13 | Sample 3 | 5/13 | Sample 1 | 7/13 |
| Sample 4 | 9/13 | Sample 1 | 9/13 | Sample 4 | 5/13 | Sample 1 | 7/13 |
| Sample 5 | 9/13 | Sample 1 | 9/13 | Sample 5 | 5/13 | Sample 1 | 7/13 |

on the rest datasets with the alternative metric set to determine the match.

The result for the comparative study is shown in Fig 8. Both the segmented statistical characterization-based method (cen-com) and SemInt are run on the eight test cases. Since they are not sensitive to data ordering, they are not tested on randomly ordered samples. The performance of the sequence similarity-based method (seq-sim) in each test case is shown as the average of five randomly ordered replicate data. The result shows that both seq-sim and cen-com significantly outperform SemInt.

The reason is that both methods take into consideration substructures (i.e., patterns) within the data, allowing for finer-grained comparison of attributes.

The advantage of cen-com to SemInt can be illustrated by a simple conceptual example: Suppose given four attributes, $A_1$, $A_2$ from one dataset and $A'_1$, $A'_2$ from another, all centered around zero (mean=0), and each dataset can be split into two equal-size clusters, the mean values of each cluster in different attributes are as follows: $A_1=[10,-10]$, $A_2=[100,-100]$, $A'_1=[10,-10]$, $A'_2=[100,-100]$, where the numbers in the brackets denote the mean values of corresponding clusters in an attribute. SemInt would use the top attribute-level distribution to represent attributes, in which case, since all attribute-level means are zero, SemInt would not be able to find a matching. On the other hand, using our approach to examine the distributions broken down to the pattern level, it is obvious to see $A_1$ matches to $A'_1$, and $A_2$ matches to $A'_2$. This argument applies to all statistical indicators other than mean as well. Our hypothesis is that similar attributes should exhibit similar per-cluster distributions.

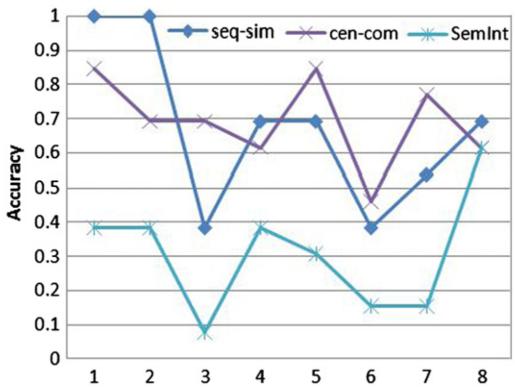


**Fig. 8.** Comparative study on the performance of three metric matching methods. Seq-sim stands for the proposed sequence-similarity-based method; cen-com stands for the segmented central tendency based method; and SemInt is an existing method that uses statistical aggregations to represent numeric attributes.

Note that seq-sim and cen-com yielded similar results. These two methods can be viewed as conceptually similar as they both utilize clustering structure in the data. It is worth noting that cen-com can be further improved by automatic weight assignment through learning. In this regard, it may prove to be more powerful, and more generalizable, than the simple comparison of means and standard deviations.

### 4.2. Pattern matching results

Tables 5–7 show the pattern matching results for the three test cases described in Section 3.3. The column and header elements

**Table 5**
Matrix of ADCO scores between all pair of test datasets in case study 1. Average ADCO is 0.842.

| | SG01_ sICA_m1 | SG01_ sICA_m2 | SG01_ tPCA_m1 | SG01_ tPCA_m2 | SG02_ sICA_m1 | SG02_ sICA_m2 | SG02_ tPCA_m1 | SG02_ tPCA_m2 |
|---|---|---|---|---|---|---|---|---|
| SG01_ sICA_m1 | 1 | 0.907 | 0.826 | 0.871 | 0.855 | 0.912 | 0.755 | 0.886 |
| SG01_ sICA_m2 | 0.907 | 1 | 0.834 | 0.827 | 0.866 | 0.972 | 0.733 | 0.901 |
| SG01_ tPCA_m1 | 0.826 | 0.834 | 1 | 0.803 | 0.881 | 0.811 | 0.77 | 0.855 |
| SG01_ tPCA_m2 | 0.871 | 0.827 | 0.803 | 1 | 0.813 | 0.823 | 0.851 | 0.835 |
| SG02_ sICA_m1 | 0.855 | 0.866 | 0.881 | 0.813 | 1 | 0.855 | 0.75 | 0.919 |
| SG02_ sICA_m2 | 0.912 | 0.972 | 0.811 | 0.823 | 0.855 | 1 | 0.743 | 0.889 |
| SG02_ tPCA_m1 | 0.755 | 0.733 | 0.77 | 0.851 | 0.75 | 0.743 | 1 | 0.752 |
| SG02_ tPCA_m2 | 0.886 | 0.901 | 0.855 | 0.835 | 0.919 | 0.889 | 0.752 | 1 |

**Table 6**
Matrix of ADCO scores between all pair of test datasets in case study 2. Average ADCO is 0.839.

| | SG01_ sICA_m1 | SG01_ sICA_m2 | SG01_ tPCA_m1 | SG01_ tPCA_m2 | SG02_ sICA_m1 | SG02_ sICA_m2 | SG02_ tPCA_m1 | SG02_ tPCA_m2 |
|---|---|---|---|---|---|---|---|---|
| SG01_ sICA_m1 | 1 | 0.909 | 0.834 | 0.873 | 0.855 | 0.911 | 0.756 | 0.883 |
| SG01_ sICA_m2 | 0.909 | 1 | 0.836 | 0.831 | 0.866 | 0.976 | 0.741 | 0.898 |
| SG01_ tPCA_m1 | 0.834 | 0.836 | 1 | 0.817 | 0.875 | 0.819 | 0.797 | 0.854 |
| SG01_ tPCA_m2 | 0.873 | 0.831 | 0.817 | 1 | 0.817 | 0.822 | 0.846 | 0.835 |
| SG02_ sICA_m1 | 0.855 | 0.866 | 0.875 | 0.817 | 1 | 0.856 | 0.753 | 0.924 |
| SG02_ sICA_m2 | 0.911 | 0.976 | 0.819 | 0.822 | 0.856 | 1 | 0.75 | 0.884 |
| SG02_ tPCA_m1 | 0.756 | 0.741 | 0.797 | 0.846 | 0.753 | 0.75 | 1 | 0.76 |
| SG02_ tPCA_m2 | 0.883 | 0.898 | 0.854 | 0.835 | 0.924 | 0.884 | 0.76 | 1 |

**Table 7**
Matrix of ADCO scores between all pair of test datasets in case study 3. 4 out of 28 matchings are incorrect (bold cells). Average ADCO is 0.605.

| | SG01_ sICA_m1 | SG01_ sICA_m2 | SG01_ tPCA_m1 | SG01_ tPCA_m2 | SG02_ sICA_m1 | SG02_ sICA_m2 | SG02_ tPCA_m1 | SG02_ tPCA_m2 |
|---|---|---|---|---|---|---|---|---|
| SG01_ sICA_m1 | 1 | 0.576 | 0.669 | 0.592 | 0.627 | 0.754 | 0.694 | 0.462 |
| SG01_ sICA_m2 | 0.576 | 1 | 0.486 | 0.528 | 0.648 | 0.656 | 0.655 | 0.4 |
| SG01_ tPCA_m1 | 0.669 | 0.486 | 1 | 0.824 | **0.472** | 0.63 | 0.642 | 0.598 |
| SG01_ tPCA_m2 | 0.592 | 0.528 | 0.824 | 1 | 0.613 | 0.708 | **0.588** | 0.665 |
| SG02_ sICA_m1 | 0.627 | 0.648 | **0.472** | 0.613 | 1 | 0.719 | **0.653** | 0.425 |
| SG02_ sICA_m2 | 0.754 | 0.656 | 0.63 | 0.708 | 0.719 | 1 | **0.661** | 0.558 |
| SG02_ tPCA_m1 | 0.694 | 0.655 | 0.642 | **0.588** | **0.653** | **0.661** | 1 | 0.423 |
| SG02_ tPCA_m2 | 0.462 | 0.4 | 0.598 | 0.665 | 0.425 | 0.558 | 0.423 | 1 |

in each table represent the eight test ERP data files. Every cell element represents the ADCO score for cluster matching between the two data files represented by the corresponding header and column elements. In all the three test cases, pattern matching was conducted on all 64 pairs of the test data files. Among the 64 pairs, 8 were trivial (self matching). Further, due to the symmetric property of matching, only a half of the remaining 56 need to be computed, leaving a total of 28 nontrivial cases for examination. In other words, Tables 5–7 are symmetric matrices, with values on diagonal cells being 1.

We achieve 100% correct matching in all tests for cases 1 and 2. It is worth noting, however, that the average ADCO score for case 1 is slightly higher than case 2. This conforms to our intuition that the more domain knowledge we leverage during the cluster analysis process, the more accurate the resulted clusters are.

The third test case is the only one where the proposed matching method outputs incorrect result (bold cells in Table 7). This is due to the fact that we employ the least amount of domain knowledge in case 3. Unconstrained EM clustering without specifying the number of clusters causes the splitting of observations of a single pattern into two or more clusters. This phenomenon leads to a many-to-many matching between clusters, which cannot be effectively handled by the density profile-based matching method. Therefore, the matching result is the worst among the three cases (however it can still be considered a good result in terms of error rate: 4 out of 28 matchings are incorrect). Note that the assignment of observations to clusters and the corresponding splitting of observation is a function of the metrics used to generate the dimensions and axis orientations of the multidimensional attribute space. Observations that are close in L2 norm along one dimension of a spatiotemporal attribute space may be farther away in L2 norm on a separate dimension of an alternate attribute space instantiated by an alternate metric set. This is consistent with mathematical topology, in which mathematical objects that are close in one topology, or one generalized measure of distance, can be far apart in another topology, or an alternate measure of distance.

## 5. Discussion

In this section we describe how our proposed methods compare with existing methods. We then summarize the contributions, as well as the assumptions and limitations, of the current study and outline some directions for future work.

*Metric and pattern matching*: As noted in Section 1.3, the problem of finding correspondences among alternative metrics can be viewed as a schema matching problem. We have proposed a method that relies on alignment of point-sequence similarity curves and examines the grouping structure of metric values through cluster analysis. Consistent with our hypotheses, we found that our method outperformed a well-known attribute matching method, SemInt. We further showed that our method performed as well as a method based on segmented central tendency. In addition, the proposed method does not require overlapping instances in different ERP datasets. Matching systems such as the iMAP [22] method that builds joint paths between two tables through which data instances can be cross-referenced perform poorly on ERP datasets because overlapping data instances from different ERP datasets are simply non-existent.

It is important to note the role of a priori knowledge (i.e., top-down methods) in our metric mapping approach: subsequences (clusters) were labeled by domain experts, prior to subsequence ordering. This was a necessary step, to ensure that metric values were compared across datasets for like patterns. As a result, we were able to calculate distance between point-sequence curves in a principled way. In future work, this labeling will be done completely automatically, through application of our ontology-based classification and labeling workflow [4–7].

In addition to the metric matching problem, our method handles a more profound challenge for ERP data sharing and integration: ERP pattern matching. This challenge requires that we discover correspondences among ERP patterns from datasets with non-overlapping observations (i.e., different study participants). For this reason, we used cluster comparison methods, which do not assume overlap in cluster membership across datasets. By contrast, most previous methods have relied on evaluation of cluster membership [13–15]. A further advantage of our method is that it takes into account the multidimensional nature of ERP data: ERP patterns are characterized by several attributes, such as time course (e.g., early or late), polarity (positive or negative), and scalp topography. We therefore chose a clustering similarity index known as ADCO (Attribute Distribution Clustering Orthogonality [16]) to represent clusters as density profiles. The ADCO measure can determine the similarity between clusters based on the distribution of data points along each attribute.

*Future work*: The present study has provided important evidence on the use of matching methods to address heterogeneities in ERP datasets from different studies of human brain function. An important next step is to extend our methods beyond the current study, where we have made several simplifying assumptions.

First, in formulating the metric and pattern matching problems, we have assumed one-to-one correspondences between alternative sets of metrics and between sets of patterns from different datasets. These assumptions may not be met in real-world applications. To address this limitation, we propose to use unconstrained EM clustering in future studies. In unconstrained EM, several clusters in one dataset may correspond to a single latent pattern in another dataset [7]. This can result in many-to-many correspondences, and thus requires that we relax the one-to-one matching constraint.

Second, the density profile (pattern matching) method requires that two datasets contain the same or similar underlying clusters (i.e., the same latent patterns). The method allows that these patterns might be characterized by distinct spatial and temporal metrics. At the same time, it assumes that these metrics are, or can be, aligned. Thus, our method presupposes that the metric matching problem has – in one way or another – already been solved. Conversely, alignment of sub-sequences (metric matching) requires that the target and source datasets contain the same subsequences (i.e., patterns). In other words, metric and pattern matching are interdependent problems. In future work, we will address variability across ERP metrics and ERP patterns simultaneously. That is, we will address the three-dimensional assignment problem. If we succeed, this could lead to a fully automatic, data-driven procedure for ERP integration.

Finally, the simulated ERP data used in the present study were carefully designed to mimic many, but not all, features of real ERP data. In particular, we minimized variability in latency and spatial distribution of patterns across the datasets so that the data decomposition and clustering of patterns would be tractable and relatively straightforward to interpret. In future work, we will carry out additional tests on simulated datasets that are progressively more complex along these dimensions. This will allow us to test the robustness of our proposed matching methods in a systematic way. After we have evaluated our methods across a wider range of simulated data, we will apply these methods to real ERP datasets, such as those that have been collected, analyzed, and stored in our NEMO ERP ontology database [5,6,8]. Again, however, it is important to note that ERP data are extremely complex and – most importantly – that they contain unknown mixtures of patterns. Therefore, it is only with the use of simulated data that we can test the robustness and validity of our methods with any confidence.

## 6. Conclusion

In this paper, we have described a data-driven solution for two key challenges in the sharing and integration of electrophysiological (brainwave) data. The first challenge, *ERP metric matching*, involves discovery of correspondences among distinct summary features (‘‘metrics’’) that are used to characterize datasets that have been collected and analyzed in different research labs. The second challenge, *ERP pattern matching*, involves discovery of matchings between spatiotemporal patterns or ‘‘components’’ of ERPs.

We have treated both problems within a unified framework that comprises multiple methods for assignment (matching). The utility of this framework has been demonstrated in a series of experiments using ERP datasets that were designed to simulate heterogeneities arising from three sources: (a) different groups of subjects, (b) different measurement methods, and (c) different spatiotemporal patterns (reflecting different pattern analysis techniques). Unlike real ERP data, the simulated data were derived from known source patterns, providing a gold standard for evaluation. We have shown that our method outperforms well-known existing methods, such as SemInt, because it utilizes cluster-based structure and thus achieves finer-grained representation of ERP attributes. We have further discussed the importance of this work in the broader context of ERP data sharing, analysis, and integration: While top-down (ontological) methods have played a key role in our project, extensions of the present work could lead a fully automatic, data-driven framework for ERP meta-analysis, enabling major breakthroughs in the science of human brain function.

## Acknowledgment

This work was supported by the NIH/NIBIB with Grant no. R01EB007684.

## References

[1] G. Gratton, M.G.H. Coles, E. Donchin, A procedure for using multi-electrode information in the analysis of components of the event-related potential: vector filter, Psychophysiology 26 (1989) 222–232.

[2] K.M. Spencer, J. Dien, E. Donchin, A componential analysis of the ERP elicited by novel events using a dense electrode array, Psychophysiology 36 (1999) 409–414.

[3] D. Dou, G. Frishkoff, J. Rong, R. Frank, A. Malony, D. Tucker, Development of NeuroElectroMagnetic ontologies (NEMO): a framework for mining brainwave ontologies, in: Proceedings of the 13th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining, KDD'07, ACM, New York, NY, USA, 2007, pp. 270–279.

[4] G. Frishkoff, P. LePendu, R. Frank, H. Liu, D. Dou, Development of neural electromagnetic ontologies (NEMO): ontology-based tools for representation and integration of event-related brain potentials, in: Proceedings of the International Conference on Biomedical Ontology, ICBO, 2009, pp. 31–34.

[5] G.A. Frishkoff, R.M. Frank, P. LePendu, Ontology-based analysis of event-related potentials, in: Proceedings of the International Conference on Biomedical Ontology, ICBO, 2011, pp. 55–62.

[6] G.A. Frishkoff, R.M. Frank, J. Sydes, K. Mueller, A.D. Malony, Minimal information for neural electromagnetic ontologies (MI-NEMO): a standards-compliant workflow for analysis and integration of human EEG, Stand. Genomic Sci. (SIGS), in press.

[7] G.A. Frishkoff, R.M. Frank, J. Rong, D. Dou, J. Dien, L.K. Halderman, A framework to support automated classification and labeling of brain electromagnetic patterns, Comput. Intell. Neurosci. (CIN) 7 (2007) 1–13. Special Issue, EEG/MEG Analysis and Signal Processing 7 (2007) 1–13.

[8] P. Lependu, D. Dou, G.A. Frishkoff, J. Rong, Ontology database: a new method for semantic modeling and an application to brainwave data, in: Proceedings of the 20th International Conference on Scientific and Statistical Database Management, SSDBM'08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 313–330.

[9] E. Donchin, E. Heffley, Multivariate analysis of event-related potential data: a tutorial review, in: D. Otto (Ed.), Multidisciplinary Perspectives in Event-Related Brain Potential Research, U.S. Government Printing Office, Washington, DC, USA, 1978, pp. 555–572.

[10] T.W. Picton, S. Bentin, P. Berg, Guidelines for using human event-related potentials to study cognition: recording standards and publication criteria, Psychophysiology 37 (2000) 127–152.

[11] H. Liu, G. Frishkoff, R. Frank, D. Dou, Ontology-based mining of brainwaves: a sequence similarity technique for mapping alternative descriptions of patterns in event related potentials (ERP) data, in: Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Springer, 2010, pp. 43–54.

[12] W.S. Li, C. Clifton, SEMINT: a tool for identifying attribute correspondences in heterogeneous databases using neural networks, Data Knowl. Eng. 33 (2000) 49–84.

[13] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (1971) 846–850.

[14] L. Hamers, Y. Hemeryck, G. Herweyers, M. Janssen, H. Keters, R. Rousseau, A. Vanhoutte, Similarity measures in scientometric research: the Jaccard index versus Salton's cosine formula, Inf. Process. Manage. 25 (1989) 315–318.

[15] A.L. Fred, A.K. Jain, Robust data clustering, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2003, p. 128.

[16] E. Bae, J. Bailey, G. Dong, A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings, Data Min. Knowl. Discovery 21 (2010) 427–471.

[17] H.W. Kuhn, The Hungarian method for the assignment problem, Nav. Res. Logistic Q. 2 (1955) 83–97.

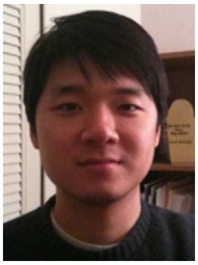
[18] E. Keogh, S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'02, ACM, New York, NY, USA, 2002, pp. 102–111.

[19] J. Dien, The ERP PCA toolkit: an open source program for advanced statistical analysis of event-related potential data, J. Neurosci. Methods 187 (2010) 138–145.

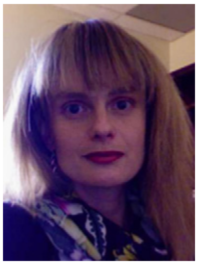
[20] A.E. Hendrickson, P.O. White, Promax: a quick method for rotation to oblique simple structure, Br. J. Stat. Psychol. 17 (1964) 65–70.

[21] A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Comput. 7 (1995) 1129–1159.

[22] R. Dhamankar, Y. Lee, A. Doan, A. Halevy, P. Domingos, iMAP: discovering complex semantic matches between database schemas, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD'04, ACM, New York, NY, USA, 2004, pp. 383–394.

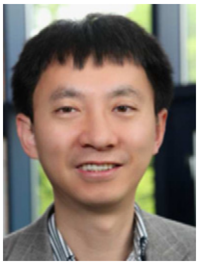


**Haishan Liu** is a Ph.D. candidate in the Computer and Information Science Department, University of Oregon, Eugene, OR. He received Bachelor degree from Shanghai Jiao Tong University, Shanghai, China, 2006. His research interests include data mining, data integration, and the Semantic Web.



**Gwen Frishkoff** is an Assistant Professor in Psychology and Neuroscience at Georgia State University and an adjunct member of the NeuroInformatics Center (NIC) at the University of Oregon. Dr. Frishkoff studies the neurocognition of language and cross-modal (EEG/MEG/fMRI) integration of cognitive neuroscience data. She is co-Investigator and scientific lead on a multi-site NIH-funded project, Neural ElectroMagnetic Ontologies (NEMO), which aims to develop methods for sharing and integration of EEG/ERP data and to use these methods for analysis of ERP responses during language comprehension.

**Robert Frank** has a B.S., Electrical Engineering, New Jersey Institute of Technology, Newark, 1987 and M.S., Applied Mathematics, University of Central Florida, Orlando, 1992, and is currently a Senior Data Analyst with the Neuroinformatics Center, University of Oregon, Eugene, OR.



**Dejing Dou** is an Associate Professor in the Computer and Information Science Department at the University of Oregon and leads the Advanced Integration and Mining (AIM) Lab. He received his bachelor degree from Tsinghua University, China in 1996 and his Ph.D. degree from Yale University in 2004. His research areas include ontologies, data integration, data mining, biomedical and health informatics, and the Semantic Web. He has published more than 40 research papers, some of which appear in prestigious conferences and journals like KDD, ICDM, SDM, CIKM, ISWC, ODBASE, JIIS and JoDS. His KDD'07 paper was nominated for the best research paper award. In addition to serving on numerous program committees, he has been invited as panelist by the NSF several times, and as an expert for grant review by the Netherlands Organization for Scientific Research (NWO). He is on the Editorial Board of Journal of Data Semantics. He has received over $3 million PI or co-PI research grants from the NSF and the NIH.