BrainEx: Interactive Visual Exploration and Discovery of Sequence Similarity in Brain Signals

ALICIA HOWELL-MUNSON, Worcester Polytechnic Institute, USA
CHRISTOPHER MICEK, Worcester Polytechnic Institute, USA
ZIHENG LI, Columbia University, USA
MICHAEL CLEMENTS, Worcester Polytechnic Institute, USA
ANDREW C. NOLAN, Worcester Polytechnic Institute, USA
JACKSON POWELL, Worcester Polytechnic Institute, USA
ERIN SOLOVEY, Worcester Polytechnic Institute, USA
RODICA NEAMTU, Worcester Polytechnic Institute, USA

Technology advances and lower equipment costs are enabling non-invasive, convenient recording of brain data outside of clinical settings in more real-world environments, and by non-experts. Despite the growing interest in and availability of brain signal datasets, most analytical tools are made for experts in the specific device technology, and have rigid constraints on the type of analysis available. We developed *BrainEx* to support interactive exploration and discovery within brain signals datasets. *BrainEx* takes advantage of algorithms that enable fast exploration of complex, large collections of time series data, while being easy to use and learn. This system enables researchers to perform similarity search, explore feature data and natural clustering, and select sequences of interest for future searches and exploration, while also maintaining the usability of a visual tool. In addition to describing the distributed architecture and visual design for *BrainEx*, this paper reports on a benchmark experiment showing that it outperforms other existing systems for similarity search. Additionally, we report on a preliminary user study in which domain experts used the visual exploration interface and affirmed that it meets the requirements. Finally, it presents a case study using *BrainEx* to explore real-world, domain-relevant data.

CCS Concepts: • **Human-centered computing → Interactive systems and tools**; **Visualization systems and tools**.

Additional Key Words and Phrases: interactive visualization, sequence similarity search, clustering, time series, fNIRS, brain signals

ACM Reference Format:
Alicia Howell-Munson, Christopher Micek, Ziheng Li, Michael Clements, Andrew C. Nolan, Jackson Powell, Erin Solovey, and Rodica Neamtu. 2022. BrainEx: Interactive Visual Exploration and Discovery of Sequence Similarity in Brain Signals. *Proc. ACM Hum.-Comput. Interact.* 6, EICS, Article 162 (June 2022), 41 pages. https://doi.org/10.1145/3534516

Authors' addresses: Alicia Howell-Munson, anhowellmunson@wpi.edu, Worcester Polytechnic Institute, Worcester, Massachusetts, USA; Christopher Micek, cjmicek@wpi.edu, Worcester Polytechnic Institute, Worcester, Massachusetts, USA; Ziheng Li, Columbia University, New York, New York, USA; Michael Clements, Worcester Polytechnic Institute, Worcester, Massachusetts, USA, 01609; Andrew C. Nolan, Worcester Polytechnic Institute, Worcester, Massachusetts, USA, 01609; Jackson Powell, Worcester Polytechnic Institute, Worcester, Massachusetts, USA, 01609; Erin Solovey, esolovey@wpi.edu, Worcester Polytechnic Institute, Worcester, Massachusetts, USA, 01609; Rodica Neamtu, rneamtu@wpi.edu, Worcester Polytechnic Institute, Worcester, Massachusetts, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
2573-0142/2022/6-ART162 $15.00
https://doi.org/10.1145/3534516

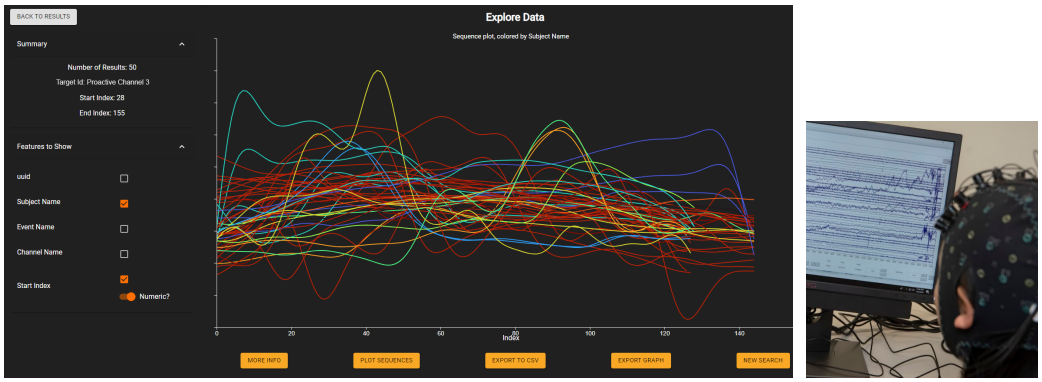

Fig. 1. BrainEx is a web-based visual analytic tool, designed for exploring sequence similarity and clusters within brain signals. On the left, we see 50 similar sequences with color used to encode metadata about the search results, and on the right is a person wearing a functional near-infrared spectroscopy brain sensing cap.

1 INTRODUCTION

Recent innovations and declining costs for non-invasive brain monitoring technologies are paving the way for future innovations in brain-computer interfaces, clinical applications, and intelligent systems that adapt to changes in an individual's dynamic cognitive state [23, 63]. While existing tools help with brain data acquisition and signal processing, most are geared toward biomedical engineers, scientists or clinicians who mainly analyze the sensor data in highly controlled and specific contexts and are experts in the underlying device technology. Their well-established practices come from traditional neuroimaging and neuroscience studies where data is collected with highly controlled timing, often while doing strictly controlled tasks, and the data is analyzed by the same research team that collected the data. While these practices are key to many of the recent discoveries about the brain, the same experimental and analytical designs do not work as well in more real-world contexts and when there is wider dissemination of datasets.

In these contexts, exploration can be valuable for gaining familiarity with the real world data and finding brain signal patterns that could indicate a common cognitive state to study further. For example, researchers may be interested in finding signature signal patterns that indicate that a driver is distracted, or a student is focused. Or they may want to scan a dataset looking for patterns that occur frequently to identify events that may have something in common within and across experiments. Researchers may also be interested in finding events or tasks that lead to similar brain signals, even if they had not been associated together prior to the data collection (e.g. doing a particular math problem or detouring while navigating). These exploratory steps could inform future confirmatory studies. Taking a data driven approach to find similar patterns within and across datasets has been difficult because the brain signal analysis tools available have not been designed to support this type of exploration, and do not leverage advances in time series data mining in a broader sense.

In this paper, we introduce *BrainEx*, a web-based, brain data analytics platform for visual exploration and discovery of sequence similarity. Its core design philosophy is *exploration at every stage. BrainEx* builds on data mining approaches for interactively finding similar sequences in large datasets, and integrates them into a workflow specifically designed for brain signals. We focus on signals from functional near infrared spectroscopy (fNIRS), a non-invasive neuroimaging tool [14, 69] that has been used to measure cognitive states in real-time while participants complete

computer-based tasks [2]. *BrainEx* preprocesses time series datasets and finds similarity structures within. From there, it allows researchers to search through their dataset interactively. It exposes the contents and metadata of each dataset, using its underlying clustering to give a sense of the overall distribution of data and patterns. It also provides interactive searching, which allows analysts to quickly retrieve potentially hundreds of sequences of interest and peruse their associated metadata. Rather than provide this information purely through numeric output, *BrainEx* provides visualizations of the relationships it describes. This way, *BrainEx* is designed not simply for statistical analysis, but broadly for empowering researchers to better understand their data, and to explore it in search of meaningful relationships for further study.

The main contributions of this work showcase both the performance and the versatility of *BrainEx*, as well as its potential impact on research using fNIRS and other brain signals. These are described below:

- *BrainEx* enables researchers to perform expansive exploration of brain signal datasets through its interactive visual interface. Researchers can use *cluster exploration* to discover patterns within the features of their dataset and then choose unique queries to find the top-k similar matches. These data-driven explorations can be customized using multiple elastic distances, which may reveal insights that would be missed by the use of one single distance.
- An experimental evaluation shows that *BrainEx* is at least 10 times faster than state-of-the-art competitors for diverse similarity-based operations. *BrainEx* consistently achieves over 99% accuracy, outperforming Piecewise Aggregate Approximation (PAA) and Symbolic Aggregate Approximation (SAX), and a query time of under 10 seconds, regardless of dataset size.
- A user study conducted with experts in data visualization, neuroscience, and human-computer interaction reveals *BrainEx*'s effectiveness at achieving five functional goals: similarity search, feature distribution exploration, cluster exploration, integration between different components of *BrainEx*, and accessibility to researchers from different backgrounds. The positive feedback shows promisee for advancing the field of brain-computer interaction.
- The case study using real-world study data with fNIRS showcases one way that *BrainEx* can be used to explore datasets and uncover relationships between cognitive states, task events, and brain regions of interest.

2 BACKGROUND

The *BrainEx* system brings together research on brain sensing with research on time series data mining and visualization to address challenges in brain signal analytics, with a focus on functional near-infrared spectroscopy (fNIRS) used in HCI settings. This section provides background on fNIRS, analytics tools for fNIRS, as well as time series similarity search and clustering.

2.1 Functional Near-Infrared Spectroscopy (fNIRS)

Functional near-infrared spectroscopy (fNIRS) is a noninvasive form of neuroimaging that provides time series data about cortical hemodynamics, which is correlated with brain activity [54], using near-infrared light. fNIRS relies on the fact that infrared light can penetrate human skin and is absorbed in different amounts dependent on the oxygenation of the blood. An fNIRS cap (Figure 1) contains multiple fNIRS light sources and light detectors on it, with each source-detector pair forming a *channel* of brain data. These measurements allow researchers to compare activity in different areas in the brain at the same time [23]. fNIRS is a useful tool for researchers due to its accurate, non-invasive, and portable properties. fNIRS research is a growing field within neuroscience [23]; in 2020, approximately 500 papers were published on the subject [29]. In addition, there has been a

growing number of HCI publications that use fNIRS brain signals [3, 4, 18, 34, 42–44, 46, 55, 71, 74]. [43].

The data generated from an fNIRS study typically consists of multivariate time series, with one time series from each channel in the fNIRS cap (e.g. [71]). These time series represent the oxygenated and deoxygenated hemoglobin in the location where the channel is placed on the head. In addition to the fNIRS data itself, a dataset usually includes additional metadata that describes the sensor locations, the study participant identifier, the activity or events that occur during the study, among other things. These characteristics are similar to other brain sensing modalities, such as electroencephalography (EEG), as well as physiological sensing channels. Only the shapes of the signals and the sampling rate would differ, depending on the sensing modality.

From these multivariate time series and the associated metadata, researchers typically search for patterns in the brain data that indicate a particular cognitive or emotional state. This can be done by using statistical methods to look for significant differences between two sets of labeled sequences (e.g. distracted vs. focused). These labels would come from the experiment design where particular states are elicited in a controlled way and then marked or labeled in the data. Machine learning approaches are also common where labeled data is used to build a classifier for future unknown data. Both statistical and machine learning approaches can be difficult when brain signal sequences are of different lengths or different scales, but workarounds exist.

2.2 Analytic Tools for fNIRS

With the growing field, specialized tools have been developed to aid researchers in the analysis of fNIRS brain data [26, 30, 32, 41, 60, 66, 75] and each device typically comes with some basic analysis software (e.g. NIRx NIRSLab). These tools generally support the calculation of oxygenated and deoxygenated hemoglobin, as well as various filtering and signal processing techniques. Visualization of co-variance and activation on a 3D model of the human brain is also common [30, 32, 75]. However, many of these tools also expect a specific experimental design so that statistical models can be built. For example, HomER [32] is a MATLAB-based graphical user interface program that supports approaches such as general linear modeling, but assumes that stimuli have precisely timed onsets and are all of the same length. This is typical for neuroscience experiments, but too constraining for many real-world applications. NIRS-KIT allows researchers to analyze resting-state fNIRS data in addition to the task-driven data which HomER and POTATo support [30]. Unlike many of these offline analysis tools, Turbo Satori [41] was designed to enable real-time visualization and classification of brain signals.

While all of these tools are valuable and widely used in particular use cases, they are designed for researchers with particular expertise. In addition, none of them are designed with visual exploration as a priority, as has been done in other medical fields [10–12]. There are few tools available for gaining a broad sense of a dataset, or that enable browsing the relationships and structures within datasets. There are even fewer methods and tools that specifically support functional near-infrared spectroscopy. Researchers need better tools for interactively searching through brain signal data and exploring the search results.

Facilitating brain data exploration is particularly critical today as there has been a recent push to publish and share datasets and to enable more robust analysis across larger datasets that are collected in diverse contexts. This means that researchers may look at brain signal data that other researchers collected, and that they are not necessarily familiar with. As more fNIRS data becomes available, there may be valuable insights to be found by leveraging advances in time series data mining, and for building on work in other domains with time series data. To date, finding insights in heterogeneous datasets from different sources and studies has been difficult.

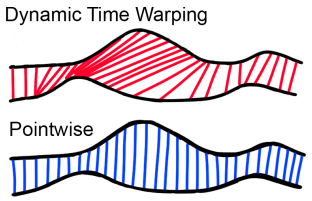

Fig. 2. Comparison of how DTW and point wise similarity matching occurs. DTW (on top) allows for a one-to-many mapping, as seen by points from the top sequence all mapping to a single area on the bottom sequence that occurs earlier. Pointwise (on bottom) only allows for a one-to-one mapping of points that occur at the same time in both sequences.

2.3 Exploring Similar Sequences in Time Series

Finding similar sequences of fNIRS data is an essential operation for identifying brain signals that might be indicative of key cognitive or emotional states. Similarity search also plays a prominent role in many other applications; search engines and weather forecasts rely on finding the most similar sequences given some query. Distance-based similarity search is a key analytical method for these datasets, which often span tens of thousands of time points at minimum [51]. However, this is far from trivial, especially when considering sequences of different lengths and with temporal misalignments.

Pointwise distance metrics, such as Euclidean or Manhattan, are easy to compute, but require a one-to-one match which necessitates the compared time series to be of the same length. *Elastic distance metrics* warp the points into a one-to-many match. One data point in the first sequence can be mapped to more than one point in the second sequence. The compared time series sequences do not need to have the same length, which makes elastic distance metrics more flexible than point-wise comparisons as they are capable of focusing more on the *shape* of sequences than their *values* [19].

One of the most widely used elastic distances is Dynamic Time Warping (DTW) [5, 58, 61], which warps Euclidean distance by compressing and expanding the time axis, allowing multiple matches to the same point. A comparison between DTW and pointwise matching is shown in Figure 2. In practice, DTW is suitable for exploring a wide array of datasets due to its flexibility and high accuracy [52]. DTW has become increasingly popular due to its expressiveness when applied to RNA expression data in bioinformatics [1] and ECG pattern matching in medicine [9]. Recently, there has been increasing interest in using DTW to match temporally misaligned sequences in brain data [15, 20, 45]. DTW has also been integrated in statistical programming languages such as R [24] and Python through *fastdtw* and *pyts* [22, 68].

However, these strengths are overshadowed by its algorithmic complexity: computation time and required memory grow quadratically in relation to the size of the input sequences [36, 48]. This makes it impractical for very large datasets, such as fNIRS and other neurological data. These shortcomings are compounded by the lack of a proven triangle inequality, making it hard to scale. Despite these challenges, thousands of research works in the past twenty years have focused on making DTW the tool of choice for key operations, such as similarity search and clustering. Countless modifications of the classic DTW have been proposed to optimize its performance by indexing, preprocessing, caching, and other optimizations [21, 38, 57, 70, 76].

DTW is a Euclidean-based approach, but it has been established that application domains often need a variety of distance measures to solve their specific problems [13]. A single distance metric may "collapse" a group of points together in a way that distorts the similarity and causes a misclassification. Previous research has shown that different distance metrics will collapse the

groups differently and reduce the number of misclassifications [50]. For example, compound classification in chemistry can use Minkowski distance to select relevant chemical descriptors [35], while image retrieval reflecting human visual perception utilizes Manhattan [64, 67] and Mahalanobis distances [62]. It is therefore necessary for any distance-based exploratory tools to include multiple distance metrics.

The examples above all use pointwise distances, failing to perform well when sequences are of different lengths or have temporal misalignments. Neamtu et al. [52] extended warping abilities to diverse point-wise distances by designing a universal alignment tool, called Generalized Dynamic Time Warping (GDTW). This framework is flexible due to its preprocessing steps. It uses multiple cheap-to-compute point-wise warped distances such as Euclidean, Manhattan and Minkowski to cluster all the sequences during a preprocessing step [50]. Then data is clustered in a reduced subset of the original dataset based on specific sequences to explore appropriate warped counterparts, namely DTW, warped Manhattan, and warped Minkowski [52]. This flexible and generalized approach for efficient similarity search has promise for interactive exploration of fNIRS signals. However, there are no existing tools that support this for general use by non-experts.

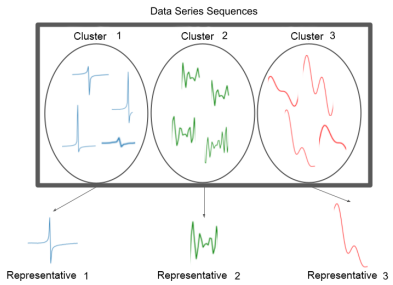

Fig. 3. Representations and groupings derived by GDTW. The colors of the sequences correspond to clusters of similar sequences and their respective representatives.

2.4 Efficient Sequence Similarity Search Using Multiple Warped Distances

BrainEx rests on the foundation created by the frameworks introduced in ONEX [51, 53] and GENEX [52], enabling researchers to perform very fast and highly accurate similarity searches in large datasets. Below, we discuss these approaches.

ONEX [51] introduced the novel idea of using a preprocessing pipeline that uses a cheap-to-compute Euclidean distance to create clusters of sequences that can then be efficiently queried using DTW to find the best match to a target sequence. This preprocessing allows for the *one-time* time-intensive work of clustering to be done offline and *many-use* fast queries of the data by researchers exploring the similarities of sequences. Such similarity search is possible due to proving a customized triangle inequality between the Euclidean distance used for clustering sequences and DTW used for similarity searches. This leads to highly accurate results, while reducing the time to find similar sequences by only using the *representatives of each cluster* for comparisons to the target sequence. This has further been extended [53] to retrieve multiple ranked similar sequences. By allowing for fast, accurate similarity search online, researchers may explore the similarity of time series more easily and interactively.

GENEX [50] is a novel framework that allows researchers to warp their own distances and then incorporate them in a cardinality reduction pipeline. It generalized the idea of combining pairs of pointwise distances and their warped counterparts, while still retaining the clustering pipeline with

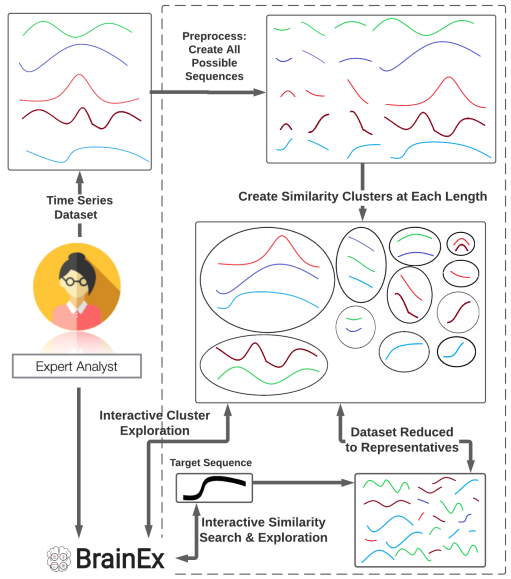

Fig. 4. *BrainEx* Pipeline Overview. When preprocessing a dataset, the time series are divided into all possible sequences of all possible lengths and then clustered into similar groups of equal length. After preprocessing, researchers can interactively explore the clusters and perform a fast similarity search by finding the cluster representatives most similar to the target sequence, and only searching the clusters they represent.

the customized triangle inequality properties of ONEX. However, the main challenges encountered by GENEX stem from the need for very large amounts of memory to preprocess the raw data and for exploring all the cluster representatives to find the ranked similar sequences. These challenges impact GENEX's ability to deal with very large datasets, both due to the high memory demands and the increased response times. A distributed system would be better suited to provide interactive response times and reasonable memory requirements.

3 BRAINEX ENGINE ARCHITECTURE

When conducting studies with brain signals such as fNIRS, researchers generate large, complex and often noisy datasets. These datasets consist of multivariate time series of brain signals coming from multiple scalp recording locations. They also contain metadata documenting the participant, the sensor channel, and any events occurring during the experiment session. A common goal in this research would be to better understand the impact of one or more of these attributes on the corresponding time series data. For example, researchers may want to answer questions such as: *What parts of the dataset look the most similar to a particular instance of user distraction? Are there particular patterns that are frequent in the dataset, in general? Are there particular patterns that are frequent for a particular study participant? Or sensor location? Or event? Or a combination of these factors?* To answer these questions, researchers cannot assume that all sequences in the dataset are the same length, since real-world tasks can vary. However, the search results should still find the most similar sequence. Many of these questions could be answered by building on the foundation of GENEX.

In this vein, we created *BrainEx*, which uses GENEX's cardinality reduction methods for similarity searches as the foundation for enabling analysts to explore complex time series datasets, yet it

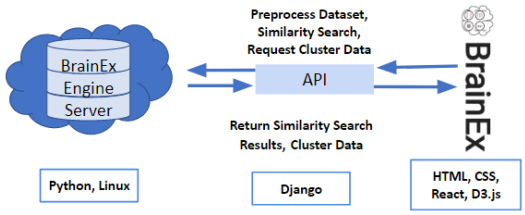

Fig. 5. The BrainEx System Architecture. The first component on the left is the *BrainEx* Engine Server, developed in Python and usable with Linux OS. The middle component is the API that preprocesses datasets, performs similarity searches, and clusters data which is implemented with Django. The last component is the *BrainEx* website interface which the user accesses and is developed in HTML, CSS, React, and D3.js.

employs a distributed architecture based on a novel preprocessing scheme to dramatically reduce both the response time and the needed amount of memory. In addition to increasing the performance and scalability, *BrainEx* provides a customized exploratory experience by taking advantage of the rich metadata and enabling researchers to perform multiple operations, including cluster exploration, filtering, and sorting. The novel preprocessing algorithm paved the way for the introduction of a much-needed user interface that makes data exploration available to researchers without requiring coding experience. In addition, the general framework and modular implementation make *BrainEx* an easy-to-expand tool, requiring merely a few lines of code to warp new distances to use to discover new insights into datasets.

BrainEx is implemented as a full stack application with three main components (Figure 5). The core *BrainEx* package is implemented in Python, and supports both Python native multiprocessing and Apache Spark [77]. In order to preprocess the dataset and perform a similarity search quickly, it relies on significant computing power. Therefore, *BrainEx* is hosted as a web application, with the *BrainEx* engine running on a server. End users can connect to a website, built using React, to interact with the tool. An API, written in Django, allows the website to communicate with the server. The BrainEx website uses the API to contact the server and load the list of preprocessed datasets, perform similarity searches, and gather the cluster data for exploration. The web browser handles the code to generate the graphs of the feature distributions returned by the server.

In the following sections, we discuss the algorithms and distributed architecture underlying *BrainEx* that enable interactive visual exploration and discovery. Its fast response time builds on the *process once, query many* paradigm [7, 72] described earlier. The two-step methodology includes first reducing the data cardinality by forming distributed similarity clusters and identifying their representatives (Section 3.1). Then, given a query sequence, the tool quickly finds the matches guided by the distances between the query and the cluster representatives (Section 3.2). The visual exploration interface that is built on top of the *BrainEx* engine is then described in Section 4.

3.1 BrainEx Engine: Distributed Preprocessing Algorithm to Compute Clusters

As noted above, to achieve fast similarity search, *BrainEx* groups the time series dataset into clusters, each of which is characterized by its "representative" time series. These clusters are defined below in Definition 3.1.

Definition 3.1. Given the set T of all possible sequences $(X_p)_j^i$—sequences of time series X_p of length i beginning at position j—in the dataset, T is divided into subsets called **clusters** C_k^i, where k is the cluster index, such that: (1) all sequences $(X_p)_j^i$ in C_k^i have the same length, and (2) each cluster C_k^i has one representative sequence R_k^i such that the normalized distance between it and any

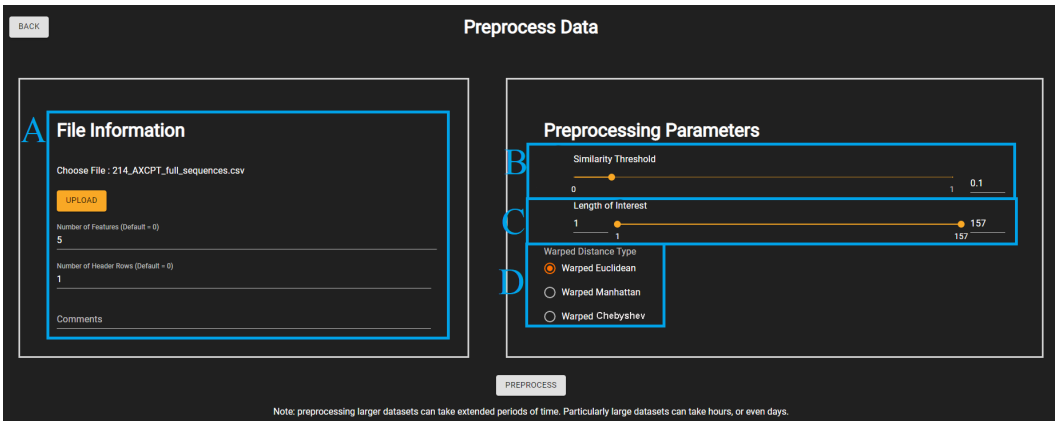

Fig. 6. The Preprocess Data page is where a user will upload a new dataset and specify the parameters for preprocessing. **A)** Users must manually provide the number of header rows and feature columns; the default for each is 0. **B)** The user-defined similarity threshold is the minimum similarity requirement between sequences in the same cluster. **C)** The length of interest is the range for subsequences to be spliced; the default is 1-n where n is the maximum sequence length in the dataset. **D)** The available warped distances for similarity matching. Currently, Warped Euclidean, Warped Manhattan, and Warped Chebyshev are available, however the code is built to easily accept more warped distances.

$(X_p)_j^i$ in C_k^i is smaller than half of the user-defined similarity threshold. The user-defined similarity threshold is the minimum similarity requirement between sequences in the same cluster (Figure 6).

Improving upon the resource requirements of GENEX, in order to form clusters in the preprocessing stage, the algorithm first slices the time series into subsequences, and distributes them onto multiple cores using the Distributive Step Slicing (DSS) schema, which ensures that a time series dataset is evenly segmented across data nodes in the distributed computing context. For the preprocessing job, the head node uses caching to store a copy of the entire time series, which each worker node can quickly access. The head node also tells each worker node which data points in each time series the worker should use as the starting point for finding all sequences for that time series that start with those particular data points. Valid sequences contain at least two data points, and are temporally ordered with no discontinuities.

In the context of distributed computing, when distributing a time series dataset D that has N time series, onto a context with a parallelism P, the naive, or default methodology would be to evenly assign $\frac{N}{P}$ time series to each executor. The drawback of this approach comes when N is not divisible by P; in this case, some executors would need to work a larger load. This load unbalancing would cause idle time for executors with smaller loads regardless of the value of P or N.

The Generalized DSS algorithm ensures that the loads are roughly balanced for computing the clusters. To form clusters, a worker node will iterate through its list of sequences of a given length that were generated by Generalized DSS. Each worker node will create a set of clusters, and sets of clusters are not aggregated by the head node (e.g. each worker node's clusters are kept as they are). The algorithm builds on a customized triangle inequality between point-wise distance and their counterpart warped distance inspired by and generalized from ONEX [51] and GENEX [50]. Clusters are built using cheap-to-compute pointwise distances such as Euclidean and Manhattan. Since they can only compare sequences of the same length, clusters will also contain such sequences.

For each sequence, the worker will check if a particular sequence is close to (as defined by the similarity threshold the end user inputted) the representative of any of the existing clusters for that sequence length. If this sequence is close (within the similarity threshold chosen by the user) to the representative of a cluster, then that sequence is added to that cluster. If the sequence is not close to the representative of any cluster, then a new cluster will be formed with this sequence as the new cluster's representative. This arbitrary selection of representatives is a method to discover the natural distribution of clusters in the dataset. Representatives are not updated throughout this process, so each cluster will have the first sequence that was sent to that cluster as representative. Once a worker node has clustered all the sequences for all the lengths specified, it then sends its partition to the head node. Once the head node has received the partition information from all the worker nodes, the preprocessing is complete.

At this point, the user can investigate each individual cluster to understand its characteristics, such as what features are present in the cluster, how many sequences it contains, and the length of the sequences. This feature will be referred to as *cluster exploration*.

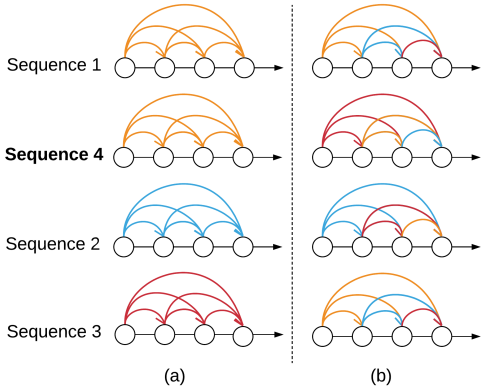

Fig. 7. Parts (a) and (b) demonstrate the difference between the naive distribution scheme and our implementation of distributive step slicing (DSS). The discs on the time series are individual data points. The curves above the data points represents the sliced subsequences. Different colored curves represent work done by different executors. Figure (a) shows the naive distribution scheme, when the number of time series is not divisible by the number of executors. This results in load imbalances. In this case, the load of the 'orange' executor is twice the amount of its fellows. Figure (b) shows a Generalized DSS for load (number of sub-sequences) balances over multiple time series; note that the executors' start index for each time series is set in a round-robin style to ensure further balancing of the loads.

3.2 *BrainEx* Engine: Distributed Similarity Search Algorithm

The similarity search functionality enables the end user to provide a search query in the form of a time series and *BrainEx* will return similar sequences from within the larger dataset. As with GENEX [50], instead of having to query all time series to find the similarity, *BrainEx* checks only the representative of each cluster. If a given representative is not within the similarity threshold of the time series used for the query, then the other time series in the corresponding cluster are not queried, thereby significantly reducing the amount of time series to query. On the other hand, if a representative is within the similarity threshold of the time series used for the query, then our algorithm will query all other time series within that cluster to find the similarity between these time series and the one used as the query. Unlike GENEX, *BrainEx* implements a distributed

algorithm for executing these similarity searches within a time series dataset. This improves both the preprocessing time and the query time on datasets as the distributed algorithm does more than simply parallelizing GENEX (Figure 7).

When the similarity search is initiated, representatives of all clusters are queried, and the distance between the target sequence and the representative of each cluster is specified. The query is broadcast to all worker nodes. If the representative is close to the target sequence, the worker will iterate through the cluster for that representative and compute the similarity between the target sequence and all sequences in the cluster. Once a worker is finished iterating through all necessary sequences, it will sort the sequences based on similarity. Finally, each worker will send the top-k sequences from its sorted list back to the head node along with its partition. The head node will then sort the final sequences based on similarity and output the top-k sequences to the user as the query result. These sequences can be a variety of lengths due to *BrainEx* using warped measurements instead of pointwise comparisons. The result output includes the sequence, the features of the sequence, and the sequence's similarity to the query.

We generalize DSS to let it iterate over multiple time series. Because the naive approach would result in the first executor having the most load, we dynamically set the begin index in a round-robin fashion as an executor working through the list of time series. Again, because the largest load unbalance is equal to P, when the number of time series and their length are sufficiently large, the impact of this imbalance becomes negligible.

3.3 *BrainEx* Engine: Time Series Indexing and Memory Optimization

Dividing time series into subsequences occupies a large amount of memory. A time series X_p of length n can be subdivided into at most $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ subsequences if we are dividing into subsequences of all lengths between 1 and n. Given that we have N such time series, caching this information is computationally prohibitive, yet *BrainEx*'s clustering scheme requires keeping tracking of all the sequences throughout. To mitigate this problem, we use an ID-start-end indexing technique.

Before preprocessing, each time series in D is assigned a universal unique identifier (UUID) $D_{ID} = \langle ID_1, ID_2, ..., ID_N \rangle$. After the subdivide stage, the resulting subsequences are represented by *Sequence* objects. For example, a subsequence X_p: $(X_p)_j^i$ will be represented by $Sequence(ID = ID_p, start = i, end = j)$

The original time series dataframe is broadcasted over all workers. When a computation needs to be performed on a *Sequence*, its data is fetched from the public dataframe and evicted from memory as soon as the computation completes. This way, the memory consumption is reduced to a linear increase, and we will later show in the experiment section the system's scalability in handling large datasets compared to competitors.

3.4 *BrainEx* Engine: Operations

BrainEx performs three main operations: *best match selection*, *ranked similarity search*, and *cluster exploration*. To assist in these operations, the user is presented with a number of parameters to filter the results. Below are descriptions of the parameters and operations they assist; more robust descriptions of similarity search and cluster exploration are found in Sections 4.4 and 4.5 respectfully, where we discuss the user interface.

3.4.1 Similarity Search. *BrainEx*'s similarity search includes four parameters to adjust either the query or the sequences that match the query. Users specify their *target sequence* from the preprocessed dataset and then specify the *start and end index* from that sequence to search for matches. The returned matches will be approximately the same length as the target sequence with

a ±1 margin of error. The user can choose any *number of matches* for their target sequence with an upper limit of the total number of sequences of similar length to the target. The last two parameters limit the sequences that can be selected as a match. First, the user can limit the *overlap* between results to prevent results from the same parent sequence that are offset by only a few data points. Secondly, the user can exclude sequences that include points from the target sequence.

3.4.2 Cluster Exploration. All clusters can be filtered by how many sequences are grouped in the cluster (*cluster size*) and the *length* of the sequences. Each cluster contains sequences of a single length. Therefore, if one sequence in the cluster is of length 40, all of its sequences have length 40.

In addition, the user can also filter clusters by values of specific *user-customized features*. In brain signal datasets (e.g., fNIRS), common features are *participant name*, *event name*, and *channel name*. By filtering by these features, the user can search for clusters that are primarily composed of a certain event or a specific subject, or look for clusters that have an equal number of sequences from different channels.

3.5 *BrainEx* Engine: Supported Datasets

BrainEx supports the input of TSV and CSV files. Columns contain features or datapoints for a sequence while each row is an individual sequence. Each sequence must contain the same number of features. The sequences do not need to be of an equal length, e.g. a dataset could contain sequences of various lengths, for example 20, 35, and 51. We show below a snapshot of an example dataset, containing one header row, two features, and sequences of length 3.

Subject	Condition			
1	*Individual*	1.1	1.2	1.3
1	*Cooperative*	2.3	2.4	2.5
2	*Individual*	1.9	2.0	2.1
2	*Cooperative*	2.8	2.9	3.0

4 BRAINEX VISUAL EXPLORATION DESIGN

The *BrainEx* engine described above provides sophisticated time series analytical tools to enable interactive exploration. All of the functionality described above can be accessed using command line APIs. However, this requires expert users with deep technical knowledge to execute. In addition, without visual representation of the data, the functionality is not particularly useful for an end user to familiarize themselves with and truly explore the data.

Thus, we aimed to build an effective visual interface on top of the *BrainEx* engine to expose the underlying cluster exploration and similarity search algorithms in a way that is easy to use by users of all skillsets. The goal is not necessarily to introduce novel ways of visualizing time series data, but rather to make the algorithms more accessible and valuable to researchers, so they can more easily explore and discover interesting relationships in time series datasets. Expert users of *BrainEx* may not be familiar with command line interfaces and would not be able to use *BrainEx* to its full potential without a visual interface.

4.1 Usage Goals

We developed the following simple usage scenario to motivate the design of the *BrainEx* interface. A researcher has performed a fNIRS study, collecting data from several participants as well as multiple sensor locations on each participant, creating multivariate time series signals. In this hypothetical study, participants were asked to complete several short tasks, some that were calibrated as *easy*, others that were calibrated as *hard*, and some task with *unknown* difficulty. The researcher is

interested in better understanding this data during the task of *unknown* difficulty, and would like to use the data collected during the other two calibrated tasks to see if there are connections.

In this scenario, a researcher would need to get a sense of the distribution of the data by understanding which sequences are similar to each other in the dataset. In this case, they would not have any particular sequences of interest in mind. Instead, they would need to explore the entirety of the data. The researcher may want to understand how the sensor location and task are related to the underlying brain activation. They may also want to get a sense of the 'shape' of a time series which represents some grouping. In the process of exploring, a researcher might come across a sequence which is of particular interest. It is important to be able to search for any number of sequences similar to the one discovered. Once these are retrieved, they will need to explore the distribution of the related metadata. When a researcher has identified any subset of the data, whether by exploration or searching, it is important that they are able to explore the distributions of one or more features of this data. This is just a simple scenario that helped to develop the functional requirements for the visual interface design. However, much more complex analysis can be performed, and some of this is illustrated in the case study described in Section 7.

4.2 Functional Requirements

By exploring the usage goals, we determined five functional requirements to incorporate from the *BrainEx* Engine into the user interface. The first requirement is the ability to compare one sequence to the rest of the dataset by finding which sequences it is most similar to.

Requirement 1: Similarity Search

a) Support retrieving and ranking any number of sequences similar to another sequence of a researcher's choice.
b) Support exploration of search results and attributes of the sequences in the result set.

The second requirement is the ability to explore the feature distribution in a set of sequences that are naturally grouped together.

Requirement 2: Feature Distribution Exploration

a) Support exploring the distribution of a single feature in a cluster or result set.
b) Support exploring the joint distribution of two features in a cluster or result set.
c) Support comparing the relationships between three or more features in a cluster or result set.
d) Support identifying a sequence shape that well-represents a cluster or result set.

Similarly, *BrainEx* should support the ability to explore the distribution of all such natural groupings in the dataset.

Requirement 3: Cluster Exploration

a) Support exploring the range of cluster sizes and sequence lengths within clusters.
b) Support finding clusters of similar sequences with skewed feature distributions (i.e. clusters that mostly contain a particular User, Channel, or Event).

From the insights gained by exploration into clusters of sequences and feature distributions, *BrainEx* should support using these insights to start new explorations and searches on sequences found to be of interest.

Requirement 4: Integration

a) Support searching and finding sequences of interest to explore further based on the results of cluster exploration.
b) Support exporting sequences of interest and other findings to explore further in other tools.

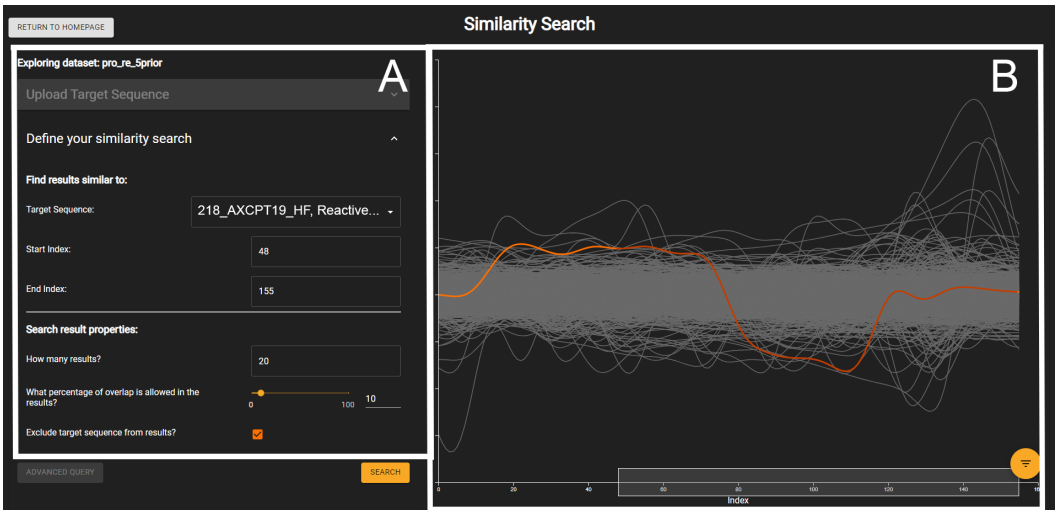

Fig. 8. Similarity Search. *BrainEx* Visual interface for conducting similarity search consisting of search options in the left panel (marked A) and a visualization of the dataset's sequences on the right (marked B).

In addition to the functional requirements above, the final requirement was that use of the tool should not be limited to a small group of highly trained researchers with access to high performance computing.

Requirement 5: Accessible to All Researchers

a) Support fast computations, regardless of researcher's computer.
b) Support researchers at all levels, from novice to expert.
c) Support diverse experiments to be explored, and remain agnostic to the particular user-customized metadata (e.g. participant, events, channels, etc.) that are associated with the dataset.

4.3 Interface Components

Based on these requirements, we built a visual interface on top of the *BrainEx* engine. It provides an integrated pipeline for researchers to move between the broad exploration of a dataset and queries for specific sets of most similar sequences. Our tool is fully agnostic, and can accommodate any number of customized, researcher-specified labels on the data. A user can select any of the preprocessed datasets and then select *Similarity Search* (Figure 8) to find the best k matches to a specific subsequence of the dataset *or* the user can select *Cluster Exploration* (Figure 9) to explore the distribution of features among clusters in the dataset.

4.4 Similarity Search

Our visual interface leverages the power of time series analysis [50, 51, 53] and expands it with interactive visualizations, as well as analytic workflows developed for fNIRS data analysis, to provide the time series exploration experience outlined in Figure 4.

BrainEx enables researchers to visually investigate the sequences in the dataset. Before initiating a similarity search, researchers can explore the sequences in the dataset via a line chart. To reduce the number of sequences visible and to enable more targeted exploration, *BrainEx* provides filtering, zooming, and panning support in the sequence view seen in Figure 8.B. For a more specific search,

Fig. 9. Cluster Explorer. *BrainEx* interface for exploring the clusters consisting of a table of filtering options in the left panel (marked A), a table of the dataset's clusters and their color-coded feature distributions (marked B), and a visualization showing the overall distribution of clusters in the dataset (marked C). When a cluster is selected, the visualization changes to show that cluster's representative.

BrainEx enables researchers to select start and end indices, specify a maximum number of results to return, and exclude the target sequence from the search results. As there may be overlap between the target sequence and a similar sequence, *BrainEx* also enables the selection of a percentage of allowed overlap. These filters can be set with the options shown on the left side in Figure 8.A.

Once a researcher has completed their search, the results are presented via a table containing details about the resulting sequences and a line chart where each sequence is visualized. The table enables the researcher to see each sequence, the features associated with it, and its distance from the target sequence. For easy comparison, the target sequence is always located at the top of the table and is highlighted in the line chart. Hovering over a sequence in either the line chart or the table highlights the sequence in both locations and scrolls to the sequence in the table if it is not already visible. To enable more refined control over the visualization of the results, the sequences can be sorted and filtered by each feature, the number of visible results can be limited, and a maximum distance can be specified. *BrainEx* also allows researchers to export the resultant sequences as a CSV file and to save the line chart as an image for further exploration and/or interpretation. Moreover, the distribution of features in the result sequences can be investigated through the feature distribution explorer.

4.5 Cluster Exploration

BrainEx enables researchers to explore the clusters of sequences through a table (Figure 9.B) containing information about every cluster in the dataset. This information includes the *number of sequences* in each cluster, the l*ength of the sequences* in each cluster, and the single *distribution of the features* of sequences in each cluster. Cells describing the feature distributions are colored to show the salience of that feature value in the sequence, allowing researchers to scan for clusters with interesting distributions to investigate further.

To ensure agnosticism regarding the features of the dataset, *BrainEx* allows the table to be expanded and contracted by the user by providing the left panel of table filter options (Figure 9.A).

This variability is available for both the number of rows and columns shown at once, allowing a researcher to customize how much information they would like immediately available. *BrainEx* also provides an option to view the average feature distributions across the dataset, providing researchers a baseline to compare when finding a cluster.

The number of clusters can grow quickly as the number of sequences and the range of sequence lengths increases. To provide researchers with the ability to target their exploration in this large space, *BrainEx* provides sorting on each column, filtering, and a visualization. Researchers may use sorting to quickly find clusters that may be more interesting, such as clusters with more sequences or those with particularly skewed feature distributions. This sorting can also be tuned by using the filtering on the length and number of sequences in the clusters that are shown in the table. As researchers may not know how best to apply these filters, we also provide a scatter plot (Figure 9.C) showing the distribution of clusters against the number and length of sequences within each cluster. This visualization can provide context for the researcher's expectations when applying the filters, while also describing more general properties of the dataset's sequences and their tendencies to be clustered together.

In addition to exploring the distribution of clusters and the features within those clusters, researchers may also explore the shapes of sequences in each cluster. These shapes are provided in the form of the cluster representative sequence to which the other sequences in the cluster are similar. Once a researcher selects a cluster, they see a line plot of this representative sequence to show this shape with a scale to allow comparison between the representatives of different clusters.

4.6 Feature Distribution Exploration

Once *BrainEx* retrieves a group of sequences, either from a similarity search or from cluster exploration, it supports visualization of relevant information. This is complicated by the fact that *BrainEx* is fully agnostic of the user-defined, customized metadata provided. When researchers preprocess the dataset, they can specify any number of attributes (e.g. channel, user id, condition), which can have different values and data types. Also, researcher-selected datasets can vary in size.

As a result, the visualization techniques must be able to show the distribution for an arbitrary number of sequences. Further, they must be able to present the distribution with respect to any number of feature labels. Thus, *BrainEx* combines several data visualization techniques to present data depending upon the sort of information a researcher is looking for.

Researchers can choose the set of features that they are interested in by using feature specification checkboxes. As they check boxes, the visualization display pane updates in real time, allowing for seamless exploration. The display may change among four states: single feature bar charts, two feature heatmaps, many feature parallel coordinates views, and a time series sequence view. Detailed information about individual visualizations is presented when a user clicks the "More Info" button.

The first visualization state is a bar chart (Figure 10.A), which is used whenever a researcher needs to display the distribution with respect to one feature. Bar charts are well-studied for the application of comparing two or more values in a single dimension [16]. The bar chart has a mouseover component, which allows researchers to explore the precise number and percent prevalence of any value presented.

The second state is a heat map (Figure 10.B) that presents the joint distribution over two user-selected features. It uses a linear saturation scale from white to the primary orange or blue color of *BrainEx*, orange when in dark mode and blue when in light mode. White corresponds to 0 percent prevalence, and the full saturation to the highest prevalence present. The use of a consistent, single-hue color scheme means researchers who examine multiple heat maps don't have to learn a

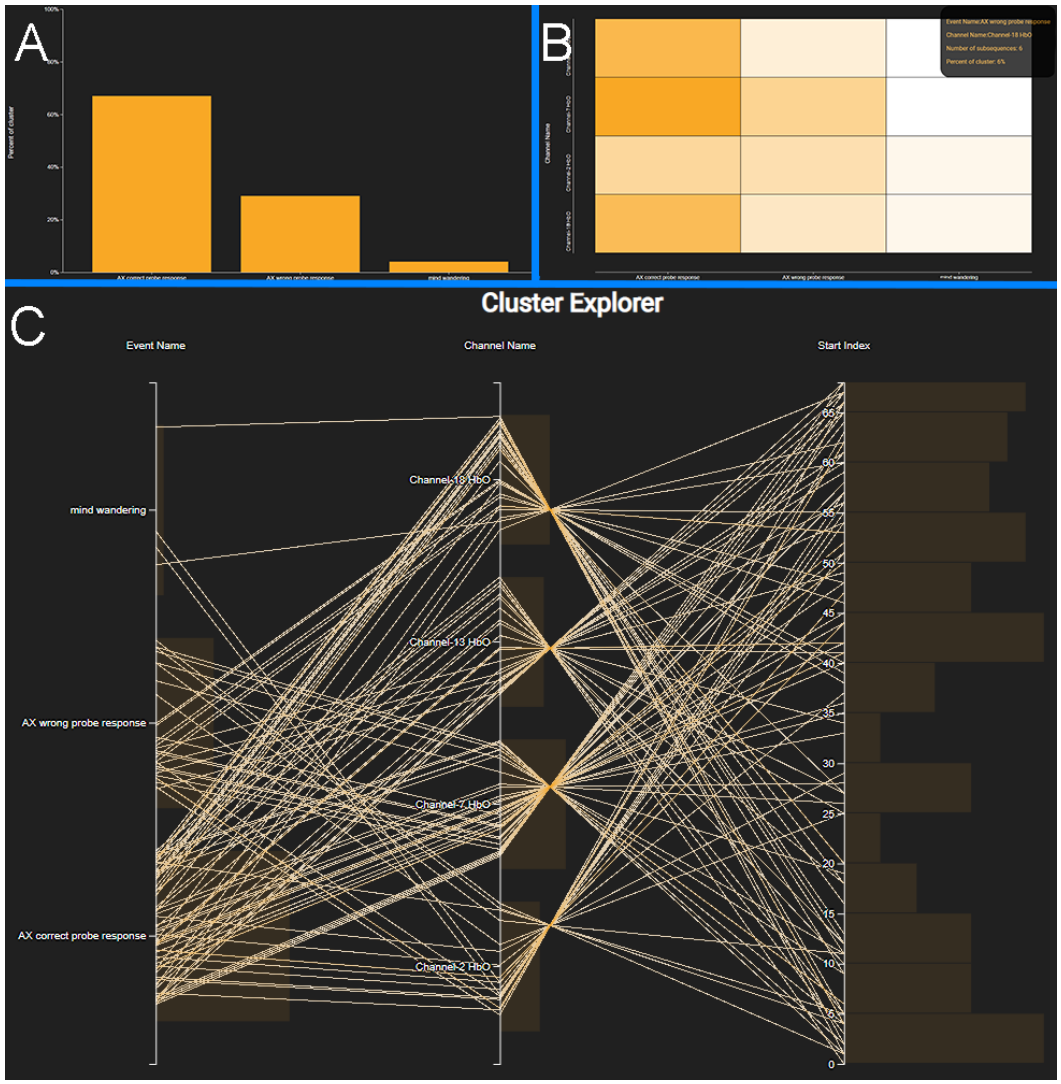

Fig. 10. The three different vizualizations to explore the content of a cluster. A user will progress from the bar chart (**A**) to the heat map (**B**) then to the parallel coordinates (**C**) as they add additional features to the visualization. **A)** A bar chart displaying the distribution of a single feature in a sample set of sequences. **B)** A heat map displaying the joint distributions of two features in a sample dataset. **C)** A parallel coordinates view displaying the joint distributions between three features in a sample dataset. This visualization will be used for any number of features ≥ 3.

new encoding every time. Mouseover text is also available, giving the precise counts and prevalence of the joint distribution.

The third is a parallel coordinates view (Figure 10.C). Parallel coordinates allow the visualization of N-dimensional data in 2-dimensional space, making them a powerful choice for researchers who could be interested in any number of data labels [33]. *BrainEx* lets researchers select any number of features, and visualize the joint distribution of all of them at once. Like many implementations,

BrainEx allows researchers to rearrange the ordering of the axes to explore different pairwise distributions. While increasing the number of features is known to increase the time necessary to explore the data [47], researchers can choose to visualize only features which they are interested in, reducing this exploration time. There are known cases where the display of thousands of data points on a single parallel coordinates map can become unreadable [47]. To accommodate this, we provide a box-select tool, which allows researchers to indicate which sequences they would like highlighted. This allows researchers to separate out a readable subset of the dataset whenever it becomes too large to read.

Once a researcher has some understanding of the overall distribution, they may want to do a more sequence-focused exploration of the data. For this reason, the tool provides a plot sequence feature. This button allows researchers to view a line plot of any set of sequences. This line plot uses color encoding to show the association between the sequences and a single feature of the researcher's selection. This allows researchers to visually check for patterns within the sequence data with respect to the features, or to get a sense of the sort of shapes of series that they are studying. An example of this is in Figure 1.

4.7 Integration Pipeline

As mentioned before, BrainEx provides researchers the ability to perform similarity searches on time series data. It also provides overviews and visualizations to allow researchers to understand the shape and content of their data. These tools cover the breadth of our first three functional requirements. All of these requirements are important for the exploration of time series data and can be used effectively on their own. However, by integrating the separate tools we can allow researchers to have a more powerful exploration workflow.

When exploring the data distribution within a cluster, researchers may discover interesting time series sequences that they want to explore further outside of the feature distribution exploration. We allow researchers to select a time series from this feature exploration view and then perform a similarity search with that sequence as the target. This allows researchers to better understand the contents of the cluster. Additionally, it allows researchers to find sequences similar to interesting time series discovered in the exploration workflow. This will allow researchers to better understand their data during exploration and potentially reveal previously unknown patterns in the dataset.

In order to integrate *BrainEx*'s exploration capabilities with other systems, our tool also enables researchers to export the contents of similarity search or cluster exploration to a CSV file. This data can then be used in other tools for further analysis of the time series data after initial exploration and pattern discovery is accomplished by *BrainEx*.

5 PERFORMANCE BENCHMARK EXPERIMENT

To demonstrate the power of *BrainEx*, we present a performance benchmark on a large number of datasets from the UCR archive [17], a well-known collection of time-series datasets from many research areas that is widely used for similarity exploration. In it, we compare the accuracy and response time of BrainEx with several competitors when performing similarity search using three different warped distances (warped Euclidean, warped Manhattan, warped Chebyshev). To comprehensively validate the precision and performance of *BrainEx*, the system was tested on the UCR archive using the pipeline described below.

5.1 Competitors

We compared the *BrainEx* data mining tool with three state-of-the-art systems that are able to employ multiple warped distances and are extensively used in the literature. These competitors are

Generalized Dynamic Time Warping (GDTW) [52], Piecewise Aggregate Approximation (PAA) [37], and Symbolic Aggregate Approximation (SAX) [40].

GDTW is the generalized warped distance framework proposed in [52], which finds exact solutions by comparing a query with all possible candidates (Table 1). We use the results of GDTW as ground-truth for our evaluation. PAA [37] is a dimensionality reduction method that compresses time series by averaging consecutive equal-length subsequences. We averaged subsequences of length 3, following the empirical practice in [50]. SAX [40] is another data reduction method that is similar to PAA in that it reduces the dimension of a time series by combining a specified number of data points. The SAX approach takes this a step further by encoding the values aggregated by PAA as strings.

As *BrainEx* employs distributed computing, in an effort to make the comparison fair, we parallelized the competitors' distance calculations to take advantage of a multi-processing context. Our preliminary study showed that with *BrainEx* being able to handle large data, the competitors, if run on a single core, quickly become impractical if we were to run an extensive experiment comparing to them. Reflecting this fact, we distributed the candidate-to-query distance calculation that happens in GDTW, PAA, and SAX. (E.g., for a dataset with 32,000 points, the distance calculation using SAX takes about 300 seconds on a single core, but only 15 seconds on 32 cores.) Doing so does not affect the accuracy of the results of these algorithms, because calculating the distance between a candidate and the query is an independent task.

5.2 Experimental Methodology

5.2.1 Preparing Datasets. Each dataset in the UCR archive is separated into training and test sets. As the training set from each dataset generally contains more time series than the test sets [17], we use these as part of the preprocessing phase for clustering. We then explore the clusters using target sequences from both the training sets and the test sets. All sequences were min-max normalized based on the minimum (min) and maximum (max) of their respective datasets. I.e., given a sequence $X = (x_1 \ldots x_n)$, the normalized value for each point x_i is $\frac{x_i - min}{max - min}$.

The response time of retrieving one or more matches for a target sequence is highly dependent on the size of the dataset. To better understand the relationship between performance and the size of the datasets, we divide the datasets into three groups and run separate experiments for each group. The grouping of the datasets is based on the number of data points they contain. In this experiment, we grouped the 128 datasets into three bins: (1) Small datasets have a range of [1, 50,000] data points and contain 69 datasets; (2) Medium datasets have a range of [50,001, 1,500,000] data points and contain 56 datasets; (3) Large datasets have over 1,500,000 data points and contain 3 datasets. This binning is heuristically determined to ensure each bin contains approximately the same number of aggregated time series (Appendix A). We use the same experimental procedure for each dataset and the outcomes for all datasets are aggregated within their respective bin when presenting the results. For the small datasets, we used 32 CPU cores and 68 gigabytes of memory. For the medium datasets, we used 80 CPU cores and 700 gigabytes of memory. The jobs were ran on a high-performance computing cluster [56] using a singularity image running *BrainEx* and containing the UCR archive datasets. The jobs had exclusive access to the nodes used, therefore no other jobs were ran on the same node as the experiments.

5.2.2 Query Selection. When selecting queries, half of query sequences are taken internally from the training set and the other half externally from the test set, and are randomly selected. This simulates searching for similar sequences with queries from both inside and outside the dataset. To ensure the system is evaluated on queries of varying length, we created three query length ranges

Table 1. Definitions of each distance

	Definition	Normalized distance
Euclidean	$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$	$\overline{ED}(X,Y) = \frac{ED(X,Y}{\sqrt{n}}$
Manhattan	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$	$\overline{MD}(X,Y) = \frac{MD(X,Y)}{n}$
Chebyshev	$\max_{i=1}^{n} \lvert x_i - y_i \rvert$	$\overline{Cheb}(X,Y) = Cheb(X,Y)$
GDTW	GDTW_d	$\overline{GDTW}_d(X,Y) = \frac{GDTW_d(X,Y)}{2n}$

(small, medium, and large), and select n_q queries from each length range from both the training and test sets, with the total number of queries equal to 20% of the number of time series in the dataset.

The experiment follows these steps:

(1) For each dataset, we selected a subset of queries as follows: the query sequences are selected from three distinct length bins. The length range for the small bin is 1 to one third of the longest time series in this dataset ($\left\lfloor \frac{l_{max}-1}{3} \right\rfloor$); the medium is from $1 + \left\lceil \frac{l_{max}-1}{3} \right\rceil$ to $\left\lfloor \frac{2\times(l_{max}-1)}{3} \right\rfloor - 1$; and the large is from $1 + \left\lceil \frac{2\times(l_{max}-1)}{3} \right\rceil$ to $l_{max} - 1$. The number of query sequences per bin, n_q, is defined as $\left\lceil \max\{1, \frac{0.1N}{3}\} \right\rceil$, where N is the number of time series in the dataset. This process is applied to both the test and training sets in the UCR Archive. For example, if the longest sequence in a dataset of 60 time series was 100 points, then we would have query ranges of [1, 33], [34, 66] and [67, 99], and $2n_q = 4$ random queries would be selected within each range limit for a total of 12 queries. Each query range contains 4 queries, $n_q = 2$ from the test set and $n_q = 2$ from the training set.
(2) We then preprocess the dataset with *BrainEx* using each of the three distances: warped Euclidean, warped Manhattan, and warped Chebyshev (Table 1).
(3) We iterate over the query sequences generated from step 1. For each query, we run three k-similar-sequences experiments: one for best-match retrieval (k=1), and two ranked-matches—one for k=5 and one for k=15. These numbers were selected because they are commonly chosen by users [50]. This iteration is performed with BrainEx, PAA, and SAX.

With the above experiment procedure, we compile the results to show: (1) how *BrainEx* performs on ranked matching performance (time and accuracy); (2) how dataset size, length of the time series, and the number of time series affect the response time and accuracy; and (3) how the preprocessing time of *BrainEx* varies with differently sized datasets.

5.3 Experiment results

Of the 128 datasets from the UCR Archive, 100 were tested with all three warped distances and 4 more were tested with warped Euclidean and warped Manhattan. We could not test these datasets with warped Chebyshev due to its longer preprocessing time paired with the long query times from PAA and SAX. In addition, 24 datasets were not tested at all, including the three datasets from the large bin, because of PAA and SAX's long query times. While *BrainEx* took approximately 5 seconds for its queries on the medium bin, SAX took over 7 minutes on average and PAA took over 10 minutes (Table 2). The competitors would not have performed better than *BrainEx* because as the size of the dataset increased, the response times increase. We limited each dataset to a maximum run time of 168 hours to complete its preprocessing and querying for each method. We did not split each method into its own session to ensure that the same node on the Turing cluster was used

Table 2. Query Time for Medium Datasets

	Warped Euclidean			Warped Manhattan			Warped Chebyshev		
Method	k = 1	k = 5	k = 15	k = 1	k = 5	k = 15	k = 1	k = 5	k = 15
PAA	674.58 s	674.58 s	674.58 s	622.20 s	622.20 s	622.20 s	352.95 s	352.95 s	352.95 s
SAX	461.51 s	461.51 s	461.51 s	427.35 s	427.35 s	427.35 s	283.61 s	283.61 s	283.61 s
BrainEx	5.32 s	5.23 s	5.20 s	5.05 s	5.01 s	5.00 s	3.93 s	3.89 s	3.88 s
GDTW	872.39 s	872.39 s	872.39 s	829.61 s	829.61 s	829.61 s	395.84 s	395.84 s	395.84 s

Table 3. Query Error for Medium Datasets

	warped Euclidean			warped Manhattan			warped Chebyshev		
Method	k = 1	k = 5	k = 15	k = 1	k = 5	k = 15	k = 1	k = 5	k = 15
PAA	0.0067	0.0087	0.0135	0.0033	0.0043	0.0068	0.0334	0.0441	0.0674
SAX	0.0746	0.0773	0.0831	0.0548	0.0572	0.0628	0.1799	0.1886	0.2023
BrainEx	0.0005	0.0007	0.0008	0.0002	0.0002	0.0003	0.0032	0.0042	0.0057

for each method. Appendix A contains the name of each UCR Archive dataset, its total number of data points, its bin classification, and if it was included in the results.

5.3.1 Ranked Matching Accuracy. We use the results from GDTW as the ground-truth. The query accuracy for the other three algorithms (BrainEx, PAA, SAX) are calculated as follows, for a single query and n number of matches:

$$\sum_{i=1}^{n} |\overline{D}^{q}_{GDTW_i} - \overline{D}^{q}_{ALGORITHM_i}|, \tag{1}$$

where $\overline{D}^{q}_{GDTW_i}$ is the normalized warped distance between the query and i-th match found by GDTW and $\overline{D}^{q}_{ALGORITHM_i}$ is the same distance but with the match found by one of the three other algorithms. Tables 4 and 5 contain the query time and errors for the small datasets while Tables 2 and 3 contain the query time and errors for the medium datasets. Even after distributing the GDTW, PAA, and SAX algorithms, *BrainEx* performs at least two orders of magnitude faster on queries in both the small and medium datasets.

For the small bin, Figures 12, 13, and 14 show that GDTW, PAA, and SAX become much slower, regardless of warped distance, for some datasets with between 30,000 and 40,000 data points. This is because the datasets in that size range were irregularly sized, which means that within the dataset the time series were of varying lengths. This irregularity caused GDTW, PAA and SAX to slow down while *BrainEx* maintained the same efficiency. In addition, Figures 13, 12, and 14 show that SAX has a substantially more errors than PAA or *BrainEx* while *BrainEx* has consistently the lowest errors.

BrainEx's largest errors on the small datasets were 0.0012 for warped Euclidean, 0.0005 for warped Manhattan, and 0.0060 for warped Chebyshev, all of which are less than 1% error rates. However, PAA's largest errors were 0.0117 for warped Euclidean, 0.0062 for warped Manhattan and 0.0538 for warped Chebyshev, while SAX's largest errors were 0.0944 for warped Euclidean, 0.0717 for warped Manhattan, and 0.2210 for warped Chebyshev. While these are competitive accuracies, *BrainEx* consistently has better performance across each elastic distance (Table 5). In addition,

Table 4. Query Time for Small Datasets

	Warped Euclidean			Warped Manhattan			Warped Chebyshev		
Method	k = 1	k = 5	k = 15	k = 1	k = 5	k = 15	k = 1	k = 5	k = 15
PAA	24.35 s	24.35 s	24.35 s	22.16 s	22.16 s	22.16 s	21.45 s	21.45 s	21.45 s
SAX	26.83 s	26.83 s	26.83 s	25.09 s	25.09 s	25.09 s	24.46 s	24.46 s	24.46 s
BrainEx	0.70 s	0.69 s	0.69 s	0.63 s	0.63 s	0.62 s	0.68 s	0.67s	0.67s
GDTW	24.90 s	24.90 s	24.90 s	21.81 s	21.81 s	21.81 s	21.05 s	21.05 s	21.05 s

Table 5. Query Error for Small Datasets

	Warped Euclidean			Warped Manhattan			Warped Chebyshev		
Method	k = 1	k = 5	k = 15	k = 1	k = 5	k = 15	k = 1	k = 5	k = 15
PAA	0.0045	0.0067	0.0117	0.0022	0.0034	0.0062	0.0216	0.0321	0.0538
SAX	0.0847	0.0876	0.0944	0.0633	0.0660	0.0721	0.1989	0.2060	0.2210
BrainEx	0.0007	0.0009	0.0012	0.0003	0.0004	0.0005	0.0032	0.0045	0.0060

BrainEx is on average 33.7x faster than GDTW, 33.8x faster than PAA, and 38x faster than SAX on the small datasets (Table 4).

For the medium bin, Figures 15, 16, and 17 show that while *BrainEx* is consistently faster than GDTW, PAA, and SAX, *BrainEx* is substantially faster on datasets greater than 140,000 data points. *BrainEx* consistnetly has the highest accuracy on all datasets in this bin, though PAA is comparable. SAX has the lowest accuracy for all datasets in this bin though it performs queries faster than PAA on average. *BrainEx* does not compromise between efficiency and speed, however, as it makes the fastest queries with the highest accuracy comparatively to PAA and SAX.

BrainEx's largest errors on the medium datasets were 0.0008 for warped Euclidean, 0.0003 for warped Manhattan, and 0.0057 for warped Chebyshev, all of which are less than 1% error rates. However, PAA's largest errors were 0.0135 for warped Euclidean, 0.0068 for warped Manhattan and 0.0674 for warped Chebyshev, while SAX's largest errors were 0.0831 for warped Euclidean, 0.0628 for warped Manhattan, and 0.2023 for warped Chebyshev. Once more, *BrainEx* consistently has better performance across each elastic distance (Table 3). *BrainEx* is on average 144.1x faster than GDTW, 114.2x faster than PAA, and 81.8x faster than SAX on the medium datasets (Table 2).

Table 6. Average clustering time for small and medium dataset bins by warped distance. Time is in seconds.

Distance	Warped Euclidean	Warped Manhattan	Warped Chebyshev
Small bin	99.30 s	67.28 s	288.72 s
Medium bin	2,776.08 s	1,810.26 s	6,544.69 s

5.3.2 BrainEx Clustering Time. "To achieve these fast query times, *BrainEx* clusters sequences together based upon their sequence length and similarity, as shown in 3.1. Figure 11 shows the clustering time for each dataset in the small bin (top) and medium bin (bottom) along with a line of best fit for each warped distance. Consistently across both bins, warped Manhattan has the fastest

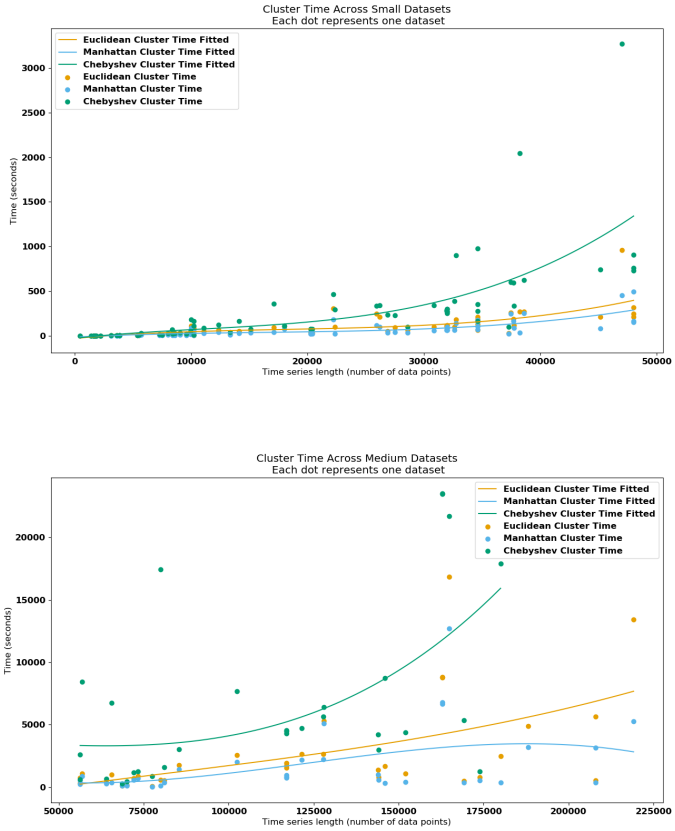

Fig. 11. (Top) The time (seconds) it takes for *BrainEx* to cluster the sequences for each small dataset, marked by dots, along with a fitted line for the clustering time for each warped distance. (Bottom) The time (seconds) it takes for *BrainEx* to cluster the sequences for each medium dataset, marked by dots, along with a fitted line for the clustering time for each warped distance.

clustering time while warped Chebyshev has the slowest. Warped Euclidean is in between the two, but is most similar to warped Manhattan.

6 PRELIMINARY USER STUDY

To get feedback on the visual exploration interface, we conducted a preliminary user study.

6.1 Study Design

For this study, participants were invited to use an instance of *BrainEx* that included a preprocessed dataset containing 8 users with activity in 4 channels. This dataset was generated to represent the usage scenario described in Section 4.1. To ensure the study delivered meaningful insights, we sought to draw on the experience of experts in the fields of fNIRS, data visualization, and/or HCI research. The participants were encouraged to spend time exploring the preprocessed dataset in *BrainEx* before answering a questionnaire. The participants were also encouraged to refer back to

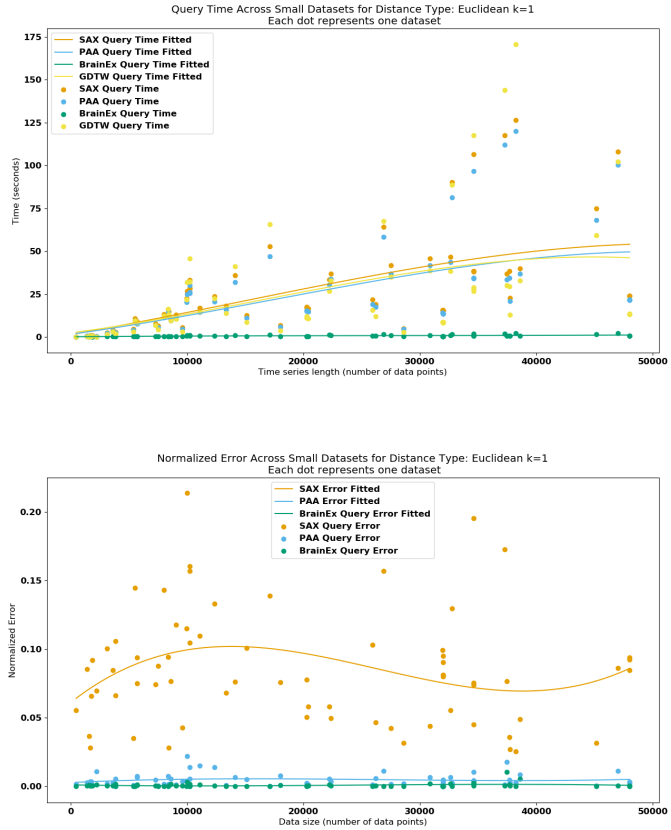

Fig. 12. (Top) The exact time (seconds) and fitted time for SAX, PAA, BrainEx, and GDTW to find the best match to a given query for the small bin and warped Euclidean distance. (Bottom) The normalized error for PAA, SAX, and BrainEx for finding the best match to a given query using the GDTW method as ground truth. This is for warped Euclidean distance on the small dataset bin.

the application while taking the survey. The study was designed to take approximately 45 minutes to an hour to complete.

6.1.1 Questionnaire. The questionnaire was an online survey developed in Qualtrics. The form consisted of three main sections: one for general demographic questions, the largest section focused on the functional requirements defined in Section 4.2, and one section for general *BrainEx* usability and usage questions. The general demographic questions collected background information about the participants' experiences with fNIRS, other brain activity tools, data visualization, and HCI. The functional requirements section of the survey asked study participants to explore a preprocessed dataset through the lens of the usage scenario described in Section 4.1. For each of the bullet points within the first four functional requirements, participants were asked to rank how much they agreed with the statement on a 5-point Likert scale. They were also asked to provide any insights they made about the dataset, and any positive or negative comments about their experience completing

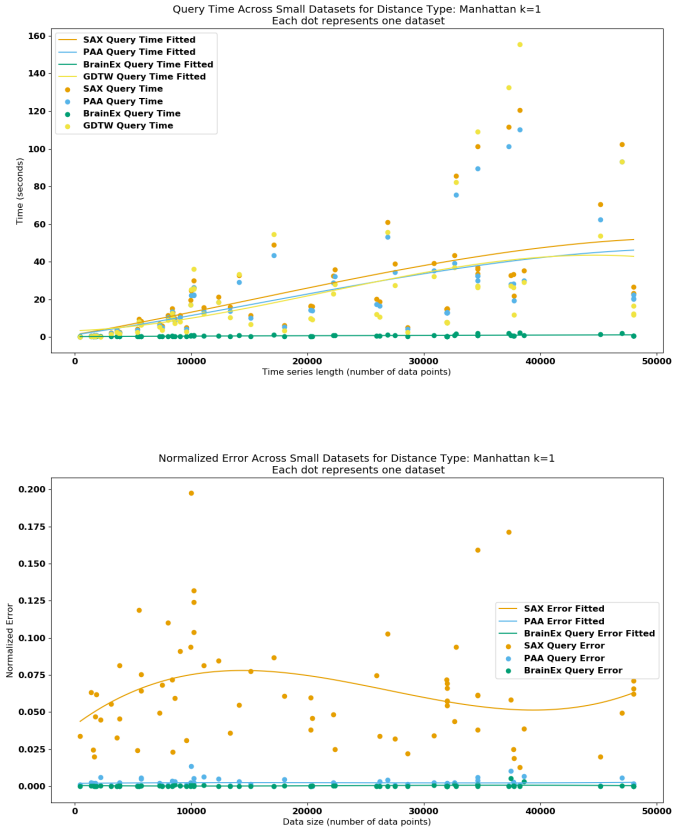

Fig. 13. (Top) The exact time (seconds) and fitted time for SAX, PAA, BrainEx, and GDTW to find the best match to a given query for warped Manhattan distance on the small dataset bin. (Bottom) The normalized error for PAA, SAX, and BrainEx for finding the best match to a given query using the GDTW method as ground truth. This is for warped Manhattan distance on the small dataset bin.

the requirement. The final section asked the study participants to rate the general usability of *BrainEx* as well as share other comments about *BrainEx*.

6.1.2 Participants. Our study was sent to 40 neuroscience researchers, data visualization experts, and HCI experts who represent our target user base. Of these, 10 responded and participated in our user study. The recruited participants represented a diverse group of target users. The education level and self reported expertise of the participants can be seen in Table 7. Six of the participants have published fNIRS or neuroscience research, and five have published HCI papers. Three of the participants had used *BrainEx* before. All participants used Chrome or Firefox to complete the study.

6.1.3 Limitations of the Study. The results of this study are predicated on the subjective responses of the survey participants. We limited the study participants to data visualization researchers, HCI experts, and fNIRS researchers because they are the most likely primary users of *BrainEx*. This is

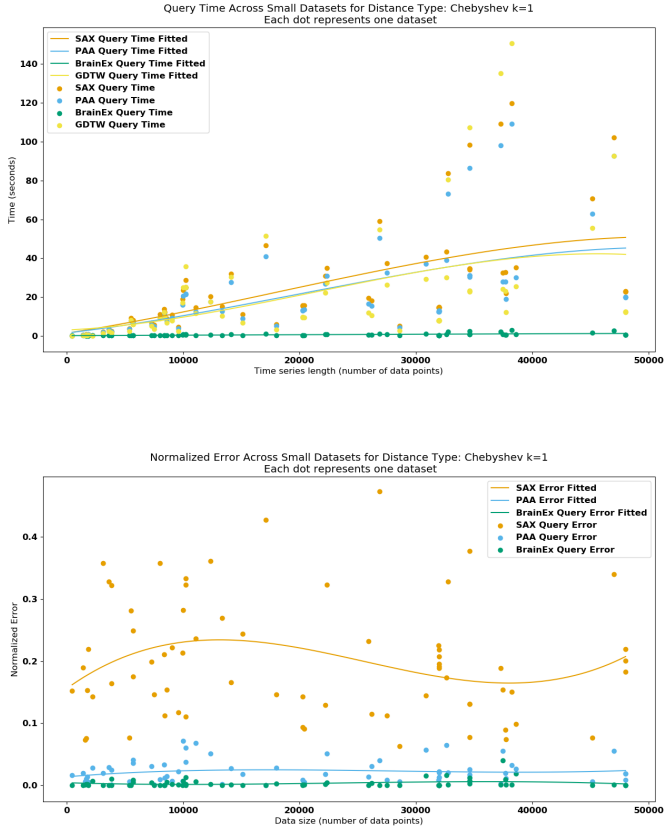

Fig. 14. (Top) The exact time (seconds) and fitted time for SAX, PAA, BrainEx, and GDTW to find the best match to a given query for warped Chebyshev distance on the small dataset bin. (Bottom) The normalized error for PAA, SAX, and BrainEx of finding the best match to a given query using the GDTW method as ground truth. This is for warped Chebyshev distance on the small dataset bin.

Table 7. Participant Demographics

This table provides demographics details for the 10 user study participants. It includes their academic positions and their self-identified expertise in several fields.

Participant	Position	fNIRS	Neural Data Analysis	HCI	Data Visualization
P1	Other	Expert	Knowledgeable	Expert	Knowledgeable
P2	PhD Candidate	Knowledgeable	Passing Knowledge	Knowledgeable	Knowledgeable
P3	Bachelor	Passing Knowledge	No Knowledge	Knowledgeable	Passing Knowledge
P4	PhD Candidate	Knowledgeable	Knowledgeable	Knowledgeable	Knowledgeable
P5	Bachelor	Knowledgeable	Knowledgeable	Passing Knowledge	Knowledgeable
P6	Post Doc	Knowledgeable	Expert	Expert	Knowledgeable
P7	Master	Passing Knowledge	Expert	Expert	Expert
P8	PhD Candidate	Knowledgeable	Passing Knowledge	Knowledgeable	Passing Knowledge
P9	PhD Candidate	Passing Knowledge	Passing Knowledge	Passing Knowledge	Passing Knowledge
P10	PhD Candidate	Knowledgeable	Passing Knowledge	Knowledgeable	Knowledgeable

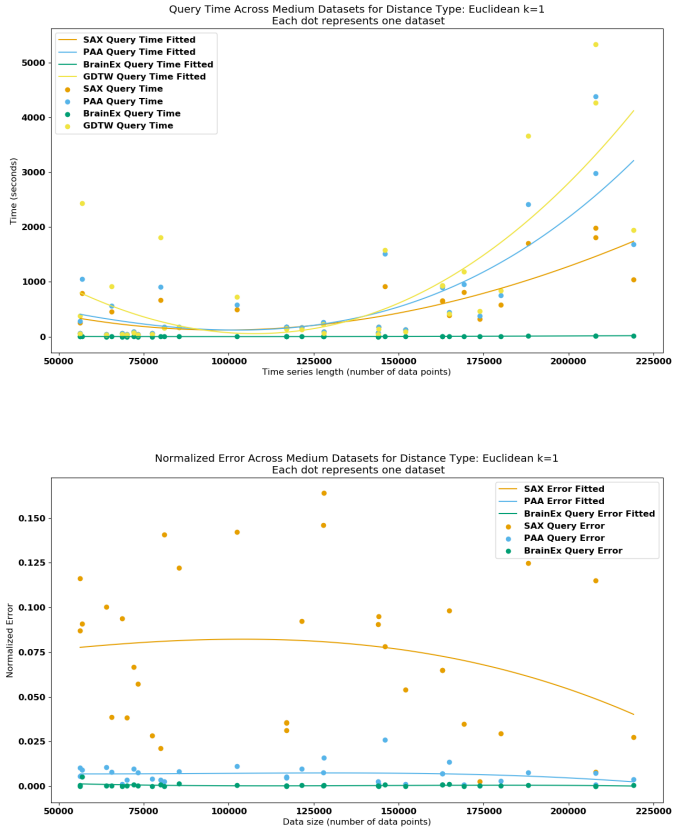

Fig. 15. (Top) The exact time (seconds) and fitted time for SAX, PAA, BrainEx, and GDTW to find the best match to a given query for warped Euclidean distance on the medium dataset bin. (Bottom) The normalized error for PAA, SAX, and BrainEx of finding the best match to a given query using the GDTW method as ground truth. This is for warped Euclidean distance on the medium dataset bin.

part of what contributed to the small sample size for this preliminary user study. It is also important to note that since the fNIRS research community is small and well-connected, the Likert scale results may experience a positive skew due to familiarity with the research team.

The dataset used for this study is smaller than most fNIRS datasets and may not be reflective of all possible brain datasets. Thus, we assume some use scenarios may result in future users interacting with *BrainEx* in ways that the study participants did not. The functional requirements defined for this paper can be abstracted from our usage scenario to cover possible use cases. In addition, the questionnaire was designed to encourage participants to explore the tool and all of its features.

6.2 User Study Results

Study participants were asked to rank their agreement with statements matching the sub-requirements discussed in Section 4.2. The Likert scale covered the range of strongly disagree to strongly agree; these were mapped to the range 1 to 5 for visualization purposes (Figure 18).

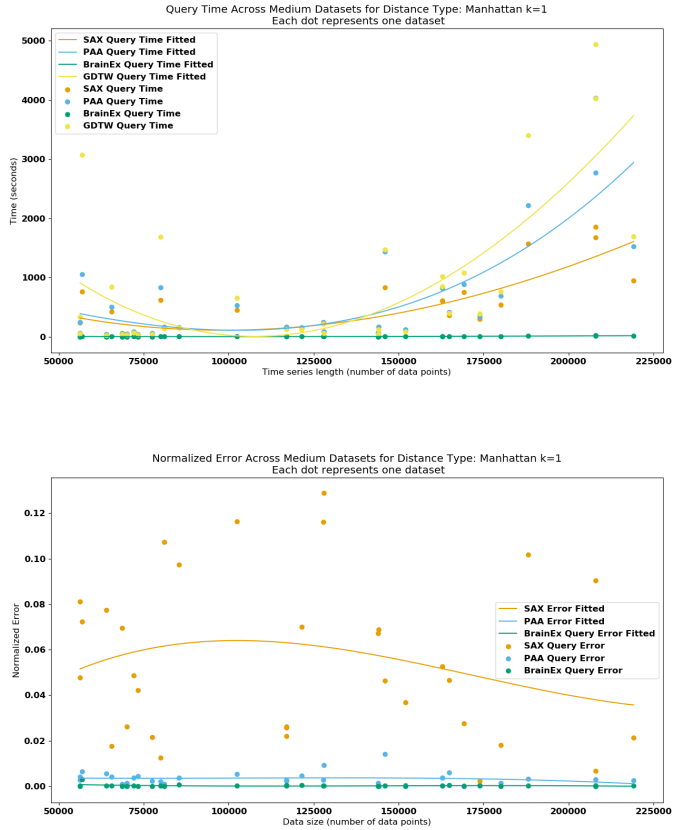

Fig. 16. (Top) The exact time (seconds) and fitted time for SAX, PAA, BrainEx, and GDTW to find the best match to a given query for warped Manhattan distance on the medium dataset bin. (Bottom) The normalized error for PAA, SAX, and BrainEx of finding the best match to a given query using the brute force method as the ground truth. This is for warped Manhattan distance on the medium dataset bin.

Requirement 1: Similarity Search. Participants agreed that *BrainEx* successfully met our functional requirements for similarity search. Participants ranked both sub-requirements very high as seen in Figure 18. Nine out of ten participants said they agreed or strongly agreed with the statements, six of the participants strongly agreed with both statements. Six of the participants (P1, P2, P4, P5, P6, P10) provided additional positive feedback in the optional text field informing us the task was very easy to perform and provided results that looked correct and useful. P10 summarized their experience: "*I loved it - very comprehensive. I liked that I could query the most important aspects of the data, and have a fine-grained level of control.*" Despite the positive feedback, P4 and P10 expressed issues with the filter feature of the similarity search.

Requirement 2: Feature Distribution Exploration. Participants were overall satisfied that the system allowed them to explore feature data through *BrainEx*. In general, the more features users attempted to explore, the less strong their agreement. In the case of exploring a single feature, all but one participant (P9) at least somewhat agreed that *BrainEx* supported them, with six participants believing

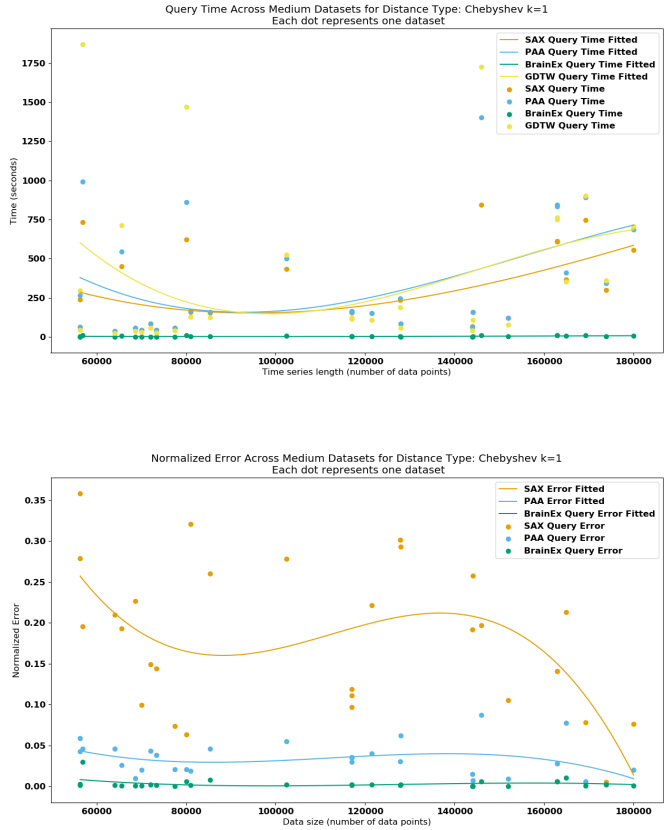

Fig. 17. (Top) The exact time (seconds) and fitted time for SAX, PAA, BrainEx, and GDTW to find the best match to a given query for warped Chebyshev distance on the medium dataset bin. (Bottom) The normalized error for PAA, SAX, and BrainEx of finding the best match to a given query using the brute force method as the ground truth. This is for warped Chebyshev distance on the medium dataset bin.

this strongly. The results were similar when considering the visualization of joint distributions. Only one participant (P9) expressed neutrality, and six participants strongly agreed.

The sub-requirements of exploring the distributions of three or more features had weaker, but still favorable results. Two participants (P7, P9) were neutral as to whether the system supported this use case. The rest agreed, but only four participants strongly agreed.

In particular, users expressed an appreciation for the options presented in *BrainEx*. P10 listed the fine level of control as a positive experience when interacting with the system. P5 found the tasks related to feature distribution more challenging than the other tasks, citing the amount of work they had to do in manually examining the data. P6 expressed confusion about the dataset presented in the trial. Despite this, they were able to use the feature-wise visualization to explore the dataset, and make statements about the different user attributes. They felt the software was "*very adaptable*," as they were not bound to specific filters. This suggests *BrainEx* may be useful to analysts who still need to learn more about their dataset of interest.

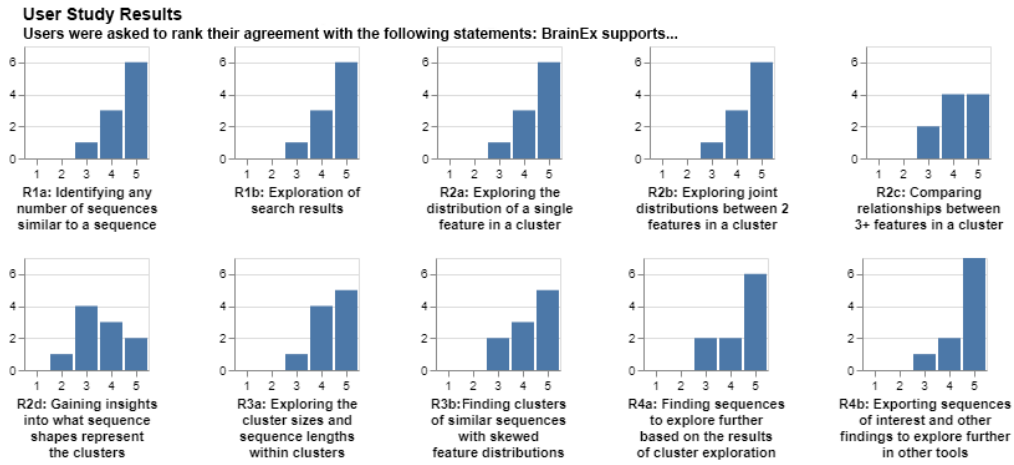

Fig. 18. The results of the user study. For each plot, the x-axis shows possible responses and the y-axis shows the frequency of each.

Requirement 3: Cluster Exploration. Both sub-requirements of the cluster exploration requirement were found to be well supported. Figure 18 shows that eight out of ten participants agreed or strongly agreed with requirements 3a and 3b. P5, P6, and P10 liked the fine control provided over what characteristics of the clusters they could view at a time when selecting the cluster. Despite the positive feedback, P2, P4, P5, P6, and P10 found the task of finding an interesting cluster overwhelming due to the number of clusters and desired more features in the cluster exploration tool.

Some of this feedback pertained to features that currently exist in the tool that participants may not have been aware of. For example, the ability to sort and filter the table to more manageable sizes exists as described in Section 4.5. Additionally, a toggle is available to show the average feature distribution for the dataset as a baseline for comparison of feature skewness in individual clusters as desired by P10.

Additional features were also suggested in the feedback to improve the efficiency of scanning the cluster list and the ability for cluster comparison. P4 and P10 suggested adding a mechanism for paging through the table quicker and to change the color gradient used for different feature groups to make the cluster table quicker to sift through and easier to separate different feature sets. To improve the ability to compare clusters, P2 suggested being able to view multiple cluster representatives at once so that they could be directly compared.

Future work can make the task of finding an interesting cluster less overwhelming and more informative. Filtering, sorting, and baseline comparison features can be made more exposed to the users and features for more powerful sorting, fine-grained filtering, and navigation through the table can be added. Additionally, the suggested features for comparing multiple clusters and cluster representatives as well as differentiating between feature groups in the table can also be added. However, the results of these survey results still indicate that *BrainEx* successfully supports the exploration of clusters within a dataset.

Requirement 4: Integration. Both of the integration requirements scored very highly. Eight out of ten participants agreed or strongly agree with requirement 4a and nine out of ten participants agreed or strongly agreed with requirement 4b. Requirement 4b was rated as strongly agree by

seven participants implying they found *BrainEx*'s ability to export findings into other tools to be very strong. The feedback for this section matched the positivity of the responses. In particular, four participants (P2, P5, P6, P10) filled out an optional text feedback section to share that they thought the integration of the cluster explorer and similarity search was very easy to use and allowed for "*exploring potentially interesting aspects of the data*" (P10). P6 rated this requirement the lowest (neither agreeing or disagreeing with the first sub-requirement), and found the user task somewhat overwhelming and hard to keep track of the patterns and data they were comparing. However, they found the concept of the integration pipeline very promising stating "The pipeline of clustering and search has a lot of potential to explore the data" (P6).

Future work could be done to polish the workflow pipeline and make it easier for analysts to remember the clusters they began the similarity search from. However, the results of this user study show that *BrainEx* successfully completes functional requirement 4 and shows that the cluster exploration and similarity search pipeline provides a novel and powerful exploration workflow for time series exploration.

Requirement 5: Accessible to All Researchers. The accessibility requirement was measured by the ability of participants to use *BrainEx* successfully during the study. Based on feedback of only two users (P2, P8) having bandwidth issues when searching for over a 1000 similar sequences, *BrainEx* supports fast computations. Additionally, based on the positive feedback for the other requirements and the varied expertise and experience of the participants, *BrainEx* can support researchers of all levels. However, improvements can be made to reduce the overwhelming nature of the presented information to further support analysts.

Usability and Additional Feedback. While not surveyed directly, based on the text feedback for each of the functional requirements, participants found *BrainEx* generally usable. Study participants particularly enjoyed the amount of detailed control they had for exploring the data as well as the ease of using the visuals. For example, P1 commented that "*very clear presentation of the results and enable users to explore different attributes*" with regards to the *Similarity Search* requirement. However, participants found the amount of information to be overwhelming at times. For example, P6 noted after the cluster exploration task that "*The number of clusters is often overwhelming*" and that "*It is not easy to identify which ones are most important to look at or to compare a specific small selection of clusters.*" BrainEx does provide filtering on the table of clusters. However, this feedback indicates that this functionality should be better exposed and expanded to allow the user more control over managing the data. While participants did not find severe usability issues with the system, there is room for improvement to expose hidden features, distinguish and clarify elements of the user interface, and allow for better management of large amounts of information.

7 FNIRS CASE STUDY

To further illustrate the potential of *BrainEx*, we describe a case study using real-world experiment data from an fNIRS study on cognitive control [31]. This dataset has been analyzed in more traditional methods with mixed effects modeling and investigating the sequence shape. In the study, participants performed the AX-Continuous Performance Task (AX-CPT) which induces different cognitive control states, such as proactive and reactive control [6]. The study concluded that proactive and reactive cognitive control can both be seen in the right dorsolateral prefrontal cortex.

For this case study, we developed a model of the expected fNIRS brain signal when a participant is in each of the two cognitive control states that we were studying. These *model sequences* were based on the expected hemodynamic response, given the tasks and timing of stimuli in the dataset

[6]. Using these model signals, we can explore the dataset's clusters to find connections to particular sequences in the dataset.

The goal of this case study was to use *BrainEx*'s cluster exploration to investigate which events and brain regions are associated with the model cognitive control sequences. We identified clusters that had higher-than-average representation of subsequences from the model cognitive control sequences. Our hypothesis was that clusters with a larger number of the model subsequences would contain brain data related to those cognitive states from the AX-CPT task. Therefore, if there was a different distribution of events or brain regions in these clusters compared to the master dataset, then we have identified which events and brain regions are most associated with cognitive control. We expect the cluster results to be different than Howell-Munson et al.'s results because they detected both cognitive control states in the *same* region, while we are looking for clusters that indicate a specific region for each cognitive control state [31].

7.1 Dataset Description

The dataset contained the two model cognitive control sequences along with 3,360 sequences from one participant divided between six brain regions. Each sequence is 157 datapoints long and spans approximately 18 seconds of neural data for a total of 527,834 points in the dataset. Along with the brain signal, there is associated metadata for each signal consisting of the *subject name*, *brain region*, *event name*, *start time*, and *end time*. The model sequences have a subject name of "representative" and a region name of "Channel-0" to designate them as different from the collected data. The collected data has six possible region names from the prefrontal cortex (PFC): dorsomedial (DMPFC), left dorsolateral (lDLPFC), right dorsolateral (rDLPFC), ventromedial (VMPFC), left orbitofrontal (lOFPFC), and right orbitofrontal (rOFPFC). There are six possible events with the percentage of the dataset they occupy in parentheses: AX cue (30%), AY cue (20%), BX cue (20%), BY cue (20%), A# cue (5%), and B# cue (5%). These names are associated with the trials in the task, and the details can be found in [6, 31]. We are most interested in AY cue and BX cue, as they can be indications of proactive and reactive cognitive control [6]. Start time is the time in milliseconds when the sequence begins in relation to data collection, and end time is the time in milliseconds when the sequence ended in relation to data collection. The dataset is available here: https://wp.wpi.edu/hcilab/brainex/.

We preprocessed the dataset using the Warped Euclidean distance metric, a similarity threshold of 0.1, and a length of interest of 1-157. In the cluster exploration, we only viewed clusters with a minimum of 20 sequences and a minimum length of 45 (approximately 5 seconds of brain data).

7.2 Case Study Results

To ensure the clusters we sampled had a large number of the model cognitive control sequences, we analyzed clusters that were at least 0.15% model sequences, which included 49 clusters. The cluster with the most model sequences had 0.56% which is 11 times greater than the distribution of model sequences in the master dataset. While the ratio of model to participant data is small, this is to be expected as there are 3,780 sequences in the master dataset of participant data and 2 model sequences, making the master dataset consist of 0.05% model sequences.

We used a Chi-square test to determine if the distribution of events and channels in the 49 clusters differed significantly from the distribution in the master dataset. There were a total of 440,895 sequences in the clustered data; the expected and observed distribution of events are located in Table 8. Our Chi-square statistic was 10,184, and with 5 degrees of freedom we can reject the null hypothesis ($p < 0.05$) and say the distribution of events in the clusters differs significantly from the master dataset. Notably, all events that start with an A cue appeared less frequently in the clusters than expected, and all events that started with a B cue appeared more frequently in

Table 8. Expected and observed distributions of events from 49 clusters that have high inclusion of modeled cognitive control sequences.

Event	AX	AY	A#	BX	BY	B#
Expected	132,268	88,179	22,044	88,179	88,179	22,044
Observed	127,083	71,795	16,229	92,471	106,561	27,524

Table 9. Expected and observed distributions of channels from 49 clusters that have high inclusion of modeled cognitive control sequences.

Region	DMPFC	lDLPFC	rDLPFC	VMPFC	lOFPFC	rOFPFC
Expected	82,888	62,166	62,166	103,610	62,166	62,166
Observed	80,208	65,808	61,023	110,237	63,139	59,353

the clusters than expected. The B cue could be indicative of reactive control, one of the modeled cognitive control sequences. Therefore, we can associate these clusters with reactive control for future similarity searches.

The observed and expected distribution for each region in the brain can be found in Table 9. Here we also had 5 degrees of freedom and our Chi-square statistic was 887, showing that we can reject the null hypothesis ($p < 0.05$) and say the distribution of brain regions in the clusters differs significantly from the master dataset. Notably, the left hemisphere and VMPFC had a higher frequency in the clustered data while the right hemisphere and DMPFC had a lower frequency in the dataset.

7.3 Case Study Conclusion

Sometimes a researcher may not know which sequences in a dataset are particularly significant, or which subsequence is a crucial element in their dataset. Through the use of the cluster exploration feature in *BrainEx*, researchers can explore the distribution of the dataset and which sequences are most similar, teasing out meaningful patterns. Through our case study, we demonstrated how one can discover events and brain regions that are correlated with model sequences. The cluster exploration can also help identify which parts of the modeled sequence were most informative by investigating the subsequences that appeared most frequently. For example, in the master dataset, all of the sequences were 157 datapoints long (18 seconds). However, the average sequence length of the clusters was only 80 datapoints (9 seconds) with a maximum sequence length of 126 (14.5 seconds) and minimum sequence length of 45 (5 seconds). By using this information, a researcher can fine-tune their query to be more meaningful to their research questions when using the similarity search feature of *BrainEx*.

8 DISCUSSION

Previous research teams have provided basic visual interfaces for DTW based engines to address the increased for interpretability and accessibility by researchers without expertise in using command-line interfaces and APIs (e.g. [51]). These basic visual interfaces for data exploration tools mark a step towards making similarity searches more available [28, 49, 59, 73]; other work focuses on visualizing the results of clustering [27, 39]. BrainEx fills a gap in this field by providing a

comparison of results across multiple elastic distances and offering insights through combined similarity searches and cluster exploration.

Due to the growing popularity of time series data, there are many other time series exploration tools [25]. *TimeSearcher* provides an interactive similarity search of time series [28]. This tool allows analysts to see a line plot depiction of time series in a dataset. Analysts can use a drag and drop box, known as a *SearchBox*, to select a part of a time series representing an interesting pattern. This pattern can then be queried to discover similar patterns in other sequences. It supports exploration of multivariate data. While it provides a similarity search feature similar to BrainEx, *TimeSearcher* does not provide the additional cluster exploration workflow. *QueryLines* is a similarity search tool for time series data that allows analysts to specify soft constraints and preferences which are then used to perform a similarity search on other sequences to discover matching or almost matching patterns [59]. These constraints are added by drawing lines to show the pattern you wish to match. Work by Buono and Simeone into extending the *SearchBox* of *TimeSearcher* showed that drawing a query is an increasingly popular approach in time series similarity searches [8]. Like *TimeSearcher*, *QueryLines* is a similarity search tool and does not provide cluster exploration. *Similan* is a visual similarity search tool for temporal data [73]. It was designed to use a similarity measure called "match and mismatch" to account for temporally misaligned records. The *Similan* researchers propose clustering as a future feature.

Not all time series tools focus on similarity search. Himberg, Hyvarinen, and Espositoc [27] designed a neuroimaging cluster visualization tool using independent component analysis. Kumar et al. proposed a time series clustering tool that represents clusters using a bitmap [39]. These bitmaps can be used for pattern recognition of time series datasets. These tools provide interesting approaches to exploring time series clusters, but do not allow for exploration via similarity search.

8.1 Future Work

Future work on the *BrainEx* engine could focus on making the preprocessing step of BrainEx even more efficient through converting the code into another language, such as Rust. Additionally, BrainEx could replicate studies with fNIRS curated specifically for validating tools, such as the n-back dataset from Wang et al. [71]. While we used the original versions of SAX and PAA to do our benchmark comparison with BrainEx, newer versions of these algorithms exist and can be tested against BrainEx [65, 78]. These versions were out of the scope of the experiment presented in Section 5.

In addition, further refinement and evaluation of the interactive visual interface could improve the user experience. Future user studies could aim to provide the participants with a larger fNIRS dataset to explore with the interface. In addition, a larger cohort of participants could be recruited from a more diverse set of expertise to be able to investigate the differences in the usability of the tool between experts in data visualization and neuroscience compared to researchers who are just starting their scientific careers.

To promote collaborative research and accessibility, we created a website (https://wp.wpi.edu/hcilab/brainex/ to make the BrainEx code available to researchers. In addition, the results from the performance experiment and the clusters from the case study can be found there.

9 CONCLUSION

We present BrainEx as a tool for visual exploration of brain signals. By combining cluster exploration, feature distribution exploration, and similarity search we provide a powerful and novel exploration workflow that existing fNIRS and time series analysis tools do not provide. In our performance experiment, we demonstrated how BrainEx is lightning fast compared to state of the art competitors as well as highly accurate. We developed five functional user requirements, and based on the results

of a preliminary user study with HCI and neuroscience researchers, we determined BrainEx meets these requirements. Finally, we used a case study to demonstrate how a researcher could use BrainEx to make inferences about real-world fNIRS brain data. Overall, we have shown that BrainEx could be an effective tool for fNIRS or other neuroscience researchers.

ACKNOWLEDGMENTS

The work is partially supported by the U.S. National Science Foundation under Grant No. 1835307 and a WPI TRIAD grant. We would also like to thank Eric Schmid, James Plante, Yufei Lin, Ellery Buntel, and the GENEX team for their help in this research.

REFERENCES

[1] John Aach and George M Church. 2001. Aligning gene expression time series with time warping algorithms. *Bioinformatics* 17, 6 (2001), 495–508.

[2] Daniel Afergan, Samuel W Hincks, Tomoki Shibata, and Robert JK Jacob. 2015. Phylter: a system for modulating notifications in wearables using physiological sensing. In *International conference on augmented cognition*. Springer, 167–177.

[3] Elizabeth M. Argyle, Adrian Marinescu, Max L. Wilson, Glyn Lawson, and Sarah Sharples. 2021. Physiological indicators of task demand, fatigue, and cognition in future digital manufacturing environments. *International Journal of Human-Computer Studies* 145 (2021), 102522. https://doi.org/10.1016/j.ijhcs.2020.102522

[4] Danushka Bandara, Senem Velipasalar, Sarah Bratt, and Leanne Hirshfield. 2018. Building predictive models of emotion with functional near-infrared spectroscopy. *International Journal of Human-Computer Studies* 110 (2018), 75 – 85. https://doi.org/10.1016/j.ijhcs.2017.10.001

[5] Donald J. Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining* (Seattle, WA) *(AAAIWS'94)*. AAAI Press, 359–370.

[6] Todd S Braver. 2012. The variable nature of cognitive control: a dual mechanisms framework. *Trends in cognitive sciences* 16, 2 (2012), 106–113.

[7] Andrei Z Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Zien. 2003. Efficient query evaluation using a two-level retrieval process. In *Proceedings of the twelfth international conference on Information and knowledge management*. 426–434.

[8] Paulo Buono and Adalberto Lafcadio Simeone. 2008. Interactive shape specification for pattern search in time series. *AVI '08: Proceedings of the working conference on Advanced visual interfaces* (2008), 480–481.

[9] Enrico Gianluca Caiani, A Porta, Giuseppe Baselli, M Turiel, S Muzzupappa, F Pieruzzi, C Crema, A Malliani, and Sergio Cerutti. 1998. Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume. In *Computers in Cardiology 1998. Vol. 25 (Cat. No. 98CH36292)*. IEEE, 73–76.

[10] Francisco M Calisto, Alfredo Ferreira, Jacinto C Nascimento, and Daniel Gonçalves. 2017. Towards touch-based medical image diagnosis annotation. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 390–395.

[11] Francisco Maria Calisto, Nuno Nunes, and Jacinto C Nascimento. 2020. Breastscreening: On the use of multi-modality in medical imaging diagnosis. In *Proceedings of the International Conference on Advanced Visual Interfaces*. 1–5.

[12] Francisco Maria Calisto, Carlos Santiago, Nuno Nunes, and Jacinto C Nascimento. 2021. Introduction of human-centric AI assistant to aid radiologists for multimodal breast image classification. *International Journal of Human-Computer Studies* 150 (2021), 102607.

[13] Sung-Hyuk Cha. 2007. Comprehensive survey on distance/similarity measures between probability density functions. *City* 1, 2 (2007), 1.

[14] Britton Chance, Endla Anday, Shoko Nioka, Shuoming Zhou, Long Hong, Katherine Worden, C Li, T Murray, Y Ovetsky, D Pidikiti, et al. 1998. A novel method for fast imaging of brain function, non-invasively, with light. *Optics express* 2, 10 (1998), 411–423.

[15] WA Chaovalitwongse and PM Pardalos. 2008. On the time series support vector machine using dynamic time warping kernel for brain activity classification. *Cybernetics and Systems Analysis* 44, 1 (2008), 125–138.

[16] William S Cleveland and Robert McGill. 1984. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association* 79, 387 (1984), 531–554.

[17] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Annh Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.

[18] Luciënne A. de With, Nattapong Thammasan, and Mannes Poel. 2022. Detecting Fear of Heights Response to a Virtual Reality Environment Using Functional Near-Infrared Spectroscopy. *Frontiers in Computer Science* 3 (2022). https://doi.org/10.3389/fcomp.2021.652550

[19] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. 2008. Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. *Proc. VLDB Endow.* 1, 2 (Aug. 2008), 1542–1552. https://doi.org/10.14778/1454159.1454226

[20] Jayesh Dubey, Mihin Sumaria, Erden Oktay, Yu Li, Ziheng Li, Rodica Neamtu, and Erin T Solovey. 2019. Towards Neuroadaptive Technology Using Time Warped Distances for Similarity Exploration of Brain Data. In *The Second Neuroadaptive Technology Conference*. 48.

[21] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. 1994. Fast subsequence matching in time-series databases. *Acm Sigmod Record* 23, 2 (1994), 419–429.

[22] Johann Faouzi and Hicham Janati. 2020. pyts: A Python Package for Time Series Classification. *Journal of Machine Learning Research* (2020), 1–6.

[23] M Ferrari and V Quaresima. 2012. A brief review on the history of human functional near-infrared spectroscopy (fNIRS) development and fields of application. *NeuroImage* 63 (2012), 921–935.

[24] Toni Giorgino. 2009. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *Journal of statistical Software* (2009), 1–24.

[25] Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, and Anastasia Bezerianos. 2019. Comparing Similarity Perception in Time Series Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 25 (2019), 523–533.

[26] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S Hämäläinen. 2014. MNE software for processing MEG and EEG data. *Neuroimage* 86 (2014), 446–460.

[27] Johan Himberg, Aapo Hyvärinen, and Fabrizio Esposito. 2004. Validating the independent components of neuroimaging time series via clustering and visualization. *Neuroimage* 22, 3 (2004), 1214–1222.

[28] H. Hochheiser and B. Shneiderman. 2004. Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration. *Information Visualization* 3 (2004). Issue 1.

[29] L Hocke, I Oni, C Duszynski, A Corrigan, B Frederick, and J Dunn. 2018. Automated Processing of fNIRS Data—A Visual Guide to the Pitfalls and Consequences. Algorithms. *Algorithms* 11 (2018).

[30] Xin Hou, Zong Zhang, Chen Zhao, Lian Duan, Yilong Gong, Zheng Li, and Chaozhe Zhu. 2021. NIRS-KIT: a MATLAB toolbox for both resting-state and task fNIRS data analysis. *Neurophotonics* 8, 1 (January 2021), 010802. https://doi.org/10.1117/1.nph.8.1.010802

[31] Alicia Howell-Munson, Deniz Sonmez, Erin Walker, Catherine Arrington, and Erin T. Solovey. 2021. Preliminary steps towards detection of proactive and reactive control states during learning with fNIRS brain signals. In *Proceedings of the First International Workshop on Multimodal Artificial Intelligence in Education (MAIED 2021)*, Vol. 2902.

[32] Theodore J Huppert, Solomon G Diamond, Maria A Franceschini, and David A Boas. 2009. HomER: a review of time-series analysis methods for near-infrared spectroscopy of the brain. *Applied optics* 48, 10 (April 2009), D280—98. https://doi.org/10.1364/ao.48.00d280

[33] A. Inselberg and B. Dimsdale. 1990. Parallel coordinates: a tool for visualizing multi-dimensional geometry. *Proceedings of the First IEEE Conference on Visualization: Visualization '90* (1990), 361–378.

[34] Weiwei Jiang, Zhanna Sarsenbayeva, Niels van Berkel, Chaofan Wang, Difeng Yu, Jing Wei, Jorge Goncalves, and Vassilis Kostakos. 2021. User Trust in Assisted Decision-Making Using Miniaturized Near-Infrared Spectroscopy. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 153, 16 pages. https://doi.org/10.1145/3411764.3445710

[35] Emre Karakoc, Artem Cherkasov, and S Cenk Sahinalp. 2006. Distance based algorithms for small biomolecule classification and structural similarity search. *Bioinformatics* 22, 14 (2006), e243–e251.

[36] Rohit J Kate. 2016. Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery* 30, 2 (2016), 283–312.

[37] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems* 3, 3 (2001), 263–286.

[38] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and information systems* 7, 3 (2005), 358–386.

[39] Nitin Kumar, Venkata Nishanth Lolla, Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, and Li Wei. 2005. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 531–535.

[40] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. 2–11.

[41] Michael Lührs and Rainer Goebel. 2017. Turbo-Satori: a neurofeedback and brain–computer interface toolbox for real-time functional near-infrared spectroscopy. *Neurophotonics* 4, 4 (2017), 041504.

[42] Kristiyan Lukanov, Horia A. Maior, and Max L. Wilson. 2016. Using fNIRS in Usability Testing: Understanding the Effect of Web Form Layout on Mental Workload. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 4011–4016. https://doi.org/10.1145/2858036.2858236

[43] Horia A Maior, Richard Ramchurn, Sarah Martindale, Ming Cai, Max L Wilson, and Steve Benford. 2019. fNIRS and Neurocinematics. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[44] Horia A. Maior, Max L. Wilson, and Sarah Sharples. 2018. Workload Alerts—Using Physiological Measures of Mental Workload to Provide Feedback During Tasks. *ACM Trans. Comput.-Hum. Interact.* 25, 2, Article 9 (apr 2018), 30 pages. https://doi.org/10.1145/3173380

[45] AI Makarova and VV Sulimova. 2017. High-performance DTW-based signals comparison for the brain electroencephalograms analysis. In *CEUR Workshop Proceedings*. 18–24.

[46] Serena Midha, Horia A. Maior, Max L. Wilson, and Sarah Sharples. 2021. Measuring Mental Workload Variations in Office Work Tasks using fNIRS. *International Journal of Human-Computer Studies* 147 (2021), 102580. https://doi.org/10.1016/j.ijhcs.2020.102580

[47] Tamara Munzner. 2014. *Visualization Analysis and Design*. CRC press.

[48] Gattigorla Nagendar and CV Jawahar. 2015. Efficient word image retrieval using fast DTW distance. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 876–880.

[49] Rodica Neamtu, Ramoza Ahsan, Charles Lovering, Cuong Nguyen, Elke Rundensteiner, and Gabor Sarkozy. 2017. Interactive time series analytics powered by ONEX. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1595–1598.

[50] Rodica Neamtu, Ramoza Ahsan, Cuong Dinh Tri Nguyen, Charles Lovering, Elke A Rundensteiner, and Gabor Sarkozy. 2020. A General Approach For Supporting Time Series Matching using Multiple-Warped Distances. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[51] Rodica Neamtu, Ramoza Ahsan, Elke Rundensteiner, and Gabor Sarkozy. 2016. Interactive time series exploration powered by the marriage of similarity distances. *Proceedings of the VLDB Endowment* 10, 3 (2016), 169–180.

[52] Rodica Neamtu, Ramoza Ahsan, Elke A Rundensteiner, Gabor Sarkozy, Eamonn Keogh, Hoang Anh Dau, Cuong Nguyen, and Charles Lovering. 2018. Generalized dynamic time warping: Unleashing the warping power hidden in point-wise distances. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 521–532.

[53] Cuong Nguyen, Charles Lovering, and Rodica Neamtu. 2017. Ranked time series matching by interleaving similarity distances. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 3530–3539.

[54] Aaron A Phillips, Franco HN Chan, Mei Mu Zi Zheng, Andrei V Krassioukov, and Philip N Ainslie. 2016. Neurovascular coupling in humans: Physiology, methodological advances and clinical implications. *Journal of Cerebral Blood Flow & Metabolism* 36, 4 (2016), 647–664. https://doi.org/10.1177/0271678X15617954 arXiv:https://doi.org/10.1177/0271678X15617954 PMID: 26661243.

[55] Kathrin Pollmann, Daniel Ziegler, Matthias Peissner, and Mathias Vukelić. 2017. A New Experimental Paradigm for Affective Research in Neuro-Adaptive Technologies. In *Proceedings of the 2017 ACM Workshop on An Application-Oriented Approach to BCI out of the Laboratory* (Limassol, Cyprus) *(BCIforReal '17)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3038439.3038442

[56] Spencer R. Pruitt. 2018. High Performance Computing. https://arc.wpi.edu/computing/hpc-clusters/

[57] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 262–270.

[58] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2013. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 7, 3 (2013), 1–31.

[59] Kathy Ryall, Neal Lesh, Tom Lanning, Darren Leigh, Hiroaki Miyashita, and Shigeru Makino. 2005. QueryLines: Approximate Query for Visual Browsing. *CHI'05 Extended Abstracts on Human Factors in Computing Systems* (2005).

[60] Hendrik Santosa, Xuetong Zhai, Frank Fishburn, and Theodore Huppert. 2018. The NIRS brain AnalyzIR toolbox. *Algorithms* 11, 5 (2018), 73.

[61] Pavel Senin. 2008. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855, 1-23 (2008), 40.

[62] Michael John Sebastian Smith. 1997. *Application-specific integrated circuits*. Vol. 7. Addison-Wesley Reading, MA.

[63] Erin T. Solovey and Felix Putze. 2021. *Improving HCI with Brain Input: Review, Trends, and Outlook*. Now Publishers Inc.

[64] Markus Andreas Stricker and Markus Orengo. 1995. Similarity of color images. In *Storage and retrieval for image and video databases III*, Vol. 2420. International Society for Optics and Photonics, 381–392.
[65] Youqiang Sun, Jiuyong Li, Jixue Liu, Bingyu Sun, and Christopher Chow. 2014. An improvement of symbolic aggregate approximation distance measure for time series. *Neurocomputing* 138 (2014), 189–198.
[66] Stephanie Sutoko, Hiroki Sato, Atsushi Maki, Masashi Kiguchi, Yukiko Hirabayashi, Hirokazu Atsumori, Akiko Obata, Tsukasa Funane, and Takusige Katura. 2016. Tutorial on platform for optical topography analysis tools. *Neurophotonics* 3, 1 (January 2016), 010801. https://doi.org/10.1117/1.nph.3.1.010801
[67] Michael J Swain and Dana H Ballard. 1991. Color indexing. *International journal of computer vision* 7, 1 (1991), 11–32.
[68] Kazuaki Tanida. 2019. [Online]. fastdtw 0.3.4. PyPI. https://pypi.org/project/fastdtw/
[69] Arno Villringer, J Planck, C Hock, L Schleinkofer, and U Dirnagl. 1993. Near infrared spectroscopy (NIRS): a new tool to study hemodynamic changes during activation of brain function in human adults. *Neuroscience letters* 154, 1-2 (1993), 101–104.
[70] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. 2003. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 216–225.
[71] Liang Wang, Zhe Huang, Ziyu Zhou, Devon McKeon, Giles Blaney, Michael C. Hughes, and Robert J. K. Jacob. 2021. *Taming FNIRS-Based BCI Input for Better Calibration and Broader Use*. Association for Computing Machinery, New York, NY, USA, 179–197. https://doi.org/10.1145/3472749.3474743
[72] Linkai Weng, Zhiwei Li, Rui Cai, Yaoxue Zhang, Yuezhi Zhou, Laurence T Yang, and Lei Zhang. 2011. Query by document via a decomposition-based two-level retrieval approach. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 505–514.
[73] Krist Wongsuphasawat and Ben Shneiderman. 2009. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *2009 IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 27–34.
[74] Hiroo Yamamura, Holger Baldauf, and Kai Kunze. 2020. Pleasant Locomotion–Towards Reducing Cybersickness using fNIRS during Walking Events in VR. In *Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 56–58.
[75] Jong Chul Ye, Sungho Tak, Kwang Eun Jang, Jinwook Jung, and Jaeduck Jang. 2009. NIRS-SPM: Statistical parametric mapping for near-infrared spectroscopy. *NeuroImage* 44, 2 (2009), 428–447. https://doi.org/10.1016/j.neuroimage.2008.08.036
[76] Byoung-Kee Yi and Christos Faloutsos. 2000. Fast time sequence indexing for arbitrary Lp norms. (2000).
[77] Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. 2016. Apache spark: a unified engine for big data processing. *Commun. ACM* 59, 11 (2016), 56–65.
[78] Chaw Thet Zan and Hayato Yamana. 2016. An improved symbolic aggregate approximation distance measure based on its statistical features. In *Proceedings of the 18th international conference on information integration and web-based applications and services*. 72–80.

A UCR ARCHIVE TIME-SERIES DATASETS

Name	Size	Bin	Completed (Y/N)
ACSF1	146,000	Medium	Y
Adiac	60,853	Medium	Y
AllGestureWiimoteX	37,473	Small	Y
AllGestureWiimoteY	37,473	Small	Y
AllGestureWiimoteZ	37,473	Small	Y
ArrowHead	9,036	Small	Y
Beef	14,100	Small	Y
BeetleFly	10,240	Small	Y
BirdChicken	10,240	Small	Y
BME	3,840	Small	Y
Car	34,620	Small	Y
CBF	3,840	Small	Y

Chinatown	480	Small	Y
ChlorineConcentration	77,522	Medium	Y
CinCECGTorso	65,562	Medium	Y
Coffee	8,008	Small	Y
Computers	180,000	Medium	Y
CricketX	117,000	Medium	Y
CricketY	117,000	Medium	Y
CricketZ	117,000	Medium	Y
Crop	331,200	Medium	N
DiatomSizeReduction	5,520	Small	Y
DistalPhalanxOutlineAgeGroup	32,000	Small	Y
DistalPhalanxOutlineCorrect	48,000	Small	Y
DistalPhalanxTW	32,000	Small	Y
DodgerLoopDay	22,228	Small	Y
DodgerLoopGame	5,695	Small	Y
DodgerLoopWeekend	5,722	Small	Y
Earthquakes	164,864	Medium	Y
ECG200	9,600	Small	Y
ECG5000	70,000	Medium	Y
ECGFiveDays	3,128	Small	Y
ElectricDevices	856,896	Medium	N
EOGHorizontalSignal	452,500	Medium	N
EOGVerticalSignal	452,500	Medium	N
EthanolLevel	882,504	Medium	N
FaceAll	73,360	Medium	Y
FaceFour	8,400	Small	Y
FacesUCR	26,200	Small	Y
FiftyWords	121,500	Medium	Y
Fish	81,025	Medium	Y
FordA	1,800,500	Large	N
FordB	1,818,500	Large	N
FreezerRegularTrain	45,150	Small	Y
FreezerSmallTrain	8,428	Small	Y
Fungi	3,618	Small	Y
GestureMidAirD1	34,623	Small	Y
GestureMidAirD2	34,623	Small	Y
GestureMidAirD3	34,623	Small	Y
GesturePebbleZ1	30,850	Small	Y
GesturePebbleZ2	32,630	Small	Y
GunPoint	7,500	Small	Y
GunPointAgeSpan	20,250	Small	Y
GunPointMaleVersusFemale	20,250	Small	Y
GunPointOldVersusYoung	20,400	Small	Y
Ham	46,979	Small	Y
HandOutlines	2,709,000	Large	N

Haptics	169,200	Medium	Y
Herring	32,768	Small	Y
HouseTwenty	80,000	Medium	Y
InlineSkate	188,200	Medium	Y*
InsectEPGRegularTrain	37,262	Small	Y
InsectEPGSmallTrain	10,217	Small	Y
InsectWingbeatSound	56,320	Medium	Y
ItalyPowerDemand	1,608	Small	Y
LargeKitchenAppliances	270,000	Medium	N
Lightning2	38,220	Small	Y
Lightning7	22,330	Small	Y
Mallat	56,320	Medium	Y
Meat	26,880	Small	Y
MedicalImages	37,719	Small	Y
MelbournePedestrian	28,574	Small	Y
MiddlePhalanxOutlineAgeGroup	32,000	Small	Y
MiddlePhalanxOutlineCorrect	48,000	Small	Y
MiddlePhalanxTW	31,920	Small	Y
MixedShapesRegularTrain	512,000	Medium	N
MixedShapesSmallTrain	102,400	Medium	Y
MoteStrain	1,680	Small	Y
NonInvasiveFetalECGThorax1	1,350,000	Medium	N
NonInvasiveFetalECGThorax2	1,350,000	Meidum	N
OliveOil	17,100	Small	Y
OSULeaf	85,400	Medium	Y
PhalangesOutlinesCorrect	144,000	Medium	Y
Phoneme	219,136	Medium	Y*
PickupGestureWiimoteZ	7,294	Small	Y
PigAirwayPressure	208,000	Medium	Y*
PigArtPressure	208,000	Medium	N
PigCVP	208,000	Medium	Y*
PLAID	173,858	Medium	Y
Plane	15,120	Small	Y
PowerCons	25,920	Small	Y
ProximalPhalanxOutlineAgeGroup	32,000	Small	Y
ProximalPhalanxOutlineCorrect	48,000	Small	Y
ProximalPhalanxTW	32,000	Small	Y
RefrigerationDevices	270,000	Medium	N
Rock	56,880	Medium	Y
ScreenType	270,000	Medium	N
SemgHandGenderCh2	450,000	Medium	N
SemgHandMovementCh2	675,000	Medium	N
SemgHandSubjectCh2	675,000	Medium	N
ShakeGestureWiimoteZ	8,594	Small	Y
ShapeletSim	10,000	Small	Y

ShapesAll	307,200	Medium	N
SmallKitchenAppliances	270,000	Medium	N
SmoothSubspace	2,250	Small	Y
SonyAIBORobotSurface1	1,400	Small	Y
SonyAIBORobotSurface2	1,755	Small	Y
StarLightCurves	1,024,000	Medium	N
Strawberry	144,055	Medium	Y
SwedishLeaf	64,000	Medium	Y
Symbols	9,950	Small	Y
SyntheticControl	18,000	Small	Y
ToeSegmentation1	11,080	Small	Y
ToeSegmentation2	12,348	Small	Y
Trace	27,500	Small	Y
TwoLeadECG	1,886	Small	Y
TwoPatterns	128,000	Medium	Y
UMD	5,400	Small	Y
UWaveGestureLibraryAll	846,720	Medium	N
UWaveGestureLibraryX	282,240	Medium	N
UWaveGestureLibraryY	282,240	Medium	N
UWaveGestureLibraryZ	282,240	Medium	N
Wafer	152,000	Medium	Y
Wine	13,338	Small	Y
WordSynonyms	72,090	Medium	Y
Worms	162,900	Medium	Y
WormsTwoClass	162,900	Medium	Y
Yoga	127,000	Medium	Y

*These datasets were completed with the warped Euclidean and warped Manhattan distances but not the warped Chebyshev.

Received February 2022; accepted April 2022