Discovering Synchronized Subsets of Sequences: A Large Scale Solution

Evangelos Sariyanidi[1], Casey J. Zampella[1], Keith G. Bartley[1], John D. Herrington[1,2],
Theodore D. Satterthwaite[2], Robert T. Schultz[1,2] and Birkan Tunc[1,2]
[1]Center for Autism Research, Children's Hospital of Philadelphia [2]University of Pennsylvania
{sariyanide,zampellac,bartleyg,herringtonj,schultzrt,tuncb}@email.chop.edu

Abstract

*Finding the largest subset of sequences (*i.e.*, time series) that are correlated above a certain threshold, within large datasets, is of significant interest for computer vision and pattern recognition problems across domains, including behavior analysis, computational biology, neuroscience, and finance. Maximal clique algorithms can be used to solve this problem, but they are not scalable. We present an approximate, but highly efficient and scalable, method that represents the search space as a union of sets called ϵ-expanded clusters, one of which is theoretically guaranteed to contain the largest subset of synchronized sequences. The method finds synchronized sets by fitting a Euclidean ball on ϵ-expanded clusters, using Jung's theorem. We validate the method on data from the three distinct domains of facial behavior analysis, finance, and neuroscience, where we respectively discover the synchrony among pixels of face videos, stock market item prices, and dynamic brain connectivity data. Experiments show that our method produces results comparable to, but up to 300 times faster than, maximal clique algorithms, with speed gains increasing exponentially with the number of input sequences.*

1. Introduction

Synchrony is observed in countless phenomena and its discovery is central to answering many computer vision and pattern recognition problems. However, a basic research problem has remained surprisingly underexplored: Given a set of sequences (*i.e.*, time series), how can we discover the largest subset in which every pair of sequences satisfies a synchrony criterion? This problem naturally arises in many domains. For example, in a video, only movement in a subset of pixels is synchronized in time, and their identification can lead to discovering events of interest (*e.g.*, facial expressions [58, 54]). In the stock market, the prices of only certain and a priori unknown assets are correlated. The brain is dynamically organized into subsets of neurons (*i.e.*, modules, systems) that have synchronous activation. Indeed, researchers have used the synchrony among sequences to mine financial data [4, 29], define the functional architecture of the brain [22, 6], and identify community structure in complex networks [36, 31]. Previous research in this area has commonly employed maximal clique algorithms. However, despite their widespread use, maximal clique algorithms have exponential complexity [48], which imposes a barrier on possible research questions that can be targeted.

In this paper, we propose an efficient and scalable framework to approximately solve the following problem: Given a set of sequences, find the largest subset such that all pairs of sequences in that subset are correlated above a given threshold. The framework first clusters all input sequences, and then refines the clusters until all sequences within the clusters are synchronized (*i.e.*, pairwise correlated). Prior to refining, the clusters are enlarged to obtain so-called ϵ-*expanded clusters*. We call our method SyncRef, as it finds synchronized subsets through refining. Herein, we demonstrate SyncRef's versatility through experiments on data from three distinct domains, namely computer vision (motion of video pixels), finance (price of market assets), and neuroscience (neuronal activity). These experiments clearly demonstrate that SyncRef yields solutions that are highly comparable to an exact solution, yet are obtained hundreds of times faster. Moreover, the speed gains of SyncRef increase exponentially with the number of sequences.

This paper makes two specific technical contributions. First, we introduce the concept of ϵ-*expansion*, a principled approach to expand a set of clusters that span the input set. The largest possible subset of synchronized sequences is theoretically guaranteed to be contained within one of the ϵ-*expanded* clusters. Second, we show how to use Jung's theorem to ensure that all possible pairs of sequences in the output solution are correlated above a given threshold. We also provide directions for how to extend the SyncRef method to discover the largest subset of locally correlated sequences within an unknown time window. The code of SyncRef is publicly available at https://github.com/sariyanidi/SyncRef.

1.1. Problem formulation

We now formulate the problem of discovering synchronized (*i.e.*, pairwise correlated) subsets of sequences (*i.e.*, time series). Let $\mathcal{X}$ be a set of N sequences, each having T time values, $\mathcal{X}=\{x_i\}_{i=1}^N$. Let $r(x_i, x_j)$ be the correlation coefficient between the sequences x_i and x_j. We call a pair of sequences *synchronized* if their correlation is not less than ρ_θ, or, equivalently, if the ℓ_2 norm of the difference of the z-normalized sequences is not more than $\epsilon_\theta := \sqrt{2T(1-\rho_\theta)}$ (see Supplementary Appendix A). Similarly, we say that a set of sequences $\mathcal{S} := \{x_i\}_{i\in\mathcal{I}}$, where $\mathcal{I}$ is a subset of $\{1,\dots,N\}$, is *synchronized* if all pairs of sequences in $\mathcal{S}$ are synchronized. We aim to find the largest synchronized set; *i.e.*, the set with the maximal cardinality $|\mathcal{S}|$. This optimization problem is expressed as follows.

Problem 1. *Given sequence set $\mathcal{X}=\{x_i\}_{i=1}^N$ and ϵ_θ,*

$$\begin{aligned} &\textit{maximize} && |\mathcal{S}| && (1)\\ &\textit{subject to} && \mathcal{S} = \{x_i\}_{i\in\mathcal{I}}, \mathcal{I} \subseteq \{1,\dots,N\}, && (2)\\ & && ||\tilde{x}_i - \tilde{x}_j||_2 \le \epsilon_\theta \ \ \forall i,j \in \mathcal{I}, && (3) \end{aligned}$$

where $\tilde{x}_i$ is the z-normalized version of sequence x_i.

This problem is equivalent to finding the largest maximal clique in a binary graph whose nodes represent sequences, and two nodes are connected if the distance between them is at most ϵ_θ—an NP-complete problem.

1.2. Related work

Correlation is arguably the most standard metric of synchrony across domains [12, 18, 38, 20, 30, 33, 28, 45, 47, 52]. However, it is not straightforward to apply the correlation metric to the task of identifying the largest subset of synchronized sequences in a given set of sequences. A naive method would be to construct all possible subsets of the given set of sequences, and test whether the pairwise correlation among all the members of each subset is larger than a predefined threshold. However, a set with n sequences has 2^n possible subsets; therefore, this method is not scalable.

A popular approach for identifying synchrony is to use warping methods (*e.g.*, dynamic time warping) [40, 41]. While methods that can be applied to multiple sequences exist [55, 56, 49, 50], warping methods do not aim to identify the largest subset of synchronized sequences, as their goal is to optimize the time warping path among sequences.

Branch and bound (B&B) frameworks have recently been proposed for unsupervised temporal commonality [10] and synchrony discovery [9, 8], but the problems addressed by them are inherently different than those addressed by our approach; those B&B methods do not aim to identify subsets of sequences that are synchronized [*i.e.*, subsets that satisfy the condition in (3)].

A problem related to ours is longest common substring (LCS) discovery [23]. However, LCS discovery applies to discrete-valued data (*e.g.*, words, DNA/protein sequences), whereas we are interested in continuous-valued sequences.

One can use clustering to identify similar sequences [46]. However, no clustering approach (including hierarchical clustering) or other space partitioning algorithms such as KD trees can ensure that the condition in (3) is satisfied—that the pairwise correlation among all members of a cluster exceeds a given threshold. The closest clustering paradigm to ours is that of density based approaches, which impose a proximity constraint –a distance threshold– to include an element in a cluster [15]. Still, these approaches do not require that *all* pairs in the cluster satisfy the proximity constraint; it suffices for any member of a cluster to be sufficiently close to any other member.

Finally, our problem can be solved using generic maximal clique algorithms [48, 5] (Section 1.1). Unfortunately, those algorithms are not practical for operation on large N, due to their computational time and space complexity (see also Section 3.1.4). One can also use an approximate maximal clique finding algorithm [17] to reduce computational complexity. In our experiments, we compare with both exact and approximate maximal clique finding algorithms. We experiment using facial videos, financial time series and brain connectivity data. SyncRef can have further applications, such as anomaly detection [3], if anomaly is manifested with synchronous behavior of a subset of time series.

2. The SyncRef method

We now describe how to approximately solve Problem 1 with the proposed SyncRef approach. The input to SyncRef is the set that contains all sequences, $\mathcal{X}$. SyncRef gradually filters out irrelevant sequences to yield a synchronized set.

Fig. 1 illustrates the entire framework. We first cast all sequences into a time-invariant principal component analysis (PCA) representation (Fig. 1c). Then, we cluster sequences efficiently by defining thresholds on their PCA coefficients (Fig. 1c, Section 2.2). We then enlarge clusters to get the so-called ϵ-*expanded* clusters (Fig. 1d). We show that the largest possible synchronized set $\mathcal{S}$ is guaranteed to be contained entirely in one of the ϵ-expanded clusters. Finally, we output a synchronized set by removing sequences from an expanded cluster until the inequalities in (3) are satisfied (Fig. 1e,f; Section 2.4). The entire process is summarized in the algorihm in Table 1.

2.1. Representing via PCA

Since SyncRef necessitates calculation of distances between sequences, reducing the dimensionality of the input sequences improves efficiency. For this purpose, we compress all sequences by applying PCA. To this end, we form a $T \times N$ matrix $\tilde{\mathbf{X}}$ whose columns are the z-normalized

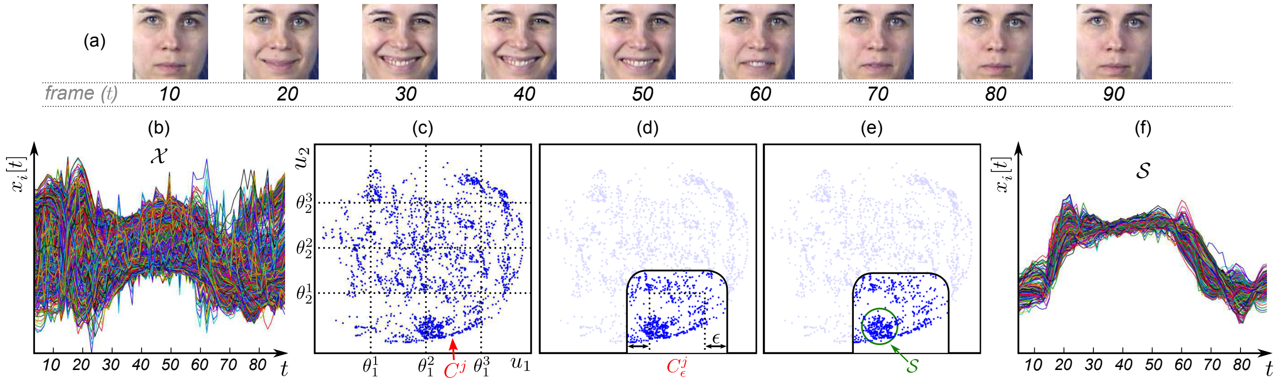

Figure 1. Illustration of how SyncRef finds a synchronized set of sequences. (a) A face video of 100×100 frames. (b) Input to SyncRef: The set $\mathcal{X}$ of 10,000 sequences, where each sequence corresponds to the optical flow magnitude of a pixel w.r.t. the first frame. (c) Illustration of the PCA representation of sequences in (b); for visualization we use two PCA coefficients u_1, u_2. Each rectangular region, defined by dashed lines (*i.e.*, thresholds θ_k^j), is a cluster. C^j is the most populated cluster. (d) The ϵ-expanded cluster C_ϵ^j. (e) The identified synchronized set $\mathcal{S}$; all points within the circle are correlated at least by $\rho_\theta = 0.80$. (f) The synchronized set of sequences illustrated back on the time domain: Those sequences correspond to pixels around the mouth region activated with the smile in (a).

Table 1. Algorithm summarizing the SyncRef method.

```
Input: Set of $N$ sequences, $\mathcal{X}=\{x_i\}_{i=1}^N$
Output: Set of synchronized sequences $\mathcal{S}$ such that $\mathcal{S} \subseteq \mathcal{X}$
  1. Compute PCA representations of all sequences, $\{\mathbf{u}^i\}_i$, using eq. (4)
  2. Determine clusters $\{C^j\}_j$ by computing the thresholds $\{\theta_k^j\}_{j,k}$ as described in Section 2.2
  3. For $j$=1, ..., $M^K$, find all the sequences that belong to cluster $C^j$, i.e. sequences whose PCA representation $\mathbf{u}^i$=$(u_1^i, \ldots, u_K^i)$ satisfies $\theta_k^{j_k} < u_k^i \leq \theta_k^{j_k+1}$ for $k = 1, \ldots, K$ (Section 2.2)
  4. Find $\epsilon$-neighbours of all clusters $\{C^j\}_j$ via ineq. (5), Section 2.3
  5. For $j$=1, ..., $M^K$, obtain expanded cluster $C_\epsilon^j$ by applying Lemma 2.2 to all points that belong to $\epsilon$-neighbours of $C^j$ (Section 2.3)
  6. Apply a refining as in Section 2.4 to the $N_C$ most populated expanded clusters to obtain $N_C$ synchronized subsets
  7. Set $\mathcal{S}$ to the largest among the $N_C$ subsets computed at step 6.
```

sequences, $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N$. Then, we approximately reconstruct this matrix using PCA as

$$\tilde{\mathbf{X}} \approx \mathbf{WU}, \tag{4}$$

where $\mathbf{W}$ is the PCA basis matrix and its columns form an orthogonal set of vectors. The size of $\mathbf{W}$ is $T \times K$ where K is chosen such that $K < \min\{T, N\}$ for compression. $\mathbf{U} = (\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^N)$ is the matrix with the PCA coefficients where each column $\mathbf{u}^i = (u_1^i, u_2^i, \ldots, u_K^i)^T$ contains the compressed representation of the sequence $\tilde{x}_i$.

2.2. Clustering via partitioning $\mathbb{R}^K$

In order to cluster sequences efficiently, we partition $\mathbb{R}^K$ –the set of all possible PCA coefficients– by defining thresholds over PCA coefficients. For each PCA coefficient u_k, we define $M + 1$ monotonically increasing threshold values $\theta_k^0, \ldots, \theta_k^M$ that partition the real line into M non-overlapping intervals (Fig. 1c). To ensure that the entire $\mathbb{R}^K$ is partitioned, we always set θ_k^0 to $-\infty$ and θ_k^M to ∞. The determination of the other thresholds is explained below.

Definition 1. *A* cluster *is a set in* $\mathbb{R}^K$ *defined by* K *threshold intervals* $(\theta_1^{j_1}, \theta_1^{j_1+1}], \ldots, (\theta_K^{j_K}, \theta_K^{j_K+1}]$ *as* $C^{j_1 j_2 \cdots j_K} := \{(u_1, \ldots, u_K) \in \mathbb{R}^K : \theta_k^{j_k} < u_k \leq \theta_k^{j_k+1}\}$, *where* $j_k \in \{0, 1, \ldots, M\}$. *For brevity, we denote the cluster* $C^{j_1 j_2 \cdots j_K}$ *as* C^j.

Since each dimension of $\mathbb{R}^K$ is divided into M intervals, we have a total of M^K possible unique clusters, and the union of all clusters covers $\mathbb{R}^K$, *i.e.*, $\bigcup_{j=1}^{M^K} C^j = \mathbb{R}^K$.

PCA coefficients of different orders may follow different activation patterns (*e.g.*, lower order components may be activated more). Therefore, we define non-uniform thresholds that divide each dimension of $\mathbb{R}^K$ into equiprobable intervals. To determine equiprobable intervals on each dimension k, we apply kernel density estimation on the PCA coefficients of the kth dimension, $\{u_k^1, u_k^2, \ldots, u_k^N\}$, and thus estimate $p(u_k)$, namely the probability density for the kth coefficient. Then, we divide the real line into $M + 1$ intervals $\mathcal{I}_k = (\theta_k^j, \theta_k^{j+1}]$ in way that the probability $\int_{\mathcal{I}_k} p_k(u_k)\, \mathrm{d}u_k$ is $1/M$ for any interval $\mathcal{I}_k = (\theta_k^j, \theta_k^{j+1}]$.

2.3. ϵ-expanded clusters

Each cluster consists of sequences that have similar PCA coefficients; however, it is possible that highly correlated sequences that satisfy (3) are separated in different clusters. Here, we explain how to *minimally* expand clusters to ensure that the largest subset of sequences that satisfy condition (3) is entirely contained within one of the clusters.

Definition 2. *The* ϵ*-expanded cluster corresponding to cluster* C^j *is* $C_\epsilon^j := \bigcup_{\mathbf{u} \in C^j} B_\epsilon[\mathbf{u}]$, *where* $B_\epsilon[\mathbf{u}]$ *is the Eu-*

clidean ball of radius ϵ centered at $\mathbf{u}$*,* i.e. $B_\epsilon[\mathbf{u}] = \{\mathbf{v} \in \mathbb{R}^K : ||\mathbf{u}-\mathbf{v}||_2 \leq \epsilon\}$.

An illustration of an ϵ-cluster in $\mathbb{R}^2$ is the rounded rectangle in Fig. 1d. The following theorem is an important result, as it demonstrates that one of the ϵ-expanded clusters is guaranteed to contain the largest set of synchronized sequences.

Theorem 2.1. *Let $\mathcal{S}{=}\{x_i\}_{i\in\mathcal{I}}$, where $\mathcal{I} \subseteq \{i\}_{i=1}^N$, be a set of T-long sequences that satisfy inequalities (3) and $\mathcal{U}$ the set that contains the K-dimensional compressed (*i.e.*, $K < \min\{T, N\}$) PCA representations of those sequences, $\mathcal{U} = \{\mathbf{u}^i\}_{i\in\mathcal{I}}$. Let ϵ be $\epsilon{:=}\epsilon_\theta\sqrt{K/(2(K+1))}$, and $\{C^j\}_j$ be a set of clusters (Definition 1) such that $\bigcup_j C^j = \mathbb{R}^K$. Then, there exists an ϵ-expanded cluster C^j_ϵ such that $\mathcal{U} \subseteq C^j_\epsilon$. Moreover, there is no $C^j_{\epsilon_0}$ with $\epsilon_0 < \epsilon$ that can in general guarantee the existence of C^j such that $\mathcal{U} \subseteq C^j_{\epsilon_0}$.*

Proof is in Supp. Appendix B. To use Theorem 2.1 in practice, we need to know when a point $\mathbf{u} \notin C^j$ belongs to C^j_ϵ.

Lemma 2.2. *A point* $\mathbf{u}$ *belongs to C^j_ϵ if and only if $\sum_{k=1}^K f\left(u_k; \theta_k^{j_k}, \theta_k^{j_k+1}\right) \leq \epsilon^2$, where*

$$f\left(u_k; \theta_k^{j_k}, \theta_k^{j_k+1}\right) := \begin{cases} 0 & \text{if } u_k \in (\theta_k^{j_k}, \theta_k^{j_k+1}] \\ \min\limits_{t\in\{j_k, j_k+1\}} \left\{(\theta_k^t - u_k)^2\right\} & \text{else.} \end{cases}$$

The proof of Lemma 2.2 is shown in Supplementary Appendix C. To efficiently identify all members of an expanded cluster C^j_ϵ, we introduce the concept of ϵ-neighbors. We say that C^i and C^j are *ϵ-neighbors* if the ℓ_2 distance of their closest points is less than ϵ (*i.e.*, $\inf\{||\mathbf{u}{-}\mathbf{v}||_2 : \mathbf{u}{\in}C^i, \mathbf{v}{\in}C^j\}{<}\epsilon$). By definition of ϵ-expanded clusters, it follows that C^i can contain members of C^j_ϵ if and only if C^i and C^j are ϵ-neighbors (*e.g.* only the five neighbors of C^j in Fig. 1c can contain points of C^j_ϵ; see also Fig. 1d). Thus, one can find all members of ϵ-expended cluster C^j_ϵ by applying Lemma 2.2 to points that belong to ϵ-neighbors of C^j, ignoring all other points, thus improving efficiency significantly. Moreover, C^i and C^j are ϵ-neighbors if

$$\sum_{k=1}^K \min_{\substack{p\in\{i_k, i_k+1\} \\ t\in\{j_k, j_k+1\}}} \left\{\left(\theta_k^p - \theta_k^t\right)^2\right\} < \epsilon^2 \qquad (5)$$

(see Supplementary Appendix D for proof). Thus, using inequality (5), one can efficiently find all the ϵ-neighbors of each cluster.

2.4. Refining ϵ-expanded cluster

We now explain how to refine ϵ-expanded clusters to output the (approximately) largest synchronized subset within a given set of sequences. This refining is the last step to solving Problem 1, and it is crucial as it ensures that the solution satisfies the condition in (3). Heuristically, we expect the optimal solution to lie in one of the most populated clusters, and experiments support this heuristic (Section 3.1.4).

Solution via maximal cliques. We can apply a maximal clique algorithm to an expanded cluster to identify the largest synchronized subset in it. The advantage of this approach compared to applying maximal clique on the entire set of sequences is reduced space complexity, due to smaller graphs. Still, this can be inefficient for very large N.

Approximate solution via ball fitting. We now present a scalable approach. According to Jung's theorem [11], the radius of the tightest (Euclidean) ball enclosing a set of points that satisfy condition (3) can be any value within the range $[\epsilon_\theta/2, \epsilon_\theta\sqrt{T/(2(T+1))}]$. This leads to the following iterative refinement approach: At each iteration, we compute the radius of the ball enclosing points in C^j_ϵ [21] and then remove from C^j_ϵ the farthest points from the mean. We terminate iterations when the radius reaches $\epsilon_\theta\sqrt{T/(2(T+1))}$. Note that points that are $\epsilon_\theta/2$ away from the ball's center may still violate (3). Therefore, we compute the pairwise distance for all remaining points in C^j_ϵ that are $\epsilon_\theta/2$ away from the ball's center, and remove the ones that violate condition (3). The same iterative procedure can be repeated by initializing not from the mean of all sequences in C^j_ϵ, but from a randomly selected point, and ignoring all points in C^j_ϵ that are $2\epsilon_\theta$ away from the randomly selected point. This turns out to be a useful strategy. Thus, we repeat the refining strategy described above for an additional randomly selected σ percent of the points in C^j_ϵ, where σ is determined empirically. Let $\mathcal{U}$ represent the set of refined points. Since we carry out the ball fitting on the PCA domain and since the columns of the PCA matrix $\mathbf{W}$ form an orthonormal set, the norm among any two points in $\mathcal{U}$ can be smaller than the actual norm of the corresponding sequences. As a result, some points in $\mathcal{U}$ can violate (3). Therefore, as a final step, we repeat the ball fitting procedure described above on the time domain; that is, we refine the set of sequences whose PCA representation lies in $\mathcal{U}$.

3. Experimental validation

Next, we evaluate the ability of SyncRef to discover the largest synchronized subset within a given set of sequences. To evaluate the versatility and generalizability of the method, we use real data from three distinct domains, namely facial analysis, finance, and neuroscience.

3.1. Optimization performance

3.1.1 Metric

We quantify (sub)optimality via mean percentage error (MPE). Let $\mathcal{X}_j$ be a set (of N sequences) whose largest synchronized subset contains S_j^* sequences, and let $\hat{S}_j^*$ be the size of the largest synchronized subset discovered with a given method. Then, the MPE over $N_{\mathcal{X}}$ sets of sequences is $\frac{1}{N_{\mathcal{X}}}\sum_{j=1}^{N_{\mathcal{X}}}\frac{S_j^* - \hat{S}_j^*}{S_j^*}$. The exact solution S_j^* is obtained via maximal clique (Section 3.1.3). We ran experiments on a cluster computer and allowed up to 100GB RAM per process (*i.e.*, per set of sequences). We ignored the set of sequences in which the maximal clique algorithm failed with memory error. We limited N to 1600 as maximal clique often failed beyond this number of sequences.

3.1.2 Datasets and tasks

Facial dataset. We use the MMI [37] dataset, which contains 327 facial expression videos at 30fps, annotated with temporal phase labels describing the evolution of expressions: neutral (i.e., expressionless), onset, apex, and offset [42]. From each video, we extract as many sequences as the number of pixels, where each sequence represents the motion (*i.e.*, optical flow [16]) magnitude of one pixel. By finding the largest subset of synchronized sequences, we are effectively identifying the largest group of pixels that move together. Since the dominant event in the videos is a facial expression, we expect the largest group of synchronized pixels to correspond to the unfolding of that expression. Thus, their identification is of potential interest for unsupervised facial expression annotation (see also Section 3.2). In sum, we have $N_{\mathcal{X}}$=327 input sets for which synchronized pixels are discovered. Video lengths vary between T=22 and 177 frames. To evaluate how computational speed changes with N, we randomly subsampled the optical flow representation to generate input sets of N=400, 800, 1200, and 1600 sequences. Obtaining ground truth for higher N was not possible (Section 3.1.1). Prior to optical flow computation, we detected facial landmarks with OpenFace [2], used a part-based representation [42] of left/right eye and mouth, and applied temporal registration [43].

Stock market dataset. We fetched the data corresponding to daily stock price for all companies on the U.S. stock market for the past three years using a financial data API (`financialmodellingrep.com`). There are approximately 22 stock trading days per month. We eliminated companies whose price was not available for a large part of this three-year period, and the final number of available companies was 6,582. The task was to identify the largest subset of companies whose stock market prices were correlated. We treated each month of each year independently, and obtained a dataset with $N_{\mathcal{X}}$=12 × 3=36 sets of sequences where each set contained $6,582$ sequences with T=22 values. Each value in a sequence contains the average of the highest and lowest stock price for the corresponding company within one day. For the same reasons with the facial dataset, we experimented on N=400, 800, 1200, and 1600 randomly subsampled sequences.

Brain dataset. We used a subset of functional magnetic resonance imaging data from the Philadelphia Neurodevelopmental Cohort [44]. This subset contains data from $\mathcal{N}_{\mathcal{X}}$=25 healthy males. Each brain is represented by $N = 400$ sequences, corresponding to mean functional activation of 400 brain regions across time (T=120). Subsampling was not necessary as N=400 was small enough to run maximal clique for obtaining ground truth.

3.1.3 Compared methods and implementation

Our method, which uses ball fitting (Section 2.4), is referred to as *SyncRef*. We compare with the following methods, some of which are variations of our method. (1) *MaxClique*: exact solution by finding the largest maximal clique [48] (Section 1.1); (2) *Approximate MaxClique* [27]; (3) *SyncRef-MaxClique*: the SyncRef method where refining is performed with maximal clique (Section 2.4); (4) *BallFit*: the ball fitting method (Section 2.4) on the entire set of input sequences (*i.e.*, no clustering); (5) *SyncRef-DCT*: the SyncRef method where PCA is replaced with discrete cosine transform (DCT). We compare with DCT to see how PCA performs compared to a non-learned but still orthogonal representation. Since DCT is not a learned representation, we compute the thresholds for the SyncRef-DCT clusters (Section 2.2) offline on a large dataset of sequences [7].

We implemented our method on python. We use the method of Gärtner [21] for ball fitting during refining. We set M=4 and K=4. We refine the N_C most populated clusters and report results for N_C=$1, 2, \ldots, 16$. To find maximal cliques, we used a state-of-the-art python package, `networkx` [27]. For approximate maximal cliques we used the `approximation` module of the same package. To our knowledge, there is no other method that we can compare SyncRef against (Section 1.2).

3.1.4 Results

Fig. 2a,b,c report the MPE of compared methods on the facial dataset, stock market dataset and brain dataset w.r.t. the number of refined clusters, N_C (MaxClique is omitted since it is the exact solution; *i.e.* its MPE is 0). The top, middle and bottom rows of Fig. 2 respectively represent results for correlation thresholds of ρ_θ=0.65, ρ_θ=0.75 and ρ_θ=0.85. SyncRef-MaxClique's MPE goes to zero if enough clusters are searched (*i.e.*, as N_C increases). This is expected from Theorem 2.1, which guarantees that one of the ϵ-expanded clusters will entirely contain the optimal solution. SyncRef and SyncRef-DCT perform better with the higher σ, as clusters are refined more thoroughly. For σ=0.05, both SyncRef and SyncRef-DCT always outperform BallFit for sufficiently large N_C. Approximate MaxClique works well on the brain dataset but relatively poorly on the other datasets.

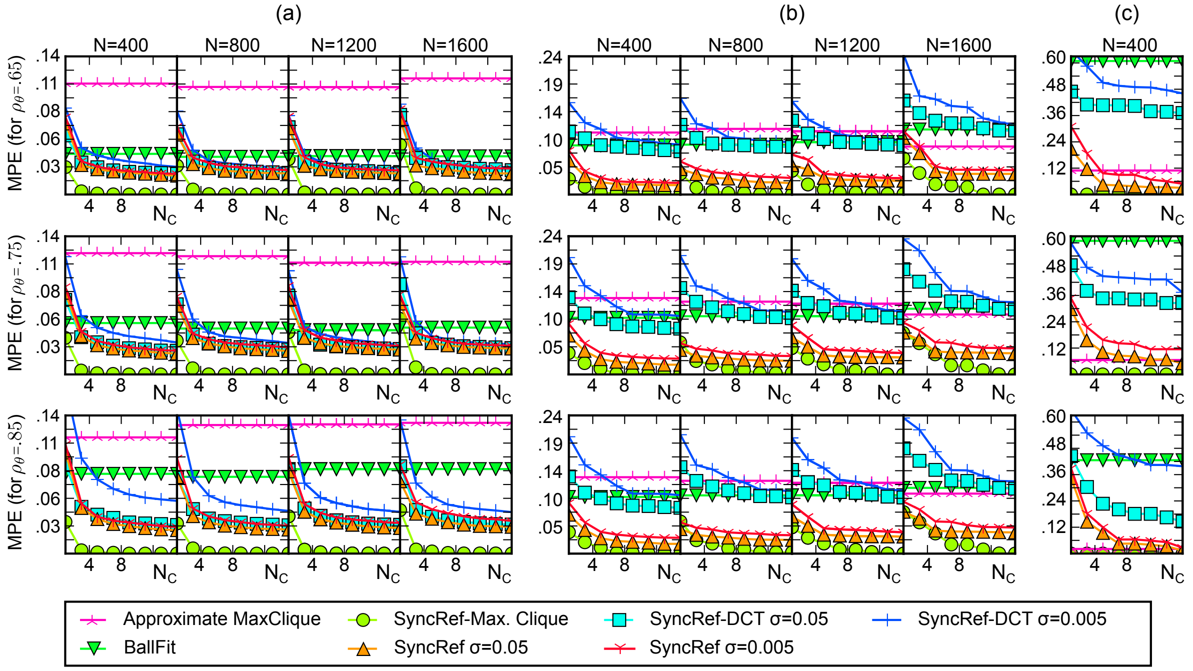

Figure 2. Mean percentage error (MPE) on facial (a), stock market (b), and brain (c) datasets against the number of expanded clusters that were refined (N_C) for varying number of input sequences (N). Top, middle, and bottom rows respectively represent results for ρ_θ=0.65, ρ_θ=0.75 and ρ_θ=0.85. BallFit and Approximate Max. Clique do not depend on N_C. MaxClique is omitted from (a–c) as its MPE is 0.

Table 2. Average computation time in seconds and speed gain relative to the exact solution (given within parentheses) for increasing numbers of input sequences (N). Speed gain for MaxClique is always 1 since it is the exact solution.

	Facial data			
Method	N=400	N=800	N=1200	N=1600
Exact (MaxClique)	1.8 (1)	27.9 (1)	205.9 (1)	392.3 (1)
Approximate MaxClique	10.9 (0.3)	72.6 (0.4)	220.25 (0.9)	477.6 (0.8)
BallFit	1.0 (1.7)	4.0 (7.1)	9.0 (22.9)	15.0 (26.1)
SyncRef-Max.Clique	5.1 (0.3)	105.7 (0.3)	822.8 (0.3)	1256.3 (0.3)
SyncRef-DCT $\sigma = .005$	**0.4 (5.1)**	1.1 (24.5)	2.5 (82.0)	5.4 (73.2)
SyncRef-DCT $\sigma = .05$	1.0 (1.9)	3.4 (8.2)	7.2 (28.3)	12.5 (31.4)
SyncRef $\sigma = .005$	0.9 (2.0)	1.1 (25.1)	**1.2 (167.5)**	**1.3 (311.2)**
SyncRef $\sigma = .05$	0.9 (2.0)	**1.1 (25.4)**	1.2 (166.6)	1.3 (303.8)

Overall, SyncRef has the best and most consistent performance across the three datasets. Error drops below 5% in all cases when N_C=16 clusters are searched and in most when N_C=8 clusters are searched. SyncRef and SyncRef-DCT perform similarly on the facial dataset, but there is a visible difference on other datasets. The high and consistent performance of SyncRef is remarkable as the characteristics of the sequences in each dataset are very different (Fig. 3). Facial and stock market sequences are relatively smooth (Fig 3a,b), but the brain dataset shows abrupt variations (Fig 3c), which explains the poor performance of DCT on the latter dataset. SyncRef works equally well on all datasets, as PCA compression does not impose assumptions regarding smoothness of sequences.

Table. 2 shows average computing times and computation gains relative to exact solution. SyncRef-MaxClique can be slower than the exact solution, as it performs searches repeatedly on partially overlapping clusters. However, SyncRef-MaxClique can be advantageous when memory restrictions are present (Section 2.4). Approximate MaxClique is also slower than MaxClique, but the former is more scalable in terms of memory: While MaxClique failed occasionally even when we allowed 100GB of RAM, no such problem was observed for other methods, even with much less RAM (*e.g.*, 16GB). SyncRef and SyncRef-DCT are always faster than all other methods. SyncRef-DCT is faster than Sync-Ref when N is not very large, as the cluster thresholds for the former are computed offline (Section 3.1.3). Importantly, SyncRef is faster when N is large (*e.g.*, N=1600), as the cost of computing thresholds becomes negligible and also because PCA uses clusters more efficiently as it adapts to data. Of note, the speed gain of Sync-Ref relative to exact solution exceeds 300 for N=1600. Table 2 shows that the computation time of SyncRef increases very modestly with N, which suggests that it can be applied to even larger N. Nevertheless, $1,600$ is already a very large number for our NP-hard problem, as obtaining exact solution not only takes very long time but also requires large computation space (*i.e.*, RAM; Section 3.1.1)

Conclusion. SyncRef achieves results comparable to the exact solution (less than 5% MPE) across three datasets

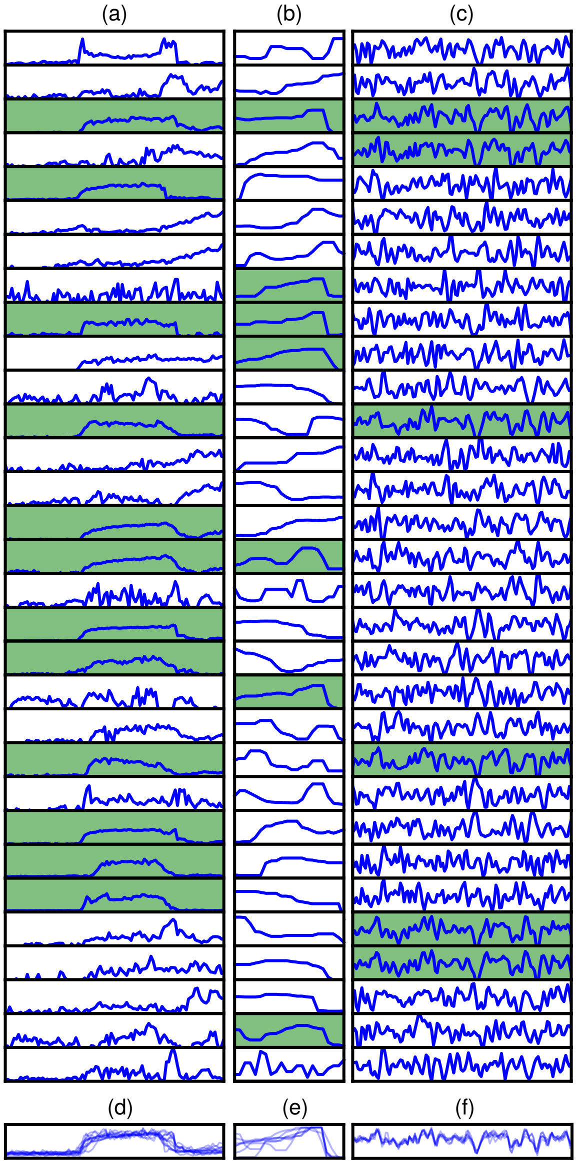

Figure 3. Illustration of synchronized sequences discovered by SyncRef ($\sigma = 0.05$; $\rho_\theta = 0.75$). 35 randomly selected sequences from the (a) facial, (b) stock market, and (c) brain datasets; the subset of synchronized sequences discovered by SyncRef are highlighted in green and also shown by overlaying in (d), (e) and (f).

from distinct domains, and is up to hundreds of times faster than the exact solution, with a speed advantage that increases exponentially with the number of sequences.

3.2. Interpretation of synchronized sequences

We discuss the interpretation of the synchronized sequences for which quantitative results were given in Section 3.1.4. We focus on the face and brain datasets, as the interpretation and research value of identifying co-movement in stock market asset prices is more evident [32, 34, 39].

Facial data. Prediction of facial expressions' temporal phases (*i.e*., neutral, onset, apex, and offset) [13] is an active research problem [57, 58, 53, 54] as annotated data is necessary for facial expression research and automation of this process eliminates or aids meticulous manual work.

We show that synchronized pixels in facial expression videos are directly related to the temporal phases of the displayed expression. We use the MMI dataset, which has manually annotated temporal phases of expressions (Section 3.1.2). We generate two input sets of sequences per video: One with pixels corresponding to the upper face (eyes, brows), and one corresponding to lower face (mouth). We then use SyncRef (σ=0.05) to find the largest synchronized set (Fig. 4ii) from each input set (Fig. 4i), and finally estimate the temporal phase as the average of sequences within the synchronized set (Fig. 4iii). Thus, we estimate two temporal phase annotations (upper and lower face) per video. We evaluate accuracy with a standard annotation evaluation metric [1], namely Pearson's correlation. Specifically, we correlate the estimated upper (lower) face annotation with the temporal phase annotations of all the upper (lower) facial action units (AUs) [13, 42], and consider an estimation successful if the correlation is at least 0.85.

SyncRef successfully estimated the temporal phase of at least one AU in 91.9% of MMI videos. Table 3 shows the ratio of AUs whose temporal phase was successfully estimated. Fig. 4 shows the estimated and true temporal phase annotation for 15 MMI sequences. Overall, this section shows that discovering the synchronized sets of pixels in a video can be useful for identifying events of interest. For proof of concept, we used the MMI dataset where events are segmented in time. In a real application, one would need additional tools to eliminate irrelevant events (*e.g*., head movements) and to identify the exact temporal location of expressions. We discuss how SyncRef can be extended to address the latter issue in Section 4.

Brain data. To show that the identified synchronized sets have neurobiological relevance, we visualized the regions of the brain that were identified as being synchronized (Fig. 5). The synchronized regions were mainly distributed within three functional systems. These included the default mode network (DMN), which has an important role in integrating different cognitive processes [24, 25]. The other two, namely somatomotor and visual systems, are known to be relatively more modular (*i.e*., densely correlated within themselves but only sparsely correlated to other systems), possibly acting as sensory modules [19, 26, 35]. Our finding is in agreement with the neuroscience literature [35], validating that SyncRef identifies biologically driven synchronized sets. Indeed, SyncRef can advance research in the field by allowing identification of synchronous neuronal clusters at the voxel-level, which is not possible with current methods due to extreme dimensionality ($\geq 10^5$).

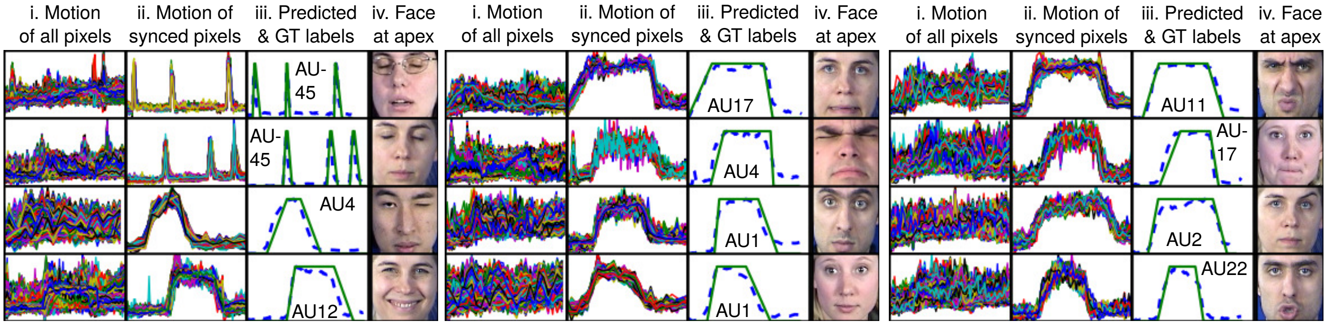

Figure 4. Temporal phase estimation for 15 MMI sequences. For each sequence, we show (i) the input set *i.e*., the motion (optical flow magnitude) of all pixels over time; (ii) the motion of synchronized pixels discovered with SyncRef, (iii) the ground truth (GT) temporal phase label (solid green lines) vs. the label predicted as the mean of synchronized pixels (dashed blue lines); and (iv) expression at apex.

Table 3. Ratio of Action Units (AUs) (in the MMI dataset) whose temporal phase annotation was predicted successfully with SyncRef.

Action unit:	1	2	4	5	6	7	9	10	11	12	13	14	15	16	17	18	20	22	23	24	25	26	27	28	43	45
Prediction rate:	.67	.77	.63	.59	.59	.43	.94	.95	.90	.89	1.0	.80	.85	.72	.86	.78	.75	.88	.80	.80	.77	.69	.88	1.0	.75	.48

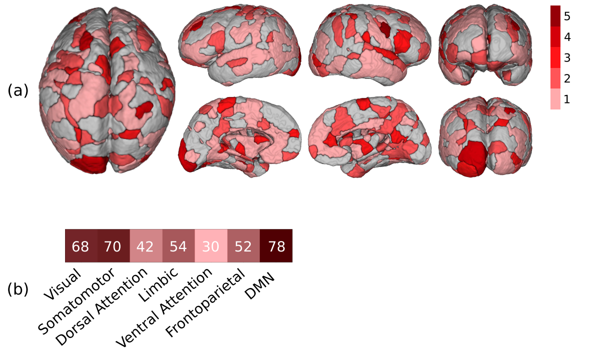

Figure 5. Brain regions with high synchrony (ρ_θ=0.6). (a) Regions are colored based on the number of times that they were included in the synchronized set. (b) Regions are assigned to the seven known functional systems of the brain. Numbers and colors indicate the number of times that the regions of these systems were included in the largest synchronized set.

4. Future directions

The range of applications for our approach of discovering synchrony can be increased significantly by allowing for local correlations; that is, identifying sequences correlated only within a time window rather than the entire sequence, and finding time windows where the number of correlated sequences is maximized. This can be done with a sliding window [51], which is a standard approach. This approach can be rendered efficient by replacing PCA with a representation that allows for the usage of integral images [51]. For Theorem 2.1 to be valid, the representation needs to also be an orthogonal transformation as this property is used in the proof (Supplementary Appendix B). Fortunately, there exist representations that satisfy these criteria. DCT is an orthogonal transformation that can be computed efficiently thanks to cosine integral images [14], and is effective for smooth sequences (Section 3.1.4). Haar basis is another representation that satisfies both criteria [51]. If those non-adaptive representations are not as effective as PCA, one can also consider a two-stage coarse-to-fine approach; that is, to use DCT or Haar to substantially reduce the candidates to time windows with a large number of synchronized sequences, and then to use SyncRef only on the remaining candidates. Another important extension to the SyncRef method is to allow for lags among sequences. While the proposed framework can be extended to allow for lag among sequences through a greedy framework, the theoretical guarantee of the ϵ-expanded clusters may not be preserved.

5. Conclusion

We presented SyncRef, a method to find the largest subset in a set of sequences such that every pair in the subset is correlated beyond a defined threshold. Data from three distinct domains clearly illustrate that SyncRef produces results comparable to the exact solution, but in a much faster and scalable manner. In addition, SyncRef is robust to the number of input sequences, the correlation threshold, and the type or source of data.

Acknowledgment

We would like to thank Yusuf Osmanlioglu for his comments on the manuscript and the technical discussions that helped shaping up the method. The work of E. Sariyanidi, C. J. Zampella, R. T. Schultz and B. Tunc is partially funded by the NIMH of US under grant R01MH118327 and by the Eagles Autism Foundation. The work of R. T. Schultz is partially funded also by the McMorris Family Foundation.

References

[1] AD Albano. Introduction to educational and psychological measurement using R, 2017. 7

[2] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. Morency. Openface 2.0: Facial behavior analysis toolkit. In *Proc. IEEE Int'l Conf. Automatic Face Gesture Recognition*, pages 59–66, May 2018. 5

[3] Jawadul H Bappy, Sujoy Paul, Ertem Tuncel, and Amit K Roy-Chowdhury. Exploiting typicality for selecting informative and anomalous samples in videos. *IEEE Trans. on Image Processing*, 28(10):5214–5226, 2019. 2

[4] Vladimir Boginski, Sergiy Butenko, and Panos M Pardalos. Mining market data: a network approach. *Computers & Operations Research*, 33(11):3171–3184, 2006. 1

[5] Coen Bron and Joep Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, Sept. 1973. 2

[6] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews. Neuroscience*, 10(3):186–98, mar 2009. 1

[7] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The UCR time series classification archive, July 2015. 5

[8] Wen-Sheng Chu, Fernando De la Torre, Jeffrey F Cohn, and Daniel S Messinger. A branch-and-bound framework for unsupervised common event discovery. *Int'l J. of Computer Vision*, 123(3):372–391, 2017. 2

[9] Wen-Sheng Chu, Jiabei Zeng, Fernando De la Torre, Jeffrey F Cohn, and Daniel S Messinger. Unsupervised synchrony discovery in human interaction. In *Proc. IEEE Int'l Conference on Computer Vision*, pages 3146–3154, 2015. 2

[10] Wen-Sheng Chu, Feng Zhou, and Fernando De la Torre. Unsupervised temporal commonality discovery. In *Proc. European Conf. on Computer Vision*, pages 373–387, 2012. 2

[11] Boris V Dekster. The Jung theorem for spherical and hyperbolic spaces. *Acta Mathematica Hungarica*, 67(4):315–331, 1995. 4

[12] Emilie Delaherche, Mohamed Chetouani, Ammar Mahdhaoui, Catherine Saint-Georges, Sylvie Viaux, and David Cohen. Interpersonal synchrony: A survey of evaluation methods across disciplines. *IEEE Transactions on Affective Computing*, 3(3):349–365, 2012. 2

[13] P. Ekman, W.V. Friesen, and J.C. Hager. *The Facial Action Coding System*. Weidenfeld and Nicolson, London, 2 edition, 2002. 7

[14] Elhanan Elboher and Michael Werman. Cosine integral images for fast spatial and range filtering. In *Proc. IEEE Int'l Conference on Image Processing*, pages 89–92, 2011. 8

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996. 2

[16] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. 2003. 5

[17] Uriel Feige. Approximating maximum clique by removing subgraphs. *SIAM Journal on Discrete Mathematics*, 18(2):219–225, 2004. 2

[18] Eileen M Finnegan, Erich S Luschei, Julie M Barkmeier, and Henry T Hoffman. Synchrony of laryngeal muscle activity in persons with vocal tremor. *Archives of Otolaryngology–Head & Neck Surgery*, 129(3):313–318, 2003. 2

[19] Jerry Alan Fodor. *Modularity of Mind: An Essay on Faculty Psychology*. MIT Press, 1983. 7

[20] Brandon K Fornwalt, Takeshi Arita, Mohit Bhasin, George Voulgaris, John D Merlino, Angel R León, Derek A Fyfe, and John N Oshinski. Cross-correlation quantification of dyssynchrony: a new method for quantifying the synchrony of contraction and relaxation in the heart. *Journal of the American Society of Echocardiography*, 20(12):1330–1337, 2007. 2

[21] Bernd Gärtner. Fast and robust smallest enclosing balls. In Jaroslav Nešetřil, editor, *Algorithms - ESA' 99*, pages 325–338, Berlin, Heidelberg, 1999. 4, 5

[22] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences*, 112(44):13455–13460, 2015. 1

[23] Szymon Grabowski. A note on the longest common substring with k-mismatches problem. *Information Processing Letters*, 115(6-8):640–642, 2015. 2

[24] Michael D Greicius, Ben Krasnow, Allan L Reiss, and Vinod Menon. Functional connectivity in the resting brain: a network analysis of the default mode hypothesis. *Proceedings of the National Academy of Sciences*, 100(1):253–258, 2003. 7

[25] Shi Gu, Theodore D Satterthwaite, John D Medaglia, Muzhi Yang, Raquel E Gur, Ruben C Gur, and Danielle S Bassett. Emergence of system roles in normative neurodevelopment. *Proceedings of the National Academy of Sciences*, 112(44):13681–13686, 2015. 7

[26] Ruben C Gur, Jan Richard, Monica E Calkins, Rosetta Chiavacci, John A Hansen, Warren B Bilker, James Loughead, John J Connolly, Haijun Qiu, Frank D Mentch, et al. Age group and sex differences in performance on a computerized neurocognitive battery in children age 8- 21. *Neuropsychology*, 26(2):251, 2012. 7

[27] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008. 5

[28] Ariel Haimovici, Enzo Tagliazucchi, Pablo Balenzuela, and Dante R Chialvo. Brain organization into resting state networks emerges at criticality on a model of the human connectome. *Physical review letters*, 110(17):178101, 2013. 2

[29] Wei-Qiang Huang, Xin-Tian Zhuang, and Shuang Yao. A network analysis of the chinese stock market. *Physica A: Statistical Mechanics and its Applications*, 388(14):2956–2964, 2009. 1

[30] Peter König. A method for the quantification of synchrony and oscillatory properties of neuronal activity. *Journal of neuroscience methods*, 54(1):31–37, 1994. 2

[31] Junqiu Li, Xingyuan Wang, and Yaozu Cui. Uncovering the overlapping community structure of complex networks by maximal cliques. *Physica A: Statistical Mechanics and its Applications*, 415:398–406, 2014. 1

[32] Shu-hsien Liao, Pei-hui Chu, and Ying-lu You. Mining the co-movement between foreign exchange rates and category stock indexes in the taiwan financial capital market. *Expert Systems with Applications*, 38(4):4608–4617, 2011. 7

[33] Francois Longin and Bruno Solnik. Extreme correlation of international equity markets. *The journal of finance*, 56(2):649–676, 2001. 2

[34] Randall Morck, Bernard Yeung, and Wayne Yu. The information content of stock markets: why do emerging markets have synchronous stock price movements? *Journal of financial economics*, 58(1-2):215–260, 2000. 7

[35] Yusuf Osmanlıoğlu, Birkan Tunç, Drew Parker, Mark A. Elliott, Graham L. Baum, Rastko Ciric, Theodore D. Satterthwaite, Raquel E. Gur, Ruben C. Gur, and Ragini Verma. System-level matching of structural and functional connectomes in the human brain. *NeuroImage*, 199:93 – 104, 2019. 7

[36] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043):814, 2005. 1

[37] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *Proc. IEEE Int'l Conf. Multimedia and Expo*, page 5, 2005. 5

[38] Emmanuel Paradis, Stephen R Baillie, William J Sutherland, and Richard D Gregory. Spatial synchrony in populations of birds: effects of habitat, population trend, and spatial scale. *Ecology*, 81(8):2112–2125, 2000. 2

[39] Robert S Pindyck and Julio J Rotemberg. The comovement of stock prices. *The quarterly journal of economics*, 108(4):1073–1104, 1993. 7

[40] Hiroaki Sakoe, Seibi Chiba, A Waibel, and KF Lee. Dynamic programming algorithm optimization for spoken word recognition. *Readings in speech recognition*, 159:224, 1990. 2

[41] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007. 2

[42] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 37(6):1113–1133, 2015. 5, 7

[43] E. Sariyanidi, H. Gunes, and A. Cavallaro. Robust registration of dynamic facial sequences. *IEEE Trans. on Image Processing*, 2016. 5

[44] Theodore D Satterthwaite, Mark A Elliott, Kosha Ruparel, James Loughead, Karthik Prabhakaran, Monica E Calkins, Ryan Hopson, Chad Jackson, Jack Keefe, Marisa Riley, Frank D Mentch, Patrick Sleiman, Ragini Verma, Christos Davatzikos, Hakon Hakonarson, Ruben C Gur, and Raquel E Gur. Neuroimaging of the Philadelphia neurodevelopmental cohort. *NeuroImage*, 86:544–53, feb 2014. 5

[45] RC Schmidt, Samantha Morr, Paula Fitzpatrick, and Michael J Richardson. Measuring the dynamics of interactional synchrony. *Journal of Nonverbal Behavior*, 36(4):263–279, 2012. 2

[46] Sohil Atul Shah and Vladlen Koltun. Robust continuous clustering. *Proceedings of the National Academy of Sciences*, 114(37):9814–9819, 2017. 2

[47] Xiaofan Sun, Khiet P Truong, Maja Pantic, and Anton Nijholt. Towards visual and vocal mimicry recognition in human-human interactions. In *Proc. IEEE Int'l Conf. on Systems, Man, and Cybernetics*, pages 367–373, 2011. 2

[48] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42, 2006. 1, 2, 5

[49] George Trigeorgis, Mihalis A Nicolaou, Björn W Schuller, and Stefanos Zafeiriou. Deep canonical time warping for simultaneous alignment and representation learning of sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 40(5):1128–1138, 2018. 2

[50] George Trigeorgis, Mihalis A Nicolaou, Stefanos Zafeiriou, and Bjorn W Schuller. Deep canonical time warping. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 5110–5118, 2016. 2

[51] Paul Viola and Michael J Jones. Robust real-time face detection. *Int'l J. of Computer Vision*, 57(2):137–154, 2004. 8

[52] Xiang Yu, Shaoting Zhang, Yang Yu, Norah Dunbar, Matthew Jensen, Judee K Burgoon, and Dimitris N Metaxas. Automated analysis of interactional synchrony using robust facial tracking and expression recognition. In *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition Workshops*, pages 1–6, 2013. 2

[53] Lazaros Zafeiriou, Mihalis A Nicolaou, Stefanos Zafeiriou, Symeon Nikitidis, and Maja Pantic. Learning slow features for behaviour analysis. In *Proc. Int'l Conf. Computer Vision*, pages 2840–2847, 2013. 7

[54] Lazaros Zafeiriou, Mihalis A Nicolaou, Stefanos Zafeiriou, Symeon Nikitidis, and Maja Pantic. Probabilistic slow features for behavior analysis. *IEEE Trans. on Neural Networks and Learning Systems*, 27(5):1034–1048, 2016. 1, 7

[55] Feng Zhou and Fernando De la Torre. Generalized time warping for multi-modal alignment of human motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1282–1289, 2012. 2

[56] Feng Zhou and Fernando De la Torre. Generalized canonical time warping. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(2):279–294, 2016. 2

[57] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *IEEE Int'l Conf. on Automatic Face & Gesture recognition*, pages 1–7, 2008. 7

[58] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(3):582–596, 2013. 1, 7