

# Dynamic Community Detection Decouples Multiple Time Scale Behavior of Complex Chemical Systems

Neda Zarayeneh, Nitesh Kumar, Ananth Kalyanaraman, and Aurora E. Clark\*

Cite This: *J. Chem. Theory Comput.* 2022, 18, 7043–7051

Read Online

ACCESS |



Metrics &amp; More

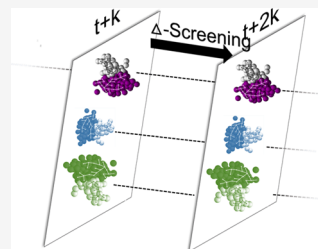


Article Recommendations



Supporting Information

**ABSTRACT:** Although community or cluster identification is becoming a standard tool within the simulation community, traditional algorithms are challenging to adapt to time-dependent data. Here, we introduce temporal community identification using the  $\Delta$ -screening algorithm, which has the flexibility to account for varying community compositions, merging and splitting behaviors within dynamically evolving chemical networks. When applied to a complex chemical system whose varying chemical environments cause multiple time scale behavior,  $\Delta$ -screening is able to resolve the multiple time scales of temporal communities. This computationally efficient algorithm is easily adapted to a wide range of dynamic chemical systems; flexibility in implementation allows the user to increase or decrease the resolution of temporal features by controlling parameters associated with community composition and fluctuations therein.



## INTRODUCTION

Network or graph representations of chemical systems have a long history that dates to the first intimations of valence and connectivity patterns within molecules.<sup>1,2</sup> Broadly, the most common applications have involved the study of static or ensemble-averaged network properties and graphs that represent intramolecular (or covalent) interactions between atoms (so-called molecular graphs). Network analysis of intermolecular interactions had perhaps its first application studying the hydrogen bond networks of water<sup>3–5</sup> and has more recently emerged as a powerful technique to study a variety of condensed phase phenomena—from the solvation and aggregation of electrolyte ions, to self-assembly of amphiphiles, and the organization of liquid–liquid interfaces.<sup>6–9</sup> These systems often exhibit structural heterogeneity, which is a manifestation of the competition of different intermolecular forces. In many systems, subgraphs can be identified that can be thought of as clusters by Chemists and communities in network science.<sup>10,11</sup> In the latter, a community is defined as a tightly knit group of vertices in an input network, such that the members of each community share a higher concentration of edges among them than the rest of the network.

While the notion of communities has been used for decades,<sup>10</sup> the seminal work by Girvan and Newman<sup>12</sup> on defining modularity<sup>13</sup> as an objective for community detection revolutionized the field. Intuitively, modularity is a real-valued score between 0 and 1 that denotes the degree to which the network is organized as “modules” or “communities”. The modularity score measures the strength of partitioning of a graph into communities, and mathematically it can be calculated by summing over all clusters of the number of edges in a community minus the number of edges expected by chance within that community. Modularity-based methods use

modularity as their optimization function for identifying the vertex community assignment. Given a graph  $G(V, E)$  with  $n$  nodes in  $V$  and  $m$  edges in  $E$ , modularity is mathematically defined as<sup>13</sup>

$$Q = \frac{1}{2m} \sum_{i,j} \left[ A_{i,j} - \frac{d_i d_j}{2m} \right] \delta(c_i, c_j) \quad (1)$$

where,  $i, j \in V$  with degrees (number of edges connected to the node)  $d_i$  and  $d_j$ , respectively;  $A_{i,j}$  denotes the weight of the edge  $(i, j)$  if the edge exists in  $E$  (0 otherwise); and  $\delta(c_i, c_j) = 1$  if the communities containing nodes  $i$  and  $j$ , respectively, are the same (i.e.,  $c_i = c_j$ ) and 0 otherwise. Modularity optimization is an NP-hard problem, but several efficient heuristics are used in practice.<sup>14,15</sup>

Recent efforts have started to explore the multiple length-scale resolution imparted by multiple pass modularity maximization procedures, as implemented in the Louvain algorithm,<sup>16</sup> which is relevant to the identification of hierarchical organization in chemical systems.<sup>17</sup> Here, the maximization function works in two primary passes: (i) iterative local greedy displacement of a node into one of its neighboring communities that maximizes its contribution to the overall modularity value and (ii) the coarse-graining of the identified communities once there is no further appreciable gain in modularity between two successive iterations. The

Received: May 3, 2022

Published: November 14, 2022



coarse-graining step collapses the identified communities into metanodes, with edges connecting metanodes and representing the strength of inter-community edges; the edges connecting each metanode to itself (i.e., loops) represent the strength of all intra-community edges for that community. The procedure is repeated on this coarsened graph representation until the modularity function reaches a peak value and cannot be increased further.

Building upon this nascent body of work, we recognize that the time evolution of these communities is an unexplored, yet potentially high impact area, where there is the potential to identify the multiple time scale evolution of a chemical system. Algorithms to identify dynamic communities over time-evolving graphs can be categorized into three main groups:<sup>18</sup> (1) static methods use the current state of the network, identify communities for this state, and retroactively map them onto the communities of the previous states of the network to track the evolution. These static-based two-step algorithms are not temporally smoothed—making tracking of communities difficult. Further, these methods incur the additional overhead of running static implementations at each timestep and mapping of communities to track the evolution also at every timestep—making the approach expensive;<sup>19–25</sup> (2) dynamic methods work across timesteps and identify communities at any given timestep based on the knowledge of the entire graph across all timesteps. The advantage of these cross-timestep algorithms is that they are better at detecting temporally persistent communities. However, these algorithms require prior knowledge of the global graph. In addition, several of these methods compare the graph and/or community states at multiple timesteps, making the approach not scalable to larger networks;<sup>26–33</sup> (3) incremental approaches follow a strategy where communities of the current timestep are identified based on the communities detected at the previous timestep(s).<sup>34–45</sup> These incremental approaches have the advantage of generating more stable communities over time, thereby facilitating an easier tracking of the evolution of communities. The  $\Delta$ -screening method<sup>46</sup> is an incremental approach where, at each timestep, a subset of nodes in the network is identified that are expected to be impacted by node or edge additions and deletions that occur in the fluctuating network. Computation of any potential community change is then restricted to only that subset of nodes. This strategy is effective at reducing the computational complexity at any step to the amount of change at that step.

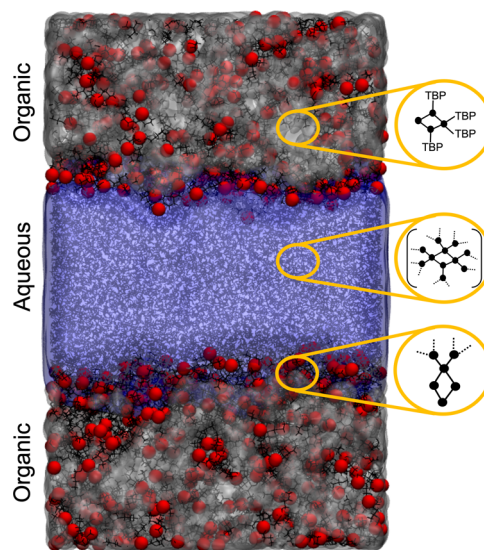
In this work, we extend the  $\Delta$ -screening approach, which is an incremental technique that applies to any given timestep, to one that works for multiple timesteps. Consequently, we define temporal communities and present a new efficient method to detect and track them across multiple timesteps of a dynamically evolving network. Importantly, the user has the ability to define a criterion for the temporal communities based upon the persistence and fluctuation of community composition (i.e., a temporal community can evolve so that nodes added or deleted up to a defined threshold before it is considered a new chemical entity). This feature allows the user to choose what is best suited for the chemical problem under study to study its multiple time scale dynamic behavior.

Using this approach, we demonstrate that dynamic community detection resolves multiple time scale behavior within complex temporal networks of intermolecular interactions. This is demonstrated for a ternary liquid system containing a liquid/liquid phase boundary. Focusing on

hydrogen bond networks of water, the temporal communities of bulk water, interfacial water, and water aggregates with amphiphilic molecules in the organic phase are identified and shown to have characteristically different durations. The ability of the  $\Delta$ -screening method to decouple the multiple time scale behavior within these structurally heterogeneous systems is computationally efficient and easily transferable to a wide range of dynamic chemical systems, providing an important new tool to study dynamic phenomena in the absence of predetermined temporal correlation functions.

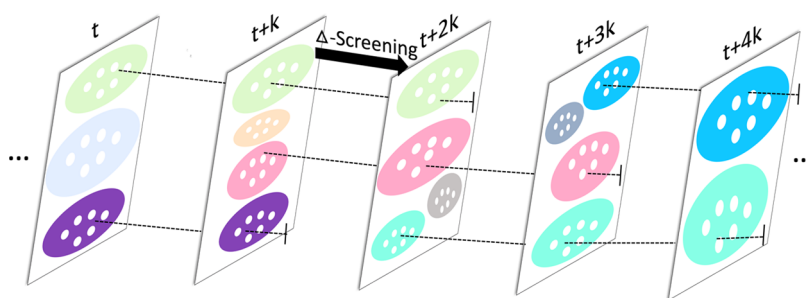
## ■ ALGORITHM DEVELOPMENT AND DATA ANALYSIS

**Time-Dependent Networks Analyzed.** Equilibrium time-dependent hydrogen bond networks of water were analyzed within a biphasic system consisting of a bulk water phase in contact with immiscible hexane laden with the surface-active amphiphile tributyl phosphate (TBP). The simulation system within this work consists of 12820 water, 1524 hexane, and 744 TBP within a periodic box, as illustrated in Figure 1. The all-atom molecular dynamics simulation



**Figure 1.** Schematic illustration of the water/TBP/hexane biphasic simulation system and associated hydrogen bond networks ( $\text{H}_2\text{O}$  as black vertices and hydrogen bonds as edges). In the organic phase, this consists of small water clusters (top) that are extracted by TBP, a percolated network in the bulk aqueous phase (middle), and  $\text{H}_2\text{O}$  at the instantaneous liquid/liquid surface forming hydrogen bonds with other water and TBP (bottom).

methodology and force fields have been previously reported by Servis and Clark.<sup>47</sup> After equilibration at 300 K, 60 ps of the production trajectory was analyzed, with the atomic coordinates recorded every 2 fs (30,000 frames within the simulation trajectory). The unweighted and undirected graph  $G(V, E)$  contains a set of vertices  $V$  corresponding to individual  $\text{H}_2\text{O}$  molecules and a set of edges  $E$  that represent hydrogen bonds (defined by a geometric criterion where an edge between two nearby  $\text{H}_2\text{O}$  is created if the  $\text{O}\cdots\text{H}$  distance  $< 2.5$  Å and the  $\text{O}\cdots\text{H}-\text{O}$  angle is  $\sim 145$ – $180^\circ$ ). An illustration is provided in Figure 1, with three major categories of hydrogen-bonding environments being present in the system: (1) the hydrogen bond network of bulk water, (2) the local hydrogen bonding about small water clusters with TBP that are



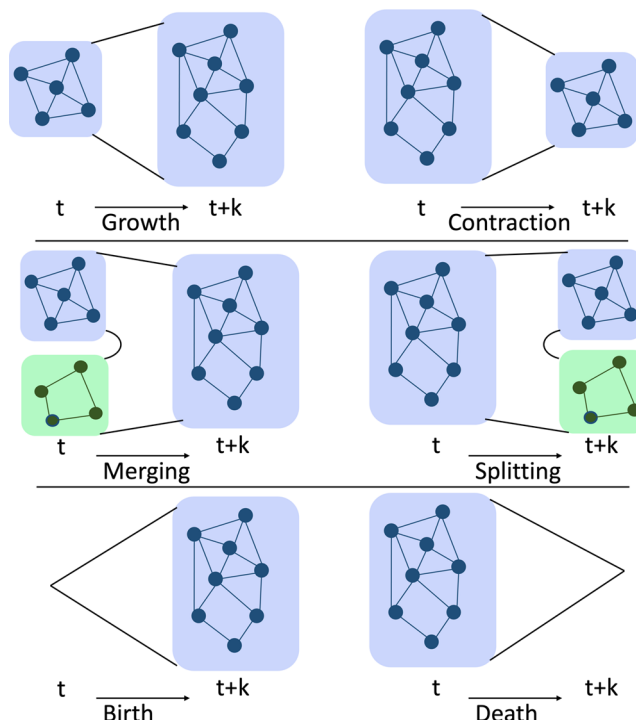
**Figure 2.** Workflow to identify temporal communities. A time sequence of  $T$  graphs employs a sampling rate  $k$ . Each community identified in the graph at the current step is updated using the community structure of the previous timestep through the  $\Delta$ -screening technique (different temporal communities are represented by a color).

extracted into the organic phase, and (3) the water hydrogen bonding at the instantaneous surface, where  $\text{H}_2\text{O}$  comes into direct contact with the surface-adsorbed TBP or hexane. The production data was subsampled for dynamic community analysis at two different time intervals: every 0.2 ps to yield 300 frames of data and every 2 ps to yield 30 frames.

#### Dynamic Chemical Community Detection Algorithm.

A dynamic network  $G(V, E)$  that evolves over  $T$  timesteps is a sequence of  $T$  graphs  $G_1(V_1, E_1), G_2(V_2, E_2), \dots, G_T(V_T, E_T)$ , where  $G_t(V_t, E_t)$  denotes the graph snapshot at timestep  $t$ . Let  $n_t$  and  $m_t$  denote the number of vertices and edges, respectively, at timestep  $t$ . Between each consecutive pair of timesteps, additions and/or removals to nodes and/or edges will occur, as illustrated in the simple example of Figure S1. The algorithmic workflow presented here<sup>48</sup> consists of two primary steps (illustrated in Figure 2): (1) community detection at each timestep  $t$  of a dynamic graph  $G(V_t, E_t)$  and (2) the subsequent step of tracking communities as they evolve across timesteps. In particular, given the network at timestep  $t$ , the goal of step (1) is to compute a set of communities  $C_t = \{C_t^1, C_t^2, \dots\}$  local to timestep  $t$ , such that  $C_t^i \cap C_t^j = \emptyset$  for any  $i \neq j$  and  $\cup_i C_t^i = V_t$ . Step (2) integrates the communities from all timesteps toward the detection of temporal communities that span across multiple (two or more) timesteps. Figure 2 illustrates this workflow for a more generic setting where the network increments are provided for every  $k$  steps (i.e., sampling rate). The forward arrow from step  $t + k$  to  $t + 2k$  illustrates step (1), which is carried out through the  $\Delta$ -screening method. The temporal communities identified from step (2) are shown using dotted horizontal lines connecting communities between individual timesteps. Note that in this formulation, for a community to be identified as temporal, it should persist for at least two consecutive timesteps (if not more). The minimum temporal persistence is a differentiating feature relative to the static community analysis that could be performed at any individual timestep and will lead to differences in the absolute value of the number of temporal communities relative to static communities.

While tracking temporal communities, the algorithm also detects various community-level events. As shown in Figure 3, for every temporal community there will be a birth timestep when the community first appears ( $t_i$ ) and a death timestep ( $t_j$ ) when it last appears—with  $t_i < t_j$ . The figure also shows other events such as growth and contraction, and merging and splitting—any of which could occur when the community is alive. In what follows, we provide the implementation details for the above two major steps of the algorithm. For ease of exposition, we assume (without loss of generality) a network



**Figure 3.** Different temporal community events are captured by the algorithm, including the birth of a community as well as its growth and contraction, the merging and splitting of communities, as well as eventual death of a community. The figure assumes a sampling rate of  $k$ , i.e., each successive network snapshots are separated by  $k$  timesteps.

sampling rate of 1 (i.e.,  $k = 1$ ) and  $T$  to denote the total number of sampled timesteps.

#### Step 1: Incremental Community Detection at Timestep $t$ .

The communities at any given timestep  $t$  are detected using the  $\Delta$ -screening algorithm,<sup>46</sup> which employs the Louvain algorithm<sup>49</sup> internally to assign communities to nodes. Importantly,  $\Delta$ -screening only uses Louvain on a selected subset of nodes that are likely to be impacted by edge additions and/or deletions that occur between time  $t$  to  $t + 1$ . Algorithm 1 presents the main steps for incremental detection. Initially, to bootstrap the incremental process, the Louvain algorithm is used to detect the set of communities ( $C_1$ ) from the graph at timestep 1, i.e.,  $G_1(V_1, E_1)$ . The set of communities at any subsequent timestep  $t > 1$  (i.e.,  $C_t$ ) are computed using the information from the communities of the previous timestep ( $C_{t-1}$ ) and the batch of edge additions (denoted by  $\Delta_{t+}$ ) and deletions (denoted by  $\Delta_{t-}$ ) in  $G_t$  relative to  $G_{t-1}$ . The  $\Delta$ -



screening algorithm internally identifies a subset of vertices (denoted by  $\mathcal{R}_t$ ) and updates the community assignments only for those nodes in that subset. The details of the selection procedure, as well as related properties and experimental analysis of the algorithm, are provided in Zarayeneh and Kalyanaraman.<sup>46</sup> After this procedure is completed for all  $T$  timesteps, the final output is a time series of community sets:  $\{C_1, C_2, \dots, C_T\}$ . Importantly, our approach may be implemented in an interleaved mode, where incremental detection of communities at any timestep is followed by the identification and tracking of temporal communities until that timestep. It is only for ease of exposition that we describe the approach as a sequential procedure where step (2) starts only after step (1) completes.

**Algorithm 1:** Dynamic community detection using  $\Delta$ -screening.<sup>46,48</sup>

---

Input:  $G_1, \{\Delta_{1+}, \Delta_{1-}, \dots, \Delta_{T+}, \Delta_{T-}\}$   
 Output:  $\{C_1, C_2, \dots, C_T\}$   
 1 Let  $C_1 \leftarrow \text{Static}(G_1)$  denote the initial step output  
 2 for each  $t \in \{2, 3, \dots, T\}$  do  
 3   Initialize  $C_t \leftarrow C_{t-1}$   
 4   // update communities based on deletions  
 5    $G_t \leftarrow \text{Update } G_{t-1} \text{ using } \Delta_{t-}$   
 6   Compute  $\mathcal{R}_{t-} \leftarrow \Delta\text{-screening}(G_t, \Delta_{t-})$   
 7   Update  $C_t \leftarrow \text{Static}(G_t, \mathcal{R}_{t-})$ —i.e., run static clustering only for vertices in  $\mathcal{R}_{t-}$   
 8   // update communities based on additions  
 9    $G_t \leftarrow \text{Update } G_t \text{ using } \Delta_{t+}$   
 10   Compute  $\mathcal{R}_{t+} \leftarrow \Delta\text{-screening}(G_t, \Delta_{t+})$   
 11   Update  $C_t \leftarrow \text{Static}(G_t, \mathcal{R}_{t+})$ —i.e., run static clustering only for vertices in  $\mathcal{R}_{t+}$   
 12 end  
 13 return  $\{C_1, C_2, \dots, C_T\}$

---

**Step 2: Temporal Community Identification and Tracking across Timesteps.** Given the time series of community sets for all  $T$  timesteps, the goal is then to identify communities that span two or more consecutive timesteps (temporal communities). Toward this end, we define the following key events at the community level:

- **Birth:** a community  $C$  is said to be born at timestep  $t$  if it has no predecessor—i.e., there is no community at timestep  $t - 1$  that shares  $\alpha$  or more fraction of its vertices in common with  $C$ . Consequently, we refer to  $\alpha$  as the predecessor threshold. When a community is born, we consider this special instance of the community to be the core for all future instances of the community (in subsequent timesteps).
- **Persistence:** a community is said to persist at timestep  $t$  if it was born at timestep  $t - 1$  or before, and if it shares at least  $\rho\%$  of its vertices with its core community. This implies satisfying two conditions: (a) that the community has a predecessor from timestep  $t - 1$  (based on  $\alpha$ ) and (b) that the community has not become too dissimilar from its core (based on  $\rho$ ). We refer to  $\rho$  as the core threshold and set  $\rho \leq \alpha$  in practice. Intuitively, thresholds  $\alpha$  and  $\rho$  ensure that the temporal community has not deviated too far from either its birth instance or node composition over the course of its life. The working hypothesis is that the temporal evolution of a community is gradual and incremental as would be expected for many chemical systems at equilibrium.
- **Death:** a community is said to die at timestep  $t$  if it persisted until timestep  $t - 1$  but ceases to exist at  $t$  (in effect violating either the predecessor condition or the core condition or both).

In the above definition of a temporal community, the notion of the predecessor threshold is intended to track the smooth evolution of a community, while the notion of the core threshold is to help us bind the divergence of a temporal

community over its entire lifetime. An alternative approach to bounding this divergence could be to compare each temporal instance against every other temporal instance of a community over its lifetime. In the literature, there are alternative forms of modularity<sup>50</sup> that are defined over this approach, where a coupling factor is defined to compare and relate each pair of timesteps. This approach, however, entails a significantly higher time complexity that would make an all-pair comparison impractical at scale for networks with thousands of timesteps. Furthermore, detecting the temporal boundaries (birth and death) of a community could also become challenging as there could be temporal gaps in communities that may have to be detected and resolved. In contrast, the combination of a single core instance alongside a predecessor instance, as defined in this paper, provides a more efficient approach. Finally, the evolution of the node composition of the community over time also has distinct chemical implications. In some chemical systems, any variation in the node composition will lead to a distinct chemical entity—having different physicochemical properties. In the system studied here, the variation in the chemical environment of the community leads to different time scale behaviors, not necessarily the internal composition of the community. The definition of the temporal community in this work provides sufficient flexibility for the chemist to tune the definition in a manner consistent with the chemical problem under study.

**Algorithmic Complexity.** Given a graph at timestep  $t$ , denoted by  $G_t(V_t, E_t)$ , taken from one of the  $T$  timesteps, the runtime complexity of our algorithmic workflow is as follows.

**Community Detection.** The cost of detecting communities at the first timestep is proportional to the size of the network  $O((|V_1| + |E_1|) \times r)$ , where  $r$  denotes the number of iterations the Louvain algorithm takes to converge (input dependent and typically a constant in tens or at most hundreds on most practical inputs<sup>49,51</sup>). For each subsequent timestep  $t$ , the  $\Delta$ -screening method<sup>46,52</sup> takes  $O(f_t(|V_t| + |E_t|) \times r_t)$ , where  $f_t \in [0, 1]$  denotes the fraction of the input graph selected by the  $\Delta$ -screening procedure and  $k_t$  denotes the number of iterations taken to converge on that reduced graph. Intuitively,  $\Delta$ -screening selects subgraphs to run Louvain based on where the changes appear, and the fractional value  $f_t$  is proportional to the sizes of the communities impacted by those changes.

**Community Tracking.** The cost of community tracking at any given timestep  $t$  is proportional to the number of community output at that timestep plus the time taken to detect predecessors for each of those communities. Note that the number of communities at any given timestep is strictly upper-bounded by the number of vertices at that timestep in the worst case (and is significantly smaller in practice). To efficiently compute the predecessor communities for a given community at step  $t$ , we keep the member list associated with each community as a sorted list so that with a linear scan of two lists we can compute the Jaccard similarity.<sup>53</sup> To avoid comparing each community  $C_t$  from timestep  $t$  with every community from timestep  $t - 1$ , we store the originating community information for each vertex of  $C_t$  provided by the  $\Delta$ -screening method—i.e.,  $C_t$  needs to be compared only against the subset of communities at  $t - 1$  from which its vertices originated. This effectively bounds the comparisons to at most the number of distinct previous communities from  $t - 1$  that contributed members to  $C_t$ . We process those candidate predecessors in the decreasing order of the contributions and

stop when we find the first predecessor that shares at least  $\alpha$  fraction of vertices in common with  $C_t$  (as there can be no others down the list when we use  $\alpha \geq 50\%$ ). Given this description, the space complexity of the algorithmic workflow is linear in the input size—i.e.,  $\sum_{t \in [1, T]} O(|V_t| + |E_t|)$ .

In the experiments for the entire hydrogen bond network (comprising 30,000 timesteps), the total runtime was 23 min, of which the first timestep took 7 s, while each subsequent timestep took 0.04 s on average. All experiments were conducted on a single CPU of a Linux server with 2.13 GHz AMD Ryzen thread ripper 1920x processors and access to 64 GB DRAM.

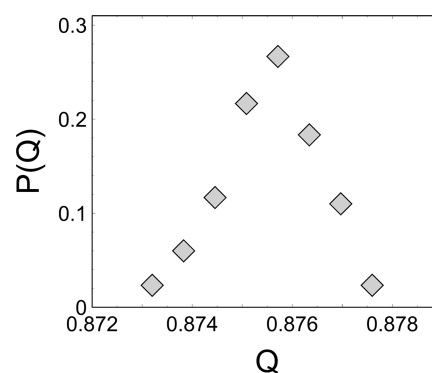
**Definitions of a Temporal Hydrogen Bond Community.** As described previously, the parameters  $\alpha$  (predecessor threshold) and  $\rho$  (core threshold) define the characteristics of temporally clustered groups of hydrogen bond networks. The predecessor threshold  $\alpha$  controls how much overlap in node composition is necessary to extend a community by a single timestep ( $t_{i-1}$  to  $t_i$ ), while the core threshold  $\rho$  ensures that the overall node composition has not far deviated from the initial instance of that community. The combination of the two influences the number of temporal communities and their duration. Here,  $\alpha$  and  $\rho$  were chosen as 0.5; however, different  $\rho$  values were also examined so as to understand how enforcing larger overlap in node composition from one timestep to the next impacted the temporal community size and duration. Based on the above scheme, each community corresponds to an active interval of timesteps when it is alive, i.e., from the timestep it was born (core) to the timestep it dies. This temporal sequence defines a temporal community and the length of the interval its duration.

## RESULTS AND DISCUSSION

We begin by presenting the results of the different stages of the dynamic community detection algorithm. The behavior of the temporal communities is then described where we demonstrate that the algorithm successfully identifies the different characteristic time scales associated with hydrogen bond networks of the biphasic chemical system: water in the bulk, at the water/TBP/hexane interface, and those water clusters that have been extracted into the organic phase. This represents a significant addition to the minimal suite of existing tools that are able to efficiently identify multiple time scale correlations within chemical simulation data.

Prior to analyzing the dynamic behavior of the temporal network, we first examined the general quality of the community structure via the modularity values reported by the Louvain algorithm at each timestep (i.e., static communities). Networks with better community-level organization typically have larger values of modularity  $Q$  (eq 1, with the theoretical maximum at 1.0). Figure 4 shows the values for the probability distribution  $P(Q)$  using the ensemble of snapshots from the sampling rate of  $\Delta t = 0.2$  ps. The modularity values are highly stable and bounded within a fine range [0.873, 0.878], implying a significant community-level structure for hydrogen bond networks.

**Temporal Evolution of the Network.** The equilibrium characteristics of the hydrogen bond network are well demonstrated through analysis of the fluctuation of network edge additions and deletions as a function of time (Figure S2). For every timestep, the edge list within the network is analyzed; all new edges at  $t + 1$  relative to  $t$  are identified, along



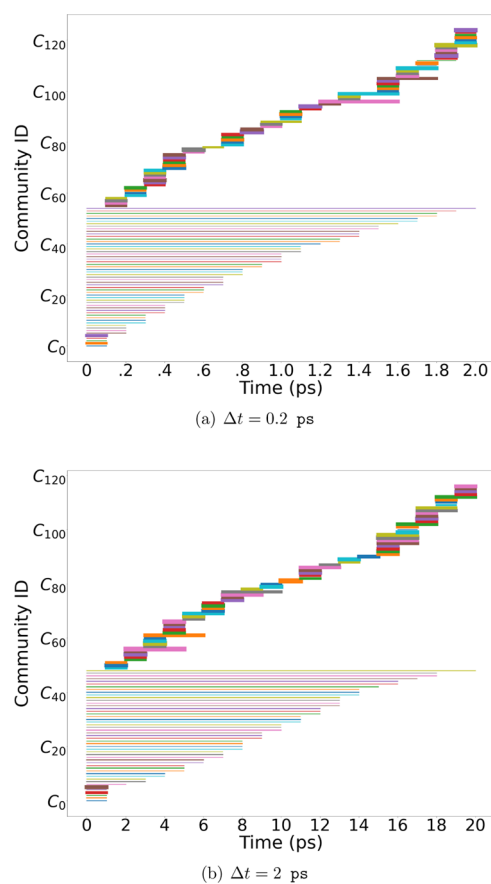
**Figure 4.** Probability distribution ( $P(Q)$ ) of the graph modularity ( $Q$ ) obtained from eq 1, illustrating the quality of the communities identified for  $\Delta t = 0.2$  ps systems.

with edges that have been lost and the set of edges present at both time  $t$  and  $t + 1$ . The total number of edges in the network at any time  $t$  is  $19863 \pm 68$ . With the more frequent sampling rate of 0.2 ps, the average number of added edges per snapshot is  $418 \pm 25$  (roughly 2% of the total edges within the network), while the number of deleted edges is  $420 \pm 25$ . On average, the number of added and deleted edges nearly cancels ( $\Delta = 10^{-3}$ ). Complementary is the number of common edges that are retained from  $t$  to  $t + 1$ , which has an average value of 19440. Thus, as the hydrogen bond network fluctuates in time, approximately 96% of the edges are maintained from one 0.2 ps increment of time to another. The average hydrogen bond edge lifetime (using 2 fs sampling) obtained from the hydrogen bond autocorrelation function fitted to a biexponential function is computed to be 5.1 ps.<sup>54,55</sup> This is consistent with the well-known dynamics of water, where multiple time scale behavior that breaks and reforms HBs is observed; vibrational motion takes place between a pair of  $\text{H}_2\text{O}$  on a time scale of 50–200 ps; hydrogen bond (HB) reorientation occurs among three  $\text{H}_2\text{O}$  at 2–3 ps; and water diffusive motion occurs among several water molecules at  $\sim 10$  ps.<sup>56–58</sup> In the case of the 2 ps sampling rate, approximately 63% of the hydrogen bond network is retained from one step to another, as would be anticipated based on the comparable time of the sampling rate and edge lifetime.

**Identification of Temporal Communities.** The hydrogen bond network structure of water has been studied since the advent of molecular dynamics and Monte Carlo to perform liquid simulations.<sup>59</sup> By defining chemical clusters, or isolated subgraphs, within the network and applying statistical physics models of site percolation,<sup>59,60</sup> a general consensus has been developed over several decades where water consists of a constantly rearranging percolating network (infinitely spanning clusters).<sup>61–64</sup> From a chemical perspective, a cluster could be considered a distinct region of the system based on the network topology that has different physicochemical characteristics (e.g., spatial density, stability, etc.). It is desirable that a community that is identified from a community-wise partitioning algorithm (which maximizes internal connectivity within a subgraph while minimizing intercommunity connectivity) is consistent with the chemical notion of a cluster. This has been recently demonstrated using the Louvain algorithm, or modularity optimization, in both Lennard Jones fluids and the biphasic system studied in this work,<sup>17</sup> as well as liquid water experiencing a shear flow,<sup>65</sup> and in electrolyte solutions.<sup>66</sup> Although the study of the dynamics and lifetimes

of chemical clusters is widely used within the simulation community, partitioning of a chemical network based on temporal features of the network topology has been relatively unexplored. Evidence for growing interest is demonstrated by the recent work of Melo et al.<sup>67</sup> that studied dynamic networks for the identification of allosteric interactions in proteins.

To identify and track temporal communities for the water hydrogen bond network, we applied dynamic chemical community detection to the two input data sets, one sampled at 2 ps and another at 0.2 ps. Figure 5 presents the birth and



**Figure 5.** Birth and death times of temporal communities (identified by their community ID), where the relative thickness of the bars indicates the number of vertices present in the community. (a) Using the  $\Delta t = 0.2$  ps data set; for the sake of clarity only 2 ps of total simulation time is illustrated. The thickest bar consists of 622 nodes, while the thinnest bar consists of 6 nodes. (b) Using the  $\Delta t = 2$  ps data set, where 20 ps is illustrated. The thickest bar consists of 593 nodes, while the thinnest bar is comprised of 6 nodes.

death times of the temporal communities identified (colored bars) in which bar thickness indicates the community size (measured in the number of participating nodes), while bar length corresponds to duration. There is significant consistency between the two sampling rates. As shown in Figure S4, maintaining a core threshold  $\rho$  of 0.5 but increasing the predecessor threshold  $\alpha$  from 0.5 to 0.6 and then 0.75 causes an interesting shift in both the size and duration of the identified temporal communities. As expected, requiring more overlap in the node composition from one timestep to the next decreases the overall number of temporal communities identified, yet within those that are identified, the overall size typically grows. Very large and short lived communities that

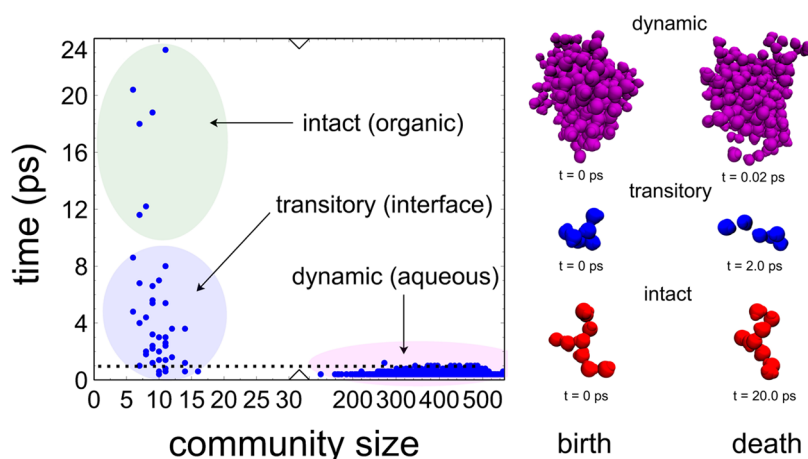
are present with  $\rho = 0.5$  are virtually absent with larger  $\rho$  values. Necessarily, this implies that the large- and short-lived communities observed with a  $\rho$  of 0.5 exhibit significant changes to node composition from one timestep to another. Complementing these data, we further examined how vertex pairs either stay together or split (or merge together) between successive pairs of timesteps. The results and corresponding observations are discussed in the Supporting Information (Figure S4). Using metrics that follow the merging and splitting events as a function of time, there is a demonstrated balance between merges and splits that occur between any two successive timesteps for vertex pairs that share a community. For most timesteps, these values are at 80%, effectively implying that for every four pairs of vertices that stay together one pair separates and one new pair enters the group.

Given the prior work using the Louvain algorithm to identify communities within bulk fluids,<sup>17,65,66</sup> we finally compare the distribution of temporal community sizes with those from the Louvain algorithm applied to the sampled snapshots (static communities). Recall that temporal communities are defined to exist only if they live for at least two snapshots, and further there are restrictions imposed based upon the core and predecessor thresholds. This will lead to a quantitative difference between the distributions of temporal vs static communities. That being said, the shapes of the two distributions should be qualitatively similar, a feature observed in Figure S5.

**Resolving Multiple Time Scale Behavior in Temporal Communities.** Figure 6 plots the time duration of a given community vs the community size. This provides three clear groupings, where cross-identification of the node IDs with their location within the simulation box delineates the different chemical environments of the temporal communities. Large communities having an average size of 400 nodes are present in the bulk aqueous phase and have exceptionally short time durations with an average value of 1.1 ps (using the 0.2 ps sampling rate). These are labeled dynamic temporal communities. Those temporal communities at the aqueous/organic interface have an average size of 10 nodes and a duration of 5 ps and are labeled transitory communities. Finally, very long lived (24 ps) and somewhat smaller (8 nodes) temporal communities are observed as clusters of  $(\text{H}_2\text{O})_n$  that are extracted by the amphiphile TBP into the organic phase (labeled intact communities). Note that intact communities do not reflect the majority of  $\text{H}_2\text{O}$  that gets extracted, as the predominant species in the organic phase is the  $\text{TBP}(\text{H}_2\text{O})\text{TBP}$  dimer and the temporal communities considered in this work consist only of  $\text{H}_2\text{O}\cdots\text{H}_2\text{O}$  hydrogen bonds as edges.

Interestingly, the average duration of the bulk aqueous dynamic communities (1.1 ps) is shorter than that of an individual hydrogen bond (calculated to be ca. 5 ps). To explain this behavior, we selected a random subset of temporal communities with a duration of 0.4 ps (two snapshots at the 0.2 ps sampling rate) and studied their ratio of the number of intracommunity edges (i.e., edges connecting two nodes that are both inside the community) to intercommunity edges (i.e., edges leaving that community to connect to nodes in a neighboring community). In all cases examined, the disintegration of the temporal communities correlated with a degradation of their respective ratios. For example, in one such community that survived only for two timesteps, the number of nodes decreased from 658 nodes (at  $t$ ) to 436





**Figure 6.** (Left) Distribution of the temporal community size and their respective duration (using 0.2 ps sampling), with three regions identified and cross-correlated to the node ID positions with the simulation box: dynamic communities of short duration present in the bulk aqueous phase, transitory communities present at the aqueous/organic interface, and intact communities of  $(\text{H}_2\text{O})_n$  bound to the amphiphile TBP that have been extracted into the organic phase. (Right) Representative structures of the three identified types of temporal communities.

nodes (at  $t + 1$ ), before eventually dying (at  $t + 2$ ). This correlated to the intra- vs intercommunity edge ratios of 5.57 (at  $t$ ) and 3.98 (at  $t + 1$ ).

These data are consistent with the observation that the large temporal communities are influenced by the value of the predecessor threshold  $\alpha$  (as explained in the Supporting Information discussion on temporal community analysis; Figure S3). Intuitively, a larger  $\alpha$  can make it harder for nodes to be retained in communities from one timestep to the next, unless the intra- vs intercommunity edge ratio is maintained. Considering that these relatively large bulk aqueous dynamic temporal communities are losing their internal edges and disintegrating very quickly in time and that there are only a small fraction of edges added or removed between successive timesteps ( $\sim 4\%$ , as shown in Figure S2 of the Supporting Information) imply that these edge removals are locally concentrated in the bulk aqueous region. Finally, with respect to the intra- vs intercommunity edge ratio, note that the modularity objective  $Q$  is closely tied to this ratio—i.e., a community is likely to be formed when the ratio increases from  $t$  to  $t + 1$  and likely to be broken when the ratio decreases. Yet, this is in strong contrast to the analysis of the modularity of the static communities shown in Figure 4 for the total network, which indicated an otherwise strong community structure. As such, there remains an opportunity to consider further improvements in modularity optimization in the context of temporal communities relative to their static analogues and in considering which algorithm is best suited to a given network structure. For example, the Louvain algorithm may have a resolution limit for modularity that prevents the detection of small communities if they are internally disconnected. To improve these issues, the Leiden community detection algorithm<sup>68</sup> in combination with the constant Potts model has been suggested.<sup>69</sup> As discussed in the Supporting Information, any modularity algorithm may be chosen within the  $\Delta$ -screening approach. Our own comparisons between Louvain vs Leiden algorithms did not yield appreciable differences in the temporal community detection.

## CONCLUSIONS

The recently developed  $\Delta$ -screening algorithm has been applied to study the dynamic evolution of intermolecular

networks of interactions within a ternary biphasic chemical system. The identified temporal communities differ from static communities present within individual timesteps as they must persist over  $>1$  instance in time and have constraints upon the allowed changes to node composition in time (relative to both the node composition at the time of community birth and the variation in node composition from one timestep to the next). These features may be tuned based on the nature of the chemical system to optimize resolution. The algorithm is computationally efficient, as it uses edge formation and deletion data to selectively apply the Louvain algorithm to subsets of nodes within the network. It further monitors the dynamic evolution of communities in the context of merging and splitting events, providing well-defined avenues for studying dynamic chemical behavior. Within the system under study, the method clearly identifies three groups of temporal communities, that further have variations in community size. These are the large highly dynamic networks of bulk aqueous phase, the mid-size longer-duration transitory communities of water at the instantaneous interface of water/hexane, and the exceptionally long-lived intact smaller communities of water that are transported into the organic phase by the surface-active amphiphile tributyl phosphate.

## ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acs.jctc.2c00454>.

Illustrations of dynamic networks; proof of equilibration in the dynamic network; and comparison of different community properties and Louvain vs Leiden modularity algorithms (PDF)

## AUTHOR INFORMATION

### Corresponding Author

Aurora E. Clark — Department of Chemistry, Washington State University, Pullman, Washington 99164, United States; Pacific Northwest National Laboratory, Richland, Washington 99354, United States; [orcid.org/0000-0001-9381-721X](https://orcid.org/0000-0001-9381-721X); Email: [aurora.clark@utah.edu](mailto:aurora.clark@utah.edu)

## Authors

Neda Zarayeneh — School of Electrical Engineering and Computer Science, Washington State University, Pullman, Washington 99164, United States

Nitesh Kumar — Department of Chemistry, Washington State University, Pullman, Washington 99164, United States; [orcid.org/0000-0003-3322-8450](https://orcid.org/0000-0003-3322-8450)

Ananth Kalyanaraman — School of Electrical Engineering and Computer Science, Washington State University, Pullman, Washington 99164, United States; [orcid.org/0000-0001-6721-233X](https://orcid.org/0000-0001-6721-233X)

Complete contact information is available at:  
<https://pubs.acs.org/10.1021/acs.jctc.2c00454>

## Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

N.Z. and A.K. were supported by the NSF awards OAC 1910213, CCF 1815467, and CCF 1919122, while N.K. and A.E.C. were supported by the DOE BES Separations Program, award DE-SC0001815.

## REFERENCES

- (1) Balaban, A. T. Applications of Graph Theory in Chemistry. *J. Chem. Inf. Comput. Sci.* **1985**, 25, 334–343.
- (2) Biggs, N.; Lloyd, E. K.; Wilson, R. J. *Graph Theory*, 1736-1936; Oxford University Press, 1986.
- (3) Luzar, A.; Chandler, D. Hydrogen-Bond Kinetics in Liquid Water. *Nature* **1996**, 379, 55–57.
- (4) Luzar, A.; Chandler, D. Effect of Environment on Hydrogen Bond Dynamics in Liquid Water. *Phys. Rev. Lett.* **1996**, 76, 928.
- (5) Poole, P. H.; Sciortino, F.; Grande, T.; Stanley, H. E.; Angell, C. A. Effect of Hydrogen Bonds on the Thermodynamic Behavior of Liquid Water. *Phys. Rev. Lett.* **1994**, 73, 1632.
- (6) Jungwirth, P.; Tobias, D. J. Ions at the Air/Water Interface. *J. Phys. Chem. B* **2002**, 106, 6361–6373.
- (7) Velichko, Y. S.; Stupp, S. I.; De La Cruz, M. O. Molecular Simulation Study of Peptide Amphiphile Self-Assembly. *J. Phys. Chem. B* **2008**, 112, 2326–2334.
- (8) Liang, Z.; Bu, W.; Schweighofer, K. J.; Walwark, D. J.; Harvey, J. S.; Hanlon, G. R.; Amoanu, D.; Erol, C.; Benjamin, I.; Schlossman, M. L. Nanoscale View of Assisted Ion Transport Across the Liquid-Liquid Interface. *Proc. Natl. Acad. Sci. U.S.A.* **2019**, 116, 18227–18232.
- (9) Benjamin, I. Molecular Structure and Dynamics at Liquid-Liquid Interfaces. *Annu. Rev. Phys. Chem.* **1997**, 48, 407–451.
- (10) Granovetter, M. S. The Strength of Weak Ties. *Am. J. Sociol.* **1973**, 78, 1360–1380.
- (11) Newman, M. E. J. The Structure of Scientific Collaboration Networks. *Proc. Natl. Acad. Sci. U.S.A.* **2001**, 98, 404–409.
- (12) Girvan, M.; Newman, M. E. Community Structure in Social and Biological Networks. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, 99, 7821–7826.
- (13) Newman, M. E. J. Modularity and Community Structure in Networks. *Proc. Natl. Acad. Sci. U.S.A.* **2006**, 103, 8577–8582.
- (14) Brandes, U.; Dellinger, D.; Gaertler, M.; Görke, R.; Hofer, M.; Nikolski, Z.; Wagner, D. On Modularity Clustering. *IEEE Trans. Knowl. Data Eng.* **2008**, 20, 172–188.
- (15) Fortunato, S. Community Detection in Graphs. *Phys. Rep.* **2010**, 486, 75–174.
- (16) Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast Unfolding of Communities in Large Networks. *J. Stat. Mech.: Theory Exp.* **2008**, 2008, P10008.
- (17) Servis, M. J.; Clark, A. E. Cluster Identification Using Modularity Optimization to Uncover Chemical Heterogeneity in Complex Solutions. *J. Phys. Chem. A* **2021**, 125, 3986–3993.
- (18) Rossetti, G.; Cazabet, R. Community Discovery in Dynamic Networks: A Survey. *ACM Comput. Surv.* **2018**, 51, 1–37.
- (19) Hopcroft, J.; Khan, O.; Kulis, B.; Selman, B. Tracking Evolving Communities in Large Linked Networks. *Proc. Natl. Acad. Sci. U.S.A.* **2004**, 101, S249–S253.
- (20) Cuzzocrea, A.; Folino, F.; Pizzuti, C. DynamicNet: An Effective and Efficient Algorithm for Supporting Community Evolution Detection in Time-Evolving Information Networks. In *Proceedings of the 17th International Database Engineering & Applications Symposium*, 2013; pp 148–153.
- (21) Asur, S.; Parthasarathy, S.; Ucar, D. An Event-Based Framework for Characterizing the Evolutionary Behavior of Interaction Graphs. *ACM Trans. Knowl. Discovery Data* **2009**, 3, 1–36.
- (22) He, J.; Chen, D.; Sun, C.; Fu, Y.; Li, W. Efficient Stepwise Detection of Communities in Temporal Networks. *Phys. A* **2017**, 469, 438–446.
- (23) Seifkar, M.; Farzi, S.; Barati, M. C-Blondel: An Efficient Louvain-Based Dynamic Community Detection Algorithm. *IEEE Trans. Comput. Soc. Syst.* **2020**, 7, 308–318.
- (24) Palla, G.; Barabási, A.-L.; Vicsek, T. Quantifying Social Group Evolution. *Nature* **2007**, 446, 664–667.
- (25) Greene, D.; Doyle, D.; Cunningham, P. Tracking the Evolution of Communities in Dynamic Social Networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, 2010; pp 176–183.
- (26) Tantipathananandh, C.; Berger-Wolf, T.; Kempe, D. A Framework for Community Identification in Dynamic Social Networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007; pp 717–726.
- (27) Bassett, D. S.; Porter, M. A.; Wymbs, N. F.; Grafton, S. T.; Carlson, J. M.; Mucha, P. J. Robust Detection of Dynamic Community Structure in Networks. *Chaos* **2013**, 23, 013142.
- (28) Matias, C.; Miele, V. Statistical Clustering of Temporal Networks Through a Dynamic Stochastic Block Model. 2015, arXiv:1506.07464. arXiv.org e-Print archive. <https://arxiv.org/abs/1506.07464> (accessed June 22, 2016).
- (29) Matias, C.; Rebafka, T.; Villers, F. A Semiparametric Extension of the Stochastic Block Model for Longitudinal Networks. *Biometrika* **2018**, 105, 665–680.
- (30) Xu, K. S.; Hero, A. O. Dynamic Stochastic Blockmodels for Time-Evolving Social Networks. *IEEE J. Sel. Top. Signal Process.* **2014**, 8, 552–562.
- (31) Viard, T.; Latapy, M.; Magnien, C. Computing Maximal Cliques in Link Streams. *Theor. Comput. Sci.* **2016**, 609, 245–252.
- (32) Himmel, A.-S.; Molter, H.; Niedermeier, R.; Sorge, M. Enumerating Maximal Cliques in Temporal Graphs. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2016; pp 337–344.
- (33) Bu, Z.; Li, H.-J.; Zhang, C.; Cao, J.; Li, A.; Shi, Y. Graph K-Means Based on Leader Identification, Dynamic Game, and Opinion Dynamics. *IEEE Trans. Knowl. Data Eng.* **2020**, 32, 1348–1361.
- (34) Maillard, P.; Görke, R.; Staudt, C.; Wagner, D. Modularity-Driven Clustering of Dynamic Graphs. In *Experimental Algorithms, 9th International Symposium, SEA 2010*, 2009; pp 436–448.
- (35) Aktunc, R.; Toroslu, I. H.; Ozer, M.; Davulcu, H. A Dynamic Modularity Based Community Detection Algorithm for Large-Scale Networks: DSLM. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2015; pp 1177–1183.
- (36) Xie, J.; Chen, M.; Szymanski, B. K. LabelrankT: Incremental Community Detection in Dynamic Networks via Label Propagation. In *Proceedings of the Workshop on Dynamic Networks Management and Mining*, 2013; pp 25–32.
- (37) Seifi, M.; Guillaume, J.-L. Community Cores in Evolving Networks. In *Proceedings of the 21st International Conference on World Wide Web*, 2012; pp 1173–1180.



- (38) Zakrzewska, A.; Bader, D. A.A Dynamic Algorithm for Local Community Detection in Graphs. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, 2015; pp 559–564.
- (39) Lin, Y.-R.; Chi, Y.; Zhu, S.; Sundaram, H.; Tseng, B. L.Facetnet: A Framework for Analyzing Communities and their Evolutions in Dynamic Networks. In *Proceedings of the 17th International Conference on World Wide Web*, 2008; pp 685–694.
- (40) Agarwal, P.; Verma, R.; Agarwal, A.; Chakraborty, T.DyPerm: Maximizing Permanence for Dynamic Community Detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2018; pp 437–449.
- (41) Zeng, X.; Wang, W.; Chen, C.; Yen, G. G. A Consensus Community-based Particle Swarm Optimization for Dynamic Community Detection. *IEEE Trans. Cybern.* **2020**, *50*, 2502–2513.
- (42) Messaoudi, I.; Kamel, N. A Multi-Objective Bat Algorithm for Community Detection on Dynamic Social Networks. *Appl. Intell.* **2019**, *49*, 2119–2136.
- (43) Zhao, Z.; Li, C.; Zhang, X.; Chiclana, F.; Viedma, E. H. An Incremental Method to Detect Communities in Dynamic Evolving Social Networks. *Knowl.-Based Syst.* **2019**, *163*, 404–415.
- (44) Wu, L.; Zhang, Q.; Guo, K.; Chen, E.; Xu, C. Dynamic Community Detection Method Based on an Improved Evolutionary Matrix. *Concurrency Comput. Pract. Exper.* **2019**, e5314.
- (45) Zhuang, D.; Chang, M. J.; Li, M. DynaMo: Dynamic Community Detection by Incrementally Maximizing Modularity. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 1934–1945.
- (46) Zarayeneh, N.; Kalyanaraman, A. Delta-Screening: A Fast and Efficient Technique to Update Communities in Dynamic Graphs. *IEEE Trans. Network Sci. Eng.* **2021**, *8*, 1614–1629.
- (47) Servis, M. J.; Clark, A. E. Surfactant-Enhanced Heterogeneity of the Aqueous Interface Drives Water Extraction into Organic Solvents. *Phys. Chem. Chem. Phys.* **2019**, *21*, 2866–2874.
- (48) [https://bitbucket.org/nzarayeneh/dynamic\\_community\\_detection](https://bitbucket.org/nzarayeneh/dynamic_community_detection).
- (49) Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast Unfolding of Communities in Large Networks. *J. Stat. Mech.: Theory Exp.* **2008**, 2008, P10008.
- (50) Mucha, P. J.; Richardson, T.; Macon, K.; Porter, M. A.; Onnela, J.-P. Community structure in time-dependent, multiscale, and multiplex networks. *Science* **2010**, *328*, 876–878.
- (51) Lu, H.; Halappanavar, M.; Kalyanaraman, A. Parallel Heuristics for Scalable Community Detection. *Parallel Comput.* **2015**, *47*, 19–37.
- (52) Zarayeneh, N.; Kalyanaraman, A.A Fast and Efficient Incremental Approach Toward Dynamic Community Detection. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2019; pp 9–16.
- (53) Jongman, E.; Jongman, S. R. R.*Data Analysis in Community and Landscape Ecology*; Cambridge University Press, 1995.
- (54) Benjamin, I. Hydrogen Bond Dynamics at Water/Organic Liquid Interfaces. *J. Phys. Chem. B* **2005**, *109*, 13711–13715.
- (55) Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A Toolkit for the Analysis of Molecular Dynamics Simulations. *J. Comput. Chem.* **2011**, *32*, 2319–2327.
- (56) Ojha, D.; Karhan, K.; Kühne, T. D. On the Hydrogen Bond Strength and Vibrational Spectroscopy of Liquid Water. *Sci. Rep.* **2018**, *8*, No. 16888.
- (57) Tokmakoff, A. Shining Light on the Rapidly Evolving Structure of Water. *Science* **2007**, *317*, 54–55.
- (58) Bakker, H. J.; Skinner, J. Vibrational Spectroscopy as a Probe of Structure and Dynamics in Liquid Water. *Chem. Rev.* **2010**, *110*, 1498–1517.
- (59) Geiger, A.; Stillinger, F. H.; Rahman, A. Aspects of the Percolation Process for Hydrogen-Bond Networks in Water. *J. Chem. Phys.* **1979**, *70*, 4185–4193.
- (60) Stanley, H. E. A Polychromatic Correlated-Site Percolation Problem with Possible Relevance to the Unusual Behaviour of Supercooled H<sub>2</sub>O and D<sub>2</sub>O. *J. Phys. A: Math. Gen.* **1979**, *12*, L329–L337.
- (61) Stanley, H. E.; Teixeira, J. Interpretation of the Unusual Behavior of H<sub>2</sub>O and D<sub>2</sub>O at Low Temperatures: Tests of a Percolation Model. *J. Chem. Phys.* **1980**, *73*, 3404–3422.
- (62) Geiger, A.; Stanley, H. E. Low-Density “Patches” in the Hydrogen-Bond Network of Liquid Water: Evidence from Molecular-Dynamics Computer Simulations. *Phys. Rev. Lett.* **1982**, *49*, 1749–1752.
- (63) Blumberg, R. L.; Stanley, H. E.; Geiger, A.; Mausbach, P. Connectivity of Hydrogen Bonds in Liquid Water. *J. Chem. Phys.* **1984**, *80*, 5230–5241.
- (64) Pártay, L. B.; Jedlovsky, P.; Brovchenko, I.; Oleinikova, A. Formation of Mesoscopic Water Networks in Aqueous Systems. *Phys. Chem. Chem. Phys.* **2007**, *9*, 1341–1346.
- (65) Gao, Y.; Fang, H.; Ni, K. A Hierarchical Clustering Method of Hydrogen Bond Networks in Liquid Water undergoing shear flow. *Sci. Rep.* **2021**, *11*, No. 9542.
- (66) Feng, Y.; Fang, H.; Gao, Y.; Ni, K. Hierarchical Clustering Analysis of Hydrogen Bond Networks in Aqueous Solutions. *Phys. Chem. Chem. Phys.* **2022**, *24*, 9707–9717.
- (67) Melo, M. C. R.; Bernardi, R. C.; de la Fuente-Nunez, C.; Luthey-Schulten, Z. Generalized Correlation-Based Dynamical Network Analysis: A New High-Performance Approach for Identifying Allosteric Communications in Molecular Dynamics Trajectories. *J. Chem. Phys.* **2020**, *153*, 134104.
- (68) Traag, V. A.; Waltman, L.; Van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **2019**, *9*, No. 5233.
- (69) Wu, F.-Y. The potts model. *Rev. Mod. Phys.* **1982**, *54*, 235.