Layer-by-layer unsupervised clustering of statistically relevant fluctuations in noisy time-series data of complex dynamical systems

PNAS RESEARCH ARTICLE | PHYSICS

Layer-by-layer unsupervised clustering of statistically relevant fluctuations in noisy time-series data of complex dynamical systems

Matteo Becchi[a], Federico Fantolino[a], and Giovanni M. Pavan[a,b,1]

Affiliations are included on p. 11.

Edited by Pratyush Tiwary, University of Maryland at College Park, College Park, MD; received February 23, 2024; accepted July 1, 2024, by Editorial Board Member Monica Olvera de la Cruz

Complex systems are typically characterized by intricate internal dynamics that are often hard to elucidate. Ideally, this requires methods that allow to detect and classify in an unsupervised way the microscopic dynamical events occurring in the system. However, decoupling statistically relevant fluctuations from the internal noise remains most often nontrivial. Here, we describe "*Onion Clustering*": a simple, iterative unsupervised clustering method that efficiently detects and classifies statistically relevant fluctuations in noisy time-series data. We demonstrate its efficiency by analyzing simulation and experimental trajectories of various systems with complex internal dynamics, ranging from the atomic- to the microscopic-scale, in- and out-of-equilibrium. The method is based on an iterative detect-classify-archive approach. In a similar way as peeling the external (evident) layer of an onion reveals the internal hidden ones, the method performs a first detection/classification of the most populated dynamical environment in the system and of its characteristic noise. The signal of such dynamical cluster is then removed from the time-series data and the remaining part, cleared-out from its noise, is analyzed again. At every iteration, the detection of hidden dynamical subdomains is facilitated by an increasing (and adaptive) relevance-to-noise ratio. The process iterates until no new dynamical domains can be uncovered, revealing, as an output, the number of clusters that can be effectively distinguished/classified in a statistically robust way as a function of the time-resolution of the analysis. *Onion Clustering* is general and benefits from clear-cut physical interpretability. We expect that it will help analyzing a variety of complex dynamical systems and time-series data.

fluctuations | time-series analysis | unsupervised clustering | complex dynamical systems

Significance

Discriminating statistically relevant fluctuations from noise is crucial in many fields, from machine learning to the analysis of signals and complex systems. We describe *Onion Clustering*, an unsupervised clustering method that can identify fluctuations and microscopic dynamical domains in noisy time-series data of any kind. The method proceeds layer-by-layer, classifying dynamical environments (clusters) in time-series data from the most evident to the least populated (hidden) ones, and iterating until no further dynamical clusters can be discriminated/classified in a statistically robust way. *Onion Clustering* analyses are fully data-driven and essentially parameter-free. Tested on simulation and experimental trajectories of different types of systems, *Onion Clustering* stands out as a general, robust, physically interpretable method useful to characterize and understand complex dynamical systems.

Understanding the dynamics of complex systems is typically a hard task and presents inherent challenges. Cause-and-effect relationships, as well as the spatial and temporal correlations, are often hidden within the noise generated by a large number of units that dynamically communicate with each other in an intricate network of interactions (1–6). The behavior of these systems is often controlled by local (rare) fluctuations, but detecting and distinguishing them from the intrinsic noise of datasets extracted from their trajectories is often nontrivial (7). This holds for a variety of systems across different scales, from the atomic- and molecular- to the macroscopic-level (8). The relevance of local microscopic fluctuations has been shown, for example, in studies of metal surfaces and nanoparticles (9, 10), supramolecular fibers (11–13), and nucleation processes (14, 15). On a macroscopic scale, the effects of local fluctuations and events on the behavior of the whole system are seen in collective phenomena such as, e.g., bird flocks (16–18), fish banks (19, 20), as well as in the dynamics of economic and stock market systems (2, 21, 22). The study of the behavior of these complex systems over time, by either computer simulations or experimental setups, typically generates a large amount of multivariate data that are often nontrivial to analyze. In particular, extracting meaningful and interpretable information from complex multibody trajectories is generally hard (23). To address this issue, common strategies involve the use of either knowledge-based or data-driven descriptors (24–26). Such descriptors, and the noisy time-series that they typically provide, serve as a crucial intermediary step to effectively reduce the raw data to an interpretable form and to facilitate the extraction of human-interpretable information useful for elucidating the underlying dynamics.

Structural descriptors—either specific and knowledge-based, or abstract, general ones—are often used to extract comprehensive insights into the structural features of complex systems. As an example, general high-dimensional structural descriptors, such

The authors declare no competing interest.

This article is a PNAS Direct Submission. P.T. is a guest editor invited by the Editorial Board.

[1]To whom correspondence may be addressed. Email: giovanni.pavan@polito.it.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2403771121/-/DCSupplemental.

Published August 7, 2024.

as, e.g., the Smooth Overlap of Atomic Positions (SOAP) descriptor (27), have been recently used to obtain a data-driven structural characterization of, e.g., water and ice systems (26, 28–31), metallic (10, 32), ionic (33), and soft (biological or artificial) molecular systems (11, 34–36) from molecular dynamics (MD) simulations. However, at the same time, pattern recognition analyses based on such structural descriptors typically struggle in capturing infrequent dynamical events and local fluctuations that play a pivotal role in determining their behavior (8, 10, 37–39). Conversely, it has been demonstrated how time-series analyses tracking the temporal evolution and fluctuations of descriptors in time allow retaining a richer amount of information all the events occurring in complex molecular systems. One recent example is the time-SOAP (*t*SOAP) descriptor, which measures the rate of change of the SOAP power spectrum of each unit in a multiunit trajectory of a dynamical molecular system (38). A time-series analysis of *t*SOAP was recently shown to retain rich information of the structural change events that occur within molecular systems, including rare local events. Another example is the Local Environments and Neighbor Shuffling (LENS) descriptor, which tracks changes in the identity of the neighbor units that surround every unit in a dynamical network (39). While these examples show the potential of studying the behavior of complex systems based on the trajectories of their individual units over time, this shifts the focus from pattern recognition on global datasets to the study of time-series data and of their dynamical fluctuations.

One key challenge in time-series analysis is clustering (40–43), and in particular, the identification and classification of fluctuations that are relevant against the background noise (44, 45). Unsupervised clustering algorithms frequently struggle in identifying rare events and sparse fluctuations due to their negligible statistical weight, and because the detection of more populated clusters implicitly sets a metric that is too coarse to discriminate well less populated ones. Typically, the higher the density of certain clusters, the more difficult it is to detect and classify the less populated ones. Detecting and retaining information on relevant fluctuations, separating them from noise, is of key importance to reconstruct the physics of the studied systems (46). Other approaches involve, for example, the detection of change points and abrupt changes in the time-series signals, which can identify, e.g., change of states, transitions, etc. (47). While powerful, these methods also suffer from characteristic limitations: e.g., they may struggle in treating nonstationary time-series, and they may suffer of limited robustness in evaluating/guaranteeing the statistical relevance of the detected events. For example, to cite a few, it is not always easy to understand to what extent a detected fluctuation is indeed statistically significant compared to noise, or the results (number and type of detected fluctuations) may depend on the resolution used in the time-series analysis.

Such issues are particularly relevant when dealing with complex systems, whose collective and adaptive behaviors often emerge locally (both in time and space) and are intimately related to rare events and local fluctuations (12, 48). Unsupervised approaches capable of providing a microscopic analysis of time-series via systematic and robust detection and clustering of the fluctuations occurring within them would be desirable to this end. However, the most common clustering algorithms are either built to handle static datasets, or to perform whole time-series clustering (49–52), and are thus not well-suited to obtain a single-point (microscopic-level) clustering of the local dynamical events occurring in the time-series (53–55).

Here, we introduce *Onion Clustering*, a general, simple, unsupervised, and physically interpretable algorithm tailored for single-point clustering of fluctuations in noisy time-series data. Our approach is founded on the general concept that every (microscopic) environment in a system is characterized by an average dynamics and by a characteristic noise (amplitude of fluctuations around the mean). As a core idea, the algorithm is based on an iterative detect-classify-archive approach where, step-by-step, the highest-density microscopic dynamical environment present in the system is detected, its dynamical features (average dynamics and characteristic noise) are classified, and its signal (and the related noise) is then removed from the time-series, which is then analyzed again according to the same iterative procedure. In a similar way as peeling the external layer of an onion reveals the internal hidden ones, after the classification of the evident dynamical environments, at every iteration the method can efficiently uncover and classify the hidden (least populated) dynamical domains thanks to an iteratively enhancing signal-to-noise ratio. In this way, *Onion Clustering* allows extracting all the features that can be effectively classified in a time-series. Noteworthy, instead of leaving the user to make an a priori choice on the resolution to be used for an analysis (which is critical, and typically requires a prior knowledge of the system under study), *Onion Clustering* reveals as an output the number of clusters that can be effectively classified in a statistically robust way in a time-series as a function of the time-resolution used in the analysis. This provides a robust unsupervised clustering algorithm with a noncommon physical interpretability that allows for a transparent intuition into the mechanism of clusters detection and an informed interpretation of the obtained results.

We demonstrate the efficiency and generality of *Onion Clustering* by analyzing a variety of complex dynamical systems, ranging from the microscopic to the mesoscopic scales, with diverse internal dynamics, in- and out-of-equilibrium conditions. *Onion Clustering* is open-source (56, 57), and is released as a Python3 package (58). We expect that this method will constitute a precious tool to study complex dynamical systems in general, and the microscopic events occurring within them and controlling their behavior.

Results and Discussion

The Method and a Test on Water-Ice Dynamic Coexistence. In this section, we illustrate the algorithm using as a first demonstrative case the clustering of data extracted from a 50 ns long MD simulation trajectory containing 2,048 TIP4P/ICE (59) molecules (1:1 liquid water:ice) in dynamic equilibrium in correspondence of the melting temperature; see Fig. 1*A*. A complete description of the algorithm is provided in *Materials and Methods* and in *SI Appendix*.

Fig. 1*B* shows, as an example, the LENS signals (39) for all 2,048 individual water molecules in the system sampled every $\Delta t = 0.1$ ns. The input dataset in this example thus consists of $N = 2{,}048$ univariate time-series $x_i(t)$, $1 \leq i \leq N$ labeling the water molecules in the simulation trajectory, each containing $0 \leq \Delta t < T$ sampled time-steps. We underline that the same analysis can be conducted also using different descriptors—see, e.g., *SI Appendix*, Fig. S1 for the consistent results obtained using the *t*SOAP descriptor (38): As it is shown in the next sections, *Onion Clustering* is general and can be applied virtually to any time-series data.

LENS is a permutation-invariant descriptor that measures how much the neighborhood of each (water) molecule in the system changes in term of neighbor molecular individuals in the sampling time-interval ($\Delta t = 0.1$ ns in this case). In detail, LENS captures local events such as reshuffling, addition, or loss of neighbor molecules, and it can be thought of as a local dynamicity

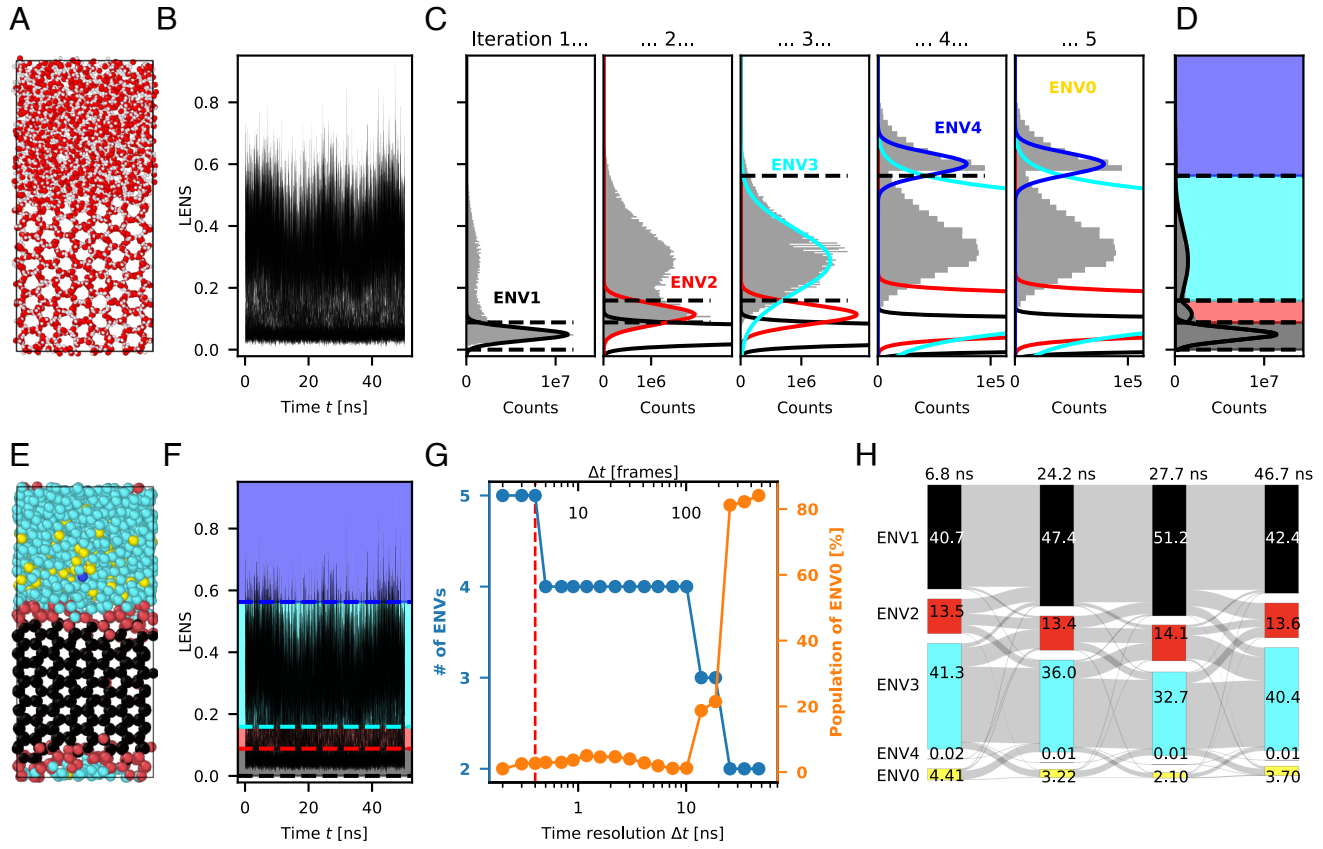

Fig. 1. Clustering of LENS signals on ice/water coexistence MD simulation. (*A*) Snapshot of the simulation of ice/water coexistence; the simulation is performed on 2,048 TIP4P/ICE molecules, lasts 50 ns and it is sampled every 0.1 ns. (*B*) LENS signals for all the oxygen atoms, as a function of time. (*C*) Data cumulative histograms at the five iterations of the algorithm, using a time resolution $\Delta t = 0.4$ ns. The solid lines are the Gaussian best fit of the maximum of the distribution. The dashed lines are the threshold between the identified clusters. At the fifth iteration, no data are assigned to the proposed cluster, and the algorithm stops. (*D*) final clustering of the LENS signals. (*E*) Snapshot of the simulation, colored according to the clustering. (*F*) Same LENS signals of panel (*B*); background is colored according to the thresholds given by the clustering algorithm. (*G*) Blue line: number of clusters identified as a function of the time resolution Δt; orange line: percentage of the data points classified in the ENV0 cluster as a function of the time resolution Δt. The red dashed line at $\Delta t = 0.4$ ns indicates at which time resolution the analysis shown in the previous panels was performed. The time resolution is expressed both in physical units (*Lower* primary *x* axis) and in number of simulation frames (*Upper* secondary *x* axis). (*H*) Sankey diagram between four different times along the simulation. Colored bars are proportional to the clusters' populations, gray lines are proportional to the number of molecules moving from one cluster to another.

parameter related, in some sense, to a local diffusivity of the molecules in the system. The LENS signal is thus expected to be lower in the solid phase (ice) and higher in the liquid water phase, where the molecular rearrangement is faster. Fig. 1*B* shows the LENS signals time-series. It is worth mentioning that typical unsupervised pattern recognition approaches used to analyze the entire dataset basically detects two main environments—liquid water and solid ice—in such a system (28, 31, 38, 60), where both states are well represented statistically (see also leftmost panel of Fig. 1*C*: two peaks in the density of the signals at LENS values of ∼0.05 and ∼0.3) (39). However, this becomes problematic in cases where there are states/environments that are present in a low fraction and that are typically overlooked in pattern recognition analyses due to their negligible statistical weight. Similarly, for the same reasons such approaches struggle in providing information on the (rare) transitions between the main states and on the involved intermediate transition states. On the other hand, recently it has been demonstrated that studying the time-series of such signals allows detecting and retaining information also of rare/local transition events that appear as outliers in the time-series (38, 39). However, to what extent one fluctuation is different from noise or from another fluctuation, how similar/different the various fluctuations are, and, in particular, with what statistical confidence it is thus possible to group them based on their similarity are typically nontrivial questions.

Using this as a first demonstrative case, we show how *Onion Clustering* is capable of performing a microscopic analysis of the time-series, subdividing them into different dynamical environments whose fluctuations have characteristic fingerprints in terms of intensities and oscillation amplitudes. The algorithm automatically identifies in an unsupervised way the dynamical microclusters that may be present in the system (the number of clusters is thus not set a priori, but is rather an output of the algorithm) and assign points to them, assessing their difference/similarity in a statistically robust way. The method follows a layer-by-layer approach, where the environments that are more evident/certain are first detected and classified and, after removing them from the signal, the algorithm proceeds iteratively in classifying the less-evident/hidden ones. In particular, in a first iteration ("Iteration 1"), *Onion Clustering* starts by computing the cumulative histogram of all the data points in the time-series (leftmost panel of Fig. 1*C*). The global maximum of the histogram is then identified: In this test case, the LENS signals have the maximum density at LENS ∼0.05 (a relatively low value, corresponding to the solid-ice phase: *vide infra*). The idea behind the algorithm is to assume that each maximum of the histogram corresponds to one well-determined dynamic environment in

the system, which is thus characterized by an average dynamics (average LENS value) and by a normally distributed characteristic noise. Based on this concept, once identified the highest-density peak, the algorithm fits a Gaussian distribution of the form

$$P(x) = \frac{A}{\sqrt{\pi}\sigma} \exp\left[-\left(\frac{x-\mu}{\sigma}\right)^2\right] \quad [1]$$

to the histogram maximum, as shown in Fig. 1*C* (black solid curve). The mean μ, the (rescaled) SD σ and the area A of the Gaussian are the fit parameters.

This identifies a first dynamical environment, labeled as "ENV1," which is characterized by LENS values within the interval $[\mu - 2\sigma, \mu + 2\sigma]$ (and that in this case identifies the solid ice domain). This criterion is equivalent to discard data points that do not belong to ENV1 with a probability higher that 99.5%. As a next step, the algorithm slices the time-series in consecutive (nonoverlapping) time-windows of length Δt. The algorithm identifies all the molecules that remain always in ENV1 (without jumping in/out ENV1) in Δt, for all Δts along the trajectory, thus classifying all molecules that, at the resolution of the analysis (Δt), appear as persistently belonging to ENV1 for time intervals at least equal to (or multiple of) Δt. After this step, all these already classified ENV1 signals are removed from the data and the time-series is analyzed again in another iteration.

It is worth noting that Δt is the only parameter required by *Onion Clustering*. In time-series analysis, the choice of the Δt is critical, as it sets de facto the time-resolution in the analysis (as it will be discussed in more detail below). Larger values of Δt correspond to a lower resolution, while smaller values of Δt correspond to a higher resolution in the study of the time-series, respectively reducing/enhancing the discretization of the events that occur along the studied trajectory. To prevent the use of the algorithm by the users as a black box (or leveraging too much prior assumptions/knowledge on/of the system), *Onion Clustering* performs the analysis at many different Δts and outputs the results that can be effectively obtained at the different resolutions (see next section for a detailed discussion on the effect of changing the Δt in the analysis). As a demonstrative case, here in Fig. 1, we show the results obtained by using a $\Delta t = 0.4$ ns, which corresponds to 4 simulation time-frames in the analysis of water-ice molecules that coexist in dynamic equilibrium (results obtained with other Δt values are available in *SI Appendix*, Fig. S2).

The second iteration starts again by computing the cumulative histogram of the data points. As can be seen in the "Iteration 2" panel of Fig. 1*C*, the removal of the points classified into ENV1 changes the histogram. Now, environments that before were difficult to identify as hidden by the ENV1 data/noise become the new prominent features of the histogram. In this way, by identifying and removing a new environment at each iteration, the algorithm automatically adjusts the data range in order to improve its efficiency in identifying environments which are less and less statistically relevant (note, in fact, the finer and finer scale on the $x-$axes of Fig. 1*C*).

The algorithm then proceeds exactly as in the previous iteration. The global maximum is identified, and fitted with a Gaussian distribution (solid red line in Fig. 1*C*), which gives the limit of the new environment, "ENV2." Then, the data-windows entirely included into ENV2 are detected, stored, and removed. The same procedures are repeated iteratively. As it is shown in Fig. 1*C*, in this specific system at this resolution ($\Delta t = 0.4$ ns) four environments can be identified, which are characterized by increasing values (and lower densities) of LENS signal.

Such *find-classify-archive* strategy builds on a hierarchical certainty approach that classifies first the data that are more certain and then, layer-by-layer, proceeds in classifying hierarchically the remaining part of the time-series. Noteworthy, eliminating the ENV1 data after the classification results also in the deletion of the associated noise, which augments in the next iteration the sensitivity of the method and the relevance-to-noise ratio. At the fifth iteration, a new environment "ENV5" is fitted. But no signal window in the remaining dataset is entirely included within it: i.e., there are no molecules that stay into such ENV5 at least for the duration of $\Delta t = 0.4$ ns. The algorithm thus meets a termination condition, and the iterative process stops. The remaining data points, which were not classified into any of the previously identified environments (at least with this choice of Δt), are classified as a last cluster, labeled as "ENV0." ENV0 contains all the data that are not persistently part of ENVs1-4 for at least Δt (e.g., transitions).

The key importance of such ENV0 environment from the physical, statistical, and methodological points of view is discussed in detail in the next section.

Once the iterative analysis terminates, the algorithm determines the thresholds between the different environments, defined as the intersection points between the various Gaussian distributions (Fig. 1 *D–F*: dashed lines). This identifies the main ENV1-4 clusters colored in Fig. 1 *D–F*. In this specific case, the algorithm finds 4 statistically relevant LENS environments (ENVs1-4), along with the ENV0 cluster. The characterization of the LENS signals within each cluster is displayed in *SI Appendix*, Fig. S3. As can be seen from the simulation snapshot in Fig. 1*E* and in Movie S1, the cluster ENV1 corresponds to the solid ice phase, ENV2 to the solid/liquid interface (ice surface), ENV3 comprises the majority of the molecules in the liquid water phase, while ENV4 contains a smaller fraction of water molecules that, as described recently (31), may occasionally form ephemeral ice-like clusters that in such conditions continuously freeze and remelt in the liquid domain.

The key importance of time-resolution. Changing the time resolution of the analysis, Δt, determines what type of information can be effectively captured and how much information is lost. Setting the Δt means setting the sensitivity and uncertainty in the analysis, in that the resolution is sufficient to classify some events but not other (faster) ones. This reflects in the number of clusters (ENVs) that are classified by the analysis. For example, reducing the Δt increases the resolution in the study of the time-series, and results in an augmented discretization and a higher number of detected clusters (ENVs). At the same time, the amount of information that remains "undetermined" at a given Δt is also exactly quantified by the ENV0 cluster. In particular, the higher is the data content of the ENV0 cluster, the higher is the amount of information that cannot be classified in a statistically robust way. In this specific case, where $\Delta t = 0.4$ ns, the molecules belonging to the ENV0 cluster and corresponding to fast transitions between the ENVs1-4 environments weight ~3.5% of the total data points.

As anticipated above, instead of making an a priori choice of the time-resolution—typically leveraging on a considerable prior knowledge of the system by an expert user, or on a "blind" unsupervised choice that risks to make the software a "black box"—*Onion Clustering* uses a different strategy that improves its transparency and physical interpretability. In particular, the software always performs the analysis at different values of Δt, ranging from the maximum resolution of $\Delta t = 2$ frames, to the minimum one, corresponding to $\Delta t = T$, where T is the entire time-series (the latter case results in a typical pattern-recognition

analysis conducted on the entire dataset). In this demonstrative case, the analysis is conducted ranging from $\Delta t = 0.2$ ns (2 frames) to $\Delta t = 47$ ns (470 frames, comparable with the entire trajectory length). At every usage, *Onion Clustering* outputs a plot such that of Fig. 1*G*. In blue and orange are respectively shown the number n of statistically relevant clusters that can be classified in a robust way (ENV1-to-n) and the fraction (in %) of unclassified data contained in the ENV0 cluster as a function of the Δt. For smaller Δt values (up to $\Delta t = 0.4$ ns) 5 clusters are found (4 statistically relevant ones—ENV1-to-4—plus the ENV0, which collects the unclassified data points). For intermediate Δt values ($0.5 \leq \Delta t \leq 10$ ns), the ENV clusters reduce to 4. Reducing the resolution of the analysis (increasing the Δt), ENV4, which corresponds to molecules with very high LENS values (identifying ephemeral ice-like domains forming/dissolving in the liquid water), merges with ENV3 (corresponding to liquid water; see also *SI Appendix*, Fig. S2). Evidently, the resolution is no more high enough to discriminate such molecules from liquid ones. Noteworthy, this outcome is also physically relevant, because it reveals the maximum time-scale at which such ephemeral ice-like domains can be effectively discriminated from a statistical point of view and provide rough information on their survival lifetime (which is shorter than 500 ps).

Increasing further the Δt (>10 ns) starts producing a loss of information that can be effectively classified. It is not possible anymore to distinguish ENV2—i.e., the solid/liquid interface—as a distinct cluster, and only ENV1 (solid ice) and ENV3 (liquid water), along with the unclassified ENV0 cluster, can be identified. This outcome provides a qualitative estimate for the average lifetime of a water molecule in ENV2 (0.5 to 10 ns), which is compatible with previous studies on the diffusion coefficient of water molecules at the ice/water interface (61).

It is worth noting how for $\Delta t > 10$ ns the fraction of data in the ENV0 cluster (unclassified data) sharply increases. This indicates that the time resolution starts to be insufficient to reconstruct the microscopic physics of the system, and a significant fraction of data points remain unclassified during the iterative process. In particular, for $\Delta t > 20$ ns the sole environments that can be detected are ENV1, corresponding to the bulk of solid ice (molecules that do not diffuse along the entire simulation) and ENV0, gathering in this case ~80% of the total data points, which includes all molecules that move in the system. Interestingly, in this case, the result of the analysis becomes consistent with the typical result that is obtained via unsupervised clustering approaches on datasets extracted from the entire trajectory (28, 31, 38, 60).

The plot of Fig. 1*G* is a key feature of *Onion Clustering*, providing relevant information. On the one hand, it sheds light on the physics underlying the system under study. On the other hand, it provides important information on the performance of the clustering algorithm and on the robustness of the classification that this provides. The correlation between the number of detected clusters and Δt unveils the characteristic time-scales of the various dynamic environments and the transitions occurring within the system. Conversely, the correlation between the population of the ENV0 cluster and Δt indicates the time resolution at which the algorithm begins to struggle, having insufficient resolution and statistics to classify large parts of the time-series. The plot of Fig. 1*G* is a key output of *Onion Clustering* in that it provides the user with a statistically robust litmus paper useful to choose a posteriori the resolution of the analysis depending on the type of events that one wants to study (instead of a priori, e.g., based on human-based assumptions). This is a noncommon feature for a fully unsupervised method, which in this way gets rid of "black box" issues/limitations and gains physical interpretability. Instead of attempting to fit all data into clusters, the philosophy of *Onion Clustering* is to determine the amount of information that cannot be statistically classified at a given resolution (starting from the concept that every measurement method has intrinsic limitations that cannot be neglected), to subtract it, and to classify only the data that can be effectively classified from a statistical point of view. This is key, as it provides an advantage in terms of transparency, reliability, robustness, and repeatability of the analysis.

Characterizing the microscopic dynamics of the system. Having assigned every data point to one of the identified clusters, it is easy, e.g., to track not only how the different cluster populations vary with time, but also the transitions of the individual water molecules between the various environments along the time-series. In the Sankey diagram of Fig. 1*H*, the height of the colored bars is proportional to the populations of the five detected clusters at four different representative time steps taken along the trajectory. The gray bands between the time steps provide a coarse-grained representation of the number of molecules that moved between any pair of clusters in the time-interval between the two represented snapshots. While the diagram of Fig. 1*G* is here purely demonstrative, and it shows just the departure and arrival clusters for water molecules between distant time intervals, a more detailed characterization of the exchange pathways and of the inner microscopic dynamics of the system can be easily attained by tracking the transitions between shorter time intervals. Nonetheless, this plot clearly shows that, as expected, the exchange of water molecules between solid and liquid phases occurs mainly via an intermediate dynamical environment (i.e., via the ice/water interface). The unclassified (ENV0) events occur mainly in connection with the liquid phase. This indicates that, among the various transitions that this encompasses, considerable part of ENV0 is related to local ephemeral ice-like domains that quickly form/dissolve in the liquid domain in these conditions (31) (events that occur too fast to be statistically classified as a distinct cluster at the time resolution of $\Delta t = 0.4$ ns).

Different Test Cases in Different Conditions. The results discussed above refer to a case of a system in dynamical equilibrium, with a rather "fluid" internal dynamics and characterized by exchange events taking place between similarly populated liquid and solid phases. While this is a particular case, to prove the generality of the method we tested *Onion Clustering* on time-series obtained from a variety of systems with diverse internal dynamics: e.g., systems far from the equilibrium, or dominated by local rare fluctuations. Finally, to prove the broad applicability of the algorithm, this is also tested on multivariate/multidimensional time-series extracted both from synthetic and experimental datasets.

Onion Clustering of out-of-equilibrium time-series: Water freezing. Analyzing time-series data that are out-of-equilibrium poses additional challenges compared to well-sampled equilibrium trajectories. Short-lived clusters and transient states may rapidly emerge and disappear, representing only a small fraction of the data with a negligible weight on the entire dataset. Furthermore, in such systems, the result of pattern recognition approaches conducted on the entire trajectory changes over time (as the density of the states does). In fact, information on transient states that, e.g., may emerge only in the beginning of the time-series disappearing rapidly, but that may be key to understand the evolution of the system, is lost when the time-series that is

analyzed becomes longer and longer. Retaining information on these short-lived states is essential for understanding the time evolution of a system, but keeping memory of these, or in some cases even realizing that they ever appeared during a trajectory, is not always easy.

We thus tested *Onion Clustering* in a prototypical test case where, starting from the equilibrium condition of Fig. 1, the temperature in the system is reduced to $T = 267$ K. Such temperature is below the melting point of the TIP4P/ICE model (see *Materials and Methods* for details). In this case, the simulation trajectory that is analyzed is approximately 5 ns long, with sampling every 1 ps (for a total of ∼5,000 frames), which is a sufficient time to observe the entire system freezing. As an example, we computed the *t*SOAP descriptor (38) for all water molecules in these out-of-equilibrium trajectories (for comparison with the same system at the equilibrium, see *SI Appendix*, Fig. S1). In brief, *t*SOAP is a scalar descriptor that measures the rate of variation of the SOAP spectrum (27) of all molecules in the system. It thus gauges the rate of structural rearrangement within the atomic environment of the molecules: lower in the ice phase, and higher in liquid water. The resulting time-series are shown in Fig. 2*A*. In this plot, it can be seen that the highest values of *t*SOAP (>0.03), identifying molecules in the liquid phase (faster structural rearrangement of their neighbors), tend to disappear after ∼2 ns, leaving only the lower *t*SOAP values corresponding to molecules in the solid ice phase. As can be seen in the leftmost cumulative histogram of Fig. 2*B*, already after ∼5 ns the statistical weight of the data points with *t*SOAP > 0.03 is negligible, which makes it hard to detect, in analysis conducted on the entire dataset, that there has even been liquid water in this system (and the problem becomes more severe if the simulation last longer).

Fig. 2*B* shows the iterations of *Onion Clustering* (here as an example, the results obtained using a $\Delta t = 25$ ps are shown). The classified clusters are shown in Fig. 2 *C* and *D*. Also in this case, at most five environments can be identified in the time-series, corresponding respectively to the ice, the ice/water interface, and two liquid water microenvironments with different *t*SOAP values (and that can be discriminated only at high resolution), along with the ENV0 cluster encompassing the unclassified data points. The significance of these clusters becomes evident in the simulation snapshots shown in Fig. 2*F* and in Movie S2. Noteworthy, using $\Delta t = 25$ ps the algorithm accurately classifies liquid water and the ice/water interface, despite these environments vanishing after only 2 ns of simulation.

Fig. 2*E* shows the number of clusters and the population of the ENV0 cluster as a function of the Δt. The number of clusters decreases from a maximum of 5 for $\Delta t < 40$ ps to 2 for $\Delta t > 0.5$ ns. At the same time, the fraction of unclassified data in the ENV0 cluster remains negligible up to $\Delta t = 0.1$ ns,

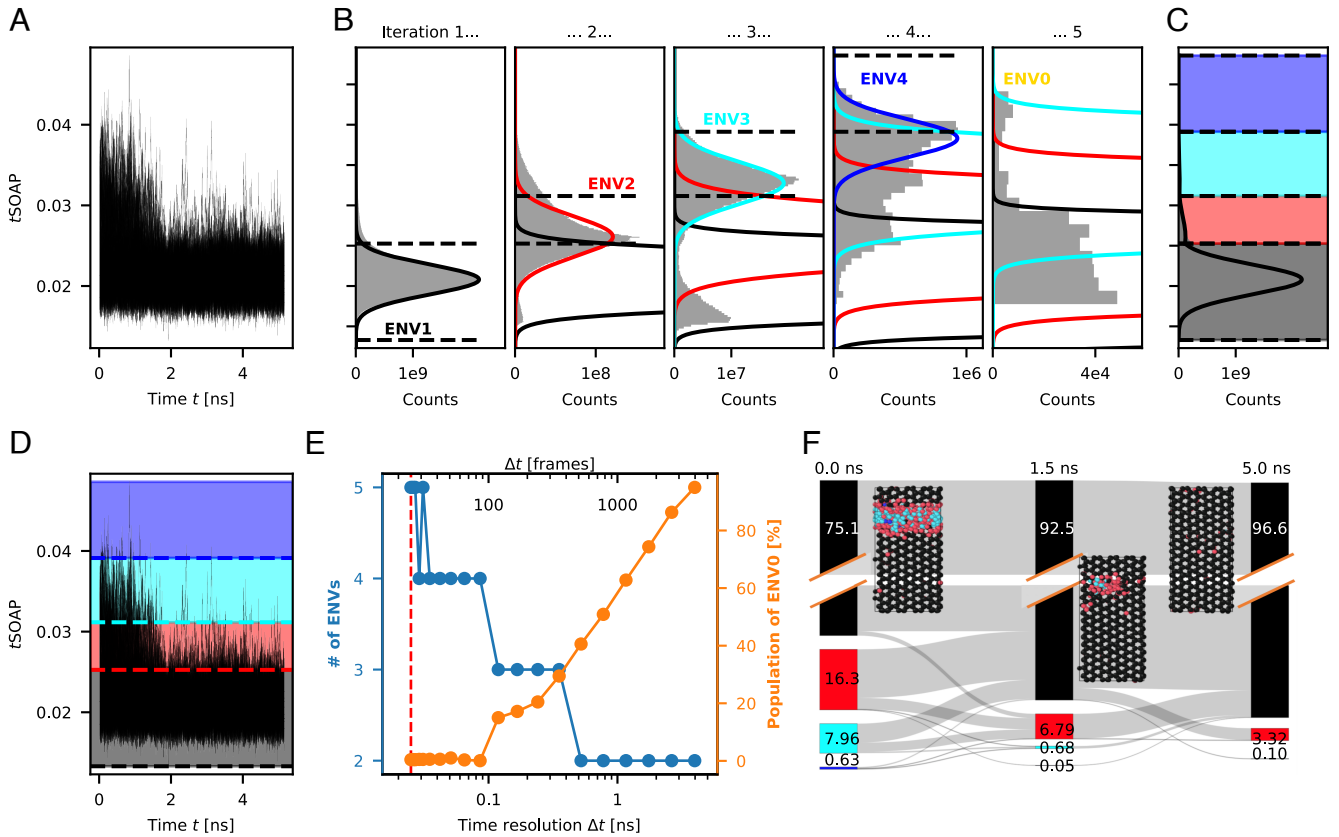

Fig. 2. Analysis of out-of-equilibrium time-series: freezing water. (*A*) *t*SOAP values for 2,048 TIP4P/ICE molecules, along the 5 ns long MD simulation sampled every ps, smoothed with a moving average with width 25 frames (25 ps). (*B*) Data cumulative histograms at the five iterations of the algorithm. The solid lines are the Gaussian best fit of the maximum of the distribution. The dashed lines are the thresholds between the identified clusters. At the fifth iteration, the Gaussian fit does not converge, and the algorithm stops. (*C*) final clustering of the *t*SOAP signals. (*D*) The *t*SOAP signals; background is colored according to the threshold given by the clustering. (*E*) Blue line: number of clusters identified as a function of the time resolution Δt; orange line: percentage of the data points classified in the ENV0 cluster as a function of the time resolution Δt. The red dashed line at $\Delta t = 0.025$ ns indicates at which time resolution the analysis shown in the previous panels was performed. (*F*) Sankey diagram between three different times along the simulation. Colored bars are proportional to the clusters' populations, gray lines are proportional to the number of molecules moving from one cluster to another.

while it begins to rise increasing the Δt, since the time-resolution is insufficient to track the fast evolution of the system. Notably, this Δt value is considerably lower than that observed in the previous section (see Fig. 1 for the LENS analysis and *SI Appendix*, Fig. S1 for the *t*SOAP analyses of the equilibrium system). Such a discrepancy is essentially due to a different relevance/noise ratio between the *t*SOAP and LENS descriptors and to the fact that the events become faster when the system evolves rapidly far-from-the-equilibrium. Anyways, such a test shows how also in this case *Onion Clustering* reveals in automatic and unsupervised way the resolution necessary to statistically characterize the events that occurred in the beginning of the trajectory, not only providing information that is nontrivial to retain but also a physical anchor to prove their relevance and robustness.

In this test case, an examination of the cluster populations and exchange rates in Fig. 2*F* offers a deeper insights, clearly demonstrating the out-of-equilibrium behavior observed in the trajectory. As the simulation time progresses, the proportion of liquid water diminishes, then followed by the interface and ultimately leading to their disappearance, while the majority of molecules undergo transition to the solid phase.

Rare local events in time-series: Atomic dynamics on metal surfaces. Another scenario where clustering algorithms often struggle is in detecting amid background noise and classifying rare events and local fluctuations that may be dominant but have a negligible statistical weight. As a prototypical example of such a system, we tested *Onion Clustering* on LENS time-series extracted from an atomistic MD simulation trajectory of a FCC(211) copper surface consisting of 2,400 Cu atoms. The simulation is conducted at a temperature $T = 600$ K using a deep-potential neural network force field that has been recently reported (10) (see *Materials and Methods* for details). The MD trajectory lasts 150 ns and is sampled every 10 ps (for a total of 15,000 frames). As shown in Fig. 3*A*, it is known that in this system, while the majority of the surface atoms vibrate within the atomic lattice, a small number of sparse atoms may undergo rapid long-distance sliding motion on the Cu surface (10, 32, 39). Specifically, such sliding motions are well captured by the LENS descriptor, which has been computed for all atoms along the trajectory, obtaining the time-series of Fig. 3*B* (LENS values $\gtrsim$ 0.1 identify atomic sliding events).

We performed an *Onion Clustering* on these LENS time-series (Fig. 3 *B* and *C*), using a time resolution for the analysis of $\Delta t = 0.12$ ns (equal to 12 simulation frames). Four statistically relevant LENS environments are identified (ENV1-4), along with the ENV0 cluster. Fig. 3*B* shows the thresholds between the LENS environments/clusters. ENV1 and ENV2 (colored in black and cyan respectively) together encompass ~99.95% of the data points, which correspond to static bulk and surface atoms in the system. Remarkably, despite this issue, the algorithm is

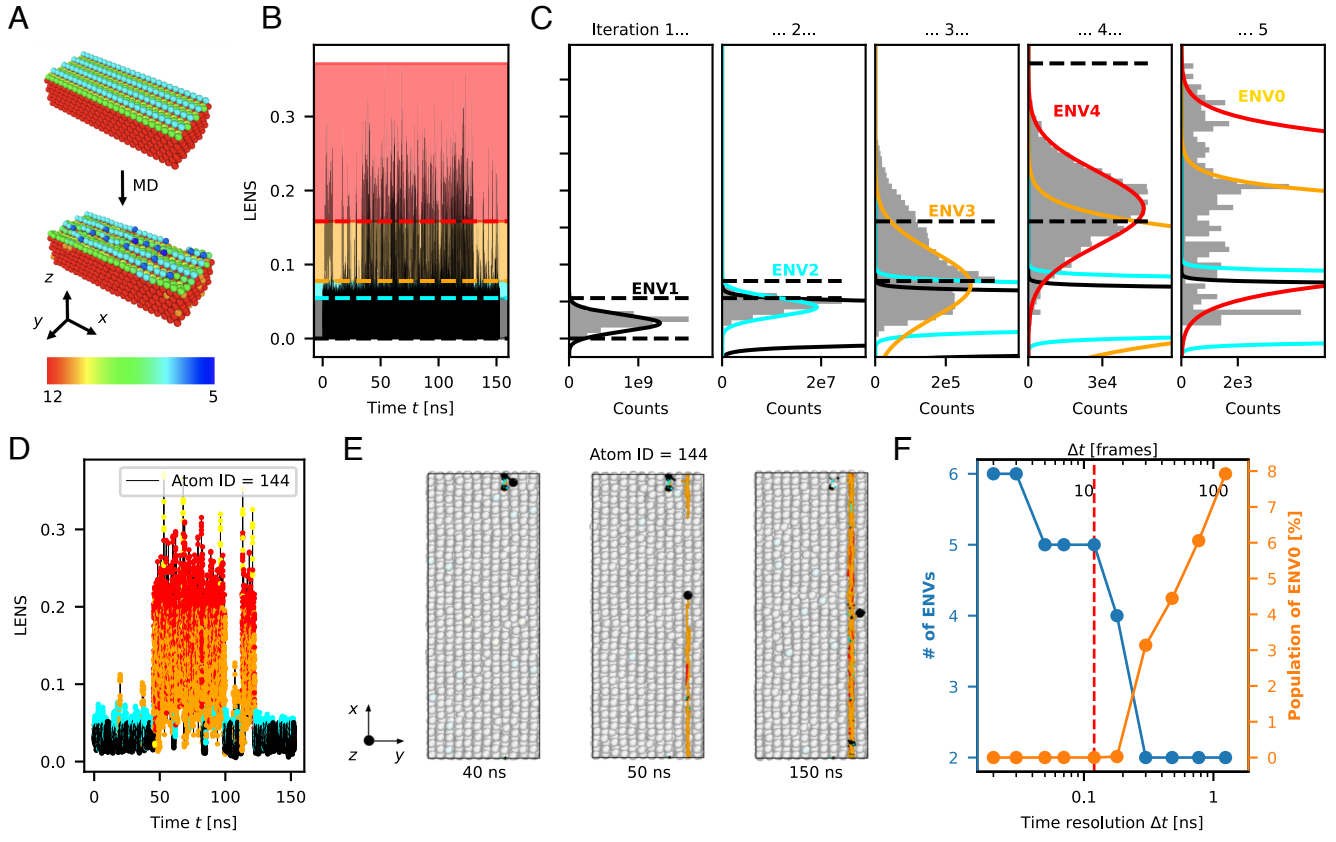

Fig. 3. Analysis of time-series dominated by rare events. (*A*) Snapshots of the MD simulation of Cu surface composed by 2,400 atoms at $T = 600$ K; atoms are colored according to their coordination number. The *Upper* snapshot is at $T = 0$ *K*, the *Lower* one during the simulation at $T = 600$ *K*. (*B*) LENS values for all the Cu atoms, along the 150 ns long simulation sampled every 10 ps, smoothed with a moving average with width 10 frames. (*C*) Data cumulative histograms at the five iterations of the algorithm. The solid lines are the Gaussian best fit of the maximum of the distribution. The dashed lines are the threshold between the identified clusters. After the fifth iteration, no data are assigned to the proposed cluster, and the algorithm stops. (*D*) the LENS signal for the atom with ID = 144, colored according to the cluster it belongs at each frame. (*E*) Top view of the simulation box, at three different times $t = 40$, 61, and 150 ns. The atom with ID = 144 is highlighted in black, and its trajectory up to that point is colored according to its environment. (*F*) Blue line: number of clusters identified as a function of the time resolution Δt; orange line: percentage of the data points classified in the ENV0 cluster as a function of the time resolution Δt. The red dashed line at $\Delta t = 0.12$ ns indicates at which time resolution the analysis shown in the previous panels was performed.

able to correctly assign the remaining data points to the other microscopic dynamical environments (ENV3-4, respectively in orange and red), which identify the rapid sliding motion of some atoms on the Cu surface. Fig. 3*D* shows a detail of the LENS time-series for one atom (ID: 144) that slides on the surface along the simulation. Fig. 3*E* shows the atom's positions at three distinct time frames along with its preceding trajectory, color-coded according to the visited LENS clusters. Until $t \sim 40$ ns, the atom remains nearly stationary on the surface (classified in ENV1-2). For $t \gtrsim 40$ ns the atom starts sliding along one of the surface facets (and is classified in ENV3-4: orange, red). From $t \sim 125$ ns, the atom is reincorporated into the surface lattice, returning to ENV1-2. Movie S3 shows the complete MD trajectory colored based on the detected clusters.

Fig. 3*F* shows how 6/5 LENS clusters can be clearly classified with a negligible information loss up to $\Delta t \sim 0.1$ to 0.2 ns time resolution. However, such atomic sliding events are so rapid that for larger Δt these get lost in the analysis. From $\Delta t > 0.3$ ns the total number of LENS clusters diminishes to 2, and the algorithm can distinguish only the static (ENV1) from the nonstatic (ENV0) atoms.

Analysis of multivariate time-series. While the examples above show the efficiency of *Onion Clustering* in analyzing univariate (unidimensional) time-series, in many case, it is desirable to conduct multidimensional analyses to minimize information loss. We thus extended the method to make it capable of processing also multivariate time-series. The main adaptation concerns the use of a multivariate Gaussian distribution for fitting the histogram maxima. As a proof of efficiency, we thus tested the method on prototypical examples of 2- or 3-dimensional time-series data, using a factorized Gaussian distribution (see *Materials and Methods* for details).

As a first test case, we constructed a synthetic 3-dimensional time-series data, generated by simulating $N = 2$ noninteracting particles. The particles move via Langevin dynamics in a three-dimensional free energy landscape featuring 5 distinct minima. Shown in Fig. 4 *A* and *B*, such a simple dataset shows 5 clear maximum density clusters separated by sparse data points. As illustrated in Fig. 4*C*, *Onion Clustering* effectively detects the 5 clusters (results obtained using $\Delta t = 5$ frames).

The plot of Fig. 4*D* shows the impact of varying the time resolution Δt. The correct number of clusters is accurately identified up to $\Delta t = 16$ frames. Reducing the resolution and using a larger Δt in the analysis of the time-series, the clusters start to merge leading to a clear information loss. In fact, the population of ENV0 remains <10% up to $\Delta t = 7$ frames, while beyond this limit it increases rapidly.

This simple example shows how *Onion Clustering* can be used also to analyze in general multivariate time-series data. This includes also datasets that are less artificial and more noisy than this synthetic example, and not necessarily coming from simulated trajectories, as discussed in the next section.

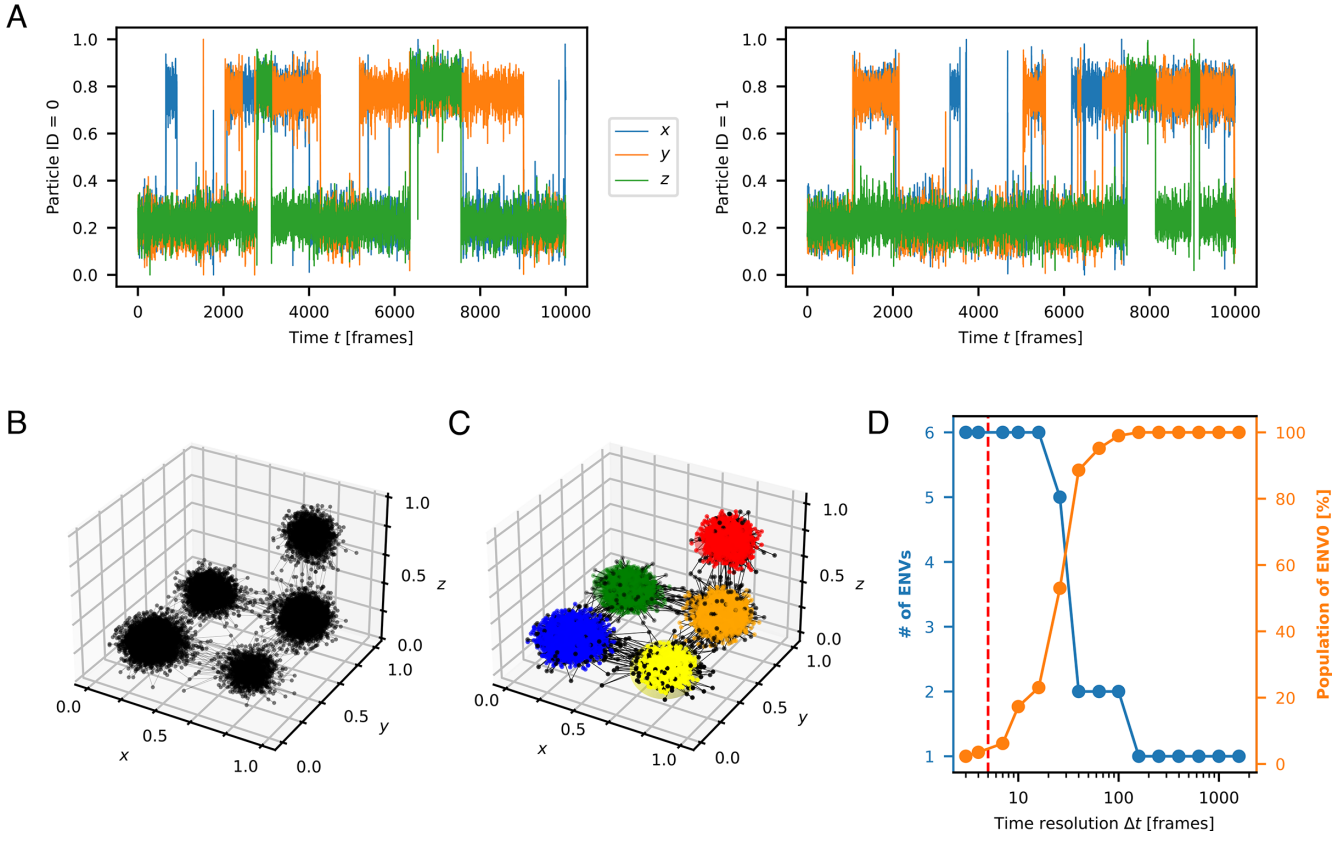

Fig. 4. Analysis of synthetic multivariate time-series. (*A*) Time-correlated data-points were generated simulating two particles (shown on the *Left* and *Right* panels) with Langevin Dynamics in a three-dimensional free-energy landscape with 5 minima. (*B*) the same trajectories, represented as a three-dimensional signal. (*C*) The output of the clustering algorithm. The identified clusters are represented as ellipsoidal surfaces, including the points closer than 2σ from the center. (*D*) Blue line: number of clusters identified as a function of the time resolution Δt. Orange line: percentage of the data points classified in the ENV0 cluster as a function of the time resolution Δt. The red dashed line at $\Delta t = 5$ frames indicates at which time resolution the analysis shown in the previous panels was performed.

Onion Clustering of experimental multidimensional time-series. As a last example, we tested *Onion Clustering* onto multivariate experimental time-series data sourced from a recent study of the complex dynamics of colloidal Quincke roller particles by Liu et al. (62). Briefly, Quincke rollers are μm scale dielectric colloidal particles suspended in a conducting fluid and exposed to a vertical DC electric field (Fig. 5*A*). While for a detailed description of these systems we refer the reader to the original publication, what is interesting to us here is that, under the stimulus of the electric field, these particles exhibit complex collective motions, eventually manifesting as collective density waves or vortexes. Noteworthy, unlike the molecular-scale examples, this test deals with a complex mesoscopic system, and the data originate from experimental observations rather than from simulations. As a proof of concept, we considered an optical microscope movie tracking $N = 6{,}921$ particles in a field of view is $700 \times 700\ \mu m^2$ for 0.25 s of real time (for a total of $T = 310$ frames), where a collective density wave emerges and runs in the system (62). From this movie, we extracted the particles' positions at each time-frame along the trajectory using the python package Trackpy (63, 64).

For each particle in the system, we extracted from the trajectory data at each sampled frame i) the minimum neighbor distance ($d_{\min}$: a proxy for the local particle density), and ii) the particles' local velocity alignment, computed as

$$\phi_i \equiv \frac{1}{n_c^i} \sum_j \frac{\vec{v}_i \cdot \vec{v}_j}{|\vec{v}_i||\vec{v}_j|} \qquad [2]$$

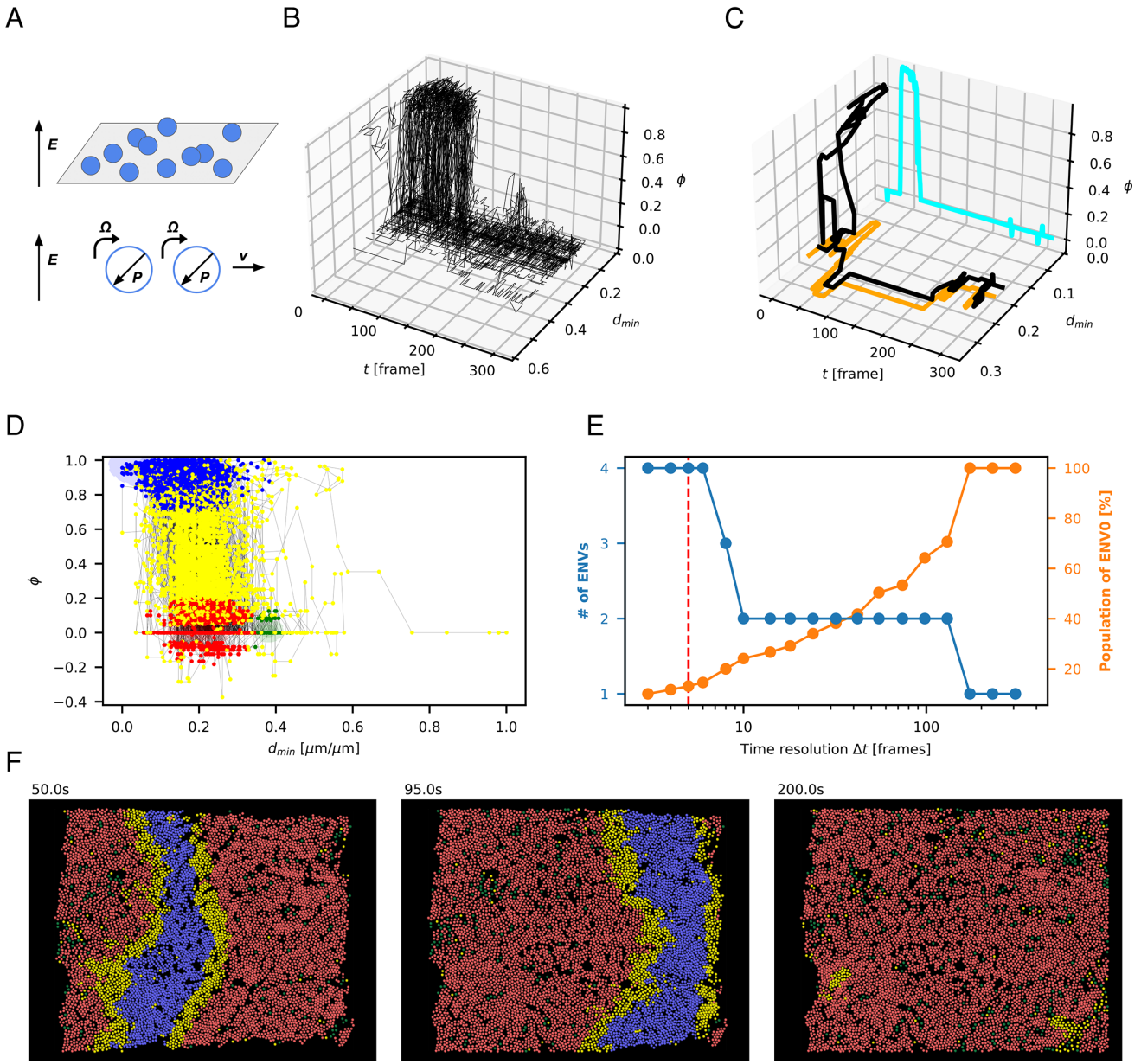

Fig. 5. Analysis of experimental multivariate time-series. (*A*) Cartoon representation of Quincke rollers, dielectric colloidal particles suspended in a conducting fluid and exposed to a vertical direct current (DC) electric field. These particles exhibit collective motion; see Liu et al. (62) for more information. (*B*) The rescaled minimum neighbor distance d_{min} and the local velocity alignment ϕ are plotted as a function of time, for all the particle in the video. (*C*) Example signal for a single particle is shown (in black), together with its two separate components (in orange and cyan). (*D*) The algorithm identifies three clusters, in red, blue, and green, respectively. (*E*) Blue line: number of clusters identified as a function of the time resolution Δt; orange line: percentage of the data points classified in the ENV0 cluster as a function of the time resolution Δt. The red dashed line at $\Delta t = 5$ frames indicates the time resolution for the show results. (*F*) Three snapshots from the video, colored according to the detected clusters.

In Eq **2**, j iterates over the n_c^i particles inside a specified cutoff distance r_c from particle i ($r_c = 15$ pixels). $\vec{v}_i$ and $\vec{v}_i$ are the velocities of particles i and j, respectively. The variable ϕ_i captures the average cosine value of the angle between the velocities of particle i and its neighboring particles: This value ranges between -1 and 1, indicating the level of alignment or orientation similarity between the velocities of the particle and its neighbors.

We thus obtained the bidimensional time-series data shown in Fig. 5*B* (showing the time-series for all particles) and Fig. 5*C* (showing a single particle, in black, and its two components i) and ii), in orange and cyan) over time. $d_{\min}$ was rescaled within the range $[0, 1]$ to facilitate the visualization and give the two variables i) and ii) a comparable weight.

Fig. 5 *D–F* show as an example the results obtained by *Onion Clustering* employing a time resolution of $\Delta t = 5$ frames. The algorithm identifies three distinct statistically relevant environments (blue, red, and green) alongside the ENV0 cluster encompassing all unclassified data points (in yellow). The significance of the clusters becomes apparent when observing the simulation snapshots in Fig. 5*F* and Movie S4. Environment ENV1 (in red) is characterized by $d_{\min} = (0.20 \pm 0.08)$ and $\phi = (0.01 \pm 0.10)$, and primarily consists of stationary particles. ENV2 (blue) is characterized by $d_{\min} = (0.12 \pm 0.15)$ and $\phi = (0.97 \pm 0.14)$, and corresponds to particles moving coherently within the wavefront. ENV3 (green) contains particles with $d_{\min} = (0.38 \pm 0.04)$ and $\phi = (0.01 \pm 0.06)$, stationary particles located in an area with very low density (exhibiting high $d_{\min}$ values). The unclassified data points (ENV0: yellow) correspond to particles situated on the two edges of the wave, whose surrounding environment is changing too rapidly to be classified as persistent clusters at this time-resolution.

Reducing Δt reduces the population of the ENV0 cluster and increases the ability of the algorithm to precisely characterize the edges of the wave (*SI Appendix*, Fig. S4). Fig. 5*E* shows the number of clusters and the population of ENV0 as a function of the Δt. Notably, 4 clusters are discernible when employing $\Delta t \leq 6$ frames, a timescale comparable to the residence time of a single particle inside the wave.

Conclusions

In this paper, we introduced *Onion Clustering*, a new unsupervised clustering algorithm for the microscopic analysis of time-series data. *Onion Clustering* automatically identifies fluctuations and microscopic dynamic environments in a time-series, and classifies the data points into microclusters. Typically, unsupervised clustering methods suffer, e.g., of lack of physical interpretability of the results, multiple parameters that have to be tuned (and that may considerably change the results), and difficulties in identifying clusters/environments that are way less sampled/populated than others, such as rare and/or local dynamical events, transient states, etc. Here, using various types of test examples, we demonstrate how *Onion Clustering* can mitigate such issues, standing out as a general and reliable unsupervised method characterized by noncommon physical interpretability of the clustering results (which are, obviously, still dependent on a good choice of the observables used to study the system), statistical robustness, ease of use, and flexibility in analyzing different types of time-series data.

Onion Clustering is based on an iterative "certainty-based" approach. The most evident and statistically populated environment is classified first, and then it is removed, together with its noise, from the time-series, which is then analyzed again in an iterative fashion. The algorithm can thus rely on an adaptive metric that, at every successive iteration, enhances the signal-to-noise ratio. This allows to unveil all the dynamical subdomains (also the least populated ones) that can be classified in a statistically robust way at a given time-resolution. In this way, the method can extract and retain all information that is statistically significant in a time-series as a function of the resolution at which this is studied. At the same time, *Onion Clustering* quantifies—via the population of the ENV0 cluster (i.e., the data points which was not possible to classify)—the amount of information that cannot be statistically classified and that gets lost at a certain time resolution Δt, which is a nontrivial added value for an unsupervised method. While in such a method, the time-resolution Δt is the sole determinant parameter, instead of choosing the time-resolution a priori, *Onion Clustering* performs the analysis at every possible resolution (the bottom limit being the time-interval between the frames in the time-series itself) and plots the results. This allows the user to make an a posteriori informed choice of the resolution at which it is best to study a time-series to analyze determined phenomena/events. This makes *Onion Clustering* a fully unsupervised, essentially parameter-free clustering method that is transparent, controllable, statistically robust, and that avoids typical problems emerging from the use of such unsupervised algorithms as a black box.

The examples discussed herein demonstrate how *Onion Clustering* can efficiently reconstruct all the statistically relevant events contained in time-series with extremely variegated features: from systems in dynamical equilibrium conditions, to systems out-of-equilibrium, to systems dominated by rare events and local fluctuations (difficult to detect by pattern recognition analyses due to their negligible statistical weight), from synthetic and simulation time-series, to experimental trajectories. Moreover, despite the examples discussed come from MD and mesoscopic colloidal system, the algorithm can be applied to any kind of time-series data. We expect that, thanks to its generality and simplicity, *Onion Clustering* will constitute a useful tool in the study of complex systems from the atomistic to the macroscopic scale.

Materials and Methods

Simulations and Data Analysis. The trajectories for Figs. 1 and 2 are obtained from MD simulations with PBC of 2,048 TIP4P/ICE molecules, starting from a configuration of 50% ice/50% liquid, at $T = 268$ K and $T = 267$ K respectively. From these trajectories, the LENS (39) and *t*SOAP (38) descriptors are computed for each molecule. The trajectories for Fig. 3 are obtained by deep-potential MD simulation, described in detail in ref. 10, of 2,400 Cu atoms at $T = 600$ K. From these trajectories, the LENS (39) descriptor is computed for each atom. Extensive details on the models and simulations' setups are provided in *SI Appendix*, Text. The data for Fig. 4 are obtained simulating two noninteracting particles with Langevin dynamics. Particles' coordinates at each frame are then given as input to the clustering algorithm. The data for Fig. 5 are obtained from experimental microscopy videos from ref. 62. We performed image recognition of the videos, and then particle tracking using Trackpy (63). For each particle, the distance from the closest neighbor d_{min} and the local alignment of the velocities ϕ are computed. Further details for all the datasets are available in *SI Appendix*, Text; the datasets used for the clustering analyses are available in Datasets S1–S5.

The Clustering Algorithm.

Univariate/monodimensional data analysis. Let's call $x_i(t)$, with $1 \leq i < N$ indexing the particle and $0 \leq t \leq T$ indexing the discrete time, the set of signals we want to cluster. The algorithm proceeds as follows:

1. The signals $x_i(t)$ are divided in windows of length Δt, the time resolution of the analysis:

$$X_{i,w} = [x_i(w\Delta t), x_i(w\Delta t + 1), x_i(w\Delta t + 2), ..., x_i(w\Delta t + \Delta t - 1)]$$

with $w \in \{0, 1, 2, ..., \text{int}(T/\Delta t)\}$.

The following procedure is then repeated iteratively, each time identifying a candidate environment E_n, until a termination condition is met:

2. The cumulative histogram H_j of all the data is computed, with $0 \leq j < n_{\text{bins}}$. n_{bins} is set automatically by Numpy (65), but can be also set to a custom value.
3. The absolute maximum of the histogram is identified, and a Gaussian distribution of the form Eq **1** is fitted on the histogram in an interval around the maximum, with μ, σ, and A as free parameters. The details about the choice of the fitting interval are reported in *SI Appendix*. If the fitting procedure does not converge, go to step 7.
4. A candidate environment E_n is identified as the signal interval

$$E_n = [\mu_n - 2\sigma_n, \mu_n + 2\sigma_n]$$

The values of μ_n, σ_n, and A_n are stored for later use.
5. For every pair (i, w), the window $X_{i,w}$ is removed from the signals if and only if it is entirely included in the environment E_n, that is, if and only if

$$\begin{cases} \min\{X_{i,w}\} \geq \mu_n - 2\sigma_n \\ \max\{X_{i,w}\} \leq \mu_n + 2\sigma_n \end{cases} \quad [3]$$

If no window satisfies these requirements, go to step 7.
6. If after this step the signals $x_i(t)$ are still not empty, the procedure is repeated from step 2. Otherwise, go on to step 7.
7. At this point, a list of environment E_n, $0 \leq n < n_{\text{states}}$, has been identified, each one described by its center μ_n, its width $4\sigma_n$ and its weight A_n. Moreover, a fraction f_n of windows $X_{i,w}$ has been assigned to each environment. From this information, strongly overlapping environments are merged together. The details about this procedure are reported in *SI Appendix*.
8. Having sliced the trajectory of each particle in time windows of size Δt, we have assigned each window to one of the clusters. This allows to assign each particle to the cluster it belongs to at every frame of the trajectory.

Multivariate/multidimensional data analysis. The case of D—dimensional signals is handled in basically the same way as in one-dimensional ones. The Gaussian used of the fit around the maxima are factorized, i.e. of the form

$$P(x_1, x_2, ..., x_D) = \prod_{d=1}^{D} \frac{A_d}{\sqrt{\pi}\sigma_d} \exp\left[-\left(\frac{x_d - \mu_d}{\sigma_d}\right)^2\right]$$

and the fit is performed inside a D—dimensional rectangular region, where the limit of the rectangle along each dimension is selected with the same procedure shown in *SI Appendix* for the univariate data. We stress that the choice of this factorized form for the Gaussian distribution does not assume that the signal components are independent; it is only done to improve the fitting performance of the algorithm in our implementation.

Data, Materials, and Software Availability. The algorithm presented in this paper is implemented as a Python3 package (58). The code is available open-source at this GitHub repository (56, 57). All the code and data necessary to reproduce the analysis of this work are available on a Zenodo repository at ref. 66.

ACKNOWLEDGMENTS. G.M.P. acknowledges the funding received by the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement no. 818776–DYNAPOL).

Author affiliations: [a]Department of Applied Science and Technology, Politecnico di Torino, Torino 10129, Italy; and [b]Department of Innovative Technologies, University of Applied Sciences and Arts of Southern Switzerland, Lugano, Viganello 6962, Switzerland

Author contributions: G.M.P. designed research; M.B. and F.F. performed research; M.B. and G.M.P. analyzed data; M.B. software implementation; and M.B. and G.M.P. wrote the paper.

1. S. Sattari *et al*., Modes of information flow in collective cohesion. *Sci. Adv.* **8**, eabj1720 (2022).
2. T. Liu, L. Ungar, K. Kording, Quantifying causality in data science with quasi-experiments. *Nat. Comput. Sci.* **1**, 24–32 (2021).
3. J. Borge-Holthoefer *et al*., The dynamics of information-driven coordination phenomena: A transfer entropy analysis. *Sci. Adv.* **2**, e1501158 (2016).
4. M. Nitzan, J. Casadiego, M. Timme, Revealing physical interaction networks from statistics of collective dynamics. *Sci. Adv.* **3**, e1600396 (2017).
5. U. S. Basak, S. Sattari, M. M. Hossain, K. Horikawa, T. Komatsuzaki, An information-theoretic approach to infer the underlying interaction domain among elements from finite length trajectories in a noisy environment. *J. Chem. Phys.* **154**, 034901 (2021).
6. M. Crippa, C. Perego, A. L. de Marco, G. M. Pavan, Molecular communications in complex systems of dynamic supramolecular polymers. *Nat. Commun.* **13**, 2162 (2022).
7. Y. Hong, S. Kwong, Y. Chang, Q. Ren, Unsupervised data pruning for clustering of noisy data. *Knowl. Based Syst.* **21**, 612–616 (2008).
8. Y. Cho, T. Christoff-Tempesta, S. J. Kaser, J. H. Ortony, Dynamics in supramolecular nanomaterials. *Soft Matter* **17**, 5850–5863 (2021).
9. F. Baletto, Structural properties of sub-nanometer metallic clusters. *J. Phys. Condens. Matter* **31**, 113001 (2019).
10. M. Cioni *et al*., Innate dynamics and identity crisis of a metal surface unveiled by machine learning of atomic environments. *J. Chem. Phys.* **158**, 124701 (2023).
11. P. Gasparotto, D. Bochicchio, M. Ceriotti, G. M. Pavan, Identifying and tracking defects in dynamic supramolecular polymers. *J. Phys. Chem. B* **124**, 589–599 (2020).
12. D. Bochicchio, S. Kwangmettatam, T. Kudernac, G. M. Pavan, How defects control the out-of-equilibrium dissipative evolution of a supramolecular tubule. *ACS Nano* **13**, 4322–4334 (2019).
13. A. L. de Marco, D. Bochicchio, A. Gardin, G. Doni, G. M. Pavan, Controlling exchange pathways in dynamic supramolecular polymers by controlling defects. *ACS Nano* **15**, 14229–14241 (2021).
14. PRt Wolde, D Frenkel,, Enhancement of protein crystal nucleation by critical density fluctuations. *Science* **277**, 1975–1978 (1997).
15. J. F. Lutsko, How crystals form: A theory of nucleation pathways. *Sci. Adv.* **5**, eaav7399 (2019).
16. M. Nagy, Z. Ákos, D. Biro, T. Vicsek, Hierarchical group dynamics in pigeon flocks. *Nature* **464**, 890–893 (2010).
17. A. Cavagna *et al*., Scale-free correlations in starling flocks. *Proc. Natl. Acad. Sci. U.S.A.* **107**, 11865–11870 (2010).
18. A. Attanasi *et al*., Information transfer and behavioural inertia in starling flocks. *Nat. Phys.* **10**, 691–696 (2014).
19. S. Butail, V. Mwaffo, M. Porfiri, Model-free information-theoretic approach to infer leadership in pairs of zebrafish. *Phys. Rev. E* **93**, 042411 (2016).
20. M. Porfiri, Inferring causal relationships in zebrafish-robot interactions through transfer entropy: A small lure to catch a big fish. *Anim. Behav. Cogn.* **5**, 341–367 (2018).
21. R. N. Mantegna, H. E. Stanley, *Introduction to Econophysics: Correlations and Complexity in Finance* (Cambridge University Press, 1999).
22. C. Duan, T. Nishikawa, D. Eroglu, A. E. Motter, Network structural origin of instabilities in large complex systems. *Sci. Adv.* **8**, eabm8310 (2022).
23. A. Jović, K. Brkić, N. Bogunović, "A review of feature selection methods with applications" in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (IEEE, 2015), pp. 1200–1205.
24. P. C. B. Kathirgamanathan, "A feature selection method for multi-dimension time-series data" in *Advanced Analytics and Learning on Temporal Data*, V. Lemaire, Eds. (Springer, 2020), pp. 220–231.
25. J. Schmidt, H. Piringer, T. Mühlbacher, J. Bernard, "Human-based and automatic feature ideation for time series data: A comparative study" in *EuroVis Workshop on Visual Analytics (EuroVA)*, M. Angelini, M. El-Assady, Eds. (The Eurographics Association, 2023), 10.2312/eurova.20231089.
26. E. D. Donkor, A. Laio, A. Hassanali, Do machine-learning atomic descriptors and order parameters tell the same story? The case of liquid water. *J. Chem. Theory Comput.* **19**, 4596–4605 (2023).
27. A. P. Bartók, R. Kondor, G. Csányi, On representing chemical environments. *Phys. Rev. B* **87**, 184115 (2013).
28. B. Monserrat, J. G. Brandenburg, E. A. Engel, B. Cheng, Liquid water contains the building blocks of diverse ice phases. *Nat. Commun.* **11**, 5757 (2020).
29. A. Offei-Danso, A. Hassanali, A. Rodriguez, High-dimensional fluctuations in liquid water: Combining chemical intuition with unsupervised learning. *J. Chem. Theory Comput.* **18**, 3136–3150 (2022).
30. N. Ansari, B. Onat, G. C. Sosso, A. Hassanali, Insights into the emerging networks of voids in simulated supercooled water. *J. Phys. Chem. B* **124**, 2180–2190 (2020).
31. R. Capelli, F. Muniz-Miranda, G. M. Pavan, Ephemeral ice-like local environments in classical rigid models of liquid water. *J. Chem. Phys.* **156**, 214503 (2022).
32. M. Crippa, A. Cardellini, M. Cioni, G. Csányi, G. M. Pavan, Machine learning of microscopic structure-dynamics relationships in complex molecular systems. *Mach. Learn. Sci. Technol.* **4**, 045044 (2023).
33. C. Lionello, C. Perego, A. Gardin, R. Klajn, G. M. Pavan, Supramolecular semiconductivity through emerging ionic gates in ion-nanoparticle superlattices. *ACS Nano* **17**, 275–287 (2022).
34. R. Capelli, A. Gardin, C. Empereur-Mot, G. Doni, G. M. Pavan, A data-driven dimensionality reduction approach to compare and classify lipid force fields. *J. Phys. Chem. B* **125**, 7785–7796 (2021).
35. A. Gardin, C. Perego, G. Doni, G. M. Pavan, Classifying soft self-assembled materials via unsupervised machine learning of defects. *Commun. Chem.* **5**, 82 (2022).

36. A. Cardellini *et al*., Unsupervised data-driven reconstruction of molecular motifs in simple to complex dynamic micelles. *J. Phys. Chem. B* **127**, 2595–2608 (2023).
37. T. A. Sharp *et al*., Machine learning determination of atomic dynamics at grain boundaries. *Proc. Natl. Acad. Sci. U.S.A.* **115**, 10943–10947 (2018).
38. C. Caruso, A. Cardellini, M. Crippa, D. Rapetti, G. M. Pavan, Timesoap: Tracking high-dimensional fluctuations in complex molecular systems via time variations of soap spectra. *J. Chem. Phys.* **158**, 214302 (2023).
39. M. Crippa, A. Cardellini, C. Caruso, G. M. Pavan, Detecting dynamic domains and local fluctuations in complex molecular systems via timelapse neighbors shuffling. *Proc. Natl. Acad. Sci. U.S.A.* **120**, e2300565120 (2023).
40. P. Rai, S. Singh, A survey of clustering techniques. *Int. J. Comput. Appl.* **7**, 1–5 (2010).
41. A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks. *Science* **344**, 1492–1496 (2014).
42. D. U. Pizzagalli, S. F. Gonzalez, R. Krause, A trainable clustering algorithm based on shortest paths from density peaks. *Sci. Adv.* **5**, eaax3770 (2019).
43. I. Barrio-Hernandez *et al*., Clustering predicted structures at the scale of the known protein universe. *Nature* **622**, 637–645 (2023).
44. E. Keogh, S. Lonardi, Bc. Chiu, "Finding surprising patterns in a time series database in linear time and space" in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York, NY, 2002), pp. 550–556.
45. M. Gupta, J. Gao, C. C. Aggarwal, J. Han, Outlier detection for temporal data: A survey. *IEEE Trans. Knowl. Data Eng.* **26**, 2250–2267 (2013).
46. D. Fernex, B. R. Noack, R. Semaan, Cluster-based network modeling-from snapshots to complex dynamical systems. *Sci. Adv.* **7**, eabf5006 (2021).
47. S. Aminikhanghahi, D. J. Cook, A survey of methods for time series change point detection. *Knowl. Inf. Syst.* **51**, 339–367 (2017).
48. L. Albertazzi *et al*., Probing exchange pathways in one-dimensional aggregates with super-resolution microscopy. *Science* **344**, 491–495 (2014).
49. X. Wang, K. Smith, R. Hyndman, Characteristic-based clustering for time series data. *Data Min. Knowl. Discov.* **13**, 335–364 (2006).
50. M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recogn. Lett.* **42**, 11–24 (2014).
51. N. S. Madiraju, "Deep temporal clustering: Fully unsupervised learning of time-domain features," PhD thesis, Arizona State University, Tempe, AZ (2018).
52. A. Javed, D. M. Rizzo, B. S. Lee, R. Gramling, Somtimes: Self organizing maps for time series clustering and its application to serious illness conversations. *Data Min. Knowl. Discov.* **38**, 1–27 (2023).
53. S. Aghabozorgi, A. S. Shirkhorshidi, T. Y. Wah, Time-series clustering-a decade review. *Inf. Syst.* **53**, 16–38 (2015).
54. E. Keogh, J. Lin, Clustering of time-series subsequences is meaningless: Implications for previous and future research. *Knowl. Inf. Syst.* **8**, 154–177 (2005).
55. A. T. Bogetti, J. M. Leung, L. T. Chong, LPATH: A semiautomated python tool for clustering molecular pathways. *J. Chem. Inf. Model.* **63**, 7610–7616 (2023).
56. M. Becchi, timeseries_analysis. Github. https://github.com/matteobecchi/timeseries_analysis. Accessed 25 February 2024.
57. M. Becchi, Gmpavanlab. Github. https://github.com/GMPavanLab/timeseries_analysis. Accessed 1 March 2024.
58. M. Becchi, onion-clustering. PyPI. https://pypi.org/project/onion-clustering/. Accessed 15 February 2024.
59. J. Abascal, E. Sanz, R. García Fernández, C. Vega, A potential model for the study of ices and amorphous water: TIP4P/ICE. *J. Chem. Phys.* **122**, 234511 (2005).
60. C. Zeni, A. Anelli, A. Glielmo, K. Rossi, Exploring the robust extrapolation of high-dimensional machine learning potentials. *Phys. Rev. B* **105**, 165141 (2022).
61. O. A. Karim, A. Haymet, The ice/water interface: A molecular dynamics simulation study. *J. Chem. Phys.* **89**, 6889–6896 (1988).
62. Z. T. Liu *et al*., Activity waves and freestanding vortices in populations of subcritical quincke rollers. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2104724118 (2021).
63. D. Allan, T. Caswell, N. Keim, C. van der Wel, trackpy: Trackpy v0.3.2. https://doi.org/10.5281/zenodo.60550. Accessed 15 December 2023.
64. J. C. Crocker, D. G. Grier, Methods of digital video microscopy for colloidal studies. *J. Colloid Interface Sci.* **179**, 298–310 (1996).
65. C. R. Harris *et al*., Array programming with NumPy. *Nature* **585**, 357–362 (2020).
66. M. Becchi, F. Fantolino, G. M. Pavan, Data for the preprint of "Layer-by-layer unsupervised clustering of statistically relevant fluctuations in noisy time-series data of complex dynamical systems". Zenodo. https://zenodo.org/doi/10.5281/zenodo.10638735. Accessed 9 February 2024.