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

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

Matteo Becchi[1], Federico Fantolino[1], and Giovanni M. Pavan*[1,2]

[1]Department of Applied Science and Technology, Politecnico di Torino, Torino 10129, Italy
[2]Department of Innovative Technologies, University of Applied Sciences and Arts of Southern Switzerland, Lugano-Viganello 6962, Switzerland

March 12, 2024

Abstract

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 unsupervised way the microscopic dynamical events occurring in the system. However, decoupling statistically relevant fluctuations from the internal noise remains most often non-trivial. 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 similar way as peeling the external (evident) layer of an onion reveals the internal hidden ones, the method performs a first detection and 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 sub-domains 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 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.

*To whom correspondence should be addressed. E-mail: giovanni.pavan@polito.it

Introduction

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, 2, 3, 4, 5, 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 non-trivial [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, 12, 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, 17, 18], fish banks [19, 20], as well as in the dynamics of economic and stock market systems [21, 22, 2]. 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 non-trivial to analyze. In particular, extracting meaningful and interpretable information from such noisy time-series is generally hard. To address this issue, common strategies involve the use of either knowledge-based or data driven descriptors. Such descriptors serve as a crucial intermediary step, effectively reducing the amount of data to an interpretable form and facilitating the extraction of useful information 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 as, *e.g.*, the Smooth Overlap of Atomic Positions (SOAP) descriptor [23], have been recently used to obtain a data-driven structural characterization of, *e.g.*, water and ice systems [24, 25, 26, 27, 28], metallic [10, 29], ionic [30], and soft (biological or artificial) molecular systems [11, 31, 32, 33] 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 [34, 8, 10, 35, 36]. 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 (tSOAP) descriptor, which measures the rate of change of the SOAP power spectrum of each unit in a multi-unit trajectory of a dynamical molecular system [35]. A time-series analysis of tSOAP was recently shown to retain rich information 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 [36]. 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 [37, 38, 39, 40], and in particular the identification and classification of fluctuations that are relevant against the background noise [41, 42]. 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 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 [43].

Furthermore, this is particularly relevant in 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, 44]. 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 [45, 46, 47, 48], and are thus not well-suited to obtain a single-point (microscopic-level) clustering of the local dynamical events occurring in the time-series [49, 50, 51].

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 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 outputs the number of clusters that can be effectively classified in a statistical 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 non-common 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 analysing 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 [52, 53], and is released as a Python3 package [54]. 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 2048 TIP4P/ICE [55] molecules (1 : 1 liquid water:ice) in dynamic equilibrium in correspondence of the melting temperature. A complete description of the algorithm is provided in the Methods section and in the Supplementary Information (SI).

Fig 1A shows, as an example, the LENS signals [36] for all 2048 individual water molecules in the system sampled every $\Delta t = 0.1$ ns. The input dataset in this example thus consists of $N = 2048$ univariate time-series $x_i(t)$, $1 \leq i \leq N$ labelling 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

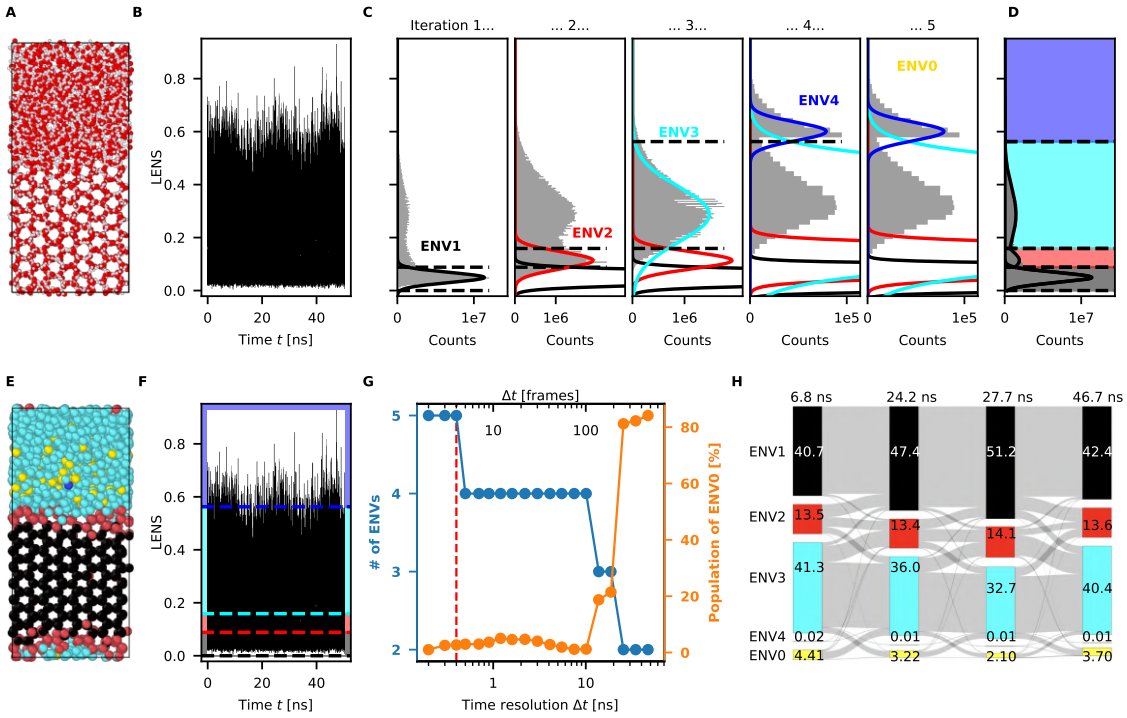

Figure 1: **Clustering of LENS signals on ice/water coexistence MD simulation**. A: Snapshot of the simulation of ice/water coexistence; the simulation is performed on 2048 TIP4P/ICE molecules, lasts 50 ns and it's sampled every 0.1 ns. B: LENS signals for all the oxygen atoms, as a function of time. 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: 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. H: 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.

conducted also using different descriptors – see, *e.g.*, Fig S1 for the consistent results obtained using the *t*SOAP descriptor [35]: 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 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 1B 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 [24, 56, 28, 35], where both states are well represented statistically (see also leftmost panel of Fig 1C: two peaks in the density of the signals at LENS values of ~ 0.05 and ~ 0.3) [36]. 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 allow detecting and retaining information also of rare/local transition events that appear as outliers in the time-series [36, 35]. 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 non-trivial 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 micro-clusters 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 an "onion-like" 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 1C). 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

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

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

This identifies a first dynamical environment, labelled 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 (non-overlapping) 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 Fig S2 in the SI).

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 1C, 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 1C).

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 1C), 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 is repeated iteratively. As it is shown in Fig 1C, 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, labelled 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 1D-F: dashed lines). This identifies the main ENV1-4 clusters colored in Fig 1D-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 Fig S3 in the SI. As can be seen from the simulation snapshot in Fig 1E and in Supplementary 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 [28], may occasionally form ephemeral ice-like clusters that in such conditions continuously freeze and re-melt 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 $\sim 3.5\%$ 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 1G. 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 Fig S2 in the SI). 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 a 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 ($0.5 - 10$ ns), which is compatible with previous studies on the diffusion coefficient of water molecules at the ice/water interface [57].

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 $\sim 80\%$ 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 [24, 56, 28, 35].

The plot of Fig 1G 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 1G 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 non-common 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 clusters 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 1H, the height of the colored bars is proportional to the populations of the 5 detected clusters at 4 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 1G 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 fastly form/dissolve in the liquid domain in these conditions [28] (events that occurs 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

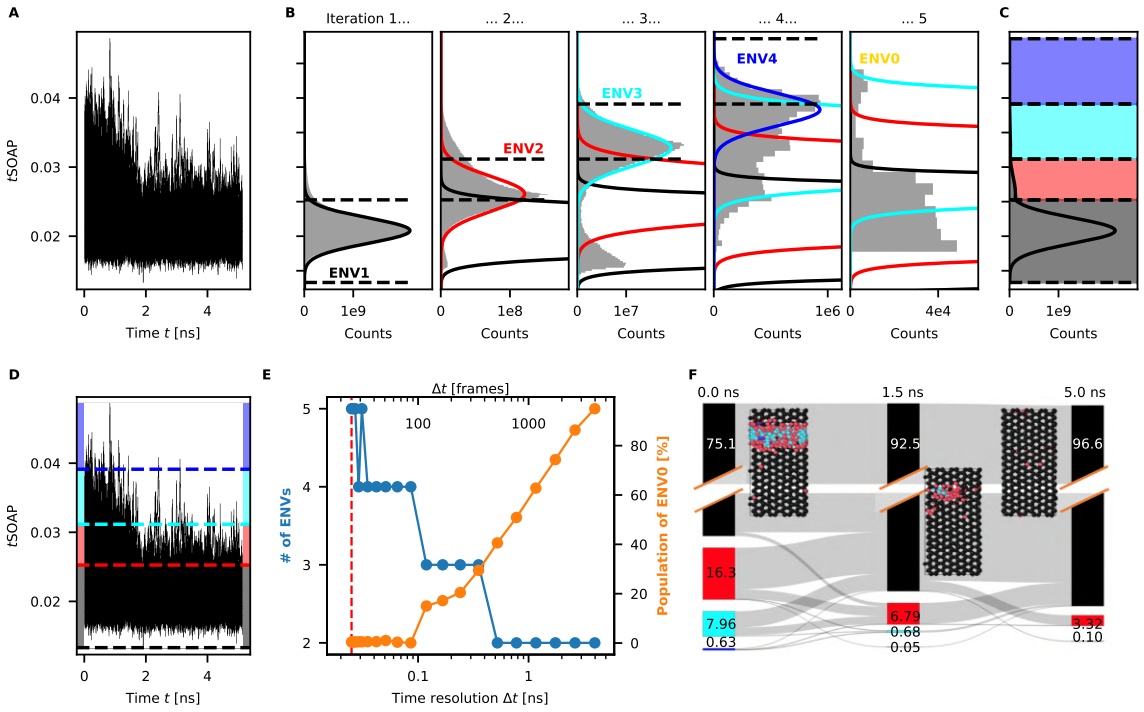

Figure 2: **Analysis of out-of-equilibrium time-series: freezing water.** A: *t*SOAP values for 2048 TIP4P/ICE molecules, along the 5 ns long MD simulation sampled every ps, smoothed with a moving average with width 25 frames (25 ps). B: Data cumulative histograms at the five iterations of the algorithm. 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.

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 Methods section 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 $\sim$ 5000 frames), which is a sufficient time to observe the entire system freezing. As an example, we computed the tSOAP descriptor [35] for all water molecules in these out-of-equilibrium trajectories (for comparison with the same system at the equilibrium, see Fig. S1). In brief, tSOAP is a scalar descriptor that measures the rate of variation of the SOAP spectrum [23] 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 2A. In this plot, it can be seen that the highest values of tSOAP (> 0.03), identifying molecules in the liquid phase (faster structural rearrangement of their neighbors), tend to disappear after $\sim$ 2 ns, leaving only the lower tSOAP values corresponding to molecules in the solid ice phase. As can be seen in the leftmost cumulative histogram of Fig 2B, already after $\sim$ 5 ns the statistical weight of the data points with tSOAP > 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 2B 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 2C-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 micro-environments with different tSOAP 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 2F and in the Supplementary 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 2E 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, while it begins to rise increasing the Δt, since the time-resolution is insufficient to track the fast evolution of the the system. Notably, this Δt value is considerably lower than that observed in the previous section (see Fig 1 for the LENS analysis and Fig S1 for the tSOAP analyses of the equilibrium system). Such a discrepancy is essentially due to a different relevance/noise ratio between the tSOAP 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 are non trivial 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 2F 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

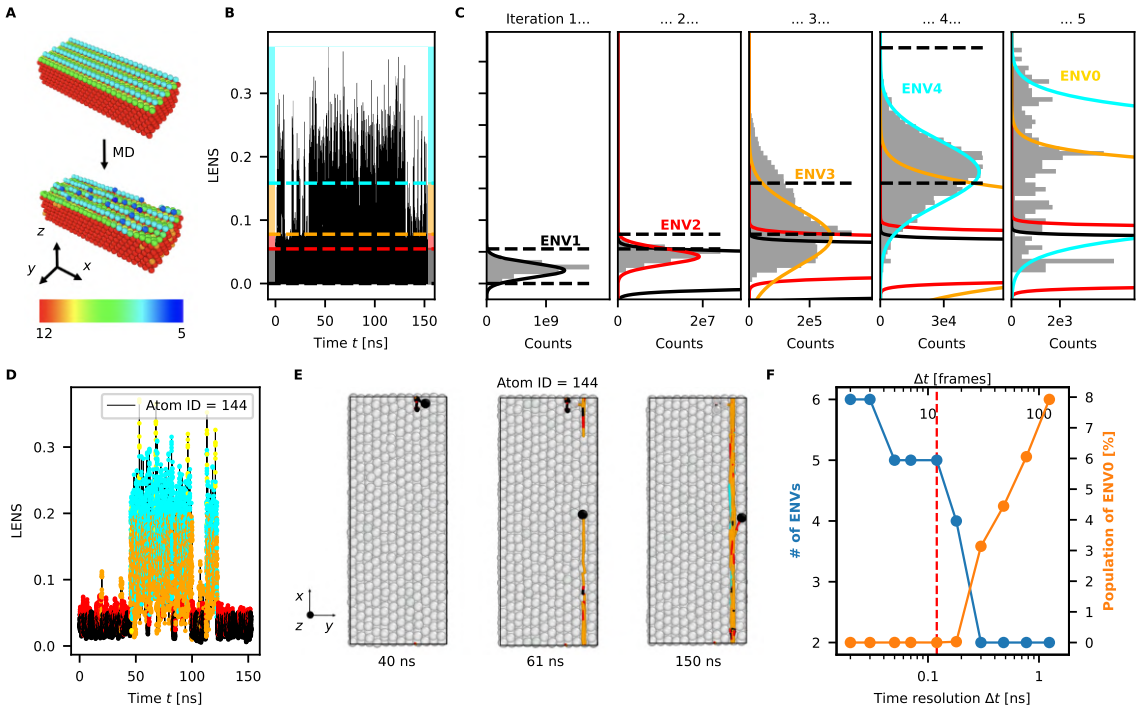

Figure 3: **Analysis of time-series dominated by rare events.** A: Snapshots of the MD simulation of Cu surface composed by 2400 atoms at $T = 600$ K; atoms are colored according to their coordination number. The upper snapshot is at $T = 0\ K$, the lower one during the simulation at $T = 600\ K$. B: 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.

Another scenario where clustering algorithms often struggle is in detecting amidst 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 2400 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 Methods section for details). The MD trajectory lasts 150 ns and is sampled every 10 ps (for a total of 15000 frames). As shown in Fig 3A, 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, 36, 29]. 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 3B (LENS values $\gtrsim 0.1$ identify atomic sliding events).

We performed an *Onion Clustering* on these LENS time-series (Fig 3B-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 3B shows the thresholds between the LENS environments/clusters. ENV1 and ENV2 together encompass $\sim 99.95\%$ of the data points, which correspond to static bulk and surface atoms in the system. Remarkably, despite this issue, the algorithm is able to correctly assign the remaining data points to the other microscopic dynamical environments (ENV3-4), which identify the rapid sliding motion of some atoms on the Cu surface. Fig 3D shows a detail of the LENS time-series for one atom (ID: 144) that slides on the surface along the simulation. Fig 3E 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, cyan). From $t \sim 125$ ns, the atom is reincorporated into the surface lattice, returning to ENV1-2. Supplementary Movie S3 shows the complete MD trajectory colored based on the detected clusters.

Fig 3F shows how 6/5 LENS clusters can be clearly classified with a negligible information loss up to $\Delta t \sim 0.1 - 0.2$ 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 non-static (ENV0) atoms.

Analysis of multivariate time-series

While the examples above show the efficiency of *Onion Clustering* in analysing univariate (unidimensional) time-series, in many case it is desirable to conduct multidimensional analyses to minimize information loss. We thus extended the method to made 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 Methods section for details).

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

The plot of Fig 4D 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.

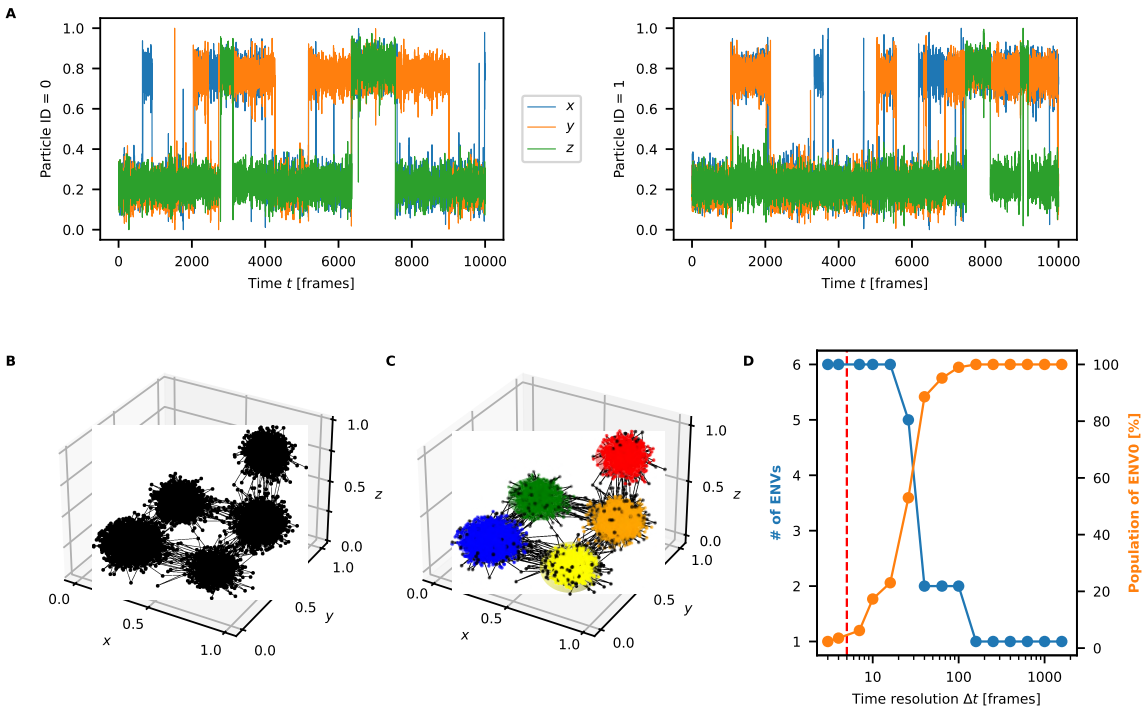

Figure 4: **Analysis of synthetic multivariate time-series.** A: Time-correlated data-points where generated simulating two particles (shown on the left and right panel) with Langevin Dynamics in a three-dimensional free-energy landscape with 5 minima. B: the same trajectories, represented as a three-dimensional 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.

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

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* [58]. Briefly, Quincke rollers are μm scale dielectric colloidal particles suspended in a conducting fluid and exposed to a vertical DC electric field (see Fig 5A). 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 $N = 6921$ particles in a field of view is 700×700 μm^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 [58]. From this movie we extracted the particles' positions at each time-frame along the trajectory using the python package Trackpy [59, 60].

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:

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

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 bi-dimensional time-series data shown in Fig 5B (showing the time-series for all particles) and Fig 5C (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 to variables (i) and (ii) a comparable weight.

Fig 5D-F show as an example the results obtained by *Onion Clustering* employing a time resolution of $\Delta t = 5$ frames. The algorithm identifies 3 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 5F and Supplementary 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 (see Fig S4). Fig 5E 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 micro-clusters. 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

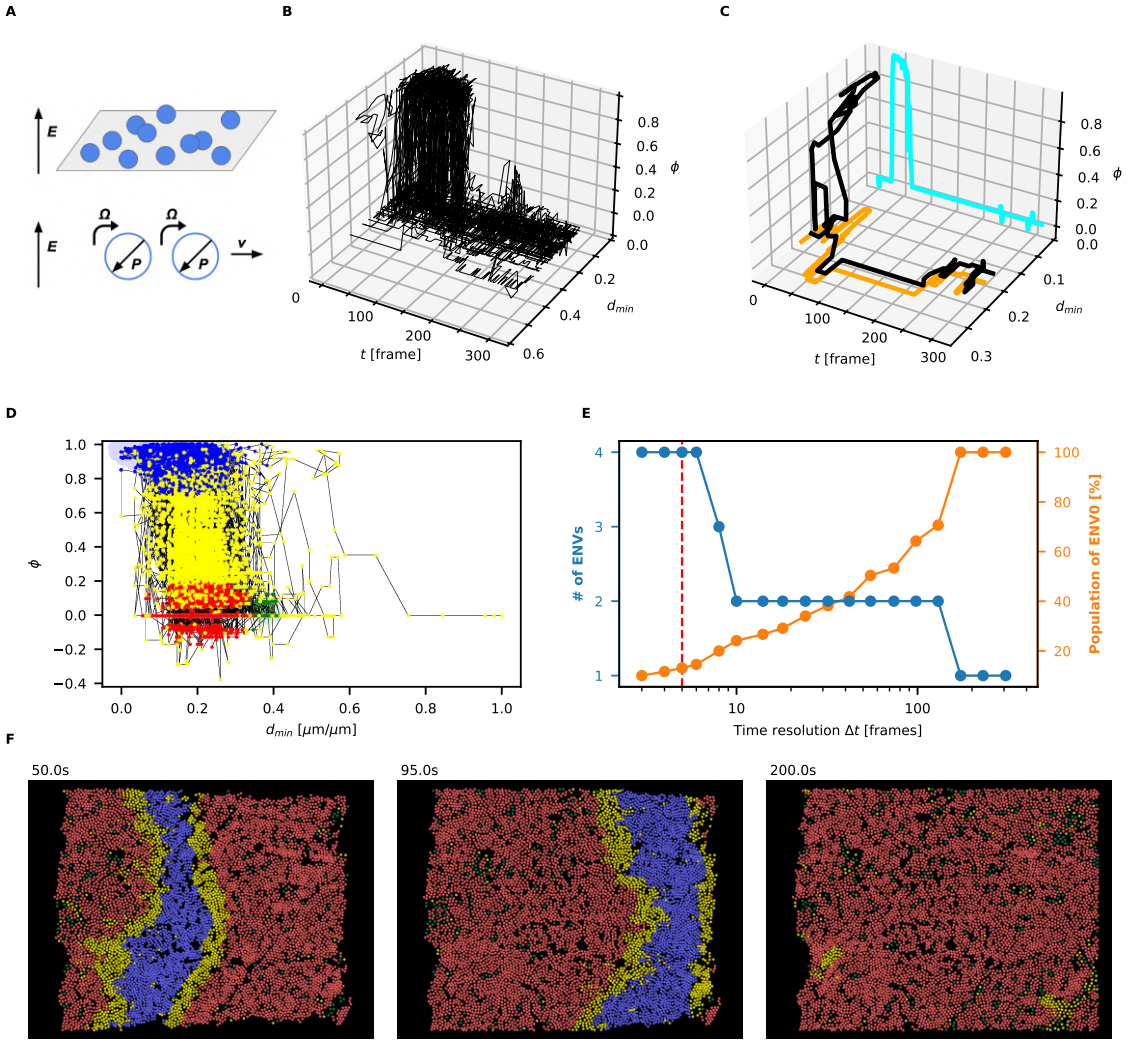

Figure 5: **Analysis of experimental multivariate time-series**. A. Cartoon representation of Quincke rollers, dielectric colloidal particles suspended in a conducting fluid and exposed to a vertical DC electric field. These particles exhibit collective motion, see Liu *et al* [58] for more information. 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 the detected clusters.

examples, we demonstrate how *Onion Clustering* can mitigate such issues, standing out as a general and reliable unsupervised method characterized by non-common physical interpretability, 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 iterative fashion. The algorithm can thus rely on an adaptive metric that, at every successive iteration, enhances the signal-to-noise ratio. In an "onion peeling" fashion, 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 information that are 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 non-trivial 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, substantially 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. 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.

Methods

Simulations and data analysis

Water-ice in dynamic coexistence

The data for Fig 1 (available in the SI: Dataset S1) are obtained by the MD simulations described in details in [36, 35, 29], of 2048 TIP4P/ICE molecules in correspondence of the melting temperature ($T = 268$ K for this force field [61]). In the initial configuration, half of the molecules are in the solid phase in hexagonal ice packing, the other half are in the liquid phase. Being at the melting temperature, solid and liquid phase are in dynamical equilibrium. The simulation lasts 50 ns with a configuration sampling interval of 0.1 ns. For every molecule in the system, the LENS signals are computed on the sampled configurations using a cutoff of 10 Å (close to the third minimum of the radial distribution function), for the 2048 oxygen atoms [36, 29].

Water freezing

The data for Fig 2 (available in the SI: Dataset S2) are obtained by continuing the simulation at coexistence using the same setup, but lowering the temperature to $T = 267$ K (just below the TIP4P/ICE melting point [61]). In this case, the system is evolved for 40 ns, sampling its configuration every ps, while only the last 5 ns of the simulation were used in the analysis: right before the freezing started, in such a way to have a prevalence of signal related to the ice domain and in order to test the algorithm in an unfavorable case, and to prove that this can keep track that some liquid water has been present in the trajectory even in the case where this has a low statistical weight in the time-series. The *t*SOAP signals [35] are computed, with a cutoff of 10 Å (close to the third minimum of the radial distribution function), for the 2048 oxygen atoms. The *t*SOAP signals are then smoothed with a moving average with 25 frames width (to reduce the noise).

FCC(211) copper surface

The data for Fig 3 (available in the SI: Dataset S3) are obtained by the MD simulation described in details in [10], of 2400 Cu atoms at $T = 600$ K . Periodic boundary conditions are applied in the xy plane, while the system is finite along the z direction, simulating *de facto* an infinte copper/vacuum surface along the (211)-plane. The system is evolved for 150 ns, sampling its configuration every 10 ps, for a total of 15000 frames. The LENS signals [36] are then computed, with a cutoff of 6 Å, for all the atoms. The signals are then smoothed with a moving average with 10 frames width (to reduce the noise).

Multivariate/multidimensional synthetic data

The data for Fig 4 (available in the SI: Dataset S4) are obtained simulating 2 non-interacting particles with Langevin dynamics, in a bounded potential energy surface with five minima, with coordinates $(0, 0, 0)$, $(0, 1, 0)$, $(1, 0, 0)$, $(1, 1, 0)$ and $(1, 1, 0)$. The system is evolved for $2 \cdot 10^6$ integration steps, sampling its configuration every 200 steps. Particles' coordinates are then given as input to the clustering algorithm.

Experimental trajectories of Quincke rollers

The data for Fig 5 (available in the SI: Dataset S5) are obtained *via* image recognition and a tracking code (trackpy [59]) from experimental microscopy videos from [58]. From the video, the $x-$ and $y-$coordinates of 6921 particles for 310 consecutive frames are extracted. For each particle at each frame, the distance from the closest neighbor d_{min} and the local alignment of the velocities ϕ (as defined in the main text, with a cutoff distance of $r_c = 15$ pixels) are computed. The two quantities are then separately smoothed with a moving average with 2 frames width (to reduce the noise).

The clustering algorithm

Univariate/monodimensional data analysis

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

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 $0 \leq j < n_{bins}$. n_{bins} is set automatically [62], 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. 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's entirely included in the environment E_n, that is, if and only if

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

   If no window satisfies this requirements, go to step 7.

6. If after this step the signals $x_i(t)$ are still not empty, the procedure is repeated from the 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. After this, each data point $x_i(t)$ is assigned to the environment i in which its window is contained.

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 are selected with the same procedure shown in SI for the univariate data.

Acknowledgements

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

Data availability

The algorithm presented in this paper is implemented as a Python3 package [54]. The code is available open-source at this GitHub repository [52, 53]. All the relevant code, Supporting Datasets and Movies for this work are available on Zenodo at [63].

Competing interests statement

The authors declare no competing interests.

References

[1] Sulimon Sattari et al. "Modes of information flow in collective cohesion". In: *Science advances* 8. 6 (2022), eabj1720.

[2] Tony Liu, Lyle Ungar, and Konrad Kording. "Quantifying causality in data science with quasi-experiments". In: *Nature computational science* 1. 1 (2021), pp. 24–32.

[3] Javier Borge-Holthoefer et al. "The dynamics of information-driven coordination phenomena: A transfer entropy analysis". In: *Science advances* 2. 4 (2016), e1501158.

[4] Mor Nitzan, Jose Casadiego, and Marc Timme. "Revealing physical interaction networks from statistics of collective dynamics". In: *Science advances* 3. 2 (2017), e1600396.

[5] Udoy S Basak et al. "An information-theoretic approach to infer the underlying interaction domain among elements from finite length trajectories in a noisy environment". In: *The Journal of Chemical Physics* 154. 3 (2021), p. 034901.

[6] Martina Crippa et al. "Molecular communications in complex systems of dynamic supramolecular polymers". In: *Nature Communications* 13 (2022), p. 2162.

[7] Yi Hong et al. "Unsupervised data pruning for clustering of noisy data". In: *Knowledge-Based Systems* 21. 7 (2008), pp. 612–616.

[8] Yukio Cho et al. "Dynamics in supramolecular nanomaterials". In: *Soft Matter* 17. 24 (2021), pp. 5850–5863.

[9] Francesca Baletto. "Structural properties of sub-nanometer metallic clusters". In: *Journal of Physics: Condensed Matter* 31. 11 (2019), p. 113001.

[10] Matteo Cioni et al. "Innate dynamics and identity crisis of a metal surface unveiled by machine learning of atomic environments". In: *The Journal of Chemical Physics* 158 (2023), p. 124701.

[11] Piero Gasparotto et al. "Identifying and tracking defects in dynamic supramolecular polymers". In: *The Journal of Physical Chemistry B* 124. 3 (2020), pp. 589–599.

[12] Davide Bochicchio et al. "How defects control the out-of-equilibrium dissipative evolution of a supramolecular tubule". In: *ACS Nano* 13. 4 (2019), pp. 4322–4334.

[13] Anna L de Marco et al. "Controlling exchange pathways in dynamic supramolecular polymers by controlling defects". In: *ACS Nano* 15. 9 (2021), pp. 14229–14241.

[14] Pieter Rein ten Wolde and Daan Frenkel. "Enhancement of protein crystal nucleation by critical density fluctuations". In: *Science* 277. 5334 (1997), pp. 1975–1978.

[15] James F Lutsko. "How crystals form: A theory of nucleation pathways". In: *Science advances* 5. 4 (2019), eaav7399.

[16] Máté Nagy et al. "Hierarchical group dynamics in pigeon flocks". In: *Nature* 464. 7290 (2010), pp. 890–893.

[17] Andrea Cavagna et al. "Scale-free correlations in starling flocks". In: *Proceedings of the National Academy of Sciences* 107. 26 (2010), pp. 11865–11870.

[18] Alessandro Attanasi et al. "Information transfer and behavioural inertia in starling flocks". In: *Nature physics* 10. 9 (2014), pp. 691–696.

[19] Sachit Butail, Violet Mwaffo, and Maurizio Porfiri. "Model-free information-theoretic approach to infer leadership in pairs of zebrafish". In: *Physical Review E* 93. 4 (2016), p. 042411.

[20] Maurizio Porfiri. "Inferring causal relationships in zebrafish-robot interactions through transfer entropy: a small lure to catch a big fish". In: *Animal Behavior and Cognition* 5. 4 (2018), pp. 341–367.

[21] Rosario N Mantegna and H Eugene Stanley. *Introduction to econophysics: correlations and complexity in finance*. Cambridge university press, 1999.

[22] Chao Duan et al. "Network structural origin of instabilities in large complex systems". In: *Science advances* 8. 28 (2022), eabm8310.

[23] Albert P Bartók, Risi Kondor, and Gábor Csányi. "On representing chemical environments". In: *Physical Review B* 87. 18 (2013), p. 184115.

[24] Bartomeu Monserrat et al. "Liquid water contains the building blocks of diverse ice phases". In: *Nature communications* 11. 1 (2020), p. 5757.

[25] Edward Danquah Donkor, Alessandro Laio, and Ali Hassanali. "Do machine-learning atomic descriptors and order parameters tell the same story? The case of liquid water". In: *Journal of Chemical Theory and Computation* 19. 14 (2023), pp. 4596–4605.

[26] Adu Offei-Danso, Ali Hassanali, and Alex Rodriguez. "High-dimensional fluctuations in liquid water: Combining chemical intuition with unsupervised learning". In: *Journal of Chemical Theory and Computation* 18. 5 (2022), pp. 3136–3150.

[27] Narjes Ansari et al. "Insights into the emerging networks of voids in simulated supercooled water". In: *The Journal of Physical Chemistry B* 124. 11 (2020), pp. 2180–2190.

[28] Riccardo Capelli, Francesco Muniz-Miranda, and Giovanni M Pavan. "Ephemeral ice-like local environments in classical rigid models of liquid water". In: *The Journal of Chemical Physics* 156 (2022), p. 214503.

[29] Martina Crippa et al. "Machine learning of microscopic structure-dynamics relationships in complex molecular systems". In: *Machine Learning: Science and Technology* 4. 4 (2023), p. 045044.

[30] Chiara Lionello et al. "Supramolecular semiconductivity through emerging ionic gates in ion–nanoparticle superlattices". In: *ACS Nano* 17. 1 (2022), pp. 275–287.

[31] Riccardo Capelli et al. "A data-driven dimensionality reduction approach to compare and classify lipid force fields". In: *The Journal of Physical Chemistry B* 125. 28 (2021), pp. 7785–7796.

[32] Andrea Gardin et al. "Classifying soft self-assembled materials via unsupervised machine learning of defects". In: *Communications Chemistry* 5 (2022), p. 82.

[33] Annalisa Cardellini et al. "Unsupervised data-driven reconstruction of molecular motifs in simple to complex dynamic micelles". In: *The Journal of Physical Chemistry B* 127. 11 (2023), pp. 2595–2608.

[34] Tristan A Sharp et al. "Machine learning determination of atomic dynamics at grain boundaries". In: *Proceedings of the National Academy of Sciences* 115. 43 (2018), pp. 10943–10947.

[35] Cristina Caruso et al. "TimeSOAP: Tracking high-dimensional fluctuations in complex molecular systems via time variations of SOAP spectra". In: *The Journal of Chemical Physics* 158 (2023), p. 214302.

[36] Martina Crippa et al. "Detecting dynamic domains and local fluctuations in complex molecular systems via timelapse neighbors shuffling". In: *Proceedings of the National Academy of Sciences* 120. 30 (2023), e2300565120.

[37] Pradeep Rai and Shubha Singh. "A survey of clustering techniques". In: *International Journal of Computer Applications* 7. 12 (2010), pp. 1–5.

[38] Alex Rodriguez and Alessandro Laio. "Clustering by fast search and find of density peaks". In: *Science* 344. 6191 (2014), pp. 1492–1496.

[39] Diego Ulisse Pizzagalli, Santiago Fernandez Gonzalez, and Rolf Krause. "A trainable clustering algorithm based on shortest paths from density peaks". In: *Science advances* 5. 10 (2019), eaax3770.

[40] Inigo Barrio-Hernandez et al. "Clustering predicted structures at the scale of the known protein universe". In: *Nature* 622. 7983 (2023), pp. 637–645.

[41] Eamonn Keogh, Stefano Lonardi, and Bill'Yuan-chi' 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*. 2002, pp. 550–556.

[42] Manish Gupta et al. "Outlier detection for temporal data: A survey". In: *IEEE Transactions on Knowledge and data Engineering* 26. 9 (2013), pp. 2250–2267.

[43] Daniel Fernex, Bernd R Noack, and Richard Semaan. "Cluster-based network modeling—From snapshots to complex dynamical systems". In: *Science Advances* 7. 25 (2021), eabf5006.

[44] Lorenzo Albertazzi et al. "Probing exchange pathways in one-dimensional aggregates with super-resolution microscopy". In: *Science* 344. 6183 (2014), pp. 491–495.

[45] Xiaozhe Wang, Kate Smith, and Rob Hyndman. "Characteristic-based clustering for time series data". In: *Data mining and knowledge Discovery* 13 (2006), pp. 335–364.

[46] Martin Längkvist, Lars Karlsson, and Amy Loutfi. "A review of unsupervised feature learning and deep learning for time-series modeling". In: *Pattern recognition letters* 42 (2014), pp. 11–24.

[47] Naveen Sai Madiraju. "Deep temporal clustering: Fully unsupervised learning of time-domain features". PhD thesis. Arizona State University, 2018.

[48] Ali Javed et al. "Somtimes: self organizing maps for time series clustering and its application to serious illness conversations". In: *Data Mining and Knowledge Discovery* (2023), pp. 1–27. DOI: 10.1007/s10618-023-00979-9.

[49] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering–a decade review". In: *Information systems* 53 (2015), pp. 16–38.

[50] Eamonn Keogh and Jessica Lin. "Clustering of time-series subsequences is meaningless: implications for previous and future research". In: *Knowledge and information systems* 8 (2005), pp. 154–177.

[51] Anthony T Bogetti, Jeremy MG Leung, and Lillian T Chong. "LPATH: A Semiautomated Python Tool for Clustering Molecular Pathways". In: *Journal of Chemical Information and Modeling* 63. 24 (2023), pp. 7610–7616.

[52] *timeseries_analysis*. `https://github.com/matteobecchi/timeseries_analysis`. 2023.

[53] *GMPavanLab GitHub*. `https://github.com/GMPavanLab/timeseries_analysis`. 2023.

[54] *onion-clustering*. `https://pypi.org/project/onion-clustering/`. 2024.

[55] JLF Abascal et al. "A potential model for the study of ices and amorphous water: TIP4P/Ice". In: *The Journal of chemical physics* 122 (2005), p. 234511.

[56] Claudio Zeni et al. "Exploring the robust extrapolation of high-dimensional machine learning potentials". In: *Physical Review B* 105. 16 (2022), p. 165141.

[57] Omar A Karim and ADJ Haymet. "The ice/water interface: A molecular dynamics simulation study". In: *The Journal of chemical physics* 89. 11 (1988), pp. 6889–6896.

[58] Zeng Tao Liu et al. "Activity waves and freestanding vortices in populations of subcritical Quincke rollers". In: *Proceedings of the National Academy of Sciences* 118. 40 (2021), e2104724118.

[59] *Trackpy*. `https://doi.org/10.5281/zenodo.60550`. 2016.

[60] John C Crocker and David G Grier. "Methods of digital video microscopy for colloidal studies". In: *Journal of colloid and interface science* 179. 1 (1996), pp. 298–310.

[61] MM Conde, M Rovere, and P Gallo. "High precision determination of the melting points of water TIP4P/2005 and water TIP4P/Ice models by the direct coexistence technique". In: *The Journal of chemical physics* 147 (2017), p. 244506.

[62] *Numpy documentation*. `https://numpy.org/doc/stable/reference/generated/numpy.histogram_bin_edges.html#numpy.histogram_bin_edges`. 2022.

[63] *Zenodo repository*. `https://zenodo.org/doi/10.5281/zenodo.10638735`. 2024.

Supporting Information for:

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

Matteo Becchi[1], Federico Fantolino[1], Giovanni M. Pavan[1,2]

Further algorithm details

Identification and selection of the fitting interval

The identification of the correct fitting interval surrounding the histogram maximum is often critical to ensure not only the convergence, but also the quality of the Gaussian fit. Our algorithm selects the best fitting interval between two candidate intervals defined as follow:

- The "*minima*" interval is defined by the positions of the two local minima immediately before and after the maxima.
- The "*half height*" interval is determined by identifying the points on either side of the maximum in the histogram where the histogram reaches half of its maximum height. If the edge of the histogram is reached before finding a point with half height, the interval will extend up to the edge.

The fit is then attempted inside both candidate intervals, and (if both fits converge) a quality score is assigned to the intervals, according to the following desired properties:

- the value of the maximum of the fitting Gaussian is close enough to the value of the maximum of the histogram;
- the Gaussian mean μ is inside the fitting interval;
- the Gaussian width σ is smaller than the fitting interval;
- the relative uncertainty on each fitting parameter is smaller than 0.5.

Finally, the fitting interval with the highest quality score is selected. If only one of the fits converges, that one is selected. If none of them converges, a termination condition is met and the algorithm exits the iterative loop.

Removing strongly overlapping clusters

Once the algorithm has identified the set of candidate environments $\{E_n\}$, all the possible pair of environments E_n and E_m are compared. If

$$\frac{A_n}{\sigma_n} > \frac{A_m}{\sigma_m}$$

(meaning, E_n has a peak higher than E_m), and

$$|\mu_n - \mu_m| < \sigma_n$$

[1]Department of Applied Science and Technology, Politecnico di Torino, Torino 10129, Italy
[2]Department of Innovative Technologies, University of Applied Sciences and Arts of Southern Switzerland, Lugano-Viganello 6962, Switzerland

(meaning, E_m is closer to E_n then E_n's typical fluctuation amplitude), then E_m is considered contained in E_n. E_m is thus removed from the list of candidate environments, and all the data points previously classified inside E_m are now consider classified inside E_n. If E_m meets the criteria for being contained in more that one different environment, the one with the mean closer to μ_m is chosen.

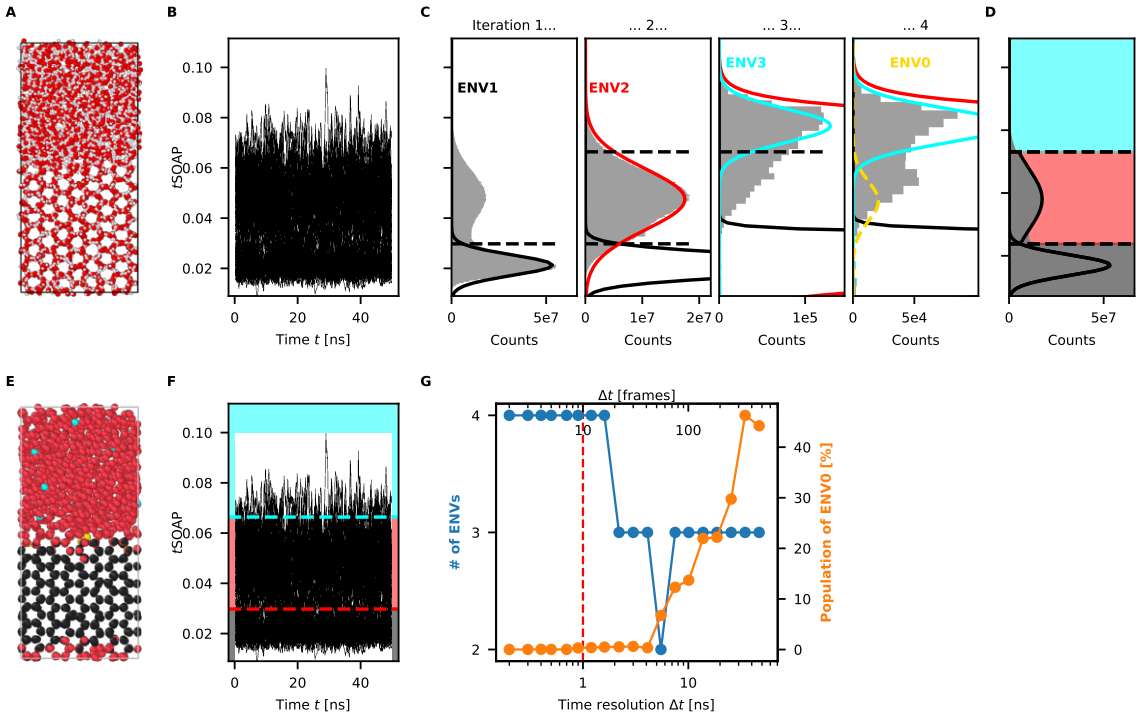

Fig. S1: **Clustering on tSOAP signals with $\Delta t = 10$ frames.** A: Snapshot of the simulation of ice/water coexistence; the simulation is performed on 2048 TIP4P/ICE molecules, lasts 50 ns and it's sampled every 0.1 ns. B: tSOAP signals for all the oxygen atoms, as a function of time. Data are smoothed with a rolling average with width of 8 frames. C: Data cumulative distribution function at the four 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. At the fourth iteration, no data are assigned to the proposed cluster, and the algorithm stops. D: final clustering of the tSOAP signals. E: Snapshot of the simulation, colored according to the clustering. F: Same tSOAP 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 $\Delta t = 1$ ns (10 frames) indicates at which time resolution the analysis shown in the previous panels was performed.

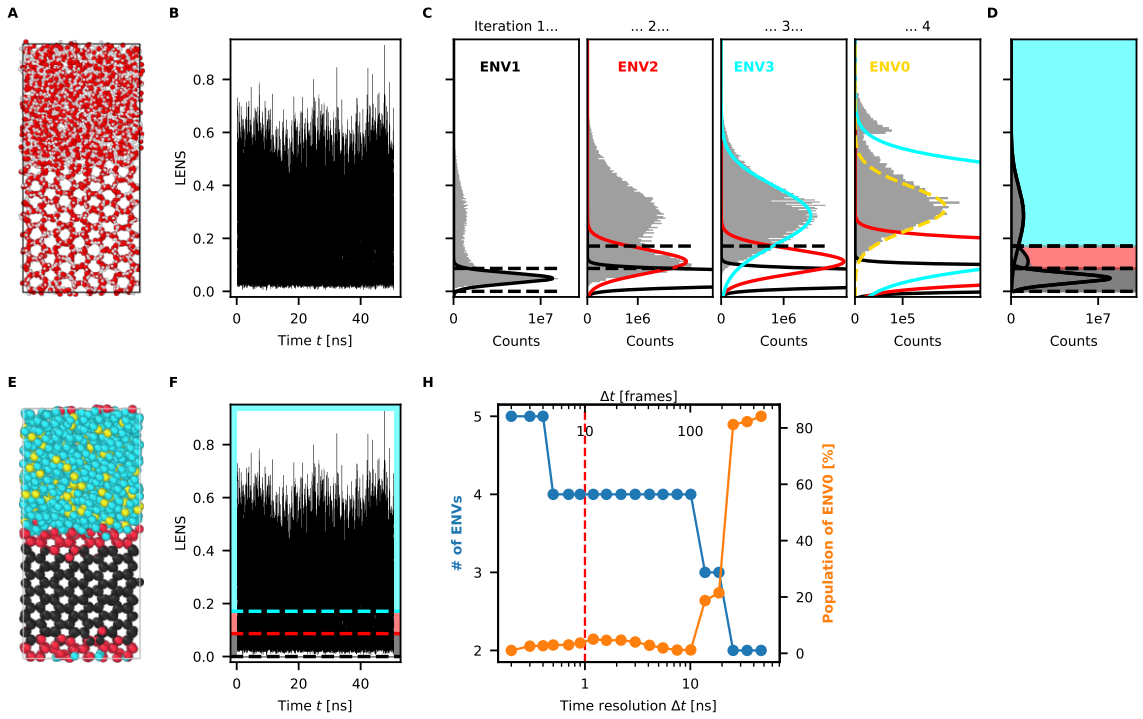

Fig. S2: **Clustering on LENS with $\Delta t = 10$ frames.** A: Snapshot of the simulation of ice/water coexistence; the simulation is performed on 2048 TIP4P/ICE molecules, lasts 50 ns and it's sampled every 0.1 ns. B: LENS signals for all the oxygen atoms, as a function of time. C: Data cumulative distribution function at the four 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. At the fourth 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 $\Delta t = 1$ ns (10 frames) indicates at which time resolution the analysis shown in the previous panels was performed.

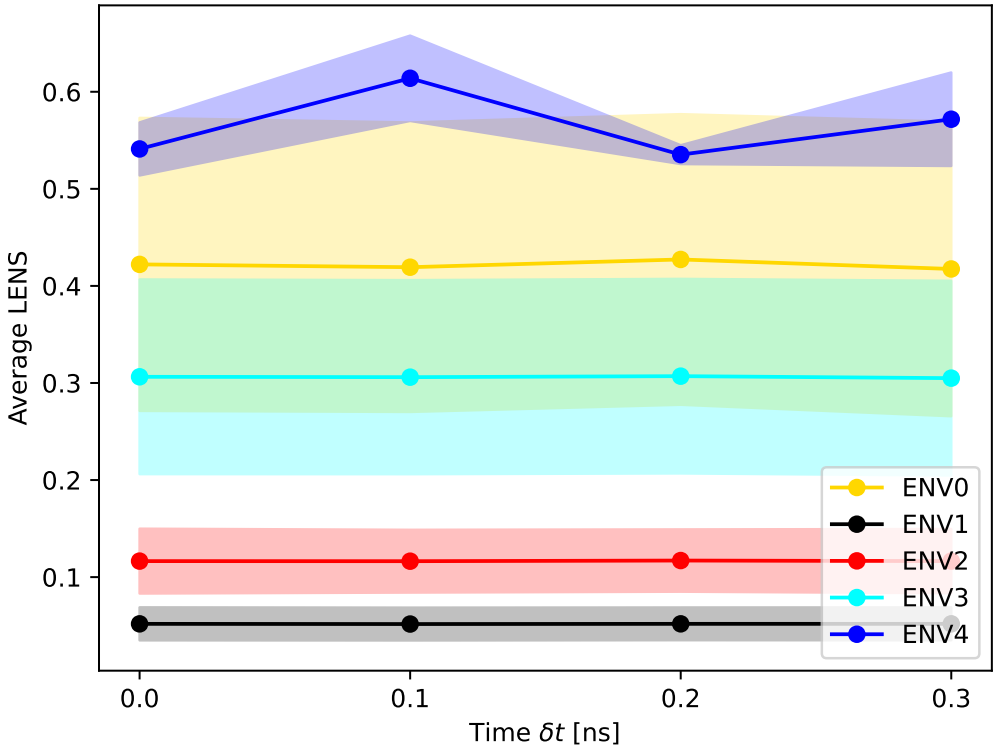

Fig. S3: **Characterization of the clusters shown in Fig1 in the main text.** The plot shows, for each cluster (ENVs0-4), the average, over all the windows classified in that cluster, of the LENS signal in the consecutive frames inside the window. The data refers to the clustering performed with a time-resolution $\Delta t = 0.4$ ns (*i.e.*, 4 simulation frames). The solid dots are the average, the transparent bands the standard deviation. It can be seen that the clusters ENV1 (in black, corresponding to solid ice), ENV2 (in red, corresponding to solid/liquid interface), ENV3 (cyan, corresponding to the majority of the liquid water) and ENV4 (blue, corresponding to liquid water with high values of LENS) are well separated with respect to their standard deviation. The ENV0 cluster instead (in yellow) includes data windows with high variability, which reflects in its large standard deviation.

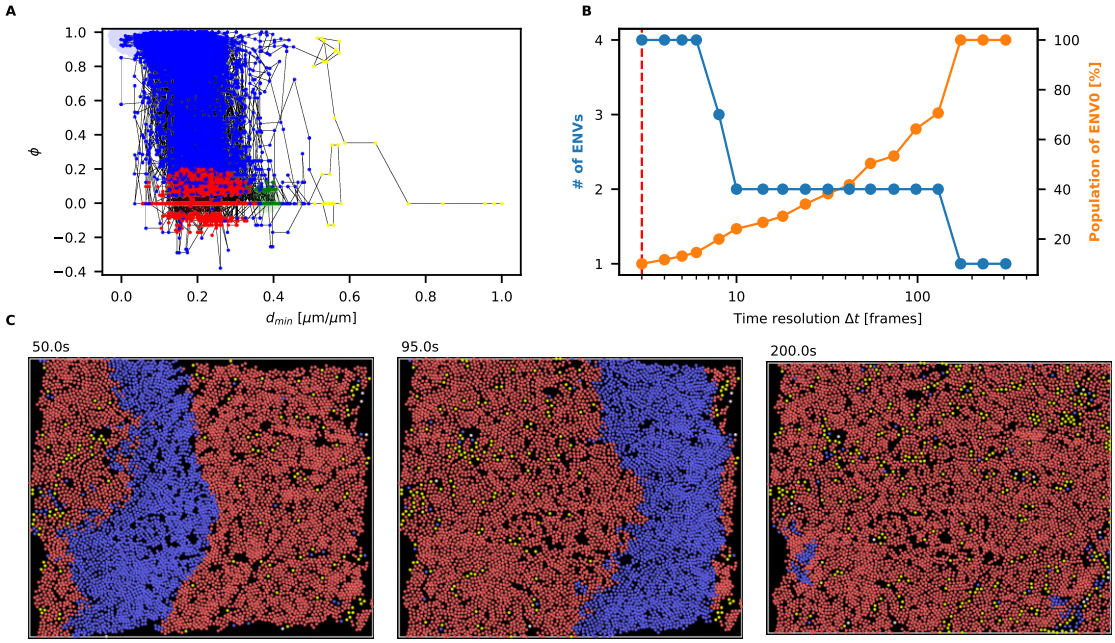

Fig. S4: **Analysis of multivariate data from experimental microscopy video of colloidal particles, using $\Delta t = 3$ frames. Compare with Fig5 in the main text.** A: the algorithm identifies three environments, in red, blue and green respectively, plus the ENV0 cluster in yellow. E: Blue line: number 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 $\Delta t = 3$ frames indicates at which time resolution the analysis shown in the previous panels was performed. F: three snapshots of the video, colored according the clustering output.

Movie S1 MD trajectory of solid/liquid water coexistence used for the analysis of Fig1 in the main text, colored according to the clustering with $\Delta t = 0.4$ ns. 5 clusters are found; the black one (ENV1) correspond to solid ice, the red one (ENV2) mainly to the solid/liquid interface, the cyan (ENV3) and blue (ENV4) ones to liquid water, and the yellow one (ENV0) to the unclassified points.

Movie S2 The MD trajectory of freezing water at $T = 267$ K used for the analysis of Fig2 in the main text, colored according to the clustering with $\Delta t = 25$ ps. 5 clusters are found; the black one (ENV1) correspond to solid ice, the red one (ENV2) mainly to the solid/liquid interface, the cyan (ENV3) and blue (ENV4) ones to liquid water, and the yellow one (ENV0) to the unclassified points.

Movie S3 The MD trajectory of copper 211 surface used for the analysis of Fig3 in the main text, colored according to the clustering with $\Delta t = 0.12$ ns. 5 clusters are found; the black one (ENV1) correspond to the atoms in the bulk, the red one (ENV2) to a fraction of the surface atoms, the orange (ENV3) and cyan (ENV4) ones to atoms sliding on the surface, and the yellow one (ENV0) to the unclassified points.

Movie S4 Reconstruction of the experimental video used for the analysis of Fig5 in the main text, colored according to the clustering with $\Delta t = 5$ frames. 4 clusters are found; the red one (ENV1) correspond to the majority of stationary rollers, the blue one (ENV2) to rollers inside the wave moving in a coherent and ordered way, the green (ENV3) one to stationary rollers in areas with exceptionally low particle density, and the white one (ENV0) to the unclassified points, mainly located before and after the wave.

Dataset S1: timeseries_Fig1.npy LENS signals for Fig1 of the main text. The file contains an array of shape (N, T), where $N = 2048$ is the number of TIP4P/ICE molecules and $T = 500$ is the number of simulation frames.

Dataset S2: timeseries_Fig2.npy tSOAP signals for Fig2 of the main text. The file contains an array of shape (N, T), where $N = 2048$ is the number of TIP4P/ICE molecules and $T = 40000$ is the number of simulation frames.

Dataset S3: timeseries_Fig3.npz LENS signals for Fig3 of the main text. The file contains an array of shape (N, T), where $N = 2400$ is the number of Cu atoms and $T = 15270$ is the number of simulation frames.

Dataset S4: timeseries_Fig4.npy Synthetic signals for Fig4 of the main text. The file contains an array of shape (D, N, T), where $D = 3$ is the number of signal components, $N = 2$ is the number of Langevin molecules and $T = 10000$ is the number of simulation frames.

Dataset S5: timeseries_Fig5.npy $d_{\min}$ and ϕ signals for Fig5 of the main text. The file contains an array of shape (D, N, T), where $D = 2$ is the number of signal components, $N = 6921$ is the number of colloidal particles and $T = 311$ is the number of video frames.