




Data-driven detection of critical points of phase transitions in complex systems

Peng Tao ^{1,2}, Chenghang Du³, Yi Xiao ²✉ & Chen Zeng ³✉

Detecting the critical points of phase transitions and their driver factors in complex systems from data is a very challenging task. In these regards, the dynamic network biomarker/marker (DNB) method derived from the bifurcation theory is currently very popular, but a unified criterion to pick the most appropriate DNBs is lacking. Here, we propose a giant-component-based DNB (GDNB) method inspired by the percolation theory, that directly selects the largest DNB as the transition core to reflect the progress of the transition. We test the effectiveness of this scheme to detect transitions on three distinct systems, differing in terms of interactions and transitions: Monte Carlo simulations of the 2D Ising model, molecular dynamics simulations of protein folding, and measured gene expression time course in mouse muscle regeneration. These results suggest that the GDNB method inherits all the advantages of the DNB method, while it improves the interpretability at a reduced computational complexity.

¹Key Laboratory of Systems Health Science of Zhejiang Province, Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Chinese Academy of Sciences, Hangzhou 310024, China. ²School of Physics, Huazhong University of Science and Technology, Wuhan 430074 Hubei, China. ³Department of Physics, The George Washington University, Washington, DC 20052, USA. ✉email: yxiao@hust.edu.cn; chenz@gwu.edu

Phase transitions are ubiquitous in almost all fields such as physical systems^{1,2}, biological systems^{3–5}, climate systems⁶, and economic systems⁷. It describes the process of a system changing from a stable state to another stable state, such as from a healthy state to a disease state in a biological system. There exists a critical point during the two steady-state transitions, and many interesting phenomena occur at this critical point such as critical opalescence⁸. Accurately pinpointing this critical point and identifying associated factors driving the phase transition may allow us to predict and even control this phase transition, thus it is of great importance to formulate a framework to develop heuristics for transition detection. Since, for a complex system, it is usually difficult to construct a concise model description but relatively easy to obtain the measurement of the system, it is imperative for a useful method to detect the critical points from the measurement data directly.

At present, several methods have been developed based on such concepts as eigenvalue spectrum⁹, critical slowing-down^{10–15}, dynamic network biomarkers/markers (DNB)^{16–25} and machine learning (ML) based methods^{26–29}. In this article, we mainly focus on the DNB and ML methods. According to the bifurcation theory, phase transition can usually be divided into three states, namely, before-transition state, pre-transition state, and after-transition state (Fig. 1a). It can be proved that at pre-transition state, at least one group of variables meets the following three conditions¹⁶: (1) the fluctuations or standard deviations of the variables in the group increase significantly; (2) the correlation between any two variables within the group increases; and (3) the correlation between the variables within the group and the variables outside the group decreases. Therefore, Chen et al.¹⁶ formulated a composite index (or DNB score) integrating these three conditions to quantify early-warning signals of complex diseases, e.g., the onset of cancers. Later, this method was further used for detecting the tipping points of phase transitions in non-disease systems, and considerable success was also reported³⁰. This evidence shows that the DNB method is a powerful model-free algorithm because it can effectively distinguish the before-transition state and pre-transition state which is essentially the limit of the before-transition state before the tipping point. However, this method has certain difficulties in practical applications. First, the DNB method usually requires two sets of data for comparison. It performs a statistical analysis on the two sets to screen for possible variables for subsequent correlation analysis, such as comparing the gene expression data of control and case samples to select the differentially expressed genes as candidate genes for DNB. Thus, this algorithm is not suitable for the situation where there is only one set of data. In addition, after clustering the variables selected by specific criteria, a multiple of potential groups satisfying these three conditions may be obtained, so a more serious issue is how to rank these groups in relevance. In the original DNB method, domain-specific knowledge is heavily relied upon to choose the group with a higher likelihood of an imminent phase transition, which is not suitable when prior knowledge is lacking. Like the DNB method, machine learning-based methods also have similar problems with prior knowledge (see Methods for details).

On the other hand, in statistical physics, one of the most important features of the water-to-ice phase transition is the scale invariance, that is, there are many pieces of ice of different sizes in the water, and the largest one (the so-called giant component in network science) reflects the progress of this transition. Inspired by this fact, we propose the giant-component-based DNB (GDNB) method, which has two main advantages over DNB, namely, (1) it does not need two sets of data for comparison since relative fluctuations are used to screen for candidate variables; and (2) it does not need prior knowledge to screen for multiple

possible candidate DNBs, as we select the giant component to represent the phase transition. In order to illustrate the effectiveness of the GDNB method, both simulated data and real data are used for validation.

Results

GDNB algorithm flow. To better understand the analysis process and results, we first introduce the entire workflow of the GDNB algorithm. Data preprocessing is excluded since it is a necessary step common to all data-driven algorithms. The input data used by GDNB can be regarded as an array $\{x_{ijk}\}$ with $(i = 1, 2, \dots, m; j = 1, 2, \dots, n; k = 1, 2, \dots, s)$ (Fig. 1b), where m represents the number of observation points of the system on the reaction coordinate, e.g., if the reaction coordinate is time, then m is the number of time points, n is the number of variables included in the system, and s is the number of repetitions at each observation point. When the input data is given, the GDNB algorithm completes the analysis through the following three steps:

1. Find out the variables with relatively large fluctuations at each observation point (Fig. 1c).

Similar to the DNB algorithm, we use the standard deviation to describe the fluctuation of each variable in the system. However, because the GDNB method does not have reference data, an adjustment needs to be given to the scale of the variable. Here we use relative fluctuations, a term in statistical physics, to describe the changes of variables with different scales, which is defined by

$$\text{RF}_{ij} = \frac{1}{\bar{x}_j} \sqrt{\sum_{k=1}^s (x_{ijk} - \bar{x}_j)^2}, \quad (1)$$

where $\bar{x}_j = \frac{1}{m \times s} \sum_{i=1}^m \sum_{k=1}^s x_{ijk}$. The one-sample Student's t-test (ttest_1samp in Python package scipy) is used to select the variables (candidate variables) with significantly large fluctuations by setting a significance level, e.g., $p = 0.05$. The group of the selected variables at i th observation point is denoted as F_i .

2. Cluster the selected variables for each observation point (Fig. 1d).

This step is the same as the DNB algorithm, in which the selected variables at i th observation point are hierarchically clustered based on the absolute value of the Pearson's correlation coefficient (PCC) between any pair of selected variables (g, h), which is calculated by

$$|\text{PCC}_i^{gh}| = \left| \frac{\sum_{k=1}^s (x_{igk} - \bar{x}_{ig})(x_{ihk} - \bar{x}_{ih})}{\sqrt{\sum_{k=1}^s (x_{igk} - \bar{x}_{ig})^2} \sqrt{\sum_{k=1}^s (x_{ihk} - \bar{x}_{ih})^2}} \right|, (g, h \in F_i; g \neq h), \quad (2)$$

where $\bar{x}_{ig} = \frac{1}{s} \sum_{k=1}^s x_{igk}$ and $\bar{x}_{ih} = \frac{1}{s} \sum_{k=1}^s x_{ihk}$. The number of the clusters could be determined by a threshold between 0 and 1 (see Methods).

3. Select the largest cluster as the transition core for each observation (Fig. 1e).

This step is the essential difference between the GDNB and DNB algorithms. In this step, the DNB algorithm may find multiple dominant groups (transition cores) that meet the three criteria at one observation point, making the algorithm require sufficient prior knowledge to filter out inapposite groups. Inspired by the water-to-ice transition, for the i th observation point, we directly select the largest cluster or giant component (denotes as G_i and its size as GC_i) as the transition core, which not only makes the computation more efficient, but also makes our algorithm easier to understand and explain.

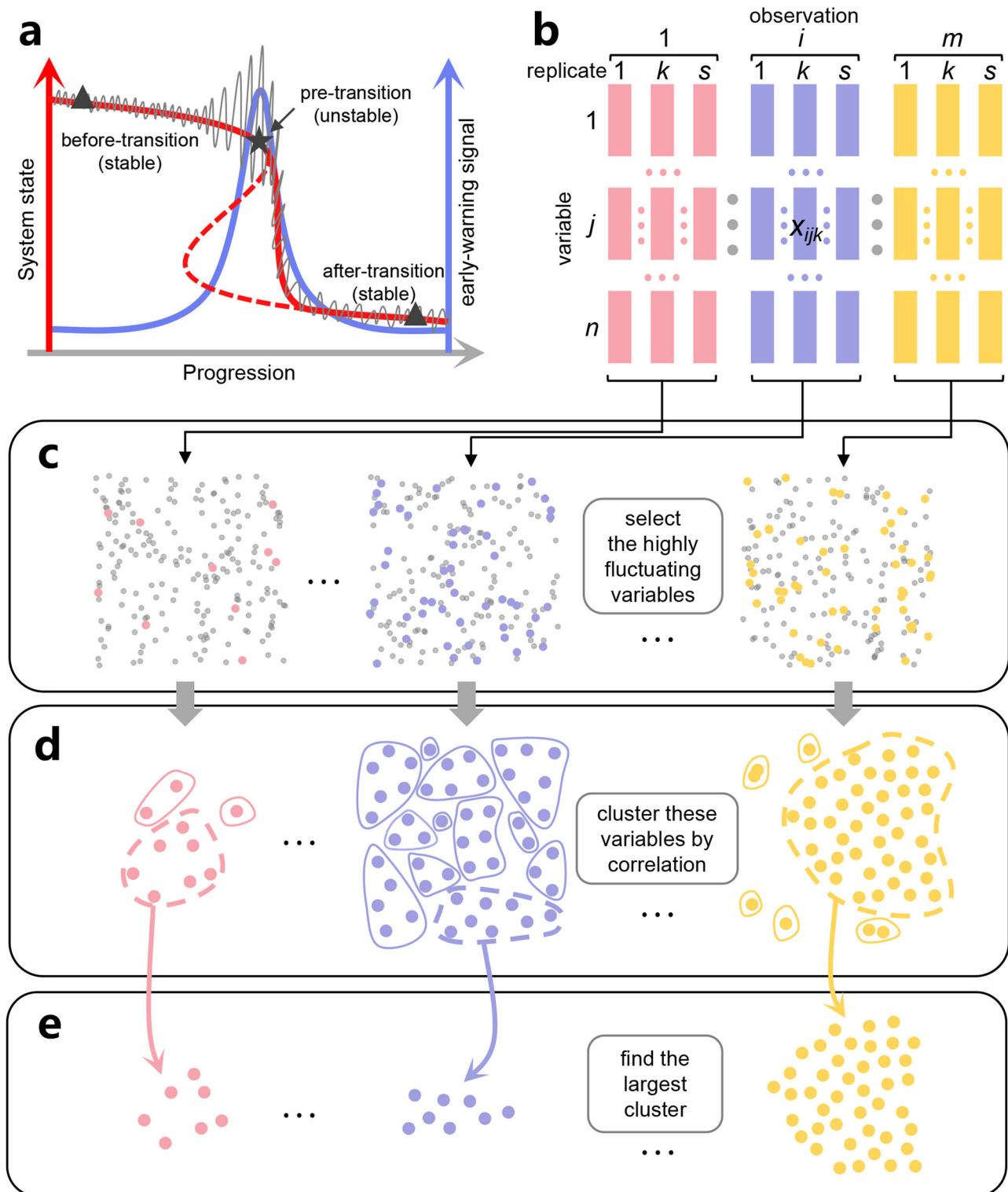


Fig. 1 The workflow of the giant-component-based dynamic network biomarker (GDNB) method. **a** A three-state model, before-, pre- and after-transition states, is used to describe a sudden phase-transition (red solid line), in which the first and last states are stable while the pre-transition state is unstable. The pre-transition state is the limit of the before-transition state before the critical or tipping point and the aim of the GDNB method is to detect the early-warning signal (the peak of the blue solid line) of the transition and corresponding driver factors based on measurement data. **b** The input data of GDNB is $\{x_{ijk}\}$, a 3D array, which consists of m observations on n variables with s replicates ($i = 1, 2, \dots, m; j = 1, 2, \dots, n; k = 1, 2, \dots, s$). **c** The first step is to select the variables with significantly high fluctuations as candidates at each observation point. **d** The second step is to cluster the variables selected from the first step according to their correlations. **e** The last step is to find the largest cluster of each observation point as DNB, which is used to further characterize the phase transition at this point.

Through these three steps, we can obtain the transition core at each observation point. To further quantify the strength of the early-warning signals for the i th observation point, we also introduce a composite index (CI) defined as

$$CI_i = GC_i \times RF_i \times |PCC_i|, \quad (3)$$

where

$$RF_i = \frac{1}{GC_i} \sum_{j \in G_i} RF_{ij} \quad (4)$$

and

$$|PCC_i| = \frac{2}{GC_i(GC_i - 1)} \sum_{p, q \in G_i; p \neq q} |PCC_i^{pq}|. \quad (5)$$

Clearly, this CI is a production of three factors, that is, the size, average relative fluctuation, and average correlation of the giant component. The larger the CI, the stronger the signal. Note that the proposed form of the CI is a linear multiplication of the three factors, but a nonlinear form could also be adapted, such as $\sqrt{GC}(RF + 1)^{|PCC|}$. Next, to demonstrate the underlying working of the GDNB method, we first apply the GDNB method to detect the known transition of the 2-dimensional (2D) Ising model.

Application to the 2D Ising model. The Ising model is a classic statistical physics model, which was first proposed in the 1920s by Wilhelm Lenz³¹ and has been widely used to describe the phase transitions of various systems because of its simple form and universal applicability³². Therefore, we first verify our algorithm on this model. The 2D Ising model can be described by the following energy function

$$E = \frac{1}{2} \sum_{j \in N_i} \sum_i J_{ij} s_i s_j + \sum_i H_i s_i, \quad (6)$$

where s stands for the spin with two possible values of $+1$ or -1 , N_i is the set of the nearest neighbors of s_i (Fig. 2a), J_{ij} is the coupling coefficient of s_i and s_j , and H_i is the external magnetic field on s_i .

Below we take the 10×10 Ising model ($n = 100$) as an example to apply the GDNB method. First, we used the Monte Carlo (MC) method to simulate the model at 41 different temperatures (or observation points $m = 41$, see Methods). The temperature T was increased from 1 to 5 with an interval of 0.1 (using the reduced unit J/k_B , where k_B is the Boltzmann constant), and 100 snapshots were generated under each temperature ($s = 100$). Figure 2b compares the index curve of the phase transition of GDNB and three machine learning-based methods (supervised learning (SL), prediction-based method (PBM) and learning by confusion (LBC), a detailed description of these methods can be found in Methods) on the Ising model (10 independent experiments for each). It can be seen that the critical temperature (T_c) predicted by GDNB (the maximum of its index) agrees with the theoretical value³³ (about 2.3, red dashed line), while the critical temperatures predicted by SL and PBM are 2.1, lower than the theoretical value. The critical temperature predicted by LBC (the local maximum of its index) is also consistent with the theoretical value, but it requires prior knowledge to exclude the larger local maxima at both ends (see Methods). Next, we clarify how the GDNB method works. As shown in Fig. 2c, d, at low temperatures (e.g., $T = 1$), there are fewer spins selected as candidates, and the correlations between these candidates are not strong. While at high temperatures (e.g., $T = 5$), although the fluctuations are significantly high, the dynamics of these spins are not related. In contrast, when the temperature is 2.3, there is a large group of spins that not only show abnormally large fluctuations, but also have a strong correlation within the group, which can serve as a giant component percolating the entire system (Fig. 2d).

Below we further compare the performance of the GDNB method and the three ML-based methods under 2D Ising models with different sizes ($L \times L$, $L = 10, 20, 30, 40, 50$ and 60) and different external noise intensities ($H_i \sim N(0, \sigma)$). Figure 3a compares the critical temperatures predicted by GDNB and the other three ML-based methods (each case was repeated 10 times). It can be seen that, when L increases from 10 to 60, the critical temperatures predicted by GDNB, PBM and LBC are similar, and all of them fluctuate around the theoretical value (red dashed line), while SL quickly loses accuracy when $L > 30$. In addition, we also consider a smaller Ising model ($L = 5$), and GDNB can still accurately predict the critical temperature in this challenging scenario (Supplementary Fig. S1). Figure 3b shows the predicted phase transition temperature of the 10×10 Ising model (results for the 30×30 Ising model can be found in Supplementary Fig. S2). It can be seen that, as σ increases from 0.001 to 0.5, the critical temperature T_c predicted by GDNB always fluctuates around the theoretical value. Compared to the three ML-based methods, the robustness of GDNB is significantly better than that of SL and PBM, and only slightly weaker than that of the LBC method. Considering that the LBC method relies on a priori knowledge, that is, it requires artificial selection of the local maximum solution that is closest to the theoretical value, the performance of GDNB is very promising. Next, we investigate the computational efficiency and scalability of GDNB. Figure 3c compares the time required by GDNB and three ML-based methods to solve Ising models of different sizes on the same hardware (two Intel Xeon Gold 6226 R CPUs @ 2.90 GHz). As shown in the figure, since the workflow of GDNB is simple and the operation with the highest computational load is just clustering variables, it runs very fast. In contrast, other ML-based methods need to train neural networks, which involve a lot of matrix operations, so they run significantly slower than GDNB, especially the LBC method that needs to train a neural network for each temperature, resulting a very high time complexity. In addition to computational efficiency, another common problem of ML-based methods is the need for prior knowledge, for example, SL requires to know in advance which temperatures correspond to different phases (for the Ising model, $T = 1$ corresponds to the ferromagnetic phase, $T = 5$ corresponds to the paramagnetic phase). Moreover, when we only observe one phase, machine learning methods usually fail, for example, when we only observe data from $T \in [1, 2.5]$, the three ML-based methods cannot be used or cannot accurately find the critical point, but as shown in Fig. 3d, the GDNB method still works normally. These results further prove the superiority of GDNB.

Application to protein folding trajectory. Protein folding refers to the process of spontaneously forming a three-dimensional structure from a one-dimensional amino acid sequence, which is also a phase transition from disorder to order³⁴. Although the Deepmind³⁵ team has recently been able to use deep learning methods to achieve high-precision protein structure prediction, the protein folding mechanism remains elusive. As an indispensable tool for studying protein folding mechanisms, molecular dynamics (MD) simulations can follow the dynamics of a structure at the atomic resolution³⁶. However, the trajectory obtained by MD simulations is a sequence of atomic coordinates, thus cannot directly show the folding mechanism. Considering that the protein folding process is a phase transition process, it is natural to ask if we can use the GDNB method to find the transition core as the folding nucleus from the folding trajectory, which may help us to elucidate the folding mechanism.

Here we apply the GDNB method to the MD simulation trajectory of the villin headpiece subdomain HP35, a small

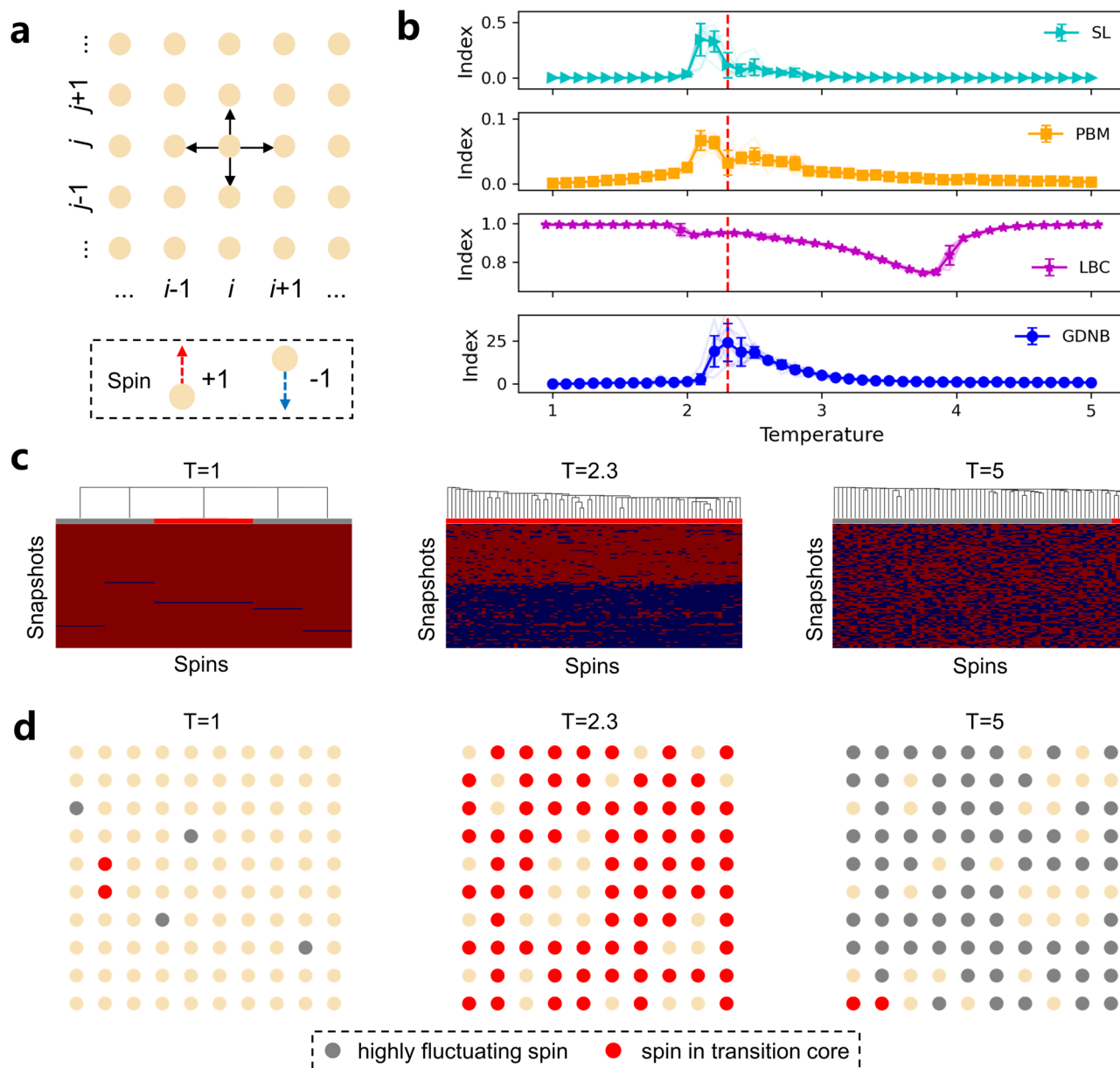


Fig. 2 Application to the 2D Ising model. **a** The 2D square-lattice Ising model is composed of some spins, whose values can only take $+1$ and -1 . Only adjacent spins can affect each other, e.g., the spin in the center and the spins pointed by arrows. **b** The computed average index (10 independent runs, error bar stands for the standard deviation) of phase transition of GDNB and three ML-based methods (supervised learning (SL): cyan triangles; prediction-based method (PBM): orange squares; learning by confusion (LBC): purple stars; GDNB: blue circles). The theoretical critical temperature (-2.3) is given by a red dashed line. **c** Hierarchical clustering results at $T=1.0, 2.3,$ and 5.0 , respectively (red: $+1$; blue: -1). Rows and columns represent the significantly highly fluctuating spins and snapshots of MC simulations, respectively. **d** The schematic diagram of phase transition at different temperatures. The spins with high fluctuations and spins in the transition core are colored in gray and red, respectively. A giant component appears at $T=2.3$.

protein with only 35 residues. As shown in Fig. 4a, its topological structure is quite simple, with only three α -helices. Therefore, the folding mechanism of this tiny protein has been extensively studied theoretically and experimentally^{37–42}. We extracted a trajectory of 10 ns from a complete simulation trajectory of $\sim 150 \mu\text{s}$ produced in our previous study³⁷ and then divided it into 10 identical time windows ($m=10$), each of them containing 50 snapshots ($s=50$). Figure 4b shows the change curves of the root-mean-square-deviations (RMSDs) of the whole protein and the two segments (N- and C-segment, as defined in Fig. 4a). By defining a successful folding event as the overall RMSD (comparing to the native structure) is less than 2 \AA , it can be seen that HP35 successfully folded at about 5 ns. Taking into

account that the spatial structure of the protein remains unchanged under translation and rotation operations, the coordinate data were converted into internal distance data, that is, the distances between the C_α atoms of different residues were used to represent the spatial structure. Since there are 35 residues for HP35, after conversion, a structure can be represented by 595 distances ($\frac{35 \times 34}{2}, n=595$, Supplementary Fig. S3). After applying the GDNB method to this trajectory, we found the CI curve remains at high values from 2 to 5 ns, with the highest value at 2 ns and the second highest value at 4 ns (Fig. 4c top and Fig. 4d). As shown in Fig. 4e, these two time points correspond to the helix1 of HP35 changing its orientation from pointing out of the paper to pointing upward and from pointing upward to pointing

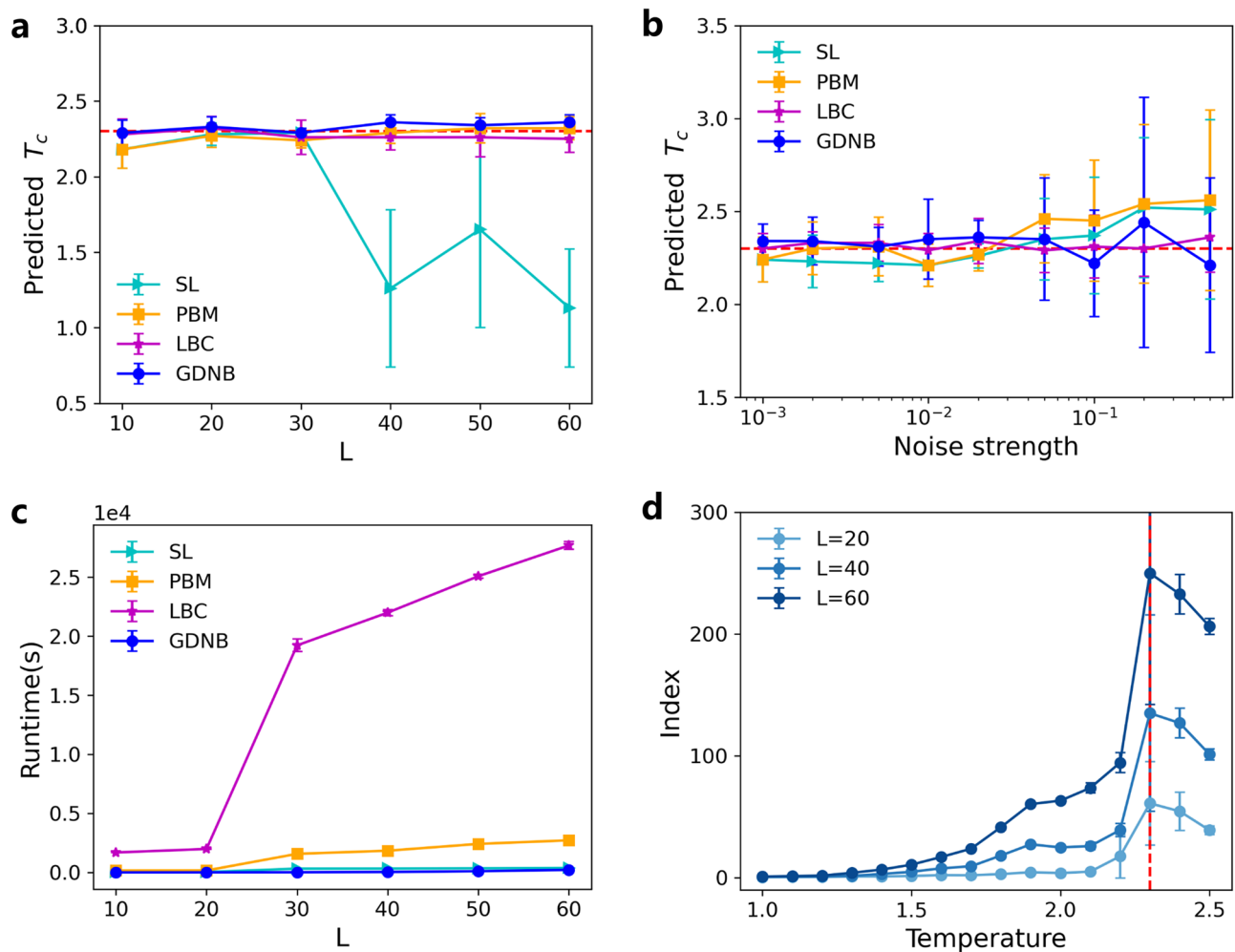


Fig. 3 Comparison of the performance of GDNB and ML-based methods. **a** The critical temperatures predicted by GDNB and the three ML-based methods under Ising models with different sizes (supervised learning (SL): cyan triangles; prediction-based method (PBM): orange squares; learning by confusion (LBC): purple stars; GDNB: blue circles). The theoretical critical temperature (~ 2.3) is given by a red dashed line. This color scheme is also used in **(b)** and **(c)**. **b** The critical temperatures predicted by GDNB and the three ML-based methods under different external noise strengths. **c** The runtimes of GDNB and the three ML-based methods to solve Ising models with different sizes. **d** The computed index of phase transition of the GDNB method under three Ising models. In this figure, error bars indicate the standard deviation of 10 independently repeated experiments.

into the paper, respectively. With the completion of the rotation of helix1, indicating HP35 entered the folded state, the transition core almost disappears after 6 ns (Fig. 4c bottom and Fig. 4e). It should be pointed out that HP35 has two types of folding pathways, one is the hierarchical folding pathway introduced above, and the other is the collaborative folding pathway³⁷. Our GDNB method is not only suitable for the first one, but also for the last, as shown in Supplementary Figs. S4 and S5.

Application to gene expression data. Up to now, we only verified the effectiveness of the GDNB method on the simulation data, and below we further demonstrate it on the real data, i.e., gene expression data, which records the regeneration process of mouse muscle after cardiotoxin injection, consisting of expression data of more than 10,000 genes at 27 time points (two samples for each, Fig. 5a shows the heatmap of the expression data)⁴³. The process of muscle regeneration can be described by the schematic diagram in Fig. 5b. Once the muscle is injured, a large number of quiescent satellite cells around the myofibers are activated and proliferated to produce a large number of myoblasts, then immature multinucleated myofibers (also known as myotube) can

form based on the differentiation and fusion of these myoblasts. After that, the nucleus of satellite cell migrates to the subsarcolemmal position, which makes the immature myofiber mature, marking the completion of the muscle regeneration. If we perform hierarchical clustering on the expression data according to 27 time points, it can be found that these time points are roughly divided into three clusters (Fig. 5c). The cyan cluster contains 5 time points, that is, days 0, 16, 20, 30, and 40. Therefore, it is reasonable to assume that the damaged muscle was recovered on day 16. In addition, the red cluster consists of 2 time points, days 0.5 and 1, and the remaining time points belong to the yellow cluster. However, the meaning of this classification is not yet clear, more in-depth analysis is required.

Considering that there are only two samples at each time point, it is impossible to perform a reasonable calculation on the standard deviations or fluctuations. Therefore, we also included the samples before and after the time point under consideration for statistical analysis, that is, 3 consecutive time points or 6 samples were used as the replicates ($s = 6$). Consequently, 27 time points were reduced to 25 ($m = 25$). After preprocessing (see Methods), 10,123 genes ($n = 10,123$) were used to do the following analysis. Figure 5d shows the results of the GDNB

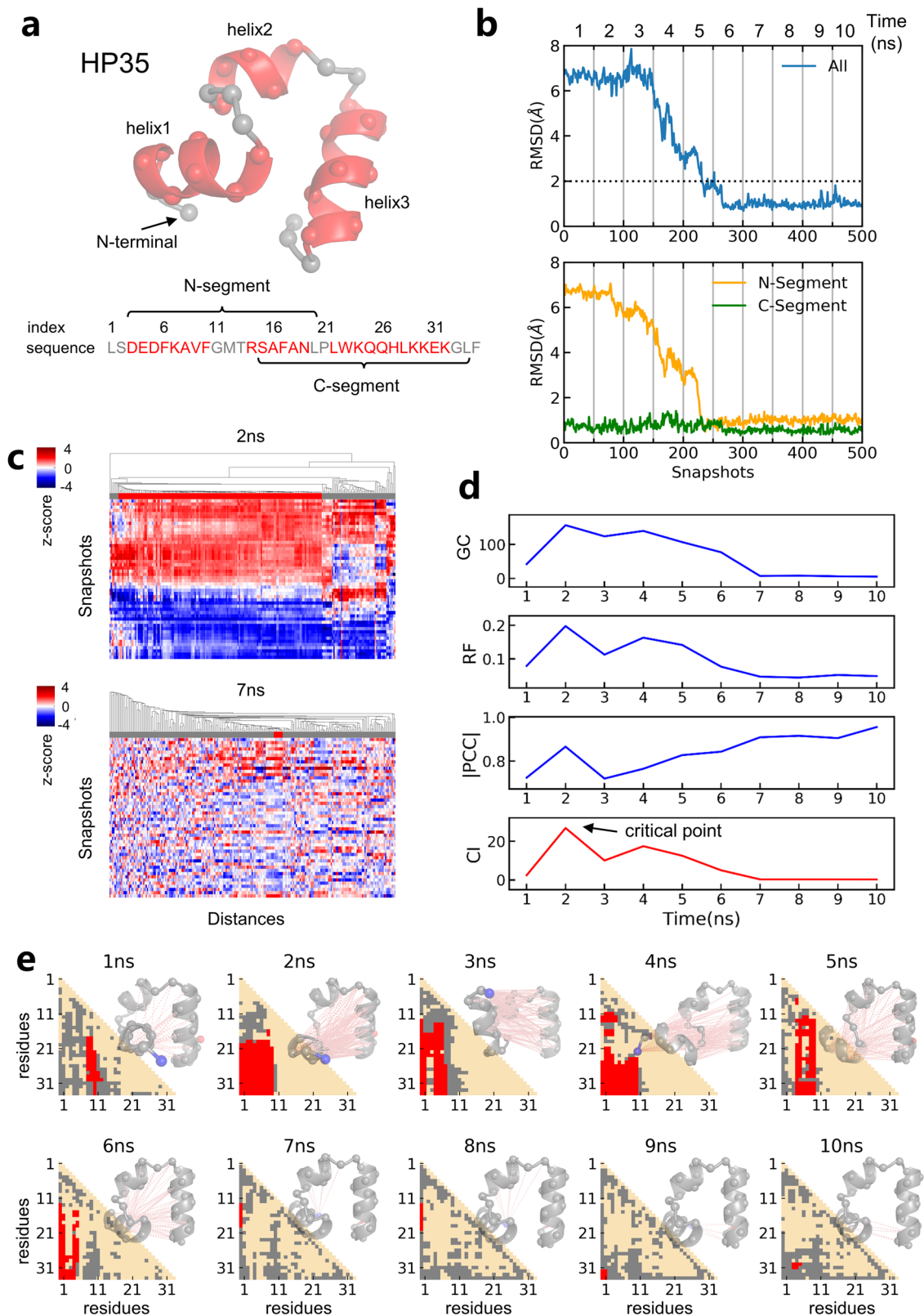


Fig. 4 Application to protein folding trajectory. a The sequence and cartoon representation of HP35. Helical regions are colored in red and structural elements are annotated. **b** The time courses of RMSDs of whole protein (top) and two segments (bottom). **c** Hierarchical clustering results at 2 and 7 ns, respectively. Rows and columns represent the significantly high fluctuating distances and snapshots of MD simulations, respectively. **d** The time courses of GC, RF, |PCC| and CI. **e** The schematic diagram of phase transition at different time points. The coloring scheme is the same as in Fig. 2 and the distances in DNB (pink dashed lines) are mapped on the 3D structures.

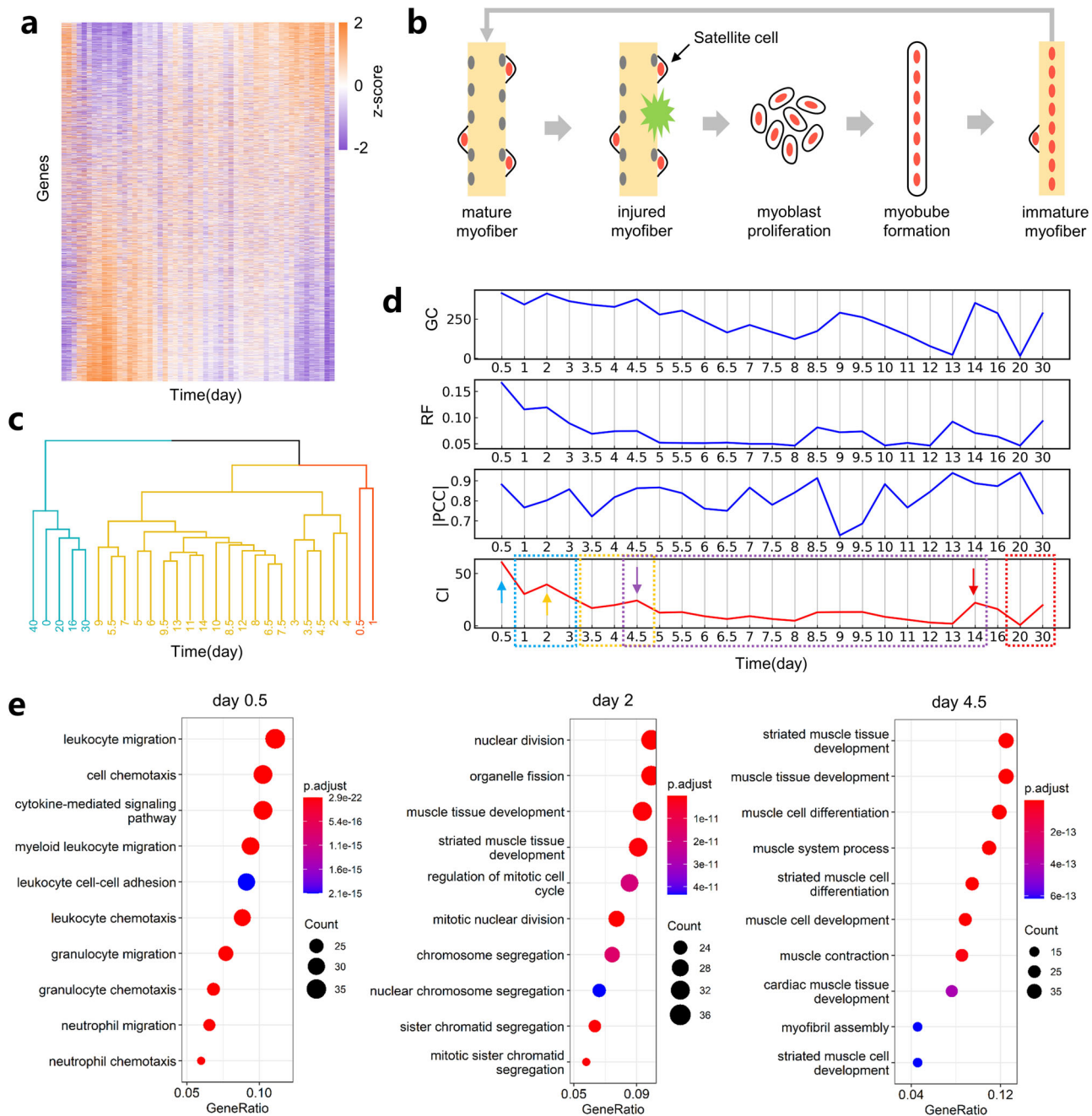


Fig. 5 Application to gene expression data. **a** The heatmap of the expression data of mouse muscle regeneration. **b** The schematic diagram of four stages of the regenerative model of muscle after injury. **c** 27 time points are clustered based on the expression data in (a). **d** The time courses of GC, RF, IPCC and CI. Four distinct peaks are indicated by arrows with four colors, and the corresponding four stages of muscle regeneration are shown in dashed boxes. **e** The results of GO functional enrichment analysis at days 0.5, 2 and 4.5, respectively. Only the top 10 significantly enriched GO terms are listed.

algorithm and it can be seen that CI has 4 distinct peaks (indicated by arrows). The first peak appears on day 0.5 (the cyan arrow). In order to explore the underlying biological function of this peak, GO functional enrichment analysis was performed on the giant component or DNB at this time point. As shown in Fig. 5e (left), the top 10 most relevant functions are listed and most of these functions are related to inflammation or immune response, which is consistent with previous studies^{44,45}. In addition, after injury, the quiescent SCs activate and proliferate, which is marked by the down-regulation of myostatin (Mstn) that inhibits the activation of SCs⁴⁶. From Supplementary Fig. S6, it can be seen that Mstn does exist in the transition cores of time points days 0.5 and 1. The second peak of CI is on day 2 (the

yellow arrow) and the functional analysis of the core genes at this time point indicates that the most significant functions are related to muscle tissue development and cell cycle (middle of Fig. 5e). This result is not difficult to understand because the activated SCs proliferate, differentiate, and fuse to form myotubes and further to form immature myofibers at this stage, which is characterized by the significant up-regulation of Myrf5, MyoD, MyoG and Pax7^{47,48}, which can also be validated from Supplementary Fig. S6. The third peak of CI appears on day 4.5 (the purple arrow) and the main function of the DNB at this time point is also related to the development of muscle tissue. In addition, compared with the functional annotations on day 2, this time point is also related to the development of myofibers, which is

Table 1 Dataset summary.

| Datasets | Space | Interaction | Source | Transition type |
|--|------------------------|-----------------------------------|----------------|------------------------------------|
| Ising model | Real 2D space | Short-range interaction | Simulated data | single-transition |
| Protein folding pathway | Real 3D space | Short- and long-range interaction | Simulated data | two single-transitions in parallel |
| Gene expression of muscle regeneration | Abstract genomic space | Unknown function interaction | Real data | multiple-transitions in series |

This table provides a summary of the characteristics (including space and scope of interactions between variables) of the datasets used in this work.

consistent with the maturation process of myofibers in Fig. 5b. The last peak of CI appears on day 14. According to the clustering results in Fig. 5c, after day 14, the muscle regeneration process has been finished, so the main function of this time point should not be directly related to the muscle, and the GO enrichment analysis results further verify our conjecture (Supplementary Figs. S7 and S8). To better understand these results, we introduce the temporal-specific pattern defined by Dadgar et al.⁴³ as the gold standard, that is, dividing the time course into 4 stages (the 4 dashed boxes in Fig. 5d), which are myoblast proliferation (days 1–3), myotube formation (days 3.5–4.5), myofiber maturation (days 4.5–14) and adaptation stage (days 20–40), respectively. It can be seen that, except for the critical point found in the maturation stage, which is located at the first time point of this stage, the critical points found by our GDNB method in the other three stages are all earlier than the corresponding actual time. Considering that the data we used for each time point contains that of previous and subsequent time points, the early-warning signals detected by the GDNB method agree well with the facts.

Discussion

Aiming at narrowing down the candidates in the widely used DNB method, we proposed an improved method GDNB based on the percolation theory. Compared with the DNB method, GDNB has two major advantages. First, the GDNB method does not need to compare two sets of data observed under different conditions, it uses relative fluctuations to select variables that may be in the DNB. Inspired by the water-ice transition in statistical mechanics, the largest DNB is selected from all possible dominant groups to represent the current transition level, which provides a plausible solution for the problem in the DNB method, that is, it does not have a unified standard to choose candidate DNBs. Since the traditional DNB method cannot handle data without a reference dataset, we did not directly compare GDNB with the DNB method, but it can still be found that the GDNB method is effective and powerful through verification on three representative datasets with diverse features (Table 1). Similar to the DNB method, GDNB is also model-free, however, the meaning of the DNBs found by our method might be unclear since there is no reference data. For example, for gene expression data, additional function annotations are needed to understand the function of these DNBs. Fortunately, this problem is not difficult to solve by utilizing some powerful databases, e.g., GO⁴⁹ and KEGG⁵⁰.

On the other hand, it can be seen from the application to gene expression data that another difficulty of the DNB method is its dependence on the number of samples because a large number of samples are needed to accurately calculate the standard deviation or fluctuation. However, obtaining enough samples is not easy in some fields, such as clinical medicine. Therefore, one of the main development directions of the current DNB method is how to use only a single sample to find DNB^{51–53}. For example, in the landscape DNB (IDNB) method proposed by Liu et al.⁵⁴, a reference sample library is constructed in advance, and then a sample-specific network is obtained by comparing the target

sample with the reference sample library, and the DNB score (or CI in this paper) of each gene in the network can be calculated. After selecting the highest k scores and averaging them to get the overall DNB score, the early-warning signal based on a single sample can be detected. Since GDNB and IDNB are trying to solve the problems of the DNB method from different aspects, the idea of IDNB can be further integrated into the GDNB method, which may be implemented in our future work.

Finally, two promising directions might be pointed out. First, a large number of researches related to DNB methods consider only the first-order correlation (nodes-based network). However, it has been shown that the second-order correlation (edges-based network) may be more robust^{55,56}. The last point is that it is widely known that the correlation is not equivalent to the causality, but this is not reflected in the DNB-based methods, which should benefit from the advance of causal science^{57–59}.

Method

Relation between DNB and GDNB. The original DNB theory¹⁶ focuses on the following dynamical system

$$Z(k+1) = f(Z(k); P), \quad (7)$$

where $Z(k) = (z_1(k), \dots, z_n(k))$ represent observed data describing the dynamical state of the system, P are parameters representing slowly changing factors, which drive the system from one state (or attractor) to another, $f = (f_1, \dots, f_n)$ are generally nonlinear functions of $Z(k)$ (note that f can contain random noise). For such a dynamical system (usually with a large number of variables and parameters), provided that the system driven by some parameters approaches the critical point, theoretically, the system can be expressed in a very simple form, i.e., one- or two-variable equations in an abstract phase space around a codim-1 bifurcation point¹⁶. The GDNB method, as a variant of DNB, retains the above theoretic foundations of the DNB method, while introducing the concept of giant component from percolation theory and applying it in the models of statistical physics. Therefore, in our view, it can be said that GDNB is a bridge that simultaneously connects the bifurcation theory, percolation theory and statistical physics.

Machine learning-based methods. Machine learning based methods have been widely used to identify phase transitions from data in recent years. As ref. ²⁹ classified, there are three main types of ML-based methods: supervised learning (SL), learning by confusion (LBC), and prediction-based method (PBM). SL assumes that there are two clear phases A and B, and the regions I and II are deep inside them. A predictive model is trained on the data from the two regions with the labels 1 and 0, respectively, using a cross-entropy (CE) loss. The model is then tested on all the data. The index for phase transitions in SL is the negative derivative of the prediction with respect to the tuning parameter, and the critical value is estimated by the global maximum. In LBC, predictive models are trained on all the data. The labels are assigned by splitting the parameter range into two adjacent

regions I and II, and labeling them as 1 and 0, respectively. For each split, a different predictive model is trained using a CE loss. The model is then evaluated on all the data points. The mean classification accuracy, or the index for phase transitions in LBC, is expected to have a local maximum at the critical point. PBM trains a predictive model on all the data to estimate the tuning parameter value for each input. The model uses a mean-square-error loss function and is tested on all the data points. The mean prediction as a function of the tuning parameter is obtained by averaging over the predictions for all the data. The index for phase transitions of PBM is the derivative of the mean prediction with respect to the tuning parameter, and the critical value is estimated by the global maximum. In this work, for these three methods, we applied a three-layer perceptron ($[L \times L, 100, 100, d]$) to train simulation data, in which the conformation of the 2D Ising model is flattened into a one-dimensional vector as the input (length is $L \times L$), and the output dimension d for SL and LBC is 2, 1 for PBM. ReLU⁶⁰ was used as the nonlinear activation function. In order to accelerate the convergence of the training process, the network parameters were optimized using the Adam⁶¹ optimizer, the learning rate was set to 0.001, and the weight decay was set to 0.01. The batch size for training was set to 100, and the maximum number of training epochs was 20.

Monte Carlo simulation of 2D Ising model. An $L \times L$ matrix ($L = 10, 20, 30, 40, 50, 60$) was randomly generated (elements are 1 or -1) as the initial configuration of the Ising model. The Monte Carlo simulations for a given temperature T consist of the following steps. (1) Randomly select a spin to flip (1 to -1 , or -1 to 1); (2) Calculate the energy difference ΔE before and after the flip according to Eq. (6); (3) If $\Delta E \leq 0$, then accept the flip, otherwise, accept the flip with a probability proportional to $e^{-\Delta E/T}$; and (4) Repeat steps 1-3 until a sufficiently large number of simulation steps is reached. The Ising models of different sizes were simulated at 41 temperatures (from 1 to 5, with an interval of 0.1). In order to make the simulations reach equilibrium, the number of simulation steps ranged from 100,000 ($L < 30$) to 10,000,000 ($L \geq 30$). 100 snapshots were extracted at even intervals from these simulated trajectories as input to the GDNB method (Supplementary Fig. S9 compares the results of different snapshot numbers, and the prediction results have converged at 100 snapshots.). For each model, 10 replications were conducted with different random number seeds.

Folding trajectory of HP35. The details of HP35 folding simulations are given in the literature³⁷, from which a 10-ns folding trajectory was extracted and then divided into 10 1-ns windows, each window contains 50 snapshots. To facilitate subsequent analysis, the three-dimensional coordinate data of the structure was converted into internal distance data, that is, the C_α atom distances between different residues. Since HP35 has 35 residues, for each structure, 595 distances were calculated ($\frac{35 \times 34}{2}$).

Gene expression data for muscle regeneration. The gene expression data can be found in the literature⁴³ (or GEO database with accession ID GSE469). The original expression data recorded the expression data of 12,422 genes at 27 time points, with two samples at each time point. 1135 records without gene symbols were filtered out and the expression data in the rest records were transformed to a logarithm scale with base 2. Because the expression levels of some genes are relatively low, this not only leads to negative numbers in the expression data after processing, but also makes the noise easy to cover the original expression signals. Therefore, in the logarithm scale, the genes with average expression values < 7 were deleted. Finally, to

calculate fluctuations more accurately, we considered not only the 2 replicate samples at the current time point, but also the samples at the previous and next time points, so 6 samples were used to calculate the relative fluctuations for each time point. In addition, we also calculated the case where the samples from the previous two and the next two time points (a total of 10 samples) are included at the same time. As shown in Supplementary Fig. S10, compared with 6 samples, the overall trend of the CI curve of 10 samples is consistent, but smoother (i.e., the time resolution is reduced), which may cause some key phase transition points to be missed, such as the peak of the CI curve near day2 disappeared.

Parameters of GDNB. The GDNB algorithm has two main parameters. The first parameter is a p -value (PV_{cut}) to select the variables with significantly high fluctuations. For the 2D Ising model and the folding trajectory, the parameters were set to 0.05 and 0.01, respectively. For gene expression data, an extremely small p -value (e.g., $1e-10$) may still produce thousands of potential genes for some time points, which makes it difficult to analyze the functions of DNBs. Therefore, a simpler and more intuitive truncation scheme was adopted, that is, the top 500 genes with the largest relative fluctuations were selected at each time point. The second parameter of the GDNB algorithm is the threshold of the absolute value of the PCC during clustering (PCC_{cut}), which should be between 0 and 1 (Note that PCC can be replaced by Spearman's rank correlation coefficient, but this has no significant impact on the results, see Supplementary Fig. S11). If this value is too small, the existing transition core may not be detected, while too large may result in many false-positive results. Generally speaking, the lesser the number of samples, the larger this value should be. In the three examples of this work, i.e., Ising model ($s = 100$), protein folding pathway ($s = 50$) and muscle regeneration ($s = 6$), the thresholds were set to 0.6, 0.05, and 0.02, respectively. Although there is no precise scheme to determine these parameters, the performances of our method are robust for different combinations of these parameters, as illustrated with the example of the Ising model (Supplementary Fig. S12). If there is only one variable in DNB, then RF and $|PCC|$ were directly set to 0.

GO function enrichment analysis. In this paper, GO function enrichment analysis (including biological process, molecular function and cellular component) was performed to identify the functions of DNBs. Before analysis, all symbols of genes were converted to Entrez ID. All functional analyses were based on the R package clusterProfiler⁶², and the significantly enriched GO terms were chosen with p -value (< 0.05) under the FDR correction.

Data availability

All datasets used in this study are available at <https://github.com/PengTao-HUST/GDNB>.

Code availability

The Python code for this study is available at <https://github.com/PengTao-HUST/GDNB>.

Received: 11 April 2023; Accepted: 13 October 2023;
Published online: 21 October 2023

References

1. Roy, S. B. First order magneto-structural phase transition and associated multi-functional properties in magnetic solids. *J. Phys. Condens Matter* **25**, 183201 (2013).

2. Bernevig, B. A., Hughes, T. L. & Zhang, S. C. Quantum spin hall effect and topological phase transition in HgTe quantum wells. *Science* **314**, 1757–1761 (2006).
3. Patel, A., Lee, H. O., Jawerth, L., Maharana, S. & Alberti, S. A liquid-to-solid phase transition of the ALS protein FUS accelerated by disease mutation. *Cell* **162**, 1066–1077 (2015).
4. Fukada, T. STAT3 orchestrates contradictory signals in cytokine-induced G1 to S cell-cycle transition. *EMBO J.* **17**, 6670–6677 (2014).
5. Houfang, L., Shuyi, Z., Yunpeng, S. & Cong, L. Biochemical and biophysical characterization of pathological aggregation of amyloid proteins. *Biophys. Rep.* **8**, 42–54 (2022).
6. Sumant, N. The annual warm to cold phase transition in the eastern equatorial pacific: diagnosis of the role of stratus cloud-top cooling. *J. Clim.* **10**, 2447–2467 (2008).
7. May, R. M., Levin, S. A. & Sugihara, G. Complex systems: ecology for bankers. *Nature* **451**, 893–895 (2008).
8. Binney J. J., Dowrick N. J., Fisher A. J. & Newman M. E. J. *The Theory of Critical Phenomena: An Introduction to the Renormalization Group* (Oxford University Press, 1992).
9. Trefethen, L. N., Trefethen, A. E., Reddy, S. C. & Driscoll, T. A. Hydrodynamic stability without eigenvalues. *Science* **261**, 578–584 (1993).
10. Drake, J. M. & Griffen, B. D. Early warning signals of extinction in deteriorating environments. *Nature* **467**, 456–459 (2010).
11. Dakos, V. et al. Slowing down as an early warning signal for abrupt climate change. *Proc. Natl Acad. Sci. USA* **105**, 14308–14312 (2008).
12. Carpenter, S. R. et al. Early warnings of regime shifts: a whole-ecosystem experiment. *Science* **332**, 1079–1082 (2011).
13. Lenton, T. M. et al. Tipping elements in the Earth’s climate system. *Proc. Natl Acad. Sci. USA* **105**, 1786–1793 (2008).
14. Marconi, M. et al. Testing critical slowing down as a bifurcation indicator in a low-dissipation dynamical system. *Phys. Rev. Lett.* **125**, 134102 (2020).
15. van Belzen, J. et al. Vegetation recovery in tidal marshes reveals critical slowing down under increased inundation. *Nat. Commun.* **8**, 15811 (2017).
16. Chen, L., Liu, R., Liu, Z.-P., Li, M. & Aihara, K. Detecting early-warning signals for sudden deterioration of complex diseases by dynamical network biomarkers. *Sci. Rep.* **2**, 342 (2012).
17. Tarazona A., Forment J. & Elena S. F. Identifying early warning signals for the sudden transition from mild to severe tobacco etch disease by dynamical network Biomarkers. *Viruses* **12**, 16 (2019).
18. Liu, R. et al. Hunt for the tipping point during endocrine resistance process in breast cancer by dynamic network biomarkers. *J. Mol. Cell Biol.* **11**, 649–664 (2019).
19. Koizumi, K. et al. Identifying pre-disease signals before metabolic syndrome in mice by dynamical network biomarkers. *Sci. Rep.* **9**, 8767 (2019).
20. Zhu, S., Gao, J., Ding, T., Xu, J. & Wu, M. Detecting early warning signal of influenza A disease using sample-specific dynamical network biomarkers. *Biomed. Res. Int.* **2018**, 6807059 (2018).
21. Vafaei, F. Using multi-objective optimization to identify dynamical network biomarkers as early-warning signals of complex diseases. *Sci. Rep.* **6**, 22023 (2016).
22. Chen, P., Liu, R., Chen, L. & Aihara, K. Identifying critical differentiation state of MCF-7 cells for breast cancer by dynamical network biomarkers. *Front. Genet.* **6**, 252 (2015).
23. Liu, R., Wang, X., Aihara, K. & Chen, L. Early diagnosis of complex diseases by molecular biomarkers, network biomarkers, and dynamical network biomarkers. *Med. Res. Rev.* **34**, 455–478 (2014).
24. Li, M., Zeng, T., Liu, R. & Chen, L. Detecting tissue-specific early warning signals for complex diseases based on dynamical network biomarkers: study of type 2 diabetes by cross-tissue analysis. *Brief. Bioinform.* **15**, 229–243 (2014).
25. Yang, B. et al. Dynamic network biomarker indicates pulmonary metastasis at the tipping point of hepatocellular carcinoma. *Nat. Commun.* **9**, 678 (2018).
26. Tanaka, A. & Tomiya, A. Detection of phase transition via convolutional neural networks. *J. Phys. Soc.* **86**, 063001 (2017).
27. Kashiwa, K., Kikuchi, Y. & Tomiya, A. Phase transition encoded in neural network. *Prog. Theor. Exp. Phys.* **2019**, 083A004 (2019).
28. Cole, A., Loges, G. J. & Shiu, G. Interpretable phase detection and classification with persistent homology. Preprint at <https://arxiv.org/abs/2012.00783> (2020).
29. Arnold, J. & Schäfer, F. Replacing neural networks by optimal analytical predictors for the detection of phase transitions. *Phys. Rev. X* **12**, 031044 (2022).
30. Chen, P., Chen, E., Chen, L., Zhou, X. J. & Liu, R. Detecting early-warning signals of influenza outbreak based on dynamic network marker. *J. Cell Mol. Med.* **23**, 395–404 (2019).
31. Brush, S. G. History of the Lenz-Ising model. *Rev. Mod. Phys.* **39**, 883 (1967).
32. McCoy, B. M. & Wu, T. T. *The two-dimensional Ising model*. Courier Corporation (2014).
33. Onsager, L. Crystal Statistics. I. A two-dimensional model with an order-disorder transition. *Phys. Rev.* **65**, 117–149 (1944).
34. Dill, K. A. & MacCallum, J. L. The protein-folding problem, 50 years on. *Science* **338**, 1042–1046 (2012).
35. Senior, A. W. et al. Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).
36. Lindorff-Larsen, K., Piana, S., Dror, R. O. & Shaw, D. E. How fast-folding proteins fold. *Science* **334**, 517–520 (2011).
37. Wang, E. C., Tao, P., Wang, J. & Xiao, Y. A novel folding pathway of the villin headpiece subdomain HP35. *Phys. Chem. Chem. Phys.* **21**, 18219–18226 (2019).
38. Tao, P. & Xiao, Y. Using the generalized Born surface area model to fold proteins yields more effective sampling while qualitatively preserving the folding landscape. *Phys. Rev. E* **101**, 062417 (2020).
39. Lei, H., Wu, C., Liu, H. & Duan, Y. Folding free-energy landscape of villin headpiece subdomain from molecular dynamics simulations. *Proc. Natl Acad. Sci. USA* **104**, 4925–4930 (2007).
40. Zoldak, G., Stigler, J., Pelz, B., Li, H. B. & Rief, M. Ultrafast folding kinetics and cooperativity of villin headpiece in single-molecule force spectroscopy. *Proc. Natl Acad. Sci. USA* **110**, 18156–18161 (2013).
41. Tao, P., Wang, E. & Xiao, Y. Pathway regulation mechanism revealed by cotranslational folding of villin headpiece subdomain HP35. *Phys. Rev. E* **101**, 052403 (2020).
42. Hao, S. et al. Protein folding mechanism revealed by single-molecule force spectroscopy experiments. *Biophys. Rep.* **7**, 399–412 (2021).
43. Dadgar, S. et al. Asynchronous remodeling is a driver of failed regeneration in Duchenne muscular dystrophy. *J. Cell Biol.* **207**, 139–158 (2014).
44. Arnold, L. et al. Inflammatory monocytes recruited after skeletal muscle injury switch into antiinflammatory macrophages to support myogenesis. *J. Exp. Med.* **204**, 1057–1069 (2007).
45. Chazaud, B. et al. Dual and beneficial roles of macrophages during skeletal muscle regeneration. *Exerc. Sport Sci. Rev.* **37**, 18–22 (2009).
46. McCroskery, S., Thomas, M., Maxwell, L., Sharma, M. & Kambadur, R. Myostatin negatively regulates satellite cell activation and self-renewal. *J. Mol. Cell Biol.* **162**, 1135–1147 (2003).
47. Chargé, S. B. & Rudnicki, M. A. Cellular and molecular regulation of muscle regeneration. *Physiol. Rev.* **84**, 209–238 (2004).
48. Relaix, F. & Zammit, P. S. Satellite cells are essential for skeletal muscle regeneration: the cell on the edge returns centre stage. *Development* **139**, 2845–2856 (2012).
49. Mi, H., Muruganujan, A., Ebert, D., Huang, X. & Thomas, P. D. PANTHER version 14: more genomes, a new PANTHER GO-slim and improvements in enrichment analysis tools. *Nucleic Acids Res.* **47**, D419–D426 (2019).
50. Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**, 27–30 (2000).
51. Liu, X., Wang, Y., Ji, H., Aihara, K. & Chen, L. Personalized characterization of diseases using sample-specific networks. *Nucleic Acids Res.* **44**, e164 (2016).
52. Liu, R. et al. Identifying critical transitions of complex diseases based on a single sample. *Bioinformatics* **30**, 1579–1586 (2014).
53. Liu, X. et al. Quantifying critical states of complex diseases using single-sample dynamic network biomarkers. *PLoS Comput. Biol.* **13**, e1005633 (2017).
54. Liu, X. et al. Detection for disease tipping points by landscape dynamic network biomarkers. *Natl Sci. Rev.* **6**, 775–785 (2019).
55. Yu, X. et al. Individual-specific edge-network analysis for disease prediction. *Nucleic Acids Res.* **45**, e170 (2017).
56. Zeng, T. et al. Big-data-based edge biomarkers: study on dynamical drug sensitivity and resistance in individuals. *Brief. Bioinform.* **17**, 576–592 (2016).
57. Pearl, J. Causal inference in statistics: an overview. *Stat. Surv.* **3**, 96–146 (2009).
58. Sugihara, G. et al. Detecting causality in complex ecosystems. *Science* **338**, 496–500 (2012).
59. Leng, S. et al. Partial cross mapping eliminates indirect causal influences. *Nat. Commun.* **11**, 2632 (2020).
60. Nair, V. & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 807–814 (2010).
61. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, (2015).
62. Yu, G., Wang, L.-G., Han, Y. & He, Q.-Y. clusterProfiler: an R package for comparing biological themes among gene clusters. *OMICS* **16**, 284–287 (2012).

Acknowledgements

This work is supported by the NSFC under Grant No. 11874162.

Author contributions

Y.X., C.Z., and P.T. designed research. P.T. and C.D. performed the research. P.T. analyzed data and made visualization, P.T., C.Z., and Y.X. wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42005-023-01429-0>.

Correspondence and requests for materials should be addressed to Yi Xiao or Chen Zeng.

Peer review information *Communications Physics* thanks Sebastian Herzog and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023