Supporting Information

Fan et al. 10.1073/pnas.1415846112

SI Text

Optimization Algorithm and Computation

The algorithm used to approximate the solution to the SIMPLE optimization problem in Eq. **2** of the main text is presented as Algorithm S1. The central idea of the algorithm is the following: Fixing detected changes in all but the jth observable, the optimization problem for the changes in the jth observable is given by

$$\arg\max_{K,\{\tau_i\}_{i=1}^{K}} \sum_{i=0}^{K} \hat{l}\left(Y_{\tau_i+1}, \ldots, Y_{\tau_{i+1}}\right) - \lambda \sum_{i=1}^{K} p_{\tau_i}, \qquad \textbf{[S1]}$$

where $Y_1, \ldots, Y_T$ denotes the data of the jth observable and where the penalty values $p_1, \ldots, p_{T-1}$ are given by $p_t = q(S_t \cup \{j\}) - q(S_t)$ for S_t the (possibly empty) set of all other observables that change at time t. Algorithm S1 is iterative, in each iteration solving the above single-observable problem for all J observables individually (lines 10–15), using penalty values computed from the changes determined in the preceding iteration (lines 35–47). Penalty values are initialized at the start of the algorithm (lines 1–6), and the iterations are deemed to have converged when the changes determined in an iteration for all observables match those of a preceding iteration (lines 29–31).

The univariate problem **[S1]** may be solved exactly, using dynamic programming (1). We use a method, recently discovered by Killick et al. (2) and based on the following proposition, to prune the dynamic programming computation:

Proposition 1. *Consider the optimization problem* **[S1]** *for univariate data* $Y_1, \ldots, Y_T$ *and penalty values* $p_1, \ldots, p_{T-1}$. *For each* $t < T$, *let* $F(t)$ *be the maximum value of the objective in* **[S1]** *for the truncated dataset* $Y_1, \ldots, Y_t$ *and penalties* $p_1, \ldots, p_{t-1}$. *If* $s < s'$ *are such that*

$$F(s) + \hat{l}(Y_{s+1}, \ldots, Y_{s'}) - \lambda p_s < F(s') - \lambda p_{s'},$$

then for any $t > s'$, *the change-point set that achieves the optimal value* $F(t)$ for data $Y_1, \ldots, Y_t$ *cannot have its last change point at time* s.

Proof:
We have

$$\begin{aligned} F(s) + \hat{l}(Y_{s+1}, \ldots, Y_t) - \lambda p_s &= F(s) + \sup_{\theta} l(Y_{s+1}, \ldots, Y_t; \theta) - \lambda p_s \\ &= F(s) + \sup_{\theta} [l(Y_{s+1}, \ldots, Y_{s'}; \theta) \\ &\quad + l(Y_{s+1}, \ldots, Y_t; \theta)] - \lambda p_s \\ &\le F(s) + \hat{l}(Y_{s+1}, \ldots, Y_{s'}) \\ &\quad + \hat{l}(Y_{s+1}, \ldots, Y_t) - \lambda p_s \\ &< F(s) + \hat{l}(Y_{s+1}, \ldots, Y_t) - \lambda p_s, \end{aligned}$$

where this last line follows by assumption. Hence the objective value if the last change point before t were at s is larger than that if the last change point were at s, so s cannot be the last change point before t. □

If the penalty values in the optimization problem **[S1]** were all equal, that is, $p_1 = \ldots = p_{T-1}$, then the problem **[S1]** is a common case of the multiple change-point problem considered in equation 1 of ref. 2 [with the cost function $\mathcal{C}$ being the negative log-likelihood, the penalty being $f(m) = m$, and $\beta = \lambda p_1 = \ldots = \lambda p_{T-1}$]. *Proposition 1* is a straightforward extension of theorem 3.1 in ref. 2 (with $K = 0$ for the negative log-likelihood cost) to the case where $p_1, \ldots, p_{T-1}$ are not all equal, and its proof is essentially the same. The dynamic programming procedure to solve problem **[S1]** operates by recursively computing $F(t)$ for $t = 1, \ldots, T$ and returning the change points that achieve the optimal value $F(T)$. For each t, $F(t)$ is computed as $F(t) = \max_s F(s) + \hat{l}(Y_{s+1}, \ldots, Y_t) - \lambda p_s$, where this maximum is taken over all times $s < t$ not prohibited by *Proposition 1* from being the last change point before time t. This procedure is represented by the "univariate_optimize" function in line 11 of Algorithm S1.

We augment this iterative approach with an additional heuristic step in each iteration, described in lines 16–28 of Algorithm S1. If change points in the observables have been identified at times $\tau_1, \ldots, \tau_K$ in a given iteration, this heuristic step potentially merges the change points at each time τ_i with those at the preceding change time τ_{i-1} or at the subsequent change time τ_{i+1} or adjusts the time of the change τ_i to a new time $\tau_i^* \in (\tau_{i-1}, \tau_{i+1})$, if either of these proposals increases the SIMPLE penalized-likelihood objective. This heuristic step allows the algorithm to adjust the time of a detected change involving multiple observables simultaneously, preventing the algorithm from reaching certain types of local minima of the objective function. We perform this step only after the number of change times has stabilized from one iteration to the next (lines 7 and 32–34).

Theorem 3.2 of ref. 2 provides conditions under which the pruned dynamic programming procedure to solve problem **[S1]** requires only $O(T)$ computational time. Essentially, these conditions require that there are $O(T)$ true change points and that after a true change point occurs at time s', the times $s < s'$ are pruned by *Proposition 1* from the list of possibilities when evaluating the location of the last change before any future time t. We note that, in multivariate change-point problems where the number of observables is large, certain observables may not exhibit any change points. In this case, the conditions of theorem 3.2 of ref. 2 would not hold for these observables, and the resulting runtime would be quadratic in T. We address this issue by introducing a small amount of randomness into the penalty values of problem **[S1]** [lines 3–4 and 44 of Algorithm S1], so that even if an observable exhibits no obvious change points, pruning according to *Proposition 1* can still occur because of differences between the penalty values p_s and $p_{s'}$. The resulting procedure no longer exactly solves the problem **[S1]** in the original nonrandomized penalties, but we observe that the difference in outcomes is small and the runtime of the resulting dynamic programming procedure is empirically reduced to $O(T)$ for all observables.

Thus, lines 10–15 of Algorithm S1 empirically require $O(JT)$ computational time. The nested loops in lines 18 and 20 require a total of $O(T)$ iterations regardless of the identified change times $\tau_1, \ldots, \tau_K$, and the value of v in line 21 for time s may be computed as an incremental update from its value at time $s-1$ in $O(J)$ time. Hence, lines 16–28 require $O(JT)$ computational time as well. Finally, each penalty value update in lines 39 and 41 may be performed in $O(1)$ time when amortized over the J observables for most reasonable penalty functions q (including those used in this paper), so that lines 35–47 also require $O(JT)$ computational time. Hence, in summary, each iteration of Algorithm S1 empirically requires $O(JT)$ runtime (i.e., linear in the size of the dataset). In all of our applications, the algorithm converged within 50 iterations.

In these calculations of runtime, we have implicitly assumed that the maximum log-likelihood $\hat{l}(Y_{j,s}, \ldots, Y_{j,t})$ may be computed

from $\hat{l}(Y_{j,s}, \ldots, Y_{j,t-1})$ in $O(1)$ time. This is true for the Gaussian likelihood model or any other model in the exponential family, due to the existence of finite-dimensional sufficient statistics. For the Laplace likelihood model, we may perform this update of the maximum log-likelihood value by maintaining a min-heap of data values greater than the sample median and a max-heap of data values less than the sample median. This requires $O(\log(t-s))$ computational time, so that each iteration of Algorithm S1 requires $O(JT \log T)$ instead of $O(JT)$ computational time.

The computations in lines 11 and 21 of Algorithm S1, which form the bulk of the total computation per iteration, are parallelizable by dividing them across up to J parallel processes, one for each observable. Our implementation of Algorithm S1 uses this parallelization scheme.

Observable and Parameter Choices for Application to Molecular Dynamics Trajectories

When analyzing molecular dynamics simulations of proteins, we typically use as inputs to SIMPLE time series of observables representing distances between pairs of atoms, as these observables constitute internal coordinates that are invariant to global rotations and translations of the protein molecules. Atomic position coordinates may alternatively be used if there is a suitable method of performing structural alignment of the proteins across different frames of the trajectory, although we caution that a poor choice of structural alignment may cause a conformational change involving one region of a protein to affect the aligned positions of atoms in a different region of the protein, making it difficult to localize such a conformational change to the correct spatial location on the protein molecule.

We consider the use of SIMPLE for two different types of analyses of molecular dynamics simulations of protein systems, exemplified by the applications to the folding trajectories and β_2AR-deactivation trajectories discussed in the main text. In the first, we seek to detect and characterize major structural changes of the protein backbone. In the second, we seek to discover more detailed and subtle conformational changes involving changes in chemical contacts formed by amino acid side chains as well as the protein backbone. In both types of analyses, we use a penalty function of the form $q(S) = (\sum_i |S \cap G_i|^{\alpha})^{\beta}$, where $\alpha, \beta \in (0,1)$ and where $\{G_i\}$ are groups of observables that we a priori believe are more likely to exhibit simultaneous change points. Setting β closer to 0 promotes the detection of changes as simultaneous between observables within the same observable groups G_i, and setting α closer to 0 promotes the detection of changes as simultaneous across all observables. Together, α, β, and $\{G_i\}$ control the shape of the penalty function q. Although both types of analyses of molecular dynamics simulations aim to detect conformational changes in the protein molecule, they are quite different in nature, and we suggest two different settings of these shape parameters for the two tasks. We note that previous literature on conformational change detection in protein simulations (e.g., refs. 3 and 4) has also explicitly differentiated these two types of analysis tasks and addressed them using different methodologies.

For the first task of characterizing major structural changes of the protein backbone, we use as observables the distances between all pairs of alpha-carbon atoms. We take $\alpha = \beta = 0.7$, and for each pair of residues A and B, an observable group G_i is constructed to consist of those observables corresponding to the distance between one alpha-carbon atom in a residue at most two away from A along the protein backbone and one at most two away from B. Hence, the term $|S \cap G_i|^{0.7}$ for each G_i promotes the simultaneous detection of changes in distances between two particular spatial regions of the molecule, and summing over G_i and exponentiating the sum by 0.7 promotes the simultaneous detection of changes globally across all analyzed observables. We applied SIMPLE, using the Laplace likelihood model with these parameters, to the folding and unfolding simulations of the 12 fast-folding protein domains reported in ref. 5, as well as to the synthetic BPTI trajectories in our comparison of performance with other methods.

For the second analysis task of discovering changes in chemical contacts involving both protein backbones and side chains, we use as observables distances between pairs of nonhydrogen atoms, transformed by a noisy sigmoid-type function to highlight changes of interatomic distances close to 4 Å (the approximate distance between two atoms in chemical contact):

$$y = \frac{1}{1 + (4/d)^5} + \mathrm{Unif}(0,0.1).$$

Here, d is a distance between two atoms in angstroms, and y is the corresponding observable used as input to SIMPLE. $\mathrm{Unif}(0,0.1)$ represents a uniform random variable between 0 and 0.1, independently drawn at each time point for each observable; we add this noise term to mask changes in distance distribution far from 4 Å. We compute one such observable for the distance between each pair of nonhydrogen atoms from two distinct residues in the protein molecule. For computational efficiency, we omit pairs of atoms that never come within 4 Å of each other in 50 evenly spaced frames between the start and end of the trajectory. We set $\beta = 0.7$, and for each pair of distinct residues A and B, a group G_i is constructed to consist of those observables corresponding to a distance between one atom in residue A and one atom in residue B. Thus, the term $|S \cap G_i|^{0.7}$ for each G_i promotes the simultaneous detection of changes in distances between atoms from two particular protein residues. As changes in chemical contacts are oftentimes very local in nature and it is important to understand the order in which specific contacts form and break, we set $\alpha \approx 1$ to preserve interpretability of results. We applied SIMPLE with these parameters, $\alpha = 0.99$, and the Laplace likelihood model to obtain the results for the β_2AR-deactivation trajectories discussed in the main text.

We note that these choices of observables and construction of observable groups G_i are motivated by a general understanding of the types of conformational changes that we aim to detect in the above two analysis tasks, rather than being particular to the specific protein systems presented as examples in the main text. When working with real trajectory data, we observed that the performance of our method is relatively robust to small variations in these parameters. We also demonstrate this property quantitatively in our synthetic dataset in Fig. S4. These settings of the penalty shape parameters thus represent our default suggestions for these two types of analysis tasks, and we have found that they are appropriate not only for the specific molecular dynamics simulation trajectories presented in the main text but also for simulations of other protein systems.

Regarding the choice of the parameter λ that controls the magnitude of the penalty values, we recommend applying SIMPLE multiple times, starting with a high initial value of λ and decreasing λ along a logarithmic scale. An observation inherent to molecular dynamics simulations of all except perhaps the simplest and smallest proteins is that conformational changes occur at many different timescales; the rare changes, which occur on longer timescales, are often of most interest. As λ is decreased, the number of change points detected by the method increases and the timescale of these changes shortens. We note that for artificial data exhibiting a single change, the difference in log-likelihood between a model with that single change and the null model with no change scales linearly with the number of observations before and after that change. Hence, varying λ on a logarithmic scale roughly corresponds to detecting changes on a logarithmic range of timescales, yielding a multiscale analysis of changes in the analyzed trajectory.

We performed this type of multiscale analysis for each of the 12 folding and unfolding trajectories and 12 β_2AR deactivation trajectories discussed here and in the main text. For clarity of the

figures, we have chosen to present results for simulations of the WW domain and Trp-cage in Figs. S1 and S2 corresponding to values of λ that yield ~100 total detected changes in each trajectory, although we note that many more conformational changes at shorter timescales were detected by SIMPLE in these trajectories for smaller settings of λ. Similarly, for the trajectory of β_2AR deactivation, changes detected at lower values of λ include side-chain contact formations and breakages other than the ones reported in the main text and in Fig. 3. We chose to depict the changes detected for a relatively large λ value ($\lambda \approx 1{,}300$) in Fig. 3 to highlight the longest-timescale changes, because these are most likely to be of biological interest and because these are also the changes that were detected consistently across all 12 different simulations of β_2AR deactivation (see below).

Details of Detected Changes in Sample Molecular Dynamics Trajectories

SIMPLE was applied to the longest simulation of each of the 12 fast-folding protein domains reported in ref. 5. The lengths of these simulations ranged between 104 μs and 1,052 μs. We present results for two of these protein domains, the WW domain variant GTT and the K8A mutant of the Trp-cage variant TC10b; SIMPLE produced similar results for the remaining proteins. All $J = 595$ pairwise distances between alpha-carbon atoms over $T = 65{,}076$ frames (at a 10-ns stride) were used as input to SIMPLE for the trajectory of the WW domain, and all $J = 190$ pairwise distances between alpha-carbon atoms over $T = 20{,}801$ frames (at a 10-ns stride) were used as input for the trajectory of Trp-cage. Figs. S1 and S2 illustrate all changes detected by SIMPLE at settings of the λ sensitivity parameter that yielded ~100 conformational change detections in each simulation. Several of these changes are highlighted in Fig. 2 of the main text. The intensity of red coloring for a residue in these figures increases with the number of detected changed distances involving the alpha-carbon atom of that residue. Ensembles of structures between detected changes in these figures are aligned using weighted alpha-carbon root-mean-square deviation (RMSD) to a representative frame within each ensemble.

The detected changes in Fig. S1 illustrate that the folding and unfolding transitions in this simulation of the WW domain variant GTT often occur through a sequence of intermediate conformations in which the two β hairpins fold and unfold sequentially. Similar behavior was reported in previous analysis of MD simulations of the WW domain variant FiP35 (6). Detected changes within the folded state in this trajectory reveal many instances in which the C-terminal and N-terminal tails of the protein partially unfold, whereas detected changes within the unfolded state reveal many instances in which partial folding occurs in one of the two β hairpins. A more detailed examination of the detected changes reveals that, among the changes occurring within the protein's unfolded state, eight correspond to the folding and unfolding of one or both β hairpins register shifted by a single residue from the native structure. The presence of such register-shifted metastable structures agrees with previous findings in ref. 7, although we note that our detected changes capture register shifts of the beta strands in both directions, whereas only shifts in a single direction were reported in ref. 7. The detected changes in Fig. S2 likewise highlight various recurrent metastable conformations of the Trp-cage variant TC10b. In particular, SIMPLE identifies the formation of the metastable structure displayed in Fig. 2*B* of the main text at least 20 times throughout this trajectory (within the unfolded portions of the trajectory).

For the second example in the main text, SIMPLE was applied to the 12 simulations of β_2AR deactivation reported in figure 3 of ref. 8. Fig. 3 of the main text displays data and results for simulation 8 (according to the numbering used in figure 3 and table S2 of ref. 8). Each trajectory was analyzed at a stride to yield $T \approx 1{,}000$ total frames, and in each trajectory, $J \approx 14{,}000$ distances between pairs of atoms in the protein backbone and residue side chains (selected to be those that come within 4 Å distance at some time in the trajectory and transformed as indicated in *SI Text*, *Observable and Parameter Choices for Application to Molecular Dynamics Trajectories*) were used as inputs to SIMPLE. In Fig. 3*B* of the main text, an ensemble of protein structures in a time window around each depicted change is shown in gray, aligned by all-atom RMSD to the active crystal structure. In these depicted changes, for each pair of residues that contains a pair of atoms corresponding to a changed observable, only one residue of the pair exhibits a significant motion relative to the rest of the protein structure; we display in red and blue only this residue. Also for clarity, we display only the protein backbone in the third change depicted in Fig. 3, although many of the time series detected as being involved in this change correspond to distances between side-chain atoms.

The four conformational changes discussed in the main text agree closely with the changes identified by manual analysis in ref. 8 as the most important conformational changes in the β_2AR structure during deactivation. In the trajectory for simulation 8 illustrated in Fig. 3, these changes were the first to be detected by SIMPLE in our multiscale analysis as we increased the method sensitivity by gradually decreasing λ. This was not the case for all 12 of the β_2AR deactivation trajectories analyzed, due to the presence of other conformational changes that occur in these trajectories. The four changes discussed in the main text were, however, the only changes consistently detected across all 12 trajectories at a common lower setting of the λ parameter, with the exception of an occasional melting of the tip of helix 1. Specifically, for a setting of $\lambda = 400$, 21 pairs of residues in β_2AR contained at least one pair of atoms corresponding to an observable with a change detected by SIMPLE in all 12 of the analyzed trajectories. Twenty of these 21 pairs correspond to the four changes discussed in the main text; the remaining pair corresponds to the melting of the tip of helix 1. We believe that this result serves both to confirm the significance of the changes identified in ref. 8 by manual analysis and to illustrate the utility of SIMPLE as a tool to automate the detection of conformational changes.

Details of Performance Comparison on Synthetic Data

Our method for constructing synthetic molecular dynamics trajectories was designed to meet the following requirements:

i) There are true changes in the distribution of the atomic positions at known times.
ii) The atomic position distributions are similar to those in real molecular dynamics trajectories.
iii) The changes are similar to conformational changes that are observed in real protein molecules.

To meet requirement *i*, we generated a synthetic sequence of conformational states, where the atomic position distributions change at each state transition. To enable comparison of SIMPLE with a Markov state model method (along with various change-point methods), we generated the sequence of conformational states according to a Markov chain. To meet requirement *ii*, we generated the data within each conformational state by sampling frames from a real molecular dynamics trajectory. We used the stationary bootstrap method of ref. 9 instead of sampling independent frames so that short-scale autocorrelations in real molecular dynamics trajectory data could be captured. We ensured that the average length of blocks of contiguous data drawn in the stationary bootstrap was much shorter than the average duration of a Markov state. To meet requirement *iii*, we based our synthetic states on kinetically distinct conformational states that were previously identified in a real molecular dynamics trajectory of the BPTI (6).

Fig. 4 of the main text presents results of various methods applied to synthetic trajectories with four state transitions each. Fig. S3 presents results on a single longer trajectory in which each Markov state is visited dozens of times. In both Fig. 4 and Fig. S3, we consider a change detected by a method to be a false detection if it is not within five synthetic frames of a true synthetic state transition or if there are multiple detections within five synthetic frames of the same synthetic state transition. (In the latter situation, all but one of the detections are considered false detections.) Likewise, we consider a true synthetic state transition to be undetected if no detected change falls within five synthetic frames of this true change. We note that this metric assesses only whether a change was detected at each given time and does not assess correctness of the subsets of observables determined to change at those times, as most of the methods we compare do not explicitly determine these subsets of observables. Qualitative examination of the output of SIMPLE, which does explicitly determine these observable subsets, indicates that the observables that change do correspond closely with those residues determined in ref. 6 to be the defining characteristics of each of the states in the real BPTI trajectory.

Details of our synthetic datasets and each of the applied methods are provided below.

Synthetic Data. Figure 4A of ref. 6 segments a 1-ms trajectory of the 58-residue protein BPTI into five states. We aligned all frames of this trajectory by alpha-carbon RMSD to the crystal structure (Protein Data Bank entry 5PTI). We then glued the segments of the trajectory in each of the blue, red, black, and green states of that figure, at a 10-ns stride, back to back, to obtain an empirical trajectory for each state. A sequence of synthetic blue, red, black, and green states was generated according to a four-state Markov chain, where the self-transition probability of each synthetic state was 0.995 and the transition probability of each state to each other state was 0.00167. Hence, the mean state duration for each of the synthetic states is 200 synthetic frames. The chain was terminated after the occurrence of 100 transitions between differing states. Data within each segment of the Markov chain having the same synthetic state were generated by sampling independent random blocks of contiguous frames from the empirical trajectory of the corresponding real state and tiling these blocks back to back. The length of each sampled block was geometrically distributed with mean length 10. Results for this synthetic trajectory are reported in Fig. S3. For Fig. 4 in the main text, 25 short trajectories were obtained by grouping the 100 state transitions of this synthetic trajectory into 25 blocks of 4 and taking the five segments of data around each block of 4 consecutive state transitions to be its own short trajectory. The short trajectories are analyzed independently; false change detections and missed true changes for each method are totaled over all 25 short trajectories to obtain the data in the figure.

SIMPLE. We applied SIMPLE to the observables of all 1,653 pairwise distances between alpha-carbon atoms of BPTI, using the Laplace likelihood model and the penalty function $q(S) = \left(\sum_i |S \cap G_i|^{0.7}\right)^{0.7}$ with G_i representing groups of observables corresponding to distances between the alpha-carbon atoms in two spatial regions on the protein molecule, as explained in *SI Text*, *Observable and Parameter Choices for Application to Molecular Dynamics Trajectories*. We varied the λ sensitivity parameter in SIMPLE to adjust the total number of detected changes.

We note that this choice of distance observables, observable groups, and penalty function $q(S)$ correspond to our default recommendations for detecting and characterizing major conformational changes of the protein backbone, as discussed in *SI Text*, *Observable and Parameter Choices for Application to Molecular Dynamics Trajectories*, and are the same as those used in our analysis of the folding–unfolding trajectories of the 12 fast-folding protein domains reported in ref. 5. Fig. S4 compares the results of applying SIMPLE with this penalty function, with the penalty functions $q(S) = \left(\sum_i |S \cap G_i|^{\alpha}\right)^{\beta}$ for a range of α and β values between 0.5 and 1, and with the penalty function $q(S) = |S|^{0.7}$ (which does not use observable groups). We recommend in the main text the use of this last penalty function for generic applications in the absence of prior information. We observe that, for this synthetic dataset, performance is similar over a large range of values of α and β and in fact is also similar between the penalty functions with and without observable groups, providing evidence for robustness of the SIMPLE method to some of these parameter choices.

On the other hand, Fig. S4 also displays the results of SIMPLE using the penalty function $q(S) = \left(\sum_i |S \cap G_i|^{0.7}\right)^{0.7}$ but with a Gaussian likelihood model, and we observe a degradation of detection accuracy in this case. The use of the Laplace likelihood rather than a Gaussian likelihood provides robustness against outlier data points that may arise in molecular dynamics simulations due to fluctuations including conformational changes at timescales shorter than the ones of interest.

Multisample Binary Segmentation. We adapted to our application domain the multisample binary segmentation (MSBS) method proposed by Zhang et al. (10) for the detection of DNA copy number variants. As the original method assumes that change points come in pairs, which is not true for our application domain, we used an analog of this method with the scan statistic

$$Z_t = \sum_{j=1}^{J} \hat{\sigma}_j^{-1} \frac{\left(S_{j,t} - t\overline{Y}_{j,T}\right)^2}{t(1-t/T)},$$

where $S_{j,t} = \sum_{i=1}^{t} Y_{j,i}$, $\overline{Y}_{j,T} = (1/T)\sum_{i=1}^{T} Y_{j,i}$, and $\hat{\sigma}_j$ is the sample SD of $Y_{j,1}, \ldots, Y_{j,T}$. This scan statistic for the case of a single observable ($J = 1$) is stated in ref. 11; compare also with equations 3 and 4 of ref. 10. We used the P-value approximation

$$\mathbb{P}\left(\max_{\varepsilon T < t < (1-\varepsilon)T} Z_t > b^2\right) \sim b^2\left(1 - \frac{J-1}{b^2}\right) f_J\left(b^2\right) \times \int_{\varepsilon}^{1-\varepsilon} \frac{1}{u(1-u)} \nu\left(\frac{b\left(1-(J-1)/b^2\right)}{(Tu(1-u))^{1/2}}\right) du, \quad \textbf{[S2]}$$

where f_J is the density of the $\chi^2_{(J)}$ distribution and where

$$\nu(x) \approx \frac{\left[(2/x)(\Phi(x/2) - 1/2)\right]}{\left[(x/2)\Phi(x/2) + \phi(x/2)\right]} \quad \textbf{[S3]}$$

is the overshoot function described in ref. 10. This P-value approximation for the case of a single observable is stated as equation **6** in ref. 11, and we believe the above generalization to the multivariate setting may be derived in the same manner as the derivation of equation 5 in ref. 10. We have confirmed through Monte Carlo simulations that this P-value approximation is accurate, assuming normally distributed observables, for ε not too small and for large T. The MSBS algorithm is then given by the following [compare with "Algorithm (Multisample Circular Binary Segmentation)" in ref. 10]:

1) Initialize $T_1 = 1$, $T_2 = T$.
2) Compute $Z_{\max} = \max_{T_1 + \varepsilon(T_2 - T_1) < t < T_1 + (1-\varepsilon)(T_2 - T_1)} Z_t$. Let t^* be the maximizing index.
3) If the P value of $Z_{\max}$, as computed using Eqs. **S2** and **S3**, is less than α, then for each $(u, v) \in \{(T_1, t^*), (t^* + 1, T_2)\}$, do:

a) Let $S = \{j : \hat{\sigma}_j^{-1}((S_{j,t} - t\overline{Y}_{j,T})^2 / t(1-t/T)) > \delta\}$. For all $t = u, \ldots, v$, if $j \in S$, let $Y'_{j,u:v} = Y_{j,u:v} - \overline{Y}_{j,u:v}$, and otherwise let $Y'_{j,u:v} = Y_{j,u:v} - \overline{Y}_{j,T_1:T_2}$.
b) Repeat steps *2* and *3* for $T_1 = u$, $T_2 = v$, and the newly normalized $Y'_{u:v}$.

We applied this algorithm to the same 1,653 distance observables used by SIMPLE. In this algorithm, we chose $\varepsilon = \min(10/(T_2 - T_1), 0.05)$, and we varied the *P*-value threshold α to adjust the total number of detected changes. We experimented with various settings of the parameter δ between 0 and 200; the results in Fig. 4 and Fig. S3 correspond to a setting of $\delta = 100$. We found that results were similar for all values of δ between 0 and 100, but worse for $\delta > 100$.

Group-Fused LASSO. We applied the group-fused LASSO (gfLASSO) change-point detection method proposed by Vert and Bleakley in ref. 12 to the same 1,653 distance observables used by SIMPLE, using the MATLAB implementation of this method released by its authors. Direct application of group-fused LASSO yielded large numbers of false change detections, so we followed the recommendation of the authors in ref. 12 to first use the group-fused LARS algorithm to select a set of candidate change times and to then prune this candidate set using a dynamic programming postprocessing step, minimizing the squared deviations of the observables from their piecewise constant signal means. For the dataset consisting of a single long trajectory, reported in Fig. S3, we used the group-fused LARS algorithm to select 500 candidate changes and applied dynamic programming to obtain the final set of k detected changes for varying values of k. For the dataset consisting of 25 short trajectories, reported in Fig. 4, we used the group-fused LARS algorithm to select 20 candidate changes for each trajectory, and we applied dynamic programming and the model selection strategy proposed in ref. 12 to select the final set of detected changes for each trajectory. We varied the threshold parameter in this model selection strategy to control the total number of detected changes across all trajectories.

Markov State Model. We applied the MSMBuilder2 algorithm in ref. 13, using the implementation provided by the authors, to construct Markov state models for our synthetic trajectories. We specified the correct number of states to use for each trajectory, such that a 4-macrostate model was constructed for the dataset consisting of a single long trajectory, reported in Fig. S3, and either a 3-macrostate or 4-macrostate model was constructed for each trajectory in the dataset consisting of 25 short trajectories, reported in Fig. 4. (The number of macrostates used in each short trajectory matched the actual number of true synthetic states present in that particular trajectory.) MSMBuilder2 identified 321 total macrostate transitions in the long synthetic trajectory and 418 total macrostate transitions across the 25 short trajectories, in both cases many more than the 100 total true state transitions. To yield a smaller number of detected changes and obtain the performance plots in Fig. 4 and Fig. S3, we filtered out spurious macrostate transitions by replacing the macrostate label of each frame with the modal label in a window centered around that frame, varying the width of this rolling window filter to adjust the number of total detected changes.

We note that our synthetic datasets give the MSM approach two artificial advantages. First, the synthetic trajectories transition among a few recurrent states according to an exact Markov process, a condition not guaranteed to hold for real MD trajectories. Second, we specified the correct number of states to use for each trajectory, rather than requiring the MSM software to discover it. In general, choosing the number of states to use when constructing an MSM may be difficult when analyzing real MD trajectories.

The Markov state models were constructed using the positions of the 58 alpha-carbon atoms in the BPTI molecule. Initial clustering into microstates was performed using 50 iterations of the hybrid *k*-medoids method suggested by the authors of ref. 13. We experimented with various settings of the intercluster distance parameter between $d = 0.18$ and $d = 0.08$, which yielded between 34 and 3,785 microstates in the long synthetic trajectory. Results displayed in Fig. 4 and Fig. S3 are for the setting $d = 0.1$, corresponding to the best performance for both datasets. The Markov transition matrix was estimated by the method described in ref. 13 for the microstates in the maximal ergodic component of the empirical microstate transition graph, using a Markov timescale parameter of $l = 1$ synthetic frame. In the long synthetic trajectory, all microstates belonged to a single ergodic component, and a 4-macrostate model was constructed from the microstate model, using PCCA+. In some of the short trajectories, the microstates were partitioned into more than one ergodic component. We reassigned each frame belonging to a microstate not in the largest ergodic component to the closest microstate belonging to this largest component, as measured by RMSD distance to the cluster center, after estimation of the microstate Markov transition matrix and before aggregation into macrostates using PCCA+.

Univariate RMSD and Principal Components Analysis. We implemented and applied the univariate change-point method described in ref. 14, using the Gaussian process model specified in appendix A of ref. 14 that was suggested by the authors for use in molecular dynamics analysis. We applied this method to two observables, the first being the alpha-carbon RMSD from the BPTI crystral structure, and the second being the coefficient of the largest principal component in a principal components analysis (PCA) of the x, y, and z coordinates of the 58 alpha-carbon atoms after RMSD alignment of all frames of the trajectory to the BPTI crystal structure. For the dataset consisting of 25 short synthetic trajectories, the PCA was performed independently on each trajectory. The algorithm of ref. 14 employs a binary segmentation procedure based on comparing the Bayes factor between the no-change and single-change models to a given threshold. We adjust the value of this threshold to control the total number of detected changes.

Discussion and Proof of *Theorem* on Asymptotic Consistency

A *Theorem* is stated in the main text regarding asymptotic consistency of the SIMPLE method, for the asymptotic regime in which the number of time points T increases while the numbers of observables and change points remain fixed. We prove this theorem here to provide theoretical support for formulating the change-point problem as the optimization problem given by Eq. **2** in the main text. We note that the *Theorem* pertains to the global optimal solution of this optimization problem, rather than to the particular solution returned by Algorithm S1. Thus, the success of the SIMPLE method rests on both this consistency result for the global optimal solution and the efficacy of the optimization algorithm used.

The asymptotic consistency *Theorem* stated in the main text may be compared with the main proposition in ref. 15, which establishes consistency for the detection of change points in univariate data, using penalized maximum likelihood with the univariate normal-likelihood model, and the main theorem in ref. 16, which extends this result to any single-parameter exponential family model. In particular, our *Theorem* for the special case of a single observable ($J = 1$) is similar to those in refs. 15 and 16 but for the Laplace likelihood model with unknown location and scale. For concreteness, we focus on this Laplace likelihood model, as this is the model we use throughout the applications in this paper. Similar ideas may be used to establish the result for the Gaussian likelihood model.

Conditions *i* and *ii* of the *Theorem* pertain to the true distribution of data between change points. Unlike in the results of refs. 15 and 16, we do not assume that the true data distributions are from the Laplace model, but only that they have bounded

density and tails that decay exponentially or faster. It is known that, under appropriate regularity conditions, a maximum-likelihood estimate in a misspecified model converges to the distribution in the model that is closest in Kullback-Leibler (KL) divergence to the true data distribution (17, 18). Condition *ii* ensures that these KL-divergence projections of the true data distributions onto the Laplace model are distinguishable, which is sufficient to guarantee consistency. Likewise, we do not assume that the observations in different observables are independent. The flexibility of these conditions is important from a practical standpoint, as the assumption of independent Laplace-distributed observables may be unrealistic in many practical situations.

Condition *iii* implies that the penalty function is strictly increasing but subadditive. Our proof of the *Theorem* in fact requires this subadditivity property to hold only for subsets of the true change sets S_i^0. One may note that if the subadditivity property were replaced by an additivity condition, $q(S \sqcup T) = q(S) + q(T)$, then the SIMPLE optimization problem would exactly decouple into J independent univariate optimization problems, and the method would not detect simultaneous change points in the observables as simultaneous even in the asymptotic limit.

Condition *iv* specifies that the magnitude of the penalty must grow asymptotically at a rate of at least $\sim(\log T)^2$. This requirement is stronger than those in refs. 15 and 16, which specify that the penalty must be at least $\sim\log T$. This stronger requirement is used in the first part of our *Proof* of *Lemma 3*, to ensure that we do not detect spurious changes around very short segments of data. Lee avoids this problem in ref. 16 by imposing an additional condition that a lower bound $r(T)$ on the minimum change-time separation is known and grows with T, but this introduces a new parameter that is problematic to specify in practice. We choose to remove this condition at the expense of strengthening the condition on the penalty. Our proof of the *Theorem* indicates that the conditions $\min_i|\tau_{i+1}^0 - \tau_i^0| \sim T$ and $\lambda(T) = o(T)$ may be replaced with the single weaker condition $\lambda(T)/\min_i|\tau_{i+1}^0 - \tau_i^0| \to 0$, and the conclusion of the *Theorem* may be strengthened to guarantee that, with probability approaching 1, $|\hat{\tau}_i - \tau_i^0| < C'\lambda(T)$ for all i and some sufficiently large constant C'.

That we constrain the solution space of the SIMPLE optimization problem so that $|\tau_{i+1} - \tau_i| \geq 2$ for all i is necessary, or else detecting two changes around a single data point would yield an infinite value of the maximum likelihood in our two-parameter model. We believe that the constraint $K \leq K_{\max}$ for some known $K_{\max}$ in condition *v* may be weakened, although our method of proof relies on this assumption. In practice, we find that our algorithm detects a reasonably small number of changes without imposing such a constraint.

The following proof of the *Theorem* follows the high-level outline of the proof of the main theorem in ref. 16; the bulk of the additional work lies in establishing *Lemma 3* below, which is a result analogous to lemma 1 in ref. 15 and lemma 2 in ref. 16, and in generalizing the final step of the proof to the setting of multiple observables with penalties of the form specified in Eq. **2** of the main text.

Lemma 1. *Let $Y_1, \ldots, Y_n$ be i.i.d. random variables with distribution function F, density f,* and *sample median $\hat{\mu}_n$. Suppose F has a unique median μ, and f satisfies $m \leq f(x) \leq M$ for all $x \in [\mu - \delta, \mu + \delta]$ and some $m > 0$, $M < \infty$, and $\delta > 0$. Then for any $\lambda > 0$,*

$$\mathbb{P}\left(\sum_{i=1}^n |Y_i - \mu| - |Y_i - \hat{\mu}_n| > \lambda\right) \leq \max\left\{4e^{-(M\lambda/8)} + 2e^{-\left(m^2\lambda/2M\right)}, 4e^{-\left(\delta^2M^2n/2\right)} + 2e^{-2\delta^2m^2n}\right\}.$$

Proof:

Let $\#(n) = \#\{Y_i \in [\mu, \hat{\mu}_n]\}$ if $\mu \leq \hat{\mu}_n$ and $\#\{Y_i \in [\hat{\mu}_n, \mu]\}$ otherwise. Note that

$$\sum_{i=1}^n |Y_i - \mu| - |Y_i - \hat{\mu}_n| \leq 2\#(n)|\hat{\mu}_n - \mu|,$$

as contributions to this sum from data values more than $\#(n)$ away from $\hat{\mu}_n$ cancel out in pairs, and the remaining data values contribute at most $|\hat{\mu}_n - \mu|$ each. Then, for any $\lambda > 0$ and $L > 0$,

$$\begin{aligned}\mathbb{P}\left(\sum_{i=1}^n |Y_i - \mu| - |Y_i - \hat{\mu}_n| > \lambda\right) &\leq \mathbb{P}(2\#(n)|\hat{\mu}_n - \mu| > \lambda) \\ &\leq \mathbb{P}\left(\#(n) > \frac{\lambda}{2L}, |\hat{\mu}_n - \mu| < L\right) + \mathbb{P}(|\hat{\mu}_n - \mu| \geq L).\end{aligned} \quad \textbf{[S4]}$$

Let $F_n(x) = (1/n)\sum_{i=1}^n 1_{\{Y_i \leq x\}}$, the empirical distribution function. To bound the first term of **[S4]**, set $L = \min\{\sqrt{\lambda/4Mn}, \delta\}$. Then

$$\begin{aligned}&\mathbb{P}\left(\#(n) > \frac{\lambda}{2L}, |\hat{\mu}_n - \mu| < L\right) \\ &\leq \mathbb{P}\left(\sup_{x \in [\mu - L, \mu]} F_n(x + L) - F_n(x) > \frac{\lambda}{2nL}\right) \\ &\leq \mathbb{P}\left(\sup_{x \in [\mu - L, \mu]} F(x) - F_n(x) > \frac{\lambda}{4nL} - \frac{LM}{2}\right) \\ &\quad + \mathbb{P}\left(\sup_{x \in [\mu - L, \mu]} F_n(x + L) - F(x) > \frac{\lambda}{4nL} + \frac{LM}{2}\right) \\ &\leq \mathbb{P}\left(\sup_{x \in [\mu - L, \mu]} F(x) - F_n(x) > \frac{\lambda}{4nL} - \frac{LM}{2}\right) \\ &\quad + \mathbb{P}\left(\sup_{x \in [\mu - L, \mu]} F_n(x + L) - F(x + L) > \frac{\lambda}{4nL} - \frac{LM}{2}\right) \\ &\leq 2\mathbb{P}\left(\sup_{x \in \mathbb{R}} |F_n(x) - F(x)| > \frac{\lambda}{4nL} - \frac{LM}{2}\right),\end{aligned}$$

where the third inequality follows from $F(x + L) \leq F(x) + LM$ for $L \leq \delta$. If $L = \sqrt{\lambda/4Mn} \leq \delta$, then $\lambda/4nL - LM/2 = \sqrt{M\lambda/16n}$, so the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality (19) gives

$$\mathbb{P}\left(\#(n) > \frac{\lambda}{2L}, |\hat{\mu}_n - \mu| < L\right) \leq 4e^{-(M\lambda/8)}. \quad \textbf{[S5]}$$

If $L = \delta \leq \sqrt{\lambda/4Mn}$, then $\lambda/4nL - LM/2 \geq \delta M/2$, so the DKW inequality gives

$$\mathbb{P}\left(\#(n) > \frac{\lambda}{2L}, |\hat{\mu}_n - \mu| < L\right) \leq 4e^{-\left(\delta^2M^2n/2\right)}. \quad \textbf{[S6]}$$

To bound the second term of **[S4]**,

$$\begin{aligned}\mathbb{P}(\hat{\mu}_n - \mu \geq L) &\leq \mathbb{P}\left(\sum_{i=1}^n \mathbb{1}_{\{Y_i \geq L + \mu\}} \geq \frac{n}{2}\right) \\ &\leq \exp\left(-2n\left(\frac{1}{2} - (1 - F(L + \mu))\right)^2\right) \leq e^{-2nm^2L^2},\end{aligned}$$

where the second inequality follows from Hoeffding's inequality (20) and the third from the bound $F(L + \mu) \geq F(\mu) + mL = 1/2 + mL$ for $L \leq \delta$. The bound $\mathbb{P}(\mu - \hat{\mu}_n \leq L) \leq e^{-2nm^2L^2}$ may be

obtained similarly, and the result follows upon substituting the two possible values of L and combining with [**S5**] and [**S6**] in [**S4**]. □

Lemma 2. *Let $Y_1, \ldots, Y_n$ be i.i.d. random variables with distribution function F and density f satisfying the conditions of Lemma 1. Suppose Y_i has exponential decay; that is,* there *exist $A, B > 0$ such that $\mathbb{P}(|Y_i| > x) \le Ae^{-Bx}$ for all $x > 0$. Let F have median μ, let $v = \mathbb{E}[|Y_i - \mu|]$ be the mean absolute deviation to median, let $Y_1, \ldots, Y_n$ have sample median $\hat{\mu}_n$, and let $\hat{v}_n = \sum_{i=1}^n |Y_i - \hat{\mu}_n|/n$ be the sample mean absolute deviation to median. Then, with m, M, and δ as defined in Lemma 1, there exist constants $C, \sigma > 0$ independent of n such that for all $\lambda > 0$ and $n \ge 2$,*

$$\begin{aligned}&\mathbb{P}(n|\hat{v}_n - v| > \lambda)\\ &\quad\le \max\left\{4e^{-(M\lambda/16)} + 2e^{-\left(m^2\lambda/4M\right)}, 4e^{-\left(\delta^2M^2n/2\right)} + 2e^{-2\delta^2m^2n}\right\}\\ &\qquad+ \max\left\{2e^{-\left(\lambda^2/16n\sigma^2\right)}, 2e^{-Cn}\right\}.\end{aligned}$$

Proof:
Let $Z_i = |Y_i - \mu| - v$. As Y_i has exponential decay, so does Z_i; let $A, B > 0$ be such that $\mathbb{P}(|Z_i| > x) \le Ae^{-Bx}$. Then for any integer $k > 1$,

$$\mathbb{E}\left[|Z_i|^k\right] = \int_0^\infty \mathbb{P}\left(|Z_i|^k > t\right)dt \le \int_0^\infty Ae^{-Bt^{1/k}}dt = AB^{-k}k!.$$

Hence, letting $\sigma^2 = \mathbb{E}[Z_i^2]$, there exists some $L > 0$ such that $\mathbb{E}[|Z_i|^k] \le \frac{\sigma^2}{2}L^{k-2}k!$ for all $k > 1$, and Bernstein's inequality (21) gives

$$\mathbb{P}\left(\left|\sum_{i=1}^n Z_i\right| > \lambda\right) \le 2e^{-\left(\lambda^2/4n\sigma^2\right)}$$

for all $\lambda \le n\sigma^2/L$. For $\lambda \ge n\sigma^2/L$,

$$\mathbb{P}\left(\left|\sum_{i=1}^n Z_i\right| > \lambda\right) \le \mathbb{P}\left(\left|\sum_{i=1}^n Z_i\right| > \frac{n\sigma^2}{L}\right) \le 2e^{-\left(n\sigma^2/4L^2\right)}.$$

Then

$$\begin{aligned}\mathbb{P}(n|\hat{v}_n - v| > \lambda) &= \mathbb{P}\left(\left|\sum_{i=1}^n |Y_i - \hat{\mu}_n| - nv\right| > \lambda\right)\\ &\le \mathbb{P}\left(\sum_{i=1}^n |Y_i - \mu| - |Y_i - \hat{\mu}_n| > \frac{\lambda}{2}\right) + \mathbb{P}\left(\left|\sum_{i=1}^n Z_i\right| > \frac{\lambda}{2}\right)\\ &\le \mathbb{P}\left(\sum_{i=1}^n |Y_i - \mu| - |Y_i - \hat{\mu}_n| > \frac{\lambda}{2}\right)\\ &\quad+ \max\left\{2e^{-\left(\lambda^2/16n\sigma^2\right)}, 2e^{-\left(n\sigma^2/4L^2\right)}\right\},\end{aligned}$$

and the result follows from applying *Lemma 1*. □

Lemma 3. *Suppose $Y_1, \ldots, Y_n$ are i.i.d. with distribution function F and density f. Suppose F has median μ_0 and mean absolute deviation to median $v_0 = \mathbb{E}[|Y_i - \mu_0|]$, $f(x) \le M$ for all $x \in \mathbb{R}$ and some $M < \infty$, $f(x) \ge m > 0$ for all $x \in [\mu_0 - \delta, \mu_0 + \delta]$ and some $m, \delta > 0$, and Y_i has exponential decay. Let $l_{\mu,v}(Y_i, \ldots, Y_j) = -(j - i + 1)\log 2v - \sum_{k=i}^j |Y_k - \mu|/v$, the log-likelihood of the Laplace(μ, v) distribution with data $Y_i, \ldots, Y_j$. Let $\hat{l}(Y_i, \ldots, Y_j)$ denote the maximum of this log-likelihood; that is, $\hat{l}(Y_i, \ldots, Y_j) = l_{\hat{\mu},\hat{v}}(Y_i, \ldots, Y_j)$, where $\hat{\mu}$ is the sample median of $Y_i, \ldots, Y_j$ and $\hat{v} = \sum_{k=i}^j |Y_k - \hat{\mu}|/(j - i + 1)$ is the sample mean absolute deviation to median. Then there exists a constant $C > 0$ (depending on M, m, δ, and the rate of exponential decay of Y_i) such that*

$$\lim_{n\to\infty} \mathbb{P}\left(\max_{1\le i<j\le n} \hat{l}(Y_i, \ldots, Y_j) - l_{\mu_0,v_0}(Y_i, \ldots, Y_j) > C(\log n)^2\right) = 0.$$

Proof:
Letting $\hat{\mu}_r$ and $\hat{v}_r$ denote the sample median and sample mean absolute deviation to median of $Y_1, \ldots, Y_r$,

$$\begin{aligned}&\mathbb{P}\left(\max_{1\le i<j\le n} \hat{l}(Y_i, \ldots, Y_j) - l_{\mu_0,v_0}(Y_i, \ldots, Y_j) > C(\log n)^2\right)\\ &\le \sum_{r=1}^{n-1} \sum_{\substack{1\le i<j\le n\\ j-i=r}} \mathbb{P}\left(\hat{l}(Y_i, \ldots, Y_j) - l_{\mu_0,v_0}(Y_i, \ldots, Y_j) > C(\log n)^2\right)\\ &= \sum_{r=2}^n (n - r + 1)\mathbb{P}\left(-r\log 2\hat{v}_r - \frac{\sum_{i=1}^r |Y_i - \hat{\mu}_r|}{\hat{v}_r} + r\log 2v_0\right.\\ &\qquad\left.+ \frac{\sum_{i=1}^r |Y_i - \mu_0|}{v_0} > C(\log n)^2\right)\\ &\le \sum_{r=2}^n n\mathbb{P}\left(-r\log\hat{v}_r - r + r\log v_0 + \frac{\sum_{i=1}^r |Y_i - \mu_0|}{v_0} > C(\log n)^2\right).\end{aligned}$$

[S7]

For any $R > 0$, the sum in [**S7**] over $r \le R\log n$ and over $r > R\log n$ may be bounded separately. The sum over $r \le R\log n$ may be bounded by the conditions on F and f. Indeed, as Y_i has exponential decay, there exist $A, B > 0$ such that $\mathbb{P}(|Y_i - \mu_0|/v_0 > x) \le Ae^{-Bx}$ for all $x > 0$. Then

$$\begin{aligned}&\mathbb{P}\left(-r\log\hat{v}_r - r + r\log v_0 + \frac{\sum_{i=1}^r |Y_i - \mu_0|}{v_0} > C(\log n)^2\right)\\ &\le \mathbb{P}\left(-r\log\hat{v}_r - r + r\log v_0 > \frac{C(\log n)^2}{2}\right)\\ &\quad+ \sum_{i=1}^r \mathbb{P}\left(\frac{|Y_i - \mu_0|}{v_0} > \frac{C(\log n)^2}{2r}\right)\\ &\le \mathbb{P}\left(\hat{v}_r < e^{-\left(C(\log n)^2/2r\right)-1+\log v_0}\right) + rAe^{-\left(BC(\log n)^2/2r\right)}\\ &\le \mathbb{P}\left(|Y_1 - Y_2| < re^{-\left(C(\log n)^2/2r\right)-1+\log v_0}\right) + rAe^{-\left(BC(\log n)^2/2r\right)}\\ &\le 2rMe^{-\left(C(\log n)^2/2r\right)-1+\log v_0} + rAe^{-\left(BC(\log n)^2/2r\right)},\end{aligned}$$

where the penultimate inequality follows from the observation that $\sum_{i=1}^r |Y_i - \hat{\mu}_r| \ge |Y_1 - Y_2|$ for $r \ge 2$, and the last inequality follows from the fact that the density of $Y_1 - Y_2$, a convolution of f with itself, is bounded by M. For any $R > 0$, there exist $N, C > 0$ such that $2rMe^{-\left(C(\log n)^2/2r\right)-1+\log v_0} + rAe^{-\left(BC(\log n)^2/2r\right)} < 1/n^2$ for all $r \le R\log n$ and $n \ge N$. Hence, for any $R > 0$, there exists $C > 0$ such that

$$\begin{aligned}&\lim_{n\to\infty} \sum_{r=2}^{\lfloor R\log n\rfloor} n\mathbb{P}\left(-r\log\hat{v}_r - r + r\log v_0 + \frac{\sum_{i=1}^r |Y_i - \mu_0|}{v_0} > C(\log n)^2\right)\\ &\quad\le \lim_{n\to\infty} \frac{Rn\log n}{n^2} = 0.\end{aligned}$$

The sum in [S7] over $r \geq R\log n$ may be bounded by the large deviation inequalities established in *Lemmas 1* and *2*. Indeed,

$$\begin{aligned}
&\mathbb{P}\left(-r\log\hat{v}_r - r + r\log v_0 + \frac{\sum_{i=1}^r |Y_i-\mu_0|}{v_0} > C(\log n)^2\right)\\
&\quad\leq \mathbb{P}\left(-r\log\hat{v}_r - r + r\log v_0 + \frac{\sum_{i=1}^r |Y_i-\hat{\mu}_r|}{v_0} > \frac{C(\log n)^2}{2}\right)\\
&\qquad + \mathbb{P}\left(\frac{\sum_{i=1}^r |Y_i-\mu_0| - |Y_i-\hat{\mu}_r|}{v_0} > \frac{C(\log n)^2}{2}\right)\\
&\quad= \mathbb{P}\left(-r\log\hat{v}_r - r + r\log v_0 + \frac{r\hat{v}_r}{v_0} > \frac{C(\log n)^2}{2}\right)\\
&\qquad + \mathbb{P}\left(\sum_{i=1}^r |Y_i-\mu_0| - |Y_i-\hat{\mu}_r| > \frac{v_0 C(\log n)^2}{2}\right)\\
&\quad\leq \mathbb{P}\left(r(\hat{v}_r - v_0)^2 > \frac{v_0^2 C(\log n)^2}{4}\right) + \mathbb{P}\left(|\hat{v}_r - v_0| > \frac{v_0}{2}\right)\\
&\qquad + \mathbb{P}\left(\sum_{i=1}^r |Y_i-\mu_0| - |Y_i-\hat{\mu}_r| > \frac{v_0 C(\log n)^2}{2}\right).
\end{aligned}$$

To see the last inequality above, let $g(v) = rv/v_0 - r\log v$ and note that, for some $\tilde{v}$ between $\hat{v}_r$ and v_0,

$$\begin{aligned}
-r\log\hat{v}_r - r + r\log v_0 + \frac{r\hat{v}_r}{v_0} &= g(\hat{v}_r) - g(v_0) = (\hat{v}_r - v_0)g'(\tilde{v})\\
&\leq \frac{r(\hat{v}_r - v_0)^2}{v_0\tilde{v}} \leq \frac{2r(\hat{v} - v_0)^2}{v_0^2}
\end{aligned}$$

if $|\hat{v}_r - v_0| \leq v_0/2$. Applying *Lemmas 1* and *2*, there exist some constants $C_1, C_2, C_3, c_1, c_2, c_3 > 0$, not depending on r and n, such that

$$\begin{aligned}
&\mathbb{P}\left(-r\log\hat{v}_r - r + r\log v_0 + \frac{\sum_{i=1}^r |Y_i-\mu_0|}{v_0} > C(\log n)^2\right)\\
&\quad\leq C_1 e^{-c_1 C(\log n)^2} + C_2 e^{-c_2\sqrt{rC(\log n)^2}} + C_3 e^{-c_3 r}\\
&\quad\leq C_1 e^{-c_1 C(\log n)^2} + C_2 e^{-c_2 C(\log n)^2} + C_2 e^{-c_2 r} + C_3 e^{-c_3 r}.
\end{aligned}$$

For any $R > \max\{1/c_2, 1/c_3\}$ and $C > 0$,

$$\sum_{r=\lfloor R\log n\rfloor+1}^n nC_1 e^{-c_1 C(\log n)^2} \leq n^2 C_1 n^{-c_1 C\log n} \to 0,$$

$$\sum_{r=\lfloor R\log n\rfloor+1}^n nC_2 e^{-c_2 C(\log n)^2} \leq n^2 C_2 n^{-c_2 C\log n} \to 0,$$

$$\sum_{r=\lfloor R\log n\rfloor+1}^n nC_2 e^{-c_2 r} \leq nC_2 \cdot \frac{e^{-c_2 R\log n}}{1-e^{-c_2}} \to 0,$$

$$\sum_{r=\lfloor R\log n\rfloor+1}^n nC_3 e^{-c_3 r} \leq nC_3 \cdot \frac{e^{-c_3 R\log n}}{1-e^{-c_3}} \to 0.$$

Putting this together gives the desired result. □

Lemma 4. *Let $Y_1, \ldots, Y_n$ be i.i.d. random variables with distribution function F and density f satisfying the conditions of Lemma 2. Let μ_0 and v_0 be the median and mean absolute deviation to median of F, and let $l_{\mu,v}$ be the log-likelihood function defined in Lemma 3. For any $\varepsilon > 0$, let $A_\varepsilon = \{(\mu, v) \in \mathbb{R}\times\mathbb{R}^+ : |\mu-\mu_0| < \varepsilon, |v-v_0| < \varepsilon\}$. Then, for any $\varepsilon > 0$, there exists $h > 0$ such that*

$$\lim_{n\to\infty} \mathbb{P}\left(l_{\mu_0,v_0}(Y_1, \ldots, Y_n) - \sup_{(\mu,v)\in(\mathbb{R}\times\mathbb{R}^+)\setminus A_\varepsilon} l_{\mu,v}(Y_1, \ldots, Y_n) > hn\right) = 1.$$

Proof:
For any $\delta > 0$,

$$\begin{aligned}
&\mathbb{P}\left(\sup_{(\mu,v)\in(\mathbb{R}\times\mathbb{R}^+)\setminus A_\varepsilon} l_{\mu,v}(Y_1, \ldots, Y_n) \geq l_{\mu_0,v_0}(Y_1, \ldots, Y_n) - hn\right)\\
&\quad\leq \mathbb{P}\left(\sup_{\substack{\mu\notin(\mu_0-\varepsilon,\mu_0+\varepsilon)\\ v>0}} l_{\mu,v}(Y_1, \ldots, Y_n) \geq l_{\mu_0,v_0}(Y_1, \ldots, Y_n) - hn\right) \qquad \textbf{[S8]}\\
&\qquad + \mathbb{P}\left(\sup_{\substack{\mu\in\mathbb{R}\\ v\geq\delta, v\notin(v_0-\varepsilon,v_0+\varepsilon)}} l_{\mu,v}(Y_1, \ldots, Y_n) \geq l_{\mu_0,v_0}(Y_1, \ldots, Y_n) - hn\right) \qquad \textbf{[S9]}\\
&\qquad + \mathbb{P}\left(\sup_{\substack{\mu\in\mathbb{R}\\ v\in(0,\delta)}} l_{\mu,v}(Y_1, \ldots, Y_n) \geq l_{\mu_0,v_0}(Y_1, \ldots, Y_n) - hn\right). \qquad \textbf{[S10]}
\end{aligned}$$

Hence, it suffices to show that there exist some $\delta > 0$ and $h > 0$ for which the limits, as n increases, of the three terms [S8], [S9], and [S10] are zero. To bound [S8],

$$\begin{aligned}
&\mathbb{P}\left(\sup_{\substack{\mu\notin(\mu_0-\varepsilon,\mu_0+\varepsilon)\\ v>0}} l_{\mu,v}(Y_1, \ldots, Y_n) \geq l_{\mu_0,v_0}(Y_1, \ldots, Y_n) - hn\right)\\
&\quad= \mathbb{P}\left(\sup_{\mu\notin(\mu_0-\varepsilon,\mu_0+\varepsilon)} -\log\frac{2\sum_{i=1}^n |Y_i-\mu|}{n} - 1\right.\\
&\qquad\left. \geq -\log 2v_0 - \frac{\sum_{i=1}^n |Y_i-\mu_0|}{nv_0} - h\right)\\
&\quad= \mathbb{P}\left(\sup_{\mu\notin(\mu_0-\varepsilon,\mu_0+\varepsilon)} \log\frac{\sum_{i=1}^n |Y_i-\mu_0|}{\sum_{i=1}^n |Y_i-\mu|} \geq 1 - \frac{\sum_{i=1}^n |Y_i-\mu_0|}{nv_0}\right.\\
&\qquad\left. + \log\frac{\sum_{i=1}^n |Y_i-\mu_0|}{nv_0} - h\right).
\end{aligned}$$

[S11]

The right-hand side of [S11] converges to $-h$ in probability. As

$$\inf_{\mu\notin(\mu_0-\varepsilon,\mu_0+\varepsilon)} \frac{\sum_{i=1}^n |Y_i-\mu|}{n} = \inf_{\mu\in[\mu_0-K,\mu_0-\varepsilon]\cup[\mu_0+\varepsilon,\mu_0+K]} \frac{\sum_{i=1}^n |Y_i-\mu|}{n}$$

for any sufficiently large $K > 0$ with probability approaching 1 and this latter quantity converges to $\inf_{\mu\in[\mu_0-K,\mu_0-\varepsilon]\cup[\mu_0+\varepsilon,\mu_0+K]} \mathbb{E}[|Y-\mu|]$ by the uniform law of large numbers, the left-hand side

of [**S11**] is upper-bounded in probability by $\sup_{\mu\notin(\mu_0-\varepsilon,\mu_0+\varepsilon)} \log(\mathbb{E}[|Y-\mu_0|]/\mathbb{E}[|Y-\mu|])+\eta$ for any $\eta>0$, which is negative for sufficiently small η. Hence, there exists $h>0$ sufficiently small such that [**S8**] converges to 0. To bound [**S9**], let $\hat{v}$ be the sample mean absolute deviation to median and note that

$$\begin{aligned}
&\mathbb{P}\left(\sup_{\substack{\mu\in\mathbb{R}\\ v\geq\delta,\, v\notin(v_0-\varepsilon,\,v_0+\varepsilon)}} l_{\mu,v}(Y_1,\ldots,Y_n)\geq l_{\mu_0,v_0}(Y_1,\ldots,Y_n)-hn\right)\\
&=\mathbb{P}\left(\sup_{v\geq\delta,\, v\notin(v_0-\varepsilon,\,v_0+\varepsilon)} -\log 2v-\frac{\hat{v}}{v}\geq -\log 2v_0-\frac{\sum_{i=1}^n|Y_i-\mu_0|}{nv_0}-h\right)\\
&\leq\mathbb{P}\left(\sup_{v\geq\delta,\, v\notin(v_0-\varepsilon,\,v_0+\varepsilon)} -\log v-\frac{v_0}{v}+\frac{|\hat{v}-v_0|}{\delta}\geq -\log v_0\right.\\
&\qquad\left.-\frac{\sum_{i=1}^n|Y_i-\mu_0|}{nv_0}-h\right)\\
&=\mathbb{P}\left(\sup_{v\geq\delta,\, v\notin(v_0-\varepsilon,\,v_0+\varepsilon)} \log v_0-\frac{v_0}{v}-\log v+1\geq 1\right.\\
&\qquad\left.-\frac{\sum_{i=1}^n|Y_i-\mu_0|}{nv_0}-\frac{|\hat{v}-v_0|}{\delta}-h\right).
\end{aligned}$$

For any $\delta>0$, the right-hand side of this last expression converges to $-h$ in probability (as a weak consequence of *Lemma 2*), whereas the left-hand side is strictly less than 0 because $\log v_0-v_0/v-\log v+1$ is decreasing as v moves away from v_0 in either direction. Hence, there exists h sufficiently small such that [**S9**] converges to 0. Finally, to bound [**S10**],

$$\begin{aligned}
&\mathbb{P}\left(\sup_{\substack{\mu\in\mathbb{R}\\ v\in(0,\,\delta)}} l_{\mu,v}(Y_1,\ldots,Y_n)\geq l_{\mu_0,v_0}(Y_1,\ldots,Y_n)-hn\right)\\
&=\mathbb{P}\left(\sup_{v\in(0,\,\delta)} -\log 2v-\frac{\hat{v}}{v}\geq -\log 2v_0-\frac{\sum_{i=1}^n|Y_i-\mu_0|}{nv_0}-h\right)\\
&\leq\mathbb{P}\left(-\log\delta-\frac{\hat{v}}{\delta}\geq -\log v_0-\frac{\sum_{i=1}^n|Y_i-\mu_0|}{nv_0}-h\right)+\mathbb{P}(\hat{v}<\delta)\\
&=\mathbb{P}\left(\hat{v}\leq\delta\log\frac{v_0}{\delta}+\frac{\delta\sum_{i=1}^n|Y_i-\mu_0|}{nv_0}+\delta h\right)+\mathbb{P}(\hat{v}<\delta).
\end{aligned}$$

As $\hat{v}\to v_0$ in probability (as a weak consequence of *Lemma 2*), we may choose δ and h sufficiently small such that both of the above terms converge to 0 in probability. □

Lemma 5. *Let $Y_1,\ldots,Y_n$ be i.i.d. random variables with distribution function F and $Y_{n+1},\ldots,Y_{2n}$ be i.i.d. random variables with distribution function G. Suppose F and G both satisfy the conditions of Lemma 3. Let (μ_F,v_F) and (μ_G,v_G) be the medians and mean absolute deviations to the median of F and G, and suppose $(\mu_F,v_F)\neq(\mu_G,v_G)$. Let $l_{\mu,v}$ and $\hat{l}$ be as defined in Lemma 3. Then there exists a constant $h>0$ such that*

$$\lim_{n\to\infty}\mathbb{P}\left(\hat{l}(Y_1,\ldots,Y_{2n})-l_{\mu_F,v_F}(Y_1,\ldots,Y_n)-l_{\mu_G,v_G}(Y_{n+1},\ldots,Y_{2n})<-hn\right)=1.$$

Proof:

Let $\varepsilon>0$ be such that $A_F=\{(\mu,v)\in\mathbb{R}\times\mathbb{R}^+:|\mu-\mu_F|<\varepsilon,|v-v_F|<\varepsilon\}$ and $A_G=\{(\mu,v)\in\mathbb{R}\times\mathbb{R}^+:|\mu-\mu_G|<\varepsilon,|v-v_G|<\varepsilon\}$ are disjoint. Then, by Lemma 4, there exists $h>0$ such that

$$\begin{aligned}
C_1(n)&:=\sup_{(\mu,v)\in(\mathbb{R}\times\mathbb{R}^+)\setminus A_F} l_{\mu,v}(Y_1,\ldots,Y_n)-l_{\mu_F,v_F}(Y_1,\ldots,Y_n)<-hn,\\
C_2(n)&:=\sup_{(\mu,v)\in(\mathbb{R}\times\mathbb{R}^+)\setminus A_G} l_{\mu,v}(Y_{n+1},\ldots,Y_{2n})-l_{\mu_G,v_G}(Y_{n+1},\ldots,Y_{2n})\\
&\quad<-hn,
\end{aligned}$$

both with probability approaching 1. As a weak consequence of Lemma 3, there exists some $C>0$ such that

$$\begin{aligned}
C_3(n)&:=\hat{l}(Y_1,\ldots,Y_n)-l_{\mu_F,v_F}(Y_1,\ldots,Y_n)<C(\log n)^2,\\
C_4(n)&:=\hat{l}\,(Y_{n+1},\ldots,Y_{2n})-l_{\mu_G,v_G}(Y_{n+1},\ldots,Y_{2n})<C(\log n)^2,
\end{aligned}$$

with probability approaching 1. The result follows from these conclusions and from the bound

$$\begin{aligned}
&\hat{l}(Y_1,\ldots,Y_{2n})-l_{\mu_F,v_F}(Y_1,\ldots,Y_n)-l_{\mu_G,v_G}(Y_{n+1},\ldots,Y_{2n})\\
&\quad\leq\max\{C_1(n)+C_4(n),C_2(n)+C_3(n)\}.
\end{aligned}$$

□

Proof of Theorem. Let K, $\{\tau_i\}_{i=1}^K$, and $\{S_i\}_{i=1}^K$ denote the number of changes, the change times, and the changed observable sets at those times, respectively, and let $\{K_j\}_{j=1}^J$ and $\{\{\tau_{j,i}\}_{i=1}^{K_j}\}_{j=1}$ denote the number of changes and the change times of the observables individually. Then either $(K,\{\tau_i\},\{S_i\})$ or $(\{K_j\},\{\tau_{j,i}\})$ specifies a change-point set uniquely, and we use both notations to denote such a change-point set as convenient throughout the *Proof*. For convenience of notation, we let $\tau_0=0$, $\tau_{K+1}=T$, $\tau_{j,0}=0$, and $\tau_{j,K_j+1}=T$ for all j.

For any interval $[s,t]$ and any j, let $l_j^0([s,t])=\sum_{r=s}^t l_{\mu_{j,i(r)},v_{j,i(r)}}(Y_{j,r})$, where $i(r)$ is the index such that $r\in[\tau_{j,i(r)}^0+1,\tau_{j,i(r)+1}^0]$ and $\mu_{j,i(r)}$ and $v_{j,i(r)}$ are the median and mean absolute deviation from the median of $F_{j,i(r)}$. Hence, l_j^0 is the log-likelihood of a segment of data under the true medians and mean absolute deviations to the median. Let $\hat{l}_j([s,t])=\hat{l}(Y_{j,s},\ldots,Y_{j,t})$, the maximum log-likelihood of a segment of data, assuming they have the same median and mean absolute deviation. Let $\mathcal{T}_j$ be the set of all subintervals of $[1,T]$ of length at least 2 that do not cross a true change point for observable j; that is, $[s,t]$ is in $\mathcal{T}_j$ if and only if $t>s$ and $[s,t-1]\cap\{\tau_{j,1}^0,\ldots,\tau_{j,K_j^0}^0\}=\emptyset$.

We may assume without loss of generality that $\min_{j,i}|\tau_{j,i+1}^0-\tau_{j,i}^0|>T\varepsilon$. We first show that, with probability approaching 1, all true change points in all observables are detected. Let $\mathcal{A}$ be the collection of possible change-point sets $(\{K_j\},\{\tau_{j,i}\})$ in the search space of the optimization problem in Eq. **2** of the main text, under the constraints of condition v, such that there exists at least one observable j_{miss} with a true change time $\tau_{\text{miss}}^0\in\{\tau_{j_{\text{miss}},1}^0,\ldots,\tau_{j_{\text{miss}},K_{j_{\text{miss}}}^0}^0\}$ for which $|\tau_{j_{\text{miss}},i}-\tau_{\text{miss}}^0|\geq T\varepsilon$ for all $i\in\{0,\ldots,K_{j_{\text{miss}}}+1\}$. For simplicity of the argument, let us for the moment consider only change-point sets for which $|\tau_{j,i}-\tau_{j,i'}^0|$ is either 0 or at least 2 for any $i\in[0,K+1]$, $i'\in[0,K^0+1]$, and $j\in[1,J]$; that is, no change point is exactly one time step away from a true change point.

For each $(\{K_j\},\{\tau_{j,i}\})\in\mathcal{A}$, consider the augmented set of change points $(\{\tilde{K}_j\},\{\tilde{\tau}_{j,i}\})$ such that

$$\left\{\tilde{\tau}_{j,0},\ \ldots,\ \tilde{\tau}_{j,\tilde{K}_j+1}\right\}=\begin{cases}\{\tau_{j,0},\ \ldots,\ \tau_{j,K_j+1}\}\ \cup\ \left\{\tau^0_{j,i}:\tau^0_{j,i}\neq\tau^0_{\mathrm{miss}}\right\}\\ \qquad\qquad\cup\ \{\tau^0_{\mathrm{miss}}-\lfloor T\varepsilon\rfloor,\tau^0_{\mathrm{miss}}+\lfloor T\varepsilon\rfloor\} & \text{if } j=j_{\mathrm{miss}}\\ \{\tau_{j,0},\ \ldots,\ \tau_{j,K_j+1}\}\ \cup\ \left\{\tau^0_{j,1},\ \ldots,\ \tau^0_{j,K^0_j}\right\} & \text{otherwise.}\end{cases}$$

Let $\tilde{\mathcal{A}}$ be the collection of all such augmented change-point sets of $\mathcal{A}$. Then

$$\begin{aligned}&\max_{(\{K_j\},\{\tau_{j,i}\})\in\mathcal{A}}\left(\sum_{j=1}^{J}\sum_{i=0}^{K_j}\hat{l}\left(Y_{j,\tau_{j,i}+1},\ \ldots,Y_{j,\tau_{j,i+1}}\right)-\sum_{i=1}^{K}\lambda(T)q(S_i)\right)\\&\quad-\left(\sum_{j=1}^{J}\sum_{i=0}^{K^0_j}\hat{l}\left(Y_{j,\tau^0_{j,i}+1},\ \ldots,Y_{j,\tau^0_{j,i+1}}\right)-\sum_{i=1}^{K^0}\lambda(T)q\left(S^0_i\right)\right)\\&\le\max_{(\{\tilde{K}_j\},\{\tilde{\tau}_{j,i}\})\in\tilde{\mathcal{A}}}\left(\sum_{j=1}^{J}\sum_{i=0}^{\tilde{K}_j}\hat{l}\left(Y_{j,\tilde{\tau}_{j,i}+1},\ \ldots,Y_{j,\tilde{\tau}_{j,i+1}}\right)\right)\\&\quad-\left(\sum_{j=1}^{J}\sum_{i=0}^{K^0_j}l_{\mu_i,\nu_i}\left(Y_{j,\tau^0_{j,i}+1},\ \ldots,Y_{j,\tau^0_{j,i+1}}\right)\right)+K^0\lambda(T)q(\{1,\ \ldots,J\})\\&=\max_{(\{\tilde{K}_j\},\{\tilde{\tau}_{j,i}\})\in\tilde{\mathcal{A}}}\left(\sum_{j=1}^{J}\sum_{i=0}^{\tilde{K}_j}\hat{l}_j\left(\left[\tilde{\tau}_{j,i}+1,\tilde{\tau}_{j,i+1}\right]\right)-l^0_j\left(\left[\tilde{\tau}_{j,i}+1,\tilde{\tau}_{j,i+1}\right]\right)\right)\\&\quad+K^0\lambda(T)q(\{1,\ \ldots,J\})\\&\le\max_{j_{\mathrm{miss}}\in\{1,\ldots,J\}}\ \max_{\tau^0_{\mathrm{miss}}\in\left\{\tau^0_{j_{\mathrm{miss}},1},\ldots,\tau^0_{j_{\mathrm{miss}},K_{j_{\mathrm{miss}}}}\right\}}\left(\hat{l}_{j_{\mathrm{miss}}}\left(\left[\tau^0_{\mathrm{miss}}-\lfloor T\varepsilon\rfloor+1,\tau^0_{\mathrm{miss}}+\lfloor T\varepsilon\rfloor\right]\right)\right.\\&\quad\left.-\ l^0_{j_{\mathrm{miss}}}\left(\left[\tau^0_{\mathrm{miss}}-\lfloor T\varepsilon\rfloor+1,\tau^0_{\mathrm{miss}}+\lfloor T\varepsilon\rfloor\right]\right)\right)\\&\quad+\ \sum_{j=1}^{J}\left(K_{\max}+K^0+1\right)\max_{[s,t]\in\mathcal{T}_j}\left(\hat{l}_j([s,t])-l^0_j([s,t])\right)\\&\quad+K^0\lambda(T)q(\{1,\ \ldots,J\}),\end{aligned}\tag{S12}$$

where the last inequality follows because the only data segment between consecutive changes of an augmented change-point set that crosses a true change point in any observable is, by construction, the segment $[\tau^0_{\mathrm{miss}}-\lfloor T\varepsilon\rfloor+1,\tau^0_{\mathrm{miss}}+\lfloor T\varepsilon\rfloor]$ in observable j_{miss}. By conditions *i* and *ii* and *Lemma 5*, for any j_{miss} and τ^0_{miss},

$$\begin{aligned}&\hat{l}_{j_{\mathrm{miss}}}\left(\left[\tau^0_{\mathrm{miss}}-\lfloor T\varepsilon\rfloor+1,\tau^0_{\mathrm{miss}}+\lfloor T\varepsilon\rfloor\right]\right)\\&\quad-l^0_{j_{\mathrm{miss}}}\left(\left[\tau^0_{\mathrm{miss}}-\lfloor T\varepsilon\rfloor+1,\tau^0_{\mathrm{miss}}+\lfloor T\varepsilon\rfloor\right]\right)<-hT\varepsilon\end{aligned}$$

for some $h>0$ with probability approaching 1. On the other hand, by condition *i* and *Lemma 3*, for any j,

$$\max_{[s,t]\in\mathcal{T}_j}\left(\hat{l}_j([s,t])-l^0_j([s,t])\right)\le C'(\log T)^2$$

for some sufficiently large $C'>0$ with probability approaching 1. Hence, when $\lambda(T)/T\to 0$ as guaranteed by condition *iv*, this implies

$$\begin{aligned}&\max_{(\{K_j\},\{\tau_{j,i}\})\in\mathcal{A}}\left(\sum_{j=1}^{J}\sum_{i=0}^{K_j}\hat{l}\left(Y_{j,\tau_{j,i}+1},\ \ldots,Y_{j,\tau_{j,i+1}}\right)-\sum_{i=1}^{K}\lambda(T)q(S_i)\right)\\&\le\left(\sum_{j=1}^{J}\sum_{i=0}^{K^0_j}\hat{l}\left(Y_{j,\tau^0_{j,i}+1},\ \ldots,Y_{j,\tau^0_{j,i+1}}\right)-\sum_{i=1}^{K^0}\lambda(T)q\left(S^0_i\right)\right)\end{aligned}$$

with probability approaching 1, so $(\hat{K},\{\hat{\tau}_i\},\{\hat{S}_i\})\notin\mathcal{A}$ with probability approaching 1. As condition *iv* guarantees $\lambda(T)/(\min_{i\in[0,K^0]}\tau^0_{i+1}-\tau^0_i)\to 0$, this also implies $\hat{K}\ge K^0$ and $\hat{K}_j\ge K^0_j$ for each j with probability approaching 1.

Now we show that, with probability approaching 1, the sets S^0_i of observables that change simultaneously are identified as simultaneous changes and no additional changes are detected. Let $\mathcal{B}$ denote the collection of possible $(K,\{\tau_i\},\{S_i\})$ in the search space of the optimization problem in Eq. **2** of the main text, under the constraints of condition *v*, that are not in $\mathcal{A}$ and such that either $\sum_i|S_i|>\sum_i|S^0_i|$ or $K>K^0$. Again, let us suppose for simplicity that $|\tau_{j,i}-\tau^0_{j,i'}|$ is either 0 or at least 2 for any $i\in[0,K+1]$, $i'\in[0,K^0+1]$, and $j\in[1,J]$. If $(K,\{\tau_i\},\{S_i\})\notin\mathcal{A}$, then for each i, the set S^0_i of observables must all have a change in the time window $[\tau^0_i-\lfloor T\varepsilon\rfloor+1,\tau^0_i+\lfloor T\varepsilon\rfloor]$. Asymptotically, condition *iv* ensures these time windows do not overlap, so condition *iii* guarantees that, for any $(K,\{\tau_i\},\{S_i\})\in\mathcal{B}$, $\sum_{i=1}^{K}q(S_i)>\sum_{i=1}^{K_0}q(S^0_i)+c$ for some constant $c>0$, depending only on the function q. For each $(\{K_j\},\{\tau_{j,i}\})\in\mathcal{B}$, let us consider the augmented set of change points

$$\left\{\tilde{\tau}_{j,0},\ \ldots,\ \tilde{\tau}_{j,\tilde{K}_j+1}\right\}=\{\tau_{j,0},\ \ldots,\tau_{j,K_j+1}\}\bigcup\left\{\tau^0_{j,1},\ \ldots,\ \tau^0_{j,K^0_j}\right\},$$

and let $\tilde{\mathcal{B}}$ denote the collection of all such augmented change-point sets. Then by similar reasoning to that above,

$$\begin{aligned}&\max_{(\{K_j\},\{\tau_{j,i}\})\in\mathcal{B}}\left(\sum_{j=1}^{J}\sum_{i=0}^{K_j}\hat{l}\left(Y_{j,\tau_{j,i}+1},\ \ldots,Y_{j,\tau_{j,i+1}}\right)-\sum_{i=1}^{K}\lambda(T)q(S_i)\right)\\&\quad-\left(\sum_{j=1}^{J}\sum_{i=0}^{K^0_j}\hat{l}\left(Y_{j,\tau^0_{j,i}+1},\ \ldots,Y_{j,\tau^0_{j,i+1}}\right)-\sum_{i=1}^{K^0}\lambda(T)q\left(S^0_i\right)\right)\\&\le\max_{(\{\tilde{K}_j\},\{\tilde{\tau}_{j,i}\})\in\tilde{\mathcal{B}}}\left(\sum_{j=1}^{J}\sum_{i=0}^{\tilde{K}_j}\hat{l}_j\left(\left[\tilde{\tau}_{j,i}+1,\tilde{\tau}_{j,i+1}\right]\right)-l^0_j\left(\left[\tilde{\tau}_{j,i}+1,\tilde{\tau}_{j,i+1}\right]\right)\right)\\&\quad-c\lambda(T)\le\sum_{j=1}^{J}\left(K_{\max}+K^0+1\right)\max_{[s,t]\in\mathcal{T}_j}\left(\hat{l}_j([s,t])-l^0_j([s,t])\right)-c\lambda(T).\end{aligned}\tag{S13}$$

By condition *i* and *Lemma 3*, for each j and some $C'>0$,

$$\max_{[s,t]\in\mathcal{T}_j}\left(\hat{l}_j([s,t])-l^0_j([s,t])\right)\le C'(\log T)^2$$

with probability approaching 1. Hence, for $\lambda(T)\ge C(\log T)^2$ and some sufficiently large C, $(\hat{K},\{\hat{\tau}_i\},\{\hat{S}_i\})\notin\mathcal{B}$ with probability approaching 1.

For change-point sets $(K,\{\tau_i\},\{S_i\})$ in which there exist j, i, and i' such that $|\tau_{j,i}-\tau^0_{j,i'}|=1$, the previous arguments must be modified because the augmentation of these sets to $(\{\tilde{K}_j\},\{\tilde{\tau}_{j,i}\})$

as described above introduces segments $[s,t]$ of only one data point, which are not in $\mathcal{T}_j$ and for which the maximum log-likelihood $\hat{l}_j([s,t])$ is infinite. If, for example, $\tau_{j,i} = \tau^0_{j,i'} - 1$, then consider instead the augmentation that adds the point $\tau^0_{j,i'} + 1$ instead of $\tau^0_{j,i'}$, if $\tau^0_{j,i'} + 2 \notin \{\tau_{j,1}, \ldots, \tau_{j,K_j}\}$, or that simply does not add this point if both $\tau^0_{j,i'} - 1$ and $\tau^0_{j,i'} + 2$ are among $\{\tau_{j,1}, \ldots, \tau_{j,K_j}\}$. Consider the analogous augmentation when $\tau_{j,i} = \tau^0_{j,i'} + 1$. This then introduces an additional term of the form $\hat{l}_j([s,t]) - l^0_j([s,t])$, for each change point $\tau_{j,i}$ that is within one time step from a true change point $\tau_{j,i'}$, in the upper-bound Eqs. **S12** and **S13**, where $[s,t]$ is an interval of two or three data points that straddles a true change point. If Y is a random variable whose distribution has median μ_i and mean absolute deviation v_i, then for any $r(T)$ such that $r(T) \to \infty$,

$$\mathbb{P}\left(-l_{\mu_i,v_i}(Y) > r(T)\right) = \mathbb{P}(|Y - \mu_i| > v_i(r(T) - \log 2v_i)) \to 0,$$

and thus $-l^0_j([s,s+1]) \le 2r(T)$ and $-l^0_j([s,s+2]) \le 3r(T)$ for any j and s with probability approaching 1. Furthermore,

$$\begin{aligned}\mathbb{P}\left(\hat{l}_j([s,s+1]) > r(T)\right) &= \mathbb{P}\left(-2 - 2\log\left(|Y_{j,s} - Y_{j,s+1}|\right) > r(T)\right) \\ &= \mathbb{P}\left(e^{-((r(T)+2)/2)} > |Y_{j,s} - Y_{j,s+1}|\right) \to 0\end{aligned}$$

when the distributions of $Y_{j,s}$ and $Y_{j,s+1}$ are continuous as guaranteed by condition i, and similarly, $\hat{l}_j([s,s+2]) \to 0$. Thus, letting $r(T) = (\log T)^2$, each additional term $\hat{l}_j([s,t]) - l^0_j([s,t])$ is at most $4(\log T)^2$ with probability approaching 1, and the remainder of the above arguments hold. This concludes the proof that $(\hat{K}, \{\hat{\tau}_i\}, \{\hat{S}_i\}) \notin \mathcal{A} \cup \mathcal{B}$ with probability approaching 1 and thus establishes the desired result. □

1. Jackson B, et al. (2005) An algorithm for optimal partitioning of data on an interval. *IEEE Signal Process Lett* 12(2):105–108.
2. Killick R, Fearnhead P, Eckley A (2012) Optimal detection of changepoints with a linear computational cost. *J Am Stat Assoc* 107(500):1590–1598.
3. Wriggers W, et al. (2009) Automated event detection and activity monitoring in long molecular dynamics simulations. *J Chem Theory Comput* 5:2595–2605.
4. Ramanathan A, Savol AJ, Agarwal PK, Chennubhotla CS (2012) Event detection and sub-state discovery from biomolecular simulations using higher-order statistics: Application to enzyme adenylate kinase. *Proteins* 80(11):2536–2551.
5. Lindorff-Larsen K, Piana S, Dror RO, Shaw DE (2011) How fast-folding proteins fold. *Science* 334(6055):517–520.
6. Shaw DE, et al. (2010) Atomic-level characterization of the structural dynamics of proteins. *Science* 330(6002):341–346.
7. Beauchamp KA, McGibbon R, Lin Y-S, Pande VS (2012) Simple few-state models reveal hidden complexity in protein folding. *Proc Natl Acad Sci USA* 109(44):17807–17813.
8. Dror RO, et al. (2011) Activation mechanism of the β_2-adrenergic receptor. *Proc Natl Acad Sci USA* 108(46):18684–18689.
9. Politis DN, Romano JP (1994) The stationary bootstrap. *J Am Stat Assoc* 89(428): 1303–1313.
10. Zhang NR, Siegmund DO, Ji H, Li JZ (2010) Detecting simultaneous changepoints in multiple sequences. *Biometrika* 97(3):631–645.
11. Siegmund D (1992) Tail approximations for maxima of random fields. *Probability Theory: Proceedings of the 1989 Singapore Probability Conference*, eds Chen LH, Choi K, Yu K, Lou J-H (deGruyter, New York), pp 147–158.
12. Vert J-P, Bleakley K (2010) Fast detection of multiple change points shared by many signals using group LARS. *Advances in Neural Information Processing Systems*, eds Lafferty J, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A (NIPS, Vancouver), Vol 23, pp 2343–2351.
13. Beauchamp KA, et al. (2011) MSMBuilder2: Modeling conformational dynamics on the picosecond to millisecond scale. *J Chem Theory Comput* 7(10):3412–3419.
14. Ensign DL, Pande VS (2010) Bayesian detection of intensity changes in single molecule and molecular dynamics trajectories. *J Phys Chem B* 114(1):280–292.
15. Yao Y-C (1988) Estimating the number of change points via Schwarz' criterion. *Stat Probab Lett* 6(3):181–189.
16. Lee C-B (1997) Estimating the number of change points in exponential families distributions. *Scand Stat Theory Appl* 24:201–210.
17. Huber PJ (1967) The behavior of maximum likelihood estimates under nonstandard conditions. *Proc Fifth Berkeley Symp Math Stat Prob* 1(1):221–233.
18. White H (1982) Maximum likelihood estimation of misspecified models. *Econometrica* 50(1):1–25.
19. Massart P (1990) The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *Ann Probab* 18:1269–1283.
20. Hoeffding W (1963) Probability inequalities for sums of bounded random variables. *J Am Stat Assoc* 58(301):13–30.
21. Bernstein S (1924) On a modification of Chebyshev's inequality and of the error formula of Laplace. *Ann Sci Inst Sav Ukraine Sect Math* 1(3):38–49.

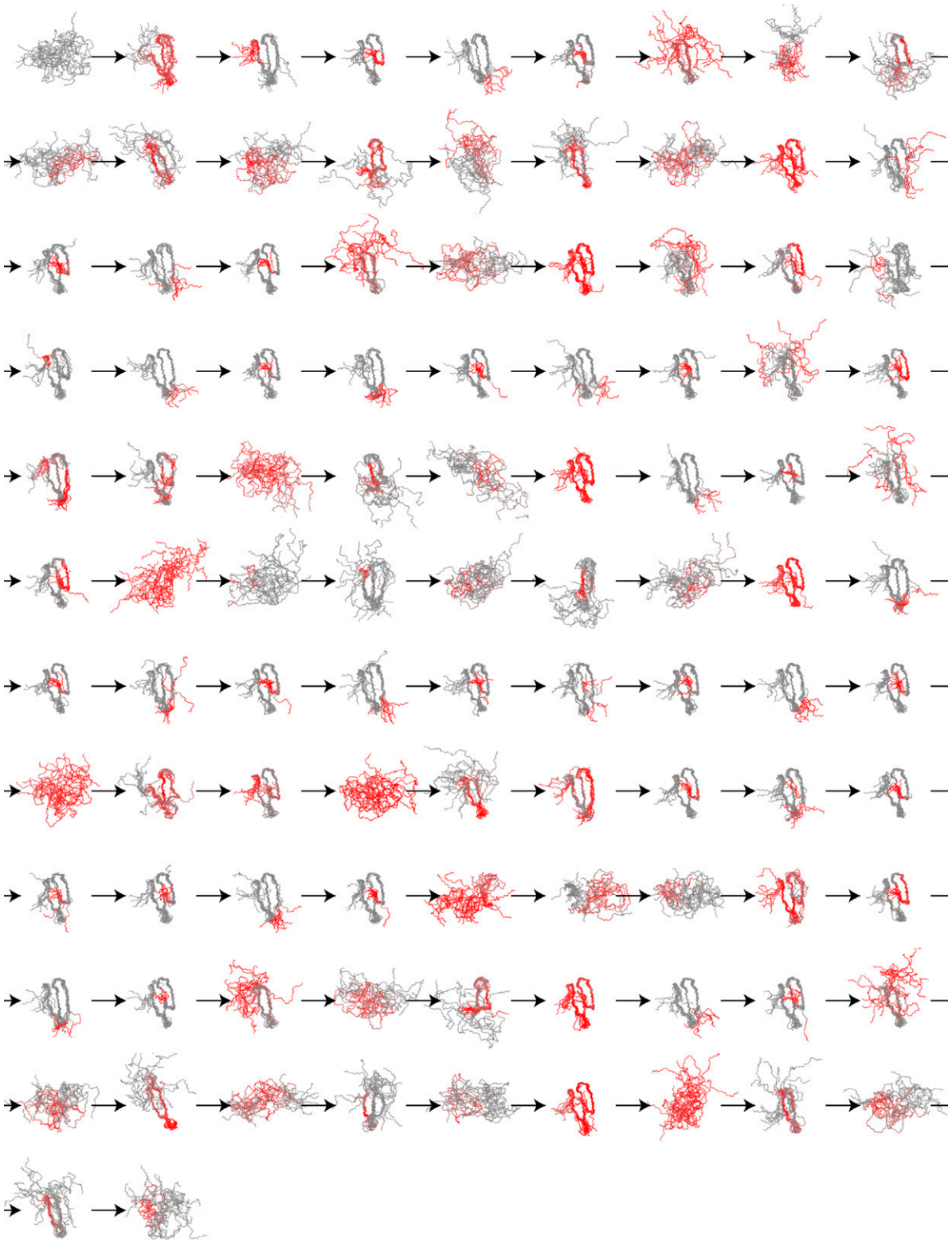

Fig. S1. Results from application of SIMPLE to an MD simulation trajectory of a WW domain repeatedly folding and unfolding (the simulation was performed at a temperature at which the protein is folded approximately half the time). All 100 changes detected by SIMPLE at a sensitivity parameter setting of $\lambda \approx 38$ are shown (three of these changes are shown in Fig. 2*A* of the main text). Each arrow represents a single detected change. Aligned ensemble images are shown for the protein backbone structures between each successive pair of detected changes, with red color used to indicate the spatial location of the change between the current ensemble and the previous ensemble as indicated by SIMPLE's output.

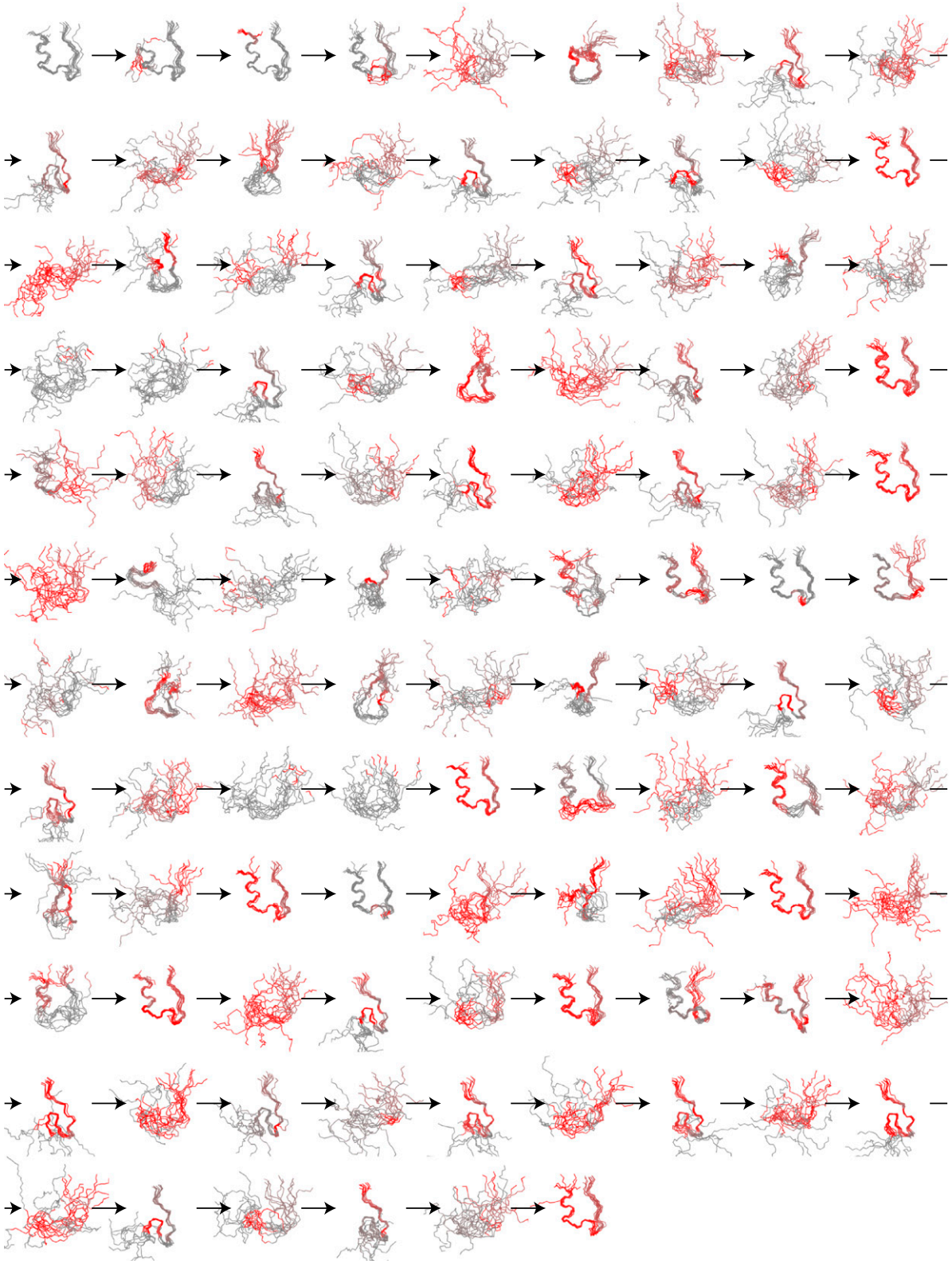

Fig. S2. Results from application of SIMPLE to an MD simulation trajectory of Trp-cage repeatedly folding and unfolding (the simulation was performed at a temperature at which the protein is folded approximately half the time). All 104 changes detected by SIMPLE at a sensitivity parameter setting of $\lambda \approx 34$ are shown (two of these changes are shown in Fig. 2*B* of the main text). Each arrow represents a single detected change. Aligned ensemble images are shown for the protein backbone structures between each successive pair of detected changes, with red color used to indicate the spatial location of the change between the current ensemble and the previous ensemble as indicated by SIMPLE's output.

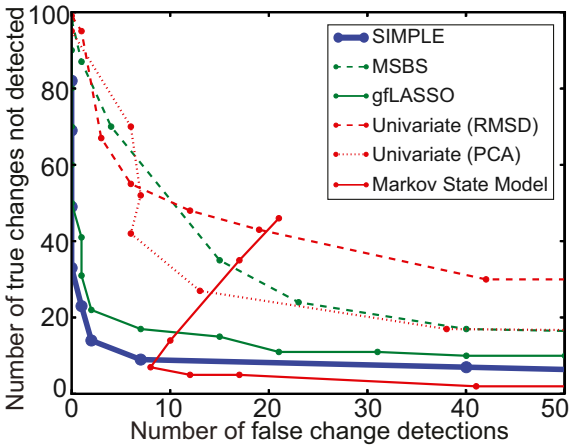

Fig. S3. Comparison of the performance of various methods for detecting conformational changes in a long synthetic protein trajectory that visits each conformational state dozens of times according to a Markov chain. Compared with results shown in Fig. 4 of the main text, the repeated visits to each conformational state improve the performance of the Markov state model (MSM) method, but have little impact on the performance of SIMPLE. As a result, the MSM outperforms SIMPLE at higher false change-detection rates (but is unable to achieve very low false change-detection rates). Due to computational limitations, few real MD simulations—even among those generated on specialized supercomputers—visit each conformational state many times.

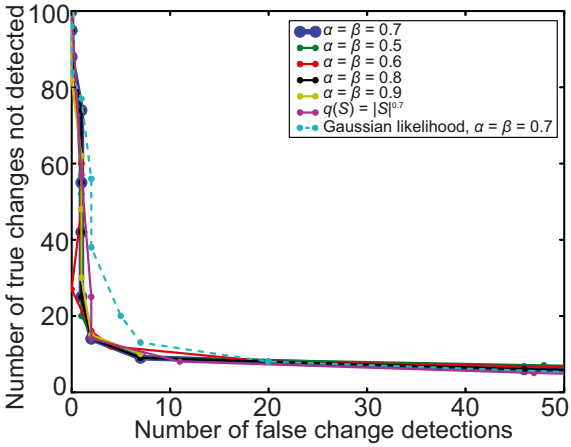

Fig. S4. Comparison of the performance of SIMPLE using different penalty functions $q(S)$ and different likelihood models, on the same synthetic protein trajectories as in Fig. 4 of the main text. For a penalty function of the form $q(S) = (\sum_i |S \cap G_i|^\alpha)^\beta$, performance is similar over a range of values of α and β between 0.5 and 0.9. Performance under these penalty functions is also similar to that under the recommended "generic" penalty function $q(S) = |S|^{0.7}$, which does not use observable groups G_i. Performance using a Gaussian likelihood model is inferior to that using a Laplace likelihood model.

Algorithm S1. SIMPLE optimization algorithm

```
for t = 1 → T − 1 do
    for j = 1 → J do
        rand[j, t] ← Unif(0.9, 1)
        penalty[j, t] ← (λq({1, . . . , J}) − λq({1, . . . , J} \ {j})) · rand[j, t]
    end for
end for
move_and_merge ← False
loop
    changes ← empty map
    for j = 1 → J do
        univariate_changes[j] ← univariate_optimize({Y_{j,1}, . . . , Y_{j,T}}, penalty[j, ·])
        for t ∈ univariate_changes[j] do
            Add j to changes[t]
        end for
    end for
    if move_and_merge then
      (τ_1, . . . , τ_K) ← sorted keys of changes
      for i = 1 → K do
          V ← −∞
          for s = τ_{i−1} → τ_{i+1} do
              v ← penalized-likelihood if changes at time τ_i were moved/merged to time s
              if v > V then
                  V ← v, τ_i* ← s
              end if
          end for
          τ_i ← τ_i*, move/merge changes[τ_i] to changes[τ_i*]
      end for
end if
if changes equals changes from any previous iteration then
    return changes
end if
if number of change times equals number from previous iteration then
    move_and_merge ← True
end if
for t = 1 → T − 1 do
    for j = 1 → J do
        if t ∈ changes then
            if j ∈ changes[t] then
                penalty[j, t] ← λq(changes[t]) − λq(changes[t] \ {j})
            else
                penalty[j, t] ← λq(changes[t] ∪ {j}) − λq(changes[t])
            end if
        else
            penalty[j, t] ← λq({j}) · rand[j, t]/0.9
        end if
    end for
  end for
end loop
```