Supplementary Information to
A Scalable Algorithm to Order and Annotate Continuous Observations Reveals the Metastable States Visited by Dynamical Systems

Nicolas Blöchliger[a], Andreas Vitalis[a,*], Amedeo Caflisch[a]

[a]*Department of Biochemistry*
University of Zurich
Winterthurerstrasse 190, 8057 Zurich, Switzerland

Abstract

Advances in IT infrastructure have enabled the generation and storage of very large data sets describing complex systems continuously in time. These can derive from both simulations and measurements. Analysis of such data requires the availability of scalable algorithms. In this contribution, we propose a scalable algorithm that partitions instantaneous observations (snapshots) of a complex system into kinetically distinct sets (termed basins). To do so, we use a combination of ordering snapshots employing the method's only essential parameter, *i.e.*, a definition of pairwise distance, and annotating the resultant sequence, the so-called progress index, in different ways. Specifically, we propose a combination of cut-based and structural annotations with the former responsible for the kinetic grouping and the latter for diagnostics and interpretation. The method is applied to an illustrative test case, and the scaling of an approximate version is demonstrated to be $O(N \log N)$ with N being the number of snapshots. Two real-world data sets from river hydrology measurements and protein folding simulations are then used to highlight the utility of the method in finding basins for complex systems. Both limitations and benefits of the approach are discussed along with routes for future research.

Keywords: Complex System, Trajectory Analysis, Scalable Algorithm, Minimum Spanning Tree, Free Energy Basins

*To Whom Correspondence Should be Addressed
Email addresses: n.bloechliger@bioc.uzh.ch (Nicolas Blöchliger), a.vitalis@bioc.uzh.ch (Andreas Vitalis), caflisch@bioc.uzh.ch (Amedeo Caflisch)

S.1. Supplementary Methods

S.1.1. Implementation of Exact Algorithm

The implementation described in the following is used to provide an algorithm for the following scheme as proposed in the main text:

Choose a starting snapshot $s_1 \in T$ and create the set $S_1 = \{s_1\}$. Initialize the cut function, $c : \{1, \ldots, N\} \to \mathbb{N}$, to 2. Then, for $i = 1, \ldots, N-1$ do the following:

1. Define s_{i+1} as the snapshot in $T \setminus S_i$ realizing the minimum of $d(\cdot, S_i) = \min_{j=1,\ldots,i} d(\cdot, s_j)$.
2. Let $S_{i+1} = S_i \cup \{s_{i+1}\}$.
3. Define $c(i+1) = \sum_{j=1}^{N-1} \zeta_{S_{i+1}}(t_j, t_{j+1})$.

Here, function ζ is given by eq. 1 in the main text. The exact **progress index** of T starting with s_1 is defined as the sequence $S(T, s_1) = (s_1, \ldots, s_N)$. Each entry i is associated with a value for the cut function, $c(i)$.

In order to guarantee the scheme to be exact, we theoretically need to know all $O(N^2)$ pairwise distances. Then, we can - for each snapshot - create a list containing all other snapshots ordered by distance along with the distance. This already has a complexity of $O(N^2 \log N)$ on account of the required sorting of a list of size $O(N)$ for every snapshot. From this set of ordered lists, the progress index is conveniently generated in $O(N^2)$ time. This is because for iteration i in the above scheme, we need to scan only the nearest, eligible neighbor for each current member of the set S, *i.e.*, we find the minimum one of $i = O(N)$ candidates. To keep track of eligibility, we utilize a pointer array to the smallest, eligible entry in each list, and updating the pointer array again has limiting complexity of $O(N)$ for each iteration.

Suppose now we assume that the density of points is homogeneous enough such that the longest edge in the underlying minimum spanning tree (MST) has a value of d_{max} that is substantially smaller than the majority of distance values found in the ordered lists. Then, the following heuristic emerges. With a reasonable guess of d_{max}, we first use an efficient clustering algorithm with controllable cluster size to find a set of cluster centroids. For each cluster k, we can compute the maximum distance from its centroid as r_{c_k}. Then, by virtue of d being a metric, we can exclude all pairwise distance comparisons for snapshots belonging to clusters k and m, whose centroids are further apart than $d_{max} + r_{c_k} + r_{c_m}$. This reduces the computational cost of the implementation twofold: first, not all $O(N^2)$ distances are evaluated (in our tests on n-butane, the fraction evaluated ranged from 5-30%); second, the sorting of the lists truncated to d_{max} needs less than $O(N^2 \log N)$ time for each list. It is important to point out that we usually expect the required value for d_{max} to decrease with increasing sampling density meaning that the overall complexity may be reduced to $O(N^2)$ or less.

Unfortunately, there are three problems associated with the above scheme. First, the required value of d_{max} is difficult to guess. To obtain a MST, the algorithm may have to be rerun a few times with increasing values for d_{max}. Second, the heuristic in use is dependent on the structure of the data, *i.e.*, it is not universally applicable. It is straightforward to construct pathological cases, in which a single edge of the MST is so much longer than the rest that the truncated lists are nearly as long and as expensive to compute as the complete $O(N)$ lists. Third, even if the data conform to the assumptions of the heuristic, the memory required for storing the lists still grows superlinearly with N. This is in contrast to the number of edges of the MST that is $N-1$.

With the lists generated, the efficiency of the generation of the progress index itself can potentially be improved by first combining all distances and snapshot pairs in the truncated lists to a single list that is then globally sorted by distance. Obviously, the complexity of this operation is favorable compared to the implementation described above only if the number of items is significantly smaller than $(N-1)N/2$. From the globally sorted list, we can derive the MST via Kruskal's algorithm [1] with lower time complexity due to the edges already being sorted. For data sets for n-butane, we found an effective scaling exponent of computational cost over a range of 900–90000 snapshots of 1.75.

S.1.2. Derivation of Combinatorial Prior for Cut Function

As shown in Fig. 3 of the main text, a progressive decrease in temporal resolution eventually yields a default profile with parabolic shape. If the saving frequency exceeds all relevant time scales of the system, the trajectory essentially looks random, and the annotation function c reports what looks like a single basin. However, there are combinatorial reasons for why the result is not flat along the progress index, and these reasons are treated explicitly next.

The idea of the analytical derivation centers around the number of different ways a trajectory can be randomly partitioned into two sets, S_i and $A_i = T \setminus S_i$. For given $i = |S_i|$ and $j = |A_i| = N - i$, there are

$$r_c = 2\binom{i-1}{q}\binom{j-1}{q}$$

trajectories with a value for c of $1 \leq c = 2q + 1 \leq 2\min\{i, j\} - 1$, and there are

$$r_c = \binom{i-1}{q}\binom{j-1}{q-1} + \binom{j-1}{q}\binom{i-1}{q-1}$$

trajectories with a value for c of $2 \leq c = 2q \leq 2\min\{i, j\}$. For a trajectory whose snapshots are randomly assigned to S_i and A_i (satisfying $|S_i| = i$), the probability that its cut value is c is given by

$$p_c = r_c \Big/ \binom{N}{i}.$$

Using Vandermonde's identity, we get the expected c_{prior} as the following expectation value:

$$\mathbb{E}(c(i)) = \sum_{c=1}^{2\min\{i,j\}} c p_c = 2i \binom{N-1}{i} \Big/ \binom{N}{i} = 2i(N-i)/N.$$

The function defined above is used in Fig. 3 of the main text (dashed line), and corresponds to a combinatorial prior function that is a direct result of the differing asymmetry in partition sizes for different values of the progress index. The resulting profile is symmetric around the point $N/2$ and is independent of the length of the trajectory in the sense that $\mathbb{E}(c_{\lambda N}(\lambda i))/\lambda N = \mathbb{E}(c_N(i))/N$ for all $\lambda \in \mathbb{N}$ (here, the subscript denotes the length of the trajectory).

S.1.3. Implementation Details for the Approximate Algorithm

Recalling the exact algorithm conceptually (see Section S.1.1), we can divide the task into two components, *i.e.*, construction of the MST and parsing of the MST to yield the annotated progress index. For the latter, for typical data sets, the computational cost will scale linearly with N given a spanning tree. In contrast, the former is prohibitively expensive for large data sets as described above. Assuming we know the ordered list of nearby snapshots for every snapshot, Borůvka's well-known algorithm [2] assembles the MST by successively joining subtrees in $O(N \log N)$ time. Therefore, approximations are introduced that are meant to replace the use of an ordered list of all other snapshots with a set of unordered and nearby snapshots and of controlled size, which allows any given merging operation to happen in constant or nearly constant time. The result is no longer an MST, but rather a short spanning tree (SST).

Specifically, this happens by data preorganization and random guessing with a fixed maximum number of guesses, which corresponds to the parameter N_g used throughout. Data preorganization allows defining a list of snapshots, τ_i, that contains candidates with small values of the distance to the set S_i describing a subtree at any given stage of Borůvka's algorithm. The notion of "small" is understood qualitatively in relation to the distribution of this quantity for all snapshots in $T \setminus S_i$. τ_i can be assembled by clustering the data set prior to the construction of the spanning tree. Specifically, τ_i is the list of unique snapshots constructed from all clusters that S_i spans into. Following the algorithm, it becomes clear that eventually all cluster members will be exhausted preventing further merging steps of subtrees. This is where the idea of a hierarchical clustering becomes critical. Hierarchical data preorganization implies that we obtain a clustering for a series of chosen resolutions of increasing coarseness. If clusters are exhausted at the finest resolution, τ_i is simply assembled at the next coarser level that yields a nonempty set.

In principle, any hierarchical clustering algorithm that does not generate cluster overlap and reflects local density could be used provided that it operates in at most $O(N \log N)$ time with data set size and in linear time with data dimensionality. In our implementation, we use a recently developed top-down, tree-based clustering algorithm meeting these requirements [3]. The resultant hierarchical tree of clusters is not to be confused with the MST or SST at the snapshot level considered here. While the reader is referred to the literature for details, a brief summary is as follows. The clustering algorithm relies on one main (a minimum threshold distance, t_1) and two auxiliary parameters (the tree height, H, and a maximum threshold distance, t_H). From the top to the bottom level the data set is clustered with increasing resolution such that parent-child relationships defining the tree of clusters can be exploited to achieve near linear scaling with data set size. Each of the H tree levels is associated with a threshold distance t_k determined by linear interpolation between t_H and t_1. The data is processed sequentially. Starting on the top level, snapshot j is added to its nearest cluster on level H if the distance does not exceed t_H, otherwise j spans a new cluster on its own. Then for each level k down to level 2, j is added to the nearest cluster on this level provided that the distance does not exceed t_k, otherwise it spans a new cluster. The key steps guaranteeing efficiency are that 1) only the children of the cluster of j on level $k+1$ are scanned, and that 2) distances of snapshots to clusters are measured as distances to the centroid of the cluster. Using simple algebra, these centroids can be updated continuously without additional cost pending that the distance function in use is Euclidean. The first pass creates a "raw" tree with minor errors caused primarily by centroid drift. Therefore, in a second pass of the data, centroids at all populated levels are kept fixed, and snapshots are simply reassigned to clusters. In addition, the clusters at the leaf level (finest resolution) are now created, which implies that the leaf level results are of higher quality. This is also a reasonable property for SST construction given that the majority of SST edges are expected to derive from neighbor relations encoded in leaf level clusters.

With the restricted list of snapshots, τ_i, in hand, a second approximation may be introduced. Specifically, if the size of τ_i exceeds the value of the parameter N_g, N_g candidates are picked randomly, and the one yielding the shortest edge becomes a putative edge of the SST; otherwise, the search is exhaustive over the set of candidate edges, and the approximation is purely at the level of reducing the search space to τ_i. Clearly, the second approximation may be severe. Consider a case where τ_i is constituted from multiple, coarse clusters spanning a large volume in data space. The distribution of points in τ_i in relation to the members of S_i may be

heavily skewed, and/or the ratio $N_g/|\tau_i|$ may be unfavorable. In either case, the likelihood of introducing a significantly inaccurate neighbor relationship is large. These types of issues imply that it is not straightforward to optimize the hierarchical clustering for SST construction, and we have not attempted to do so rigorously. The most important property is probably that the leaf clusters be tight, free of overlap, and somewhat matched in size to the choice for N_g.

In addition, there are some technical points to consider. First, the assembly of snapshot lists for each subtree must be handled efficiently by using dedicated, but straightforward data structures such as different types of linked lists. Memory is allocated dynamically, but with sufficient buffering to prevent slowdown by frequent allocation events. A heuristic is used to determine whether to use partial snapshot lists created during clustering or whether to recreate the list for a given cluster. Second, larger subtrees will eventually span multiple clusters. It may happen that one or more, but not all of these clusters are exhausted at a given level requiring a jump to the next coarser resolution. In this case, τ_i remains restricted to the finest level for which any cluster still contains eligible snapshots. This strategy is meant to exploit the fact that neighbor relationships will be most meaningful at the finest resolution (leaf) level. Third, note that at each step of the algorithm there is only a single minimum distance considered for every subtree, and that the corresponding edge is added to the SST only if it does not introduce a cycle. Cycles are avoided by updating an index array denoting subtree memberships for all snapshots immediately after each merging event.

S.1.4. Data Sets

S.1.4.1. n-butane

We used torsional space, stochastic dynamics simulations in the gas phase at 400 K to obtain trajectories of varying length. In all cases, only a single molecule of *n*-butane was present, the integration time step was 5 fs, and the number of trajectory snapshots that were analyzed was 30000. The only term to the potential energy were the torsional potentials native to the OPLS-AA force field [4]. The chosen representation consisted of the three dihedral angles themselves with appropriate corrections for computing Euclidean distances between snapshots [3].

S.1.4.2. Beta3S Miniprotein

Data were taken from equilibrium sampling of the polypeptide Thr-Trp-Ile-Gln-Asn-Gly-Ser-Thr-Lys-Trp-Tyr-Gln-Asn-Gly-Ser-Thr-Lys-Ile-Tyr-Thr with the terminal residues in zwitterionic state. At least in the limits of a specific continuum description of solvation [5], this peptide undergoes reversible folding transitions [6, 7] between a well-ordered, three-stranded β-sheet conformation and a coil-like unfolded state ensemble. This is augmented by various enthalpically stabilized, non-native basins, most prominently a partially ordered ensemble of α-helix rich conformations.

The simulation data were obtained from prior work [8], and a data set of size $8 \cdot 10^6$ trajectory snapshots was considered for analysis. The pairwise, Euclidean distance function is defined on a chosen representation that here consisted of 273 interatomic distances between backbone nitrogen and oxygen atoms. This representation deemphasizes fast degrees of freedom such as side chain rotamer states. For the purpose of runtime analysis (Fig. 5A) of the approximate algorithm, the entire $8 \cdot 10^6$ snapshots were read with increasing skip to arrive at data sets of reduced size. In order to make the results comparable, we kept the number of guesses, N_g, the tree height, and the maximum threshold criterion constant at values of 20, 16, and 8 Å, respectively. To obtain a constant average cluster size, the finest threshold value was set to 1.82, 1.67, 1.52, 1.35, 1.15, 1.03, 0.88, and 0.76, for increasing snapshot numbers from 62500 to $8 \cdot 10^6$. It should be noted that this parameter is also the only one that shows a weakly systematic impact on the total weight of the SST, which is a measure of the quality of the approximation (see Fig. S.5). For Fig. 5C we used a data set of fixed size corresponding to the case with $N = 10^6$ in Fig. 5A. N_g was varied to investigate runtime dependency on this parameter. Quantifying the influence of the dimensionality of representation is more complicated, and we chose to first transform the 273 interatomic distances into principal components sorted by decreasing total variance (see Fig. 5B). For the full dimensionality, this has no impact on the results of the clustering or the SST construction. It does, however, allow a more straightforward reduction in dimensionality by simply discarding more and more of those dimensions with the smallest variance, which are presumed to encapsulate the least information.

For the data in Fig. 7, we used data on the identical system and physical model, but from a different set of simulations [7]. This is meant to facilitate comparisons to published work [7, 9, 3]. The data set is comprised of $N = 10^6$ snapshots saved at an interval of 20 ps, and the representation consists of the Cartesian coordinates of the backbone nitrogen and oxygen atoms of residues 3–18. A pairwise distance is defined as the root mean square deviation of atomic coordinates after pairwise alignment. It should be pointed out that the inclusion of translation and rotation operators poses technical challenges in the hierarchical clustering underlying the approximate approach as discussed [3]. The clustering used a tree height of 16, and a maximum and minimum threshold radius of 10.0 and 1.5 Å, respectively (identical to Fig. 6A in [3]). It yielded 161778 clusters, and the resultant network of conformational transitions was used to derive the τ_{MFP} annotations in Figs. 7, S.7, and S.8 as well as the cut-based free energy profile in Fig. S.7.

S.1.4.3. Hydrology Data Set

Because of storage constraints, it is difficult to find non-synthetic data on an accessible topic hosted on public servers such that there are continuous recordings of quantities or parameters with both time resolution and recurrence that allows one to make

statements regarding basins. Here, we have chosen river hydrology parameters (temperature in °C, pH (unfiltered), specific conductance in µS/cm at 25°C, discharge in cubic feet per second, and dissolved oxygen in mg/ml) available from the following stations in Oregon, USA:

Site number	River	Coordinates	Altitude
USGS 14211010	Clackamas River	45°22'46"N 122°34'34"W	0 ft.
USGS 14209710	Clackamas River	45°10'02"N 122°09'18"W	840 ft.
USGS 14138870	Fir Creek	45°28'49"N 122°01'28"W	1440 ft.
USGS 14138850	Bull Run River	45°29'54"N 122°00'40"W	1080 ft.

Table S1: The first column lists the station ID within the system of the United States Geological Survey (USGS) [10]. For each of the four stations, we here list the river (column 2) it measures along with complete geographic coordinates as latitude, longitude, and altitude (columns 3–4) [10]. The stations are listed in the same order as their data are shown in Figs. 6 and S.6. All these rivers are ultimately indirect tributaries to the Columbia river, which drains into the Pacific Ocean. The Portland area has mild, wet winters, comparatively cool summers, and is classified to be part of the cool, dry-summer subtropical (Csb) zone in the Köppen–Geiger classification system [11]. The rivers form part of the system that is relevant to the freshwater supply of the Portland metropolitan area and to the generation of hydroelectric power. In 2008, the lower basin of the Clackamas river was subject to a USGS investigation regarding river pollution from pesticides and herbicides based on data from years 2000–2005 [12].

These hydrology data are available with a temporal resolution of 30 minutes and over a period of about 5 years (from October 2007 to the present). In some cases, homogenization of the time axis required shifts of a few minutes for the actual time of measurement. Due to malfunctioning equipment, severe weather, or scheduled outages, data are incomplete (ca. 2.6% of points). In such cases, we interpolated linearly between the two measured data points bracketing a stretch of missing data. This is reasonable even for missing stretches of multiple days since river hydrology data are neither prone to strong random fluctuations nor to pronounced diurnal patterning. After homogenization and completion of the data, uniform noise was added to compensate for the lack of resolution in measurements. The width of the noise function was centered at the measured value and chosen in accordance with the dominant apparent resolution for each type of measurement. Discharge (streamflow) data were converted to logarithmic space before centering all the data. To achieve similar impact of each dimension, data were then normalized by their apparent standard deviations. These complete, centered, and normalized data are what is shown as color annotations in Fig. 6 of the main text and Fig. S.6.

S.2. Supplementary Results

S.2.1. Hydrology Data for Rivers Near Portland, Oregon

As outlined in the main text, we use the hydrology data (see S.1.4.3) for two main purposes: 1) highlighting data-dependent difficulties in applying the algorithm; 2) demonstrating the algorithm's utility on a real-world data set. There are some finer details regarding both points that are presented here instead of in 3.1 in the main text. To preserve clarity, some results are repeated here.

Figs. 6 and S.6 both reveal two major basins corresponding to warm and cold seasons, respectively. The cold season is the more heterogeneous of the two, and this is manifested predominantly by a broader range of discharge (flow) levels. 2008 appears to have been a year with anomalous conditions and is largely excluded from both major basins. The rest of the plot is partially comprised of a number of "entropic" regions constituted by mixed conditions from throughout the year. This is probably an aspect specific to data following an annual rhythm that generally cycles between two sets of extreme values, and it is expected that fall and spring are overrepresented in these "entropic" regions. The remainder are well-defined regions of homogeneous conditions that often come from specific years. As outlined in the main text, these tend to be resolved rather poorly in terms of the cut functions c or l on account of a lack of recurrence. Using the example of the winter and spring of 2008 found at progress index values of 5 to $6 \cdot 10^4$ in Fig. 6, we note that the conditions are unique with a very high pH at site #3 and very low water temperatures in winter. The linear correlation of progress index and real time indicates poor recurrence. This is because adding snapshots in their exact temporal sequence will leave the cut function invariant, *i.e.*, the number of transitions between sets S_i and $T \setminus S_i$ is constant for a range of consecutive i. Difficulties notwithstanding, the method allows identification of 2008 as a year with an unusually cold first half and friendly and dry weather deep into fall (see values for the progress index around $6.3 \cdot 10^4$) [13].

As alluded to in the main text, we also explore an alternative approach to the identification of barrier regions. This approach utilizes the locality of the progress index, *i.e.*, the difference in progress index position between a snapshot $i + 1$ and the snapshot in set S_i that it shares an MST edge with. This is discussed in detail in Fig. S.4 and its caption. With a suitable amount of averaging, this produces a plot that highlights putative barrier regions. These are then plotted as circles in Fig. 6. The aforementioned winter and spring basins of 2008 at progress index values of 5 to $6 \cdot 10^4$ are a good example for the utility of this approach. Specifically, the cut functions do not allow delineation of the winter basin from the data immediately to the left, whereas an identification via

nonlocality of the progress index is successful. In summary, these results emphasize the need to combine several annotations to extract meaningful information from a challenging data set.

As a final test, we want to utilize this realistic data set to evaluate how much the results depend on the spanning tree used to generate the progress index. Specifically, we are interested in identifying potential problems with the approximate algorithm that uses a parameter-dependent SST in place of the MST. First, we consider the SST on its own. In this context, the total weight of the spanning tree is a useful quantifier, given that this quantity is minimal for the MST. Fig. S.5 shows a comparative analysis for various choices of the number of guesses, N_g, used to construct the SST (see 2.3 in the main text and S.1.3 above). Clearly, N_g can be used to systematically decrease the weight of the SST. However, the value of the MST in not reached in an asymptotic manner, which must be on account of search restrictions introduced by data preorganization (see S.1.3). This means that the influence of the implied approximation on both progress index and annotation functions is difficult to predict. Fig. S.6 and its caption describe an example application of the approximate algorithm to the hydrology data. The conclusion is that, for this particular example, the SST actually provides better resolution in terms of the annotation functions, because it introduces artificial recurrence within basins. This comes at a cost, however, that is manifested in increased variability between plots starting from different snapshots and between plots starting from the same snapshot for different SSTs (not shown).

S.3. Supplementary Figures

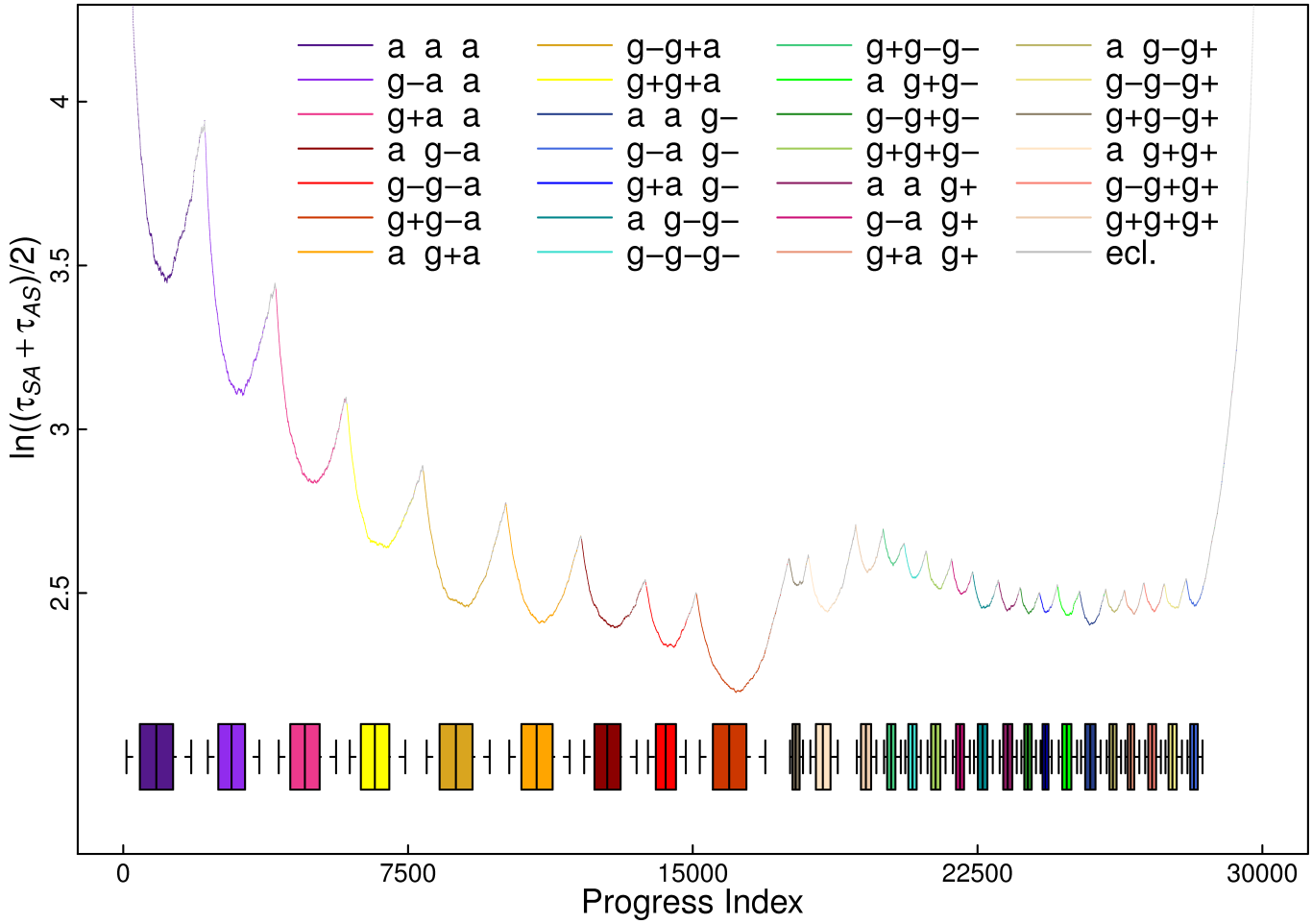

Figure S.1: Illustration of the approach using *n*-butane. This figure is largely similar to Fig. 2 in the main text. Our proposed approach yields a curve that resolves all 27 basins of the system. Microstates are annotated by color, and box plots are shown that quantify the distribution of snapshots annotated via the dihedral-angle based binning. For each basin, a box is drawn indicating the interval that the central 50% of the snapshots belonging to that basin are confined to. Whiskers indicate the central 90% of the snapshots in that basin. Medians are shown as black, vertical lines. Compared to the maxima in the curve, boxes and medians appear skewed to the left. This emphasizes that within each basin eclipsed microstates are concentrated toward the right (larger progress index), which is a natural result of the way the progress index is constructed. This is also qualitatively apparent from the gray dots indicating such eclipsed microstates. The implied unit of time on the y-axis is a single snapshot, *i.e.*, 250 fs.

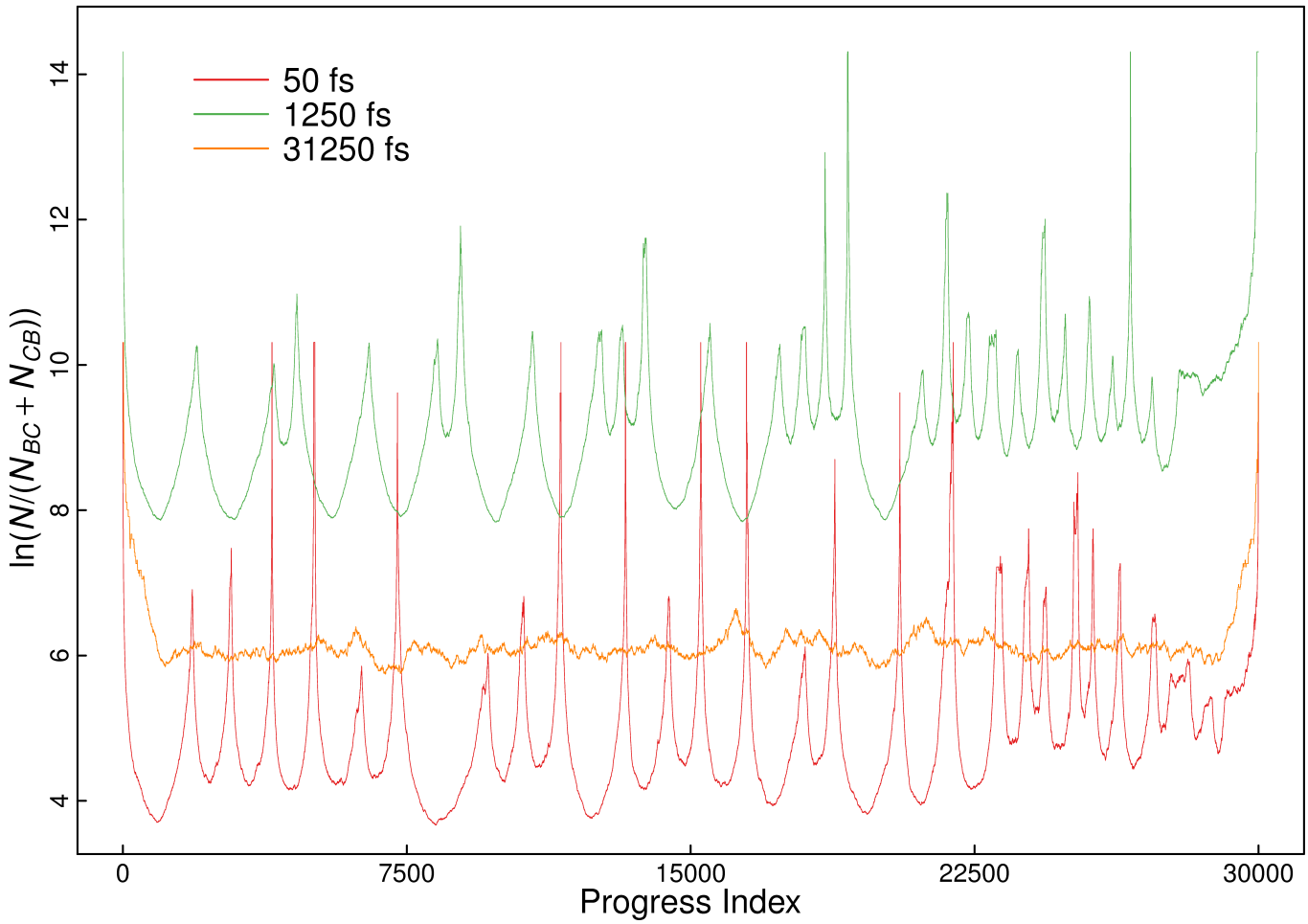

Figure S.2: Influence of temporal resolution on the alternative cut function l with fixed n_l of 1000. This figure is similar to Fig. 3 in the main text, but only three cases are shown. The profile at 1.25 ps is shifted by 4 units for better readability. In analogy to Fig. 3, it can be seen that at 31.25 ps the profile loses its salient features. In contrast to Fig. 3, however, the underlying prior function appears to be flat for all points outside of the first and last n_l points. The cases with finer time resolution therefore highlight two important advantages of annotation function l when compared to function c (see Fig. 3). First, the lack of inherent curvature improves the ability to resolve basins. Second, the localization of the cut improves the signal-to-noise ratio when considering the ratio of values at barriers to those in the bottom of basins. Both advantages are contingent upon finding appropriate values for n_l. For each curve the implied unit of time on the y-axis is a single snapshot of the respective trajectory, *i.e.*, the saving frequency or temporal resolution itself.

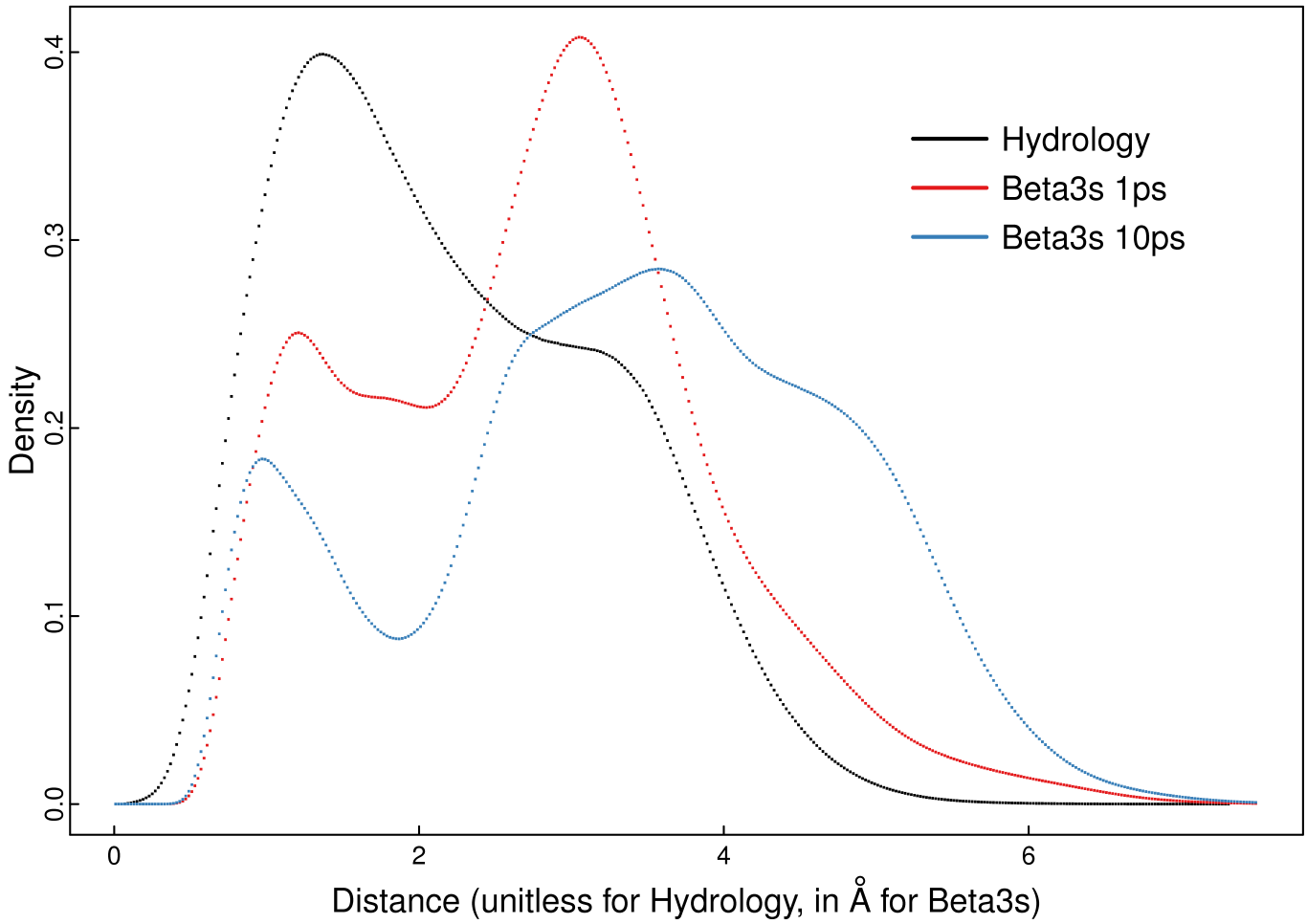

Figure S.3: Histograms of pairwise distance for the hydrology data set in comparison to protein folding data sets. As discussed in Section 3.1 of the main text, the distance spectrum for the hydrology data is expected to be relatively featureless. This is confirmed by the comparison shown here to continuous data obtained for the Beta3S miniprotein [8] at two different time resolutions. In all cases, all possible, unique distance values are computed for $N = 87840$. The figure shows that the hydrology data give rise to a dominant fraction of similar pairs of snapshots. This creates degeneracy in establishing near-neighbor relations that the MST relies on. Even for a total length of just 87.84 ns, the protein data exhibit a much smaller fraction of similar pairs providing more meaningful neighbor relations. This is despite the fact that 87.84 ns are not enough to ensure recurrence *between* major basins. The spectrum becomes increasingly discriminatory if the total time considered is increased (here, up to 878.4 ns).

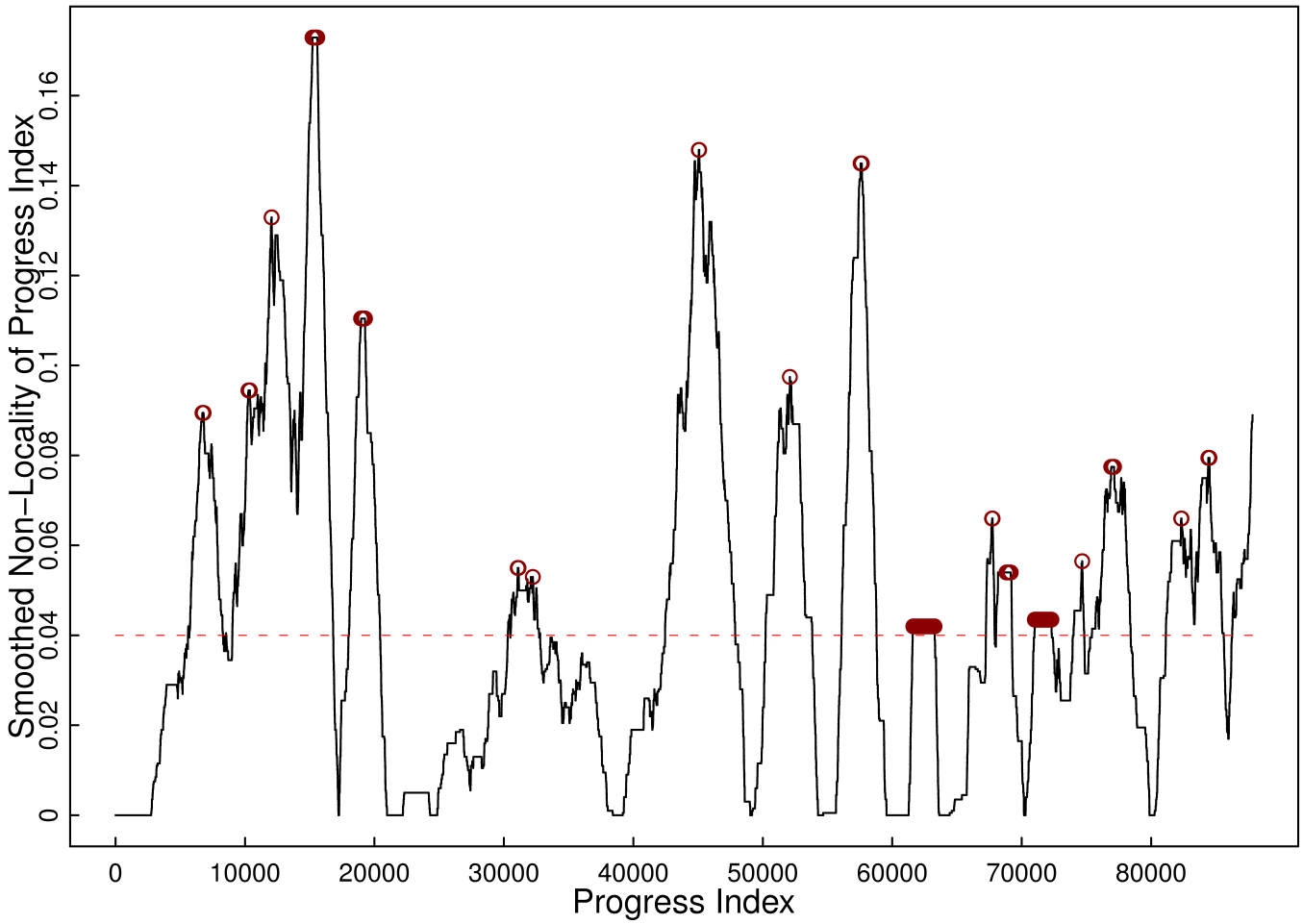

Figure S.4: Identification of barrier regions via non-locality in progress index generation. As discussed in Sections 3.1 of the main text and S.2.1 above, the hydrology data have a poor signal-to-noise ratio, which makes it challenging to identify basins purely based on the annotation functions c or l. During progress index generation, the MST provides information regarding the source or parent vertex (snapshot) that the currently added snapshot is indeed closest to. The value of the progress index for this parent is also known (necessarily smaller), and the difference provides information whether parent and child are likely to be members of the same basin. With a difference threshold of 2000 snapshots, we generate a bit-sequence (1/0) of nonlocality. This bit sequence is then smoothed using (sliding) window averaging with a window size of 2000 snapshots, and the resultant curve is plotted here. Clearly, there are well-defined regions where the function peaks, and we can define a threshold (red dashed line) to select a number of candidate points. It is important to note that the sliding window is not centered at each point, but rather extends only to the left (lower progress index). This is to compensate for the intrinsic property of the progress index in accumulating fringe and transition region points at the right end (toward higher progress index). Due to construction, several points within a sliding window may have the same maximal value. In such a case, we consider only that point yielding the maximal value of the annotation function of interest (c or l).

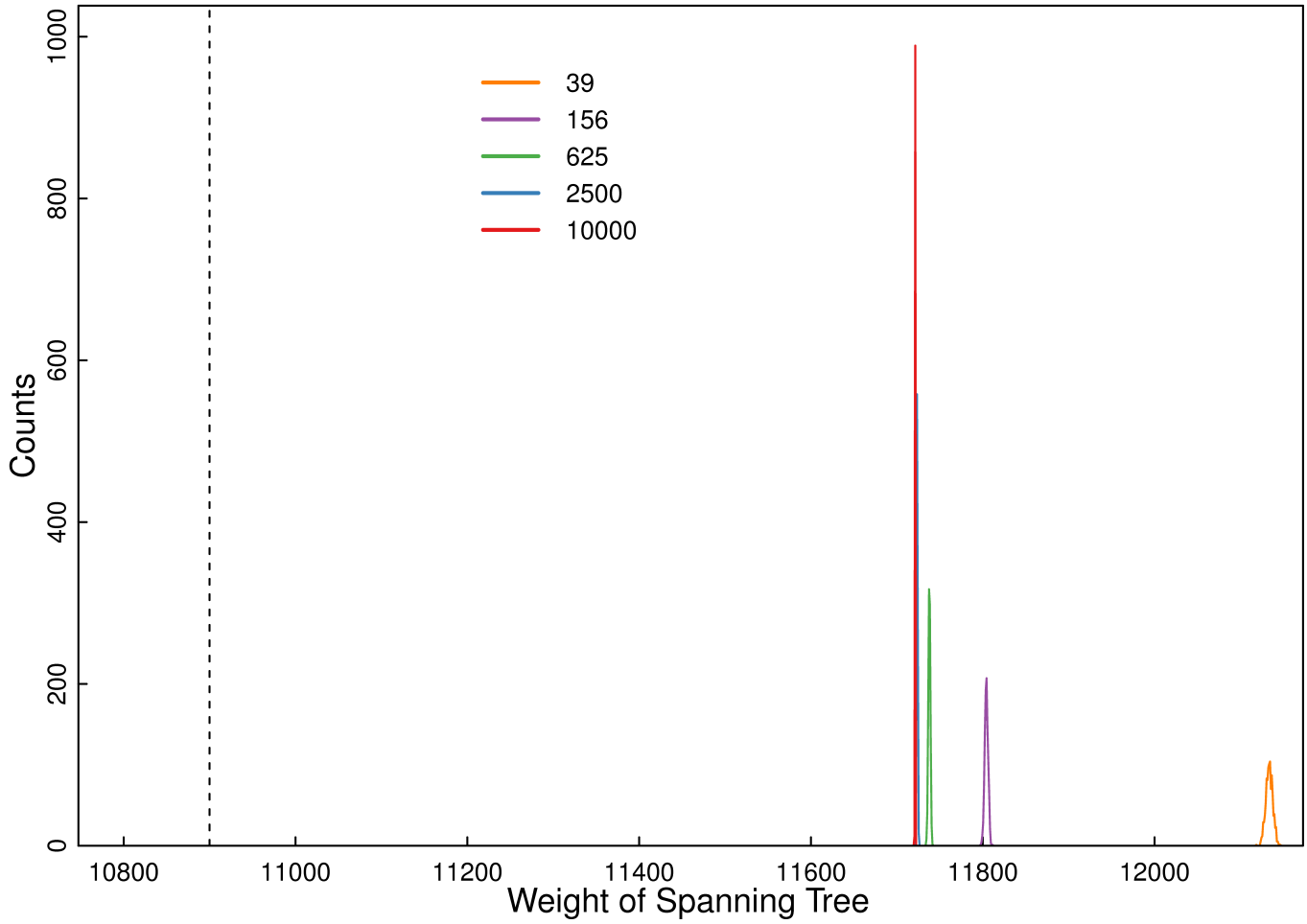

Figure S.5: Quality of the short spanning tree (SST) as a function of the number of guesses. Since the approximate algorithm has a random component and a parameter that is expected to control the total weight of the SST in systematic fashion, this figure shows histograms from 1000 samples each for various choices of N_g. The underlying data are the hydrology data as described in Section S.1.4.3. As expected, larger values for N_g systematically shift the mean of the corresponding histogram toward lower total SST weights. The effect is more pronounced toward low values of N_g with a level of saturation being reached toward high values. The stochasticity of results is also expected to decrease with increasing N_g, and this is evident in the decreasing width of histograms. Lastly, the unique total weight of the MST is indicated as a vertical, dashed line. Clearly, the MST does not appear to coincide with the asymptotic limit for $N_g \longrightarrow \infty$. There are several putative reasons for this, but most likely it is a direct result of the data preorganization exploited during SST construction that limits the search space for a new edge.

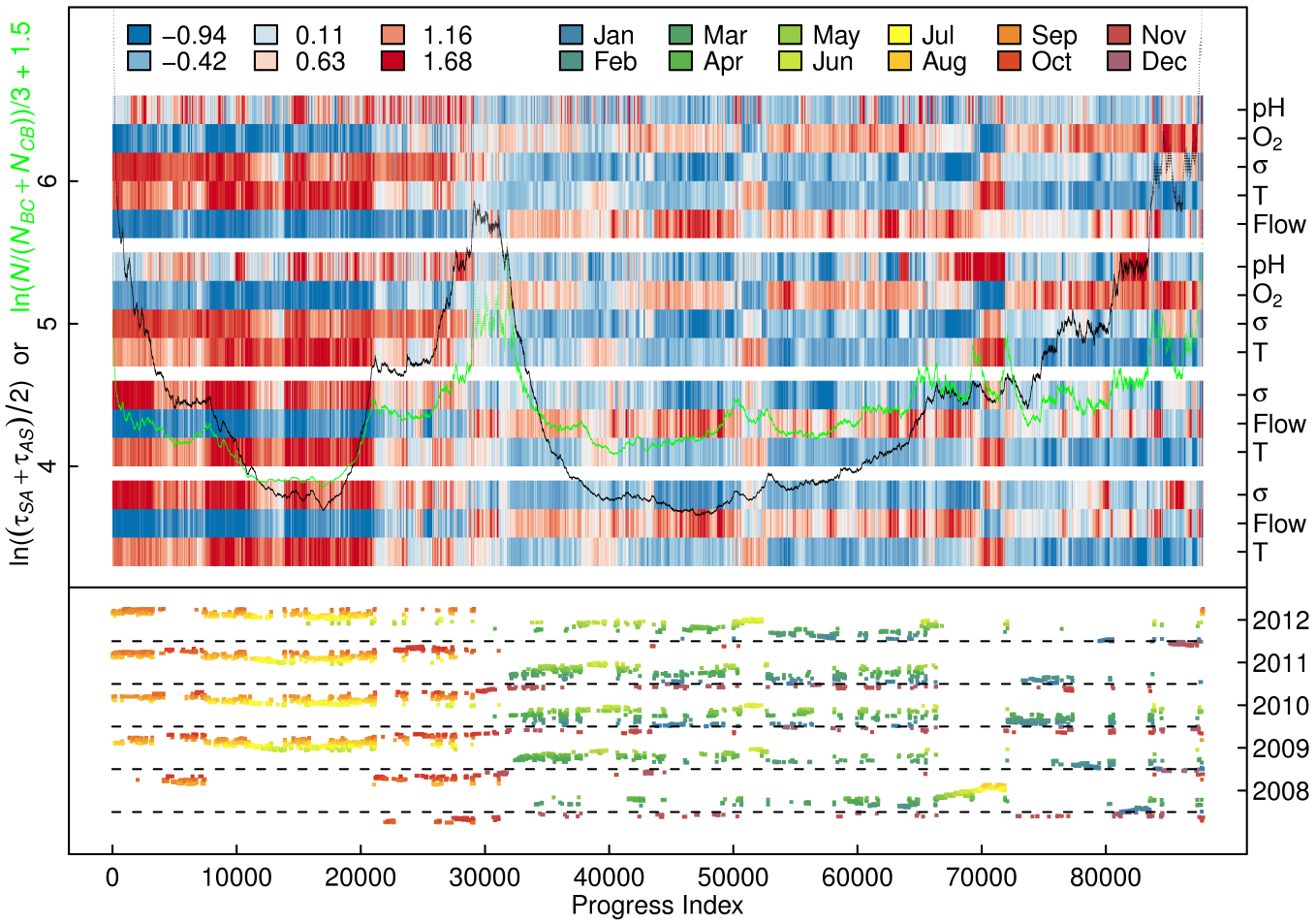

Figure S.6: Comparison of exact and approximate algorithm for the hydrology data set. This figure is analogous to Fig. 6 in the main text, and the reader is referred to the caption of Fig. 6 for understanding what is plotted. Here, we show a progress index generated from the same starting snapshot as in Fig. 6 with the approximate algorithm using a setting of $N_g = 1000$ along with the resultant annotations. The parameters for the tree-based clustering were chosen as 24, 7.0, and 0.36 for the tree height, coarsest threshold, and finest threshold, respectively. The resultant SST had a total weight of 11987 compared to the MST weight of 10906. The plot shows a similar partitioning and arrangement of basins to Fig. 6 with the warm months first followed by a broad basin of the cold season. The remainder of the plot is a series of smaller basins often restricted to individual years. There are two major differences compared to Fig. 6. First, here the transition seasons are more closely grouped and/or are integrated into the large basins leading to a lack of "entropic" regions. Second, the delineation of basins using function l and in particular using function c is much improved. Note for example how close to progress index values of $7 \cdot 10^4$ the spring and mid-summer basins of 2008 are resolved and separated from one another in both annotation functions. This is likely the result of randomness in picking the next snapshot within a group of similar microstates, which creates "artificial" recurrence. It is important to point out that this is neither automatically the case (see winter basin of 2008 just beyond values of the progress index of $8 \cdot 10^4$) nor necessarily desirable. Overall, this figure and Fig. 6 in the main text are similar, however.

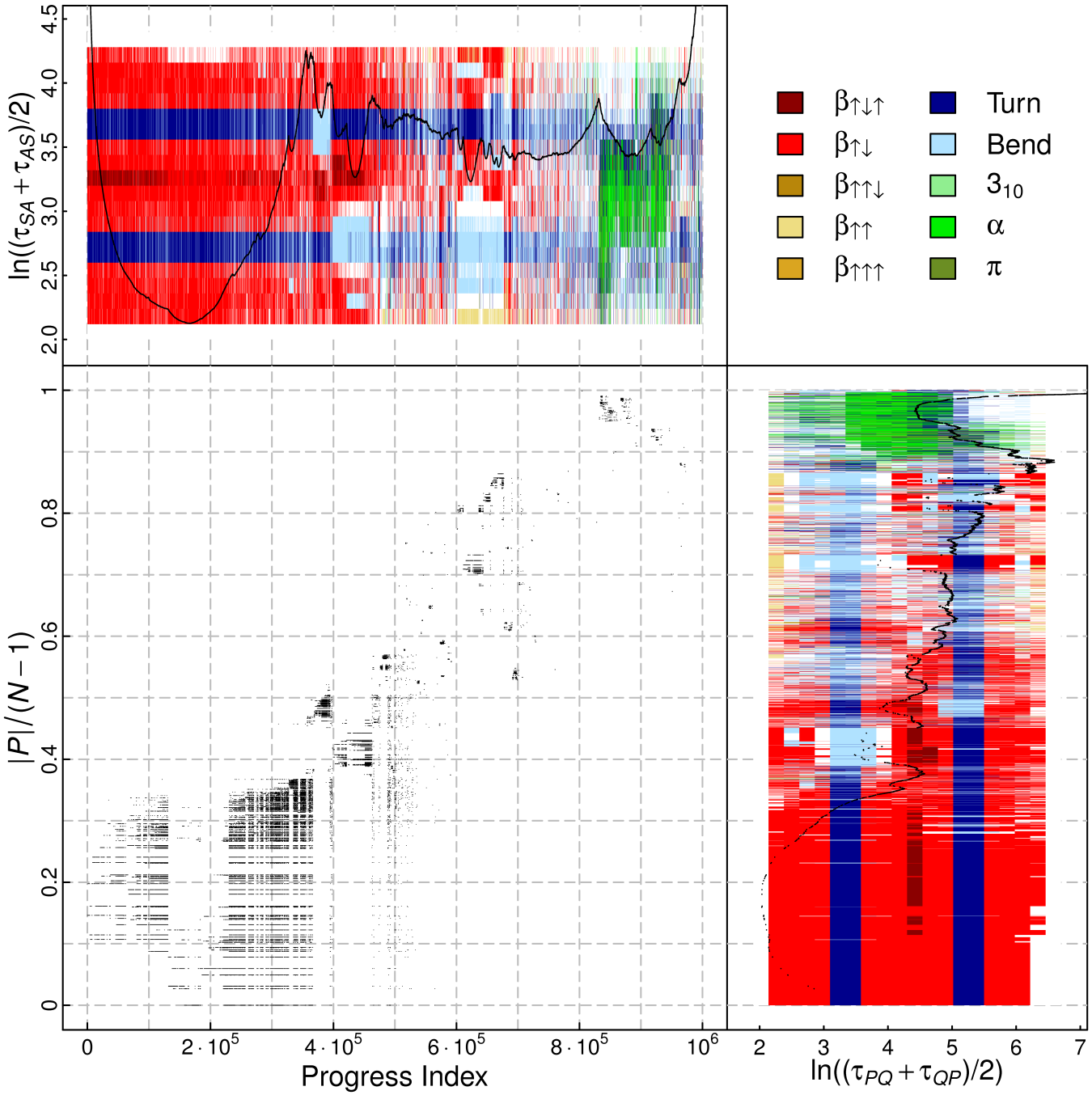

Figure S.7: Comparison of proposed scheme to cut-based free energy profile for Beta3S. The top portion of the plot reproduces the annotated progress index found in Fig. 7 of the main text. Only annotations with function c and DSSP secondary structure assignments [14] are shown for every 20th snapshot. In the DSSP case, this implies that the plotted DSSP strings are extended correspondingly, which means that they are not necessarily exact for 95% of snapshots. At typical resolutions, the similarity is high enough, however, that the visual appearance is not altered by this (this also holds for Figs. 7 and S.8). The right-hand side shows an analogous plot with a cut-based free energy profile [15] instead. Here, the progress index is replaced by a cumulative probability ($|P|/(N-1)$) computed from the equilibrium probabilities of all the nodes of the underlying conformational network that are part of the growing set P. The ordering principle is kinetic distance from a reference state (here, the native basin). A network cut is used to annotate this sequence of kinetically ordered nodes, and the similarity is apparent. To reduce image size, annotations are only shown for those clusters with at least 10 members (they encompass about 74% of snapshots). For DSSP color annotations, this means specifically that the plotted DSSP string of the centroid of a given cluster extends until the integrated weight reaches a new cluster of a minimum size of 10. Because the omitted clusters are tiny, the visual appearance of the plot is not altered at typical resolutions. The lower left portion plot shows a scatter plot emphasizing the correspondence between network nodes of size 100 or

larger and the positions of their constituent snapshots in the progress index. By means of the DSSP annotations, it is easily seen that the same basins are resolved with the same widths (total weights). This suggests that the proposed algorithm is unlikely to suffer from false positives or false negatives in terms of partitioning the data into basins if the underlying trajectory is sufficiently recurrent. Additionally, clusters of points are largely close to the diagonal indicating good correlation between the explicit, kinetic ordering and the progress index for this system. The implied unit of time for the mean first passage times is one snapshot, *i.e.*, 20 ps.

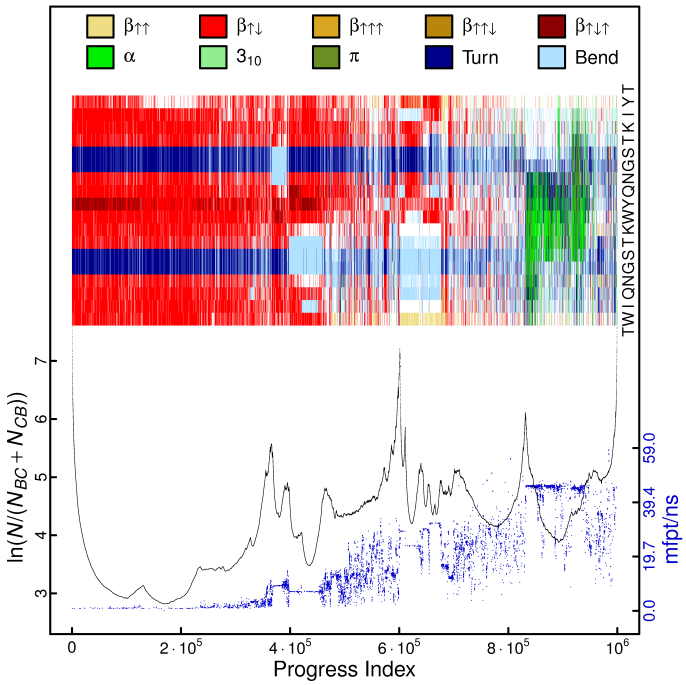

Figure S.8: Results for Beta3S with the alternative cut function l and fixed n_l of 10^5. This figure is identical to Fig. 7 in the main text with the exception that the primary annotation function is not c, but rather l (most importantly, it uses the exact same SST). Because of the wide distribution of sizes of basins for this system, a flat choice for n_l is not expected to enhance resolution uniformly. Interesting changes compared to Fig. 7 involve a pronounced barrier at $6 \cdot 10^5$ and substructure within the native basin, the latter of which is more apparent here, but also discernible in Fig. 7. While not clearly visible in the plot, the data allude to the fact that, for small enough and fixed n_l, the number of direct transitions between partitions B and C starts to *decrease* for increasing basin size. This leads to a flattening effect that makes it difficult distinguish large basins from "entropic" regions, and this effect is much more apparent for $n_l = 2 \cdot 10^4$ (not shown).

References

[1] J. B. Kruskal Jr., On the shortest spanning subtree of a graph and the traveling salesman problem, Proc. Am. Math. Soc. 7 (1956) 48–50.

[2] J. Nešetřil, E. Milková, H. Nešetřilová, Otakar Borůvka on minimum spanning tree problem, Translation of both the 1926 papers, comments, history, Discrete Math. 233 (2001) 3–36.

[3] A. Vitalis, A. Caflisch, Efficient construction of mesostate networks from molecular dynamics trajectories, J. Chem. Theory Comput. 8 (2012) 1108–1120.

[4] W. L. Jorgensen, D. S. Maxwell, J. Tirado-Rives, Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids, J. Am. Chem. Soc. 118 (1996) 11225–11236.

[5] P. Ferrara, J. Apostolakis, A. Caflisch, Evaluation of a fast implicit solvent model for molecular dynamics simulations, Proteins: Struct., Funct., Bioinf. 46 (2002) 24–33.

[6] S. Muff, A. Caflisch, Kinetic analysis of molecular dynamics simulations reveals changes in the denatured state and switch of folding pathways upon single-point mutation of a β-sheet miniprotein, Proteins: Struct., Funct., Bioinf. 70 (2008) 1185–1195.

[7] S. V. Krivov, S. Muff, A. Caflisch, M. Karplus, One-dimensional barrier-preserving free-energy projections of a β-sheet miniprotein: New insights into the folding process, J. Phys. Chem. B 112 (2008) 8701–8714.

[8] T. Zhou, A. Caflisch, Free energy guided sampling, J. Chem. Theory Comput. 8 (2012) 2134–2140.

[9] B. Qi, S. Muff, A. Caflisch, A. R. Dinner, Extracting physically intuitive reaction coordinates from transition networks of a β-sheet miniprotein, J. Phys. Chem. B 114 (2010) 6979–6989.

[10] U.S. Geological Survey, Site inventory for the nation, http://waterdata.usgs.gov/nwis/inventory, accessed October 26, 2012.

[11] M. Kottek, J. Grieser, C. Beck, B. Rudolf, F. Rubel, World map of the Köppen-Geiger climate classification updated, Meteorol. Z. 15 (2006) 259–263.

[12] K. D. Carpenter, S. Sobieszczyk, A. J. Arnsberg, F. A. Rinella, Pesticide occurrence and distribution in the lower Clackamas river basin, Oregon, 2000–2005, Scientific Investigations Report 2008-5027, U.S. Department of the Interior – U.S. Geological Survey, 2008.

[13] WeatherSpark, Historical weather for 2008 in Portland, Oregon, USA, http://weatherspark.com/history/30477/2008/Portland-Oregon-United-States, accessed October 29, 2012.

[14] W. Kabsch, C. Sander, Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features, Biopolymers 22 (1983) 2577–2637.

[15] S. V. Krivov, M. Karplus, One-dimensional free-energy profiles of complex systems: Progress variables that perserve the barriers, J. Phys. Chem. B 110 (2006) 12689–12698.