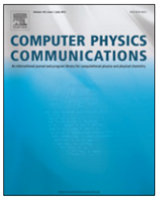
Computer Physics Communications 184 (2013) 2446–2453

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



A scalable algorithm to order and annotate continuous observations reveals the metastable states visited by dynamical systems

Nicolas Blöchliger, Andreas Vitalis *, Amedeo Caflisch

Department of Biochemistry, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 30 October 2012
Received in revised form
7 June 2013
Accepted 12 June 2013
Available online 25 June 2013

Keywords:
Complex system
Trajectory analysis
Scalable algorithm
Minimum spanning tree
Free energy basins

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 $\mathcal{O}(N \log N)$ 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.

1. Introduction

With present day computing resources, large-scale temporal simulations of complex systems can be performed routinely, and time-resolved, experimental data in many dimensions are collected and stored. In both cases, the resultant, very large amounts of data require dedicated, scalable protocols to handle access and analysis [1–3]. Examples can be found in fields such as protein science [4,5], astronomy [6], cell biology [7], or climatology [8] to name just a few.

For a complex system evolving in time, data are present in the form of sequences of instantaneous snapshots (microstates in the language of statistical mechanics) of this complex system, and such a sequence will be referred to as a trajectory throughout. Depending on whether data are synthetic or real, the implied projection of the system to obtain a snapshot may differ, and this may limit spatial resolution. Temporal resolution is limited directly by the instruments or numerical schemes if storage space is not a concern. Even though continuous evolution need not be observed explicitly as a function of time, we will restrict our terminology to this case. Routine analyses of trajectory data may involve computing average properties and their estimated distribution functions in $\mathcal{O}(N)$ time, where N is the number of snapshots. Distribution functions offer hints toward the diversity of states visited by the complex system and their relative weights. Time-resolved analyses provide insight regarding state connectivity and transition rates. Projection onto low-dimensional properties is necessary to render such analyses statistically meaningful and visualizable by conventional means.

If we assume that snapshots follow a well-defined distribution (such as the Boltzmann distribution for particles in the classical limit), these analyses look for spatial domains that are highly populated under the given conditions, *i.e.*, those for which a finite sample yields higher-than-average densities of microstates, preferably through recurrence [9]. Here, recurrence refers to the trajectory's property of entering and exiting subdomains within high density regions several times. The motivation behind this is twofold: (1) characterization of the complex system and communication of results in terms fit for human consumption [10]; (2) derivation of simplified models that provide a meaningful representation of the complex system [11,12]. Such models can preserve coarse-grained dynamical and static properties of the system and enable predictions to be made over vastly extended temporal or spatial domains.

When analyzing trajectories in projected spaces, high density regions are prone to overlap, and plots rarely resolve all of them [13]. This overlap phenomenon can lead to incorrect conclusions regarding the diversity and connectivity of coarse states. Consequently, affordable protocols that require little knowledge of the system *a priori* and that decrease the likelihood of such overlap are of interest. Techniques such as principal component analysis,

* Corresponding author. Tel.: +41 446355597; fax: +41 446356862.
E-mail addresses: n.bloechliger@bioc.uzh.ch (N. Blöchliger), a.vitalis@bioc.uzh.ch (A. Vitalis), caflisch@bioc.uzh.ch (A. Caflisch).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.06.009

spectral clustering [14] and the related diffusion maps [15], locally linear embeddings [16], cut-based free energy profiles [17], kinetic groupings based on networks [18–21], which are specific cases of community detection algorithms in graphs [22], *etc.* are all in use, but many of them scale superlinearly with N.

Data clustering [23] offers a simple route to the identification of high density domains. Clusters are defined as groups of mutually similar snapshots. Similarity is assessed by a criterion of distance generally requiring an *ad hoc* selection of both a subset of features [24] and a functional form. However, a grouping meant to describe an evolving system should also encode dynamic proximity [25], *i.e.*, given a time resolution, which snapshots constitute a kinetically distinct state? If the system is of atomic scale and at equilibrium, this question aims to identify free energy basins and barriers in a generally high-dimensional phase space [26,27]. Positional coordinates of atoms are often used exclusively given that momenta can likely be ignored out on account of their much shorter autocorrelation times. We note that the language and concepts of statistical physics have proven useful in the analysis of nonphysical systems as well [28], *i.e.*, our adaptation of this language does not imply a restricted domain of application.

In this contribution, we present an algorithm that operates directly on a trajectory. With just the definition of a pairwise distance between snapshots, we are able to generate a one-dimensional plot that allows the identification of states in a joint geometric and kinetic sense, which we will refer to as basins. With standard metrics derived from microstate representations (such as interatomic distances in a flexible molecule), the method relies on the continuity of geometric representations in time. This implies that it may fail for certain classes of discrete systems. The main benefits of our algorithm are that it does not rely on any parameters *per se*, that it is very likely to resolve all basins, and that with the help of reasonable approximations to the exact procedure, the total running time approaches $\mathcal{O}(N \log N)$. The combination of these points is worth emphasizing, since we believe that they constitute a desirable and unique fingerprint of our approach.

The rest of this manuscript is structured as follows. First, we present the key ideas behind the procedure (Section 2.1) and illustrate its utility with a suitable model system (2.2). Next, we describe a computationally efficient and robust approximation to the exact procedure. The scaling of computational cost with data set size and dimensionality is tested explicitly (2.3). This is followed by applying the method to two complex real-world data sets, the first from hydrology (3.1) and the second from protein folding (3.2). We conclude by discussing the advantages and possible problems in comparison with related approaches (4).

2. Methods and proof of concept

2.1. The exact algorithm

Let $T = \{t_1, \ldots, t_N\}$ be a set (trajectory) of N unique snapshots, which usually are representations of the system in $\mathbb{R}^D$, which is the chosen subspace of the original system representation with $D \leq D_{system}$. We use any pairwise distance $d : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}_{\geq 0}$ to measure the similarity between two snapshots. This need not be a purely coordinate-dependent function. Below it will prove beneficial for d to be a metric yielding a continuous number space with all $\mathcal{O}(N^2)$ values of d being unique.

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

1. Define s_{i+1} as the snapshot in $T \backslash S_i$ realizing the minimum of $d(\cdot, S_i) = \min_{j=1,\ldots,i} d(\cdot, s_j)$.

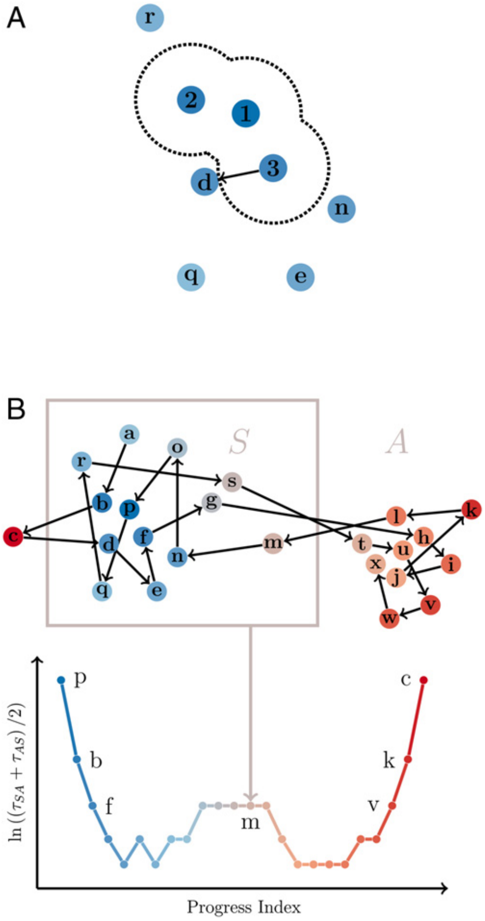


Fig. 1. Schematic highlighting the fundamental components of the algorithm. **A**. A set of points in two dimensions is shown as circles. See 2.1 for details. **B**. The points in **A** are shown as a subset of a larger data set. Arrows and letters indicate progression in time. The color scheme follows the order in which points are added when starting with point **p**, *i.e.*, colors trace the progress index itself. The schematic on the bottom shows values for the inverse logarithm of c at each value of the progress index. An example point and the cut to obtain the respective partitions S_i and A_i are highlighted. Point **c** illustrates an outlier, which are prone to be added last to S. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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, the function ζ is defined as

$$\zeta_X(t,u) = \begin{cases} 0 & \text{if neither or both } t \text{ and } u \text{ are part of set } X \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

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 words, given a starting snapshot, the algorithm finds a unique ordering of the snapshots, and annotates it with the number of transitions between the two partitions defined by all the snapshots that are currently part of the set (S_i) and those that are not yet part of the set ($A_i = T \backslash S_i$). The cut function c is related to the mean first passage time in the implied two-state Markov model via

$$\tau_{\mathrm{MFP}}(A_i \rightarrow S_i) + \tau_{\mathrm{MFP}}(S_i \rightarrow A_i) = 2N/c(i). \tag{2}$$

We use τ_{AS} as shorthand notation for $\tau_{\mathrm{MFP}}(A_i \rightarrow S_i)$ throughout. In Fig. 1(A), we show an illustration of a trajectory in 2D space with the current set of snapshots 1–3. The order of adding further snapshots would then be **d**, **n**, **r**, **e**, and **q** based on the mutual distance relations. There are no free parameters beyond having to