Estimating the Dynamics of Biological Systems from Unordered Samples

Paul M. Magwene (Email: paul.magwene@yale.edu), Paul Lizardi, and Junhyong Kim

1Department of Ecology and Evolutionary Biology, 3Department of Molecular, Cellular, and Development Biology, 4Department of Statistics, Yale University
2Department of Pathology, Yale University School of Medicine

Introduction

Biological systems are inherently dynamical. A common approach to studying the behavior of dynamic biological phenomena is to sample individuals, tissues, or other relevant units at intervals throughout the temporal progression of the system under study. An ordered collection of such samples is called a time-series. Time-series measurements are often critical data in the study of dynamic biological phenomena such as cell cycles, developmental programs, and disease. Well-characterized time-series can be used as benchmarks (e.g. cell and developmental stages) for measuring the temporal progress of biological systems, or as diagnostic tools for assessing and treating disease, and provide a basis for comparative studies of such phenomena among closely related taxa.

Several problems associated with sampling from dynamic biological systems—synchronization, temporal sampling, and rate heterogeneity—often make it difficult to obtain accurate time-series. Both for methodological and analytical reasons time-series data are usually drawn from a population. Unless all members of the population are synchronized, the samples will contain mixtures of the temporal process. Even if synchronization is possible rate heterogeneity among the members of the population will further complicate the temporal samples. Rate heterogeneity can arise from both intrinsic and extrinsic factors, and can lead to situations in which orderings of samples which are correct with respect to absolute time are incorrect with respect to the dynamics of the biological process under study. A more difficult problem arises when explicitly time indexed samples can not be acquired. For example, cancerous cells are only identified at their late stages. If we want to investigate the early process of carcinogenesis, we must blindly sample from the cell population and hope to reconstruct the temporal sequence.

Newly available microarray data provides us with a novel approach to this ab initio temporal ordering problem for cell populations. Microarray data is multi-dimensional with large numbers of measurements. The large measurements arrays for each cell or collection of cells for a given time point can be used to reconstruct the temporal ordering assuming that the temporal changes in the transcriptome are relatively
smooth and continuous. Here, we present an algorithmic method for estimating temporal orderings from unordered sets of sample elements. The approach we advocate is based on modifications of a minimum spanning tree calculated from a weighted, undirected graph, and provides useful heuristics for assessing relevant properties of samples such as noise and density. We demonstrate the efficacy of our approach by applying these techniques to artificial datasets and to gene expression data derived from DNA microarray experiments.

**Ordering and Curve Reconstruction**

Approaches to ordering sample elements have been developed in a number of different fields. The alternate approaches and methodologies that are in use reflect differences in the types of data, the quality of sampling, the intended applications or uses for the reconstructed orderings, as well as the historical development of particular fields of study.

If one assumes that the biological process under study can be treated as a continuous function with respect to time, then the problem of ordering samples is equivalent to the problem of reconstructing vector-valued functions of the form, \( \tilde{f}(t) = [x_1(t), x_2(t), \ldots, x_d(t)] \), where each \( d \)-dimensional point on the function represents the state of a system at a particular point in time. Estimating the geometry of \( \tilde{f}(t) \) from finite points has been referred to as the curve reconstruction problem. The techniques that have been applied in such contexts can be roughly categorized into two classes – polygonal reconstruction approaches and principal curves techniques.

Polygonal reconstruction techniques have been the purview of computational geometers because of their application to problems in computer vision, image processing, and pattern recognition. If \( F \) is a smooth, twice differentiable curve in \( \mathbb{R}^d \), and \( S \) is a finite sample of points on \( F \), then Amenta et al. [Amenta, 1998 #16] defined a polygonal reconstruction of \( F \) from \( S \) as a graph that connects every pair of samples which are adjacent on \( F \), and no others. Two points, \( \tilde{f}(t_0) \) and \( \tilde{f}(t_1) \) are defined as being adjacent if no point in \( S \) exists that is a point on the arc \( \{ \tilde{f}(s) : t_0 \leq s \leq t_1 \} \). That is, adjacency, here and below, refers to adjacency in terms of the time parameter.

A variety of different algorithms for generating polygonal reconstructions have been advocated, but common to these are the assumptions that the observations in the sample are drawn from a smooth curve embedded in \( \mathbb{R}^d \) dimensions, that the points are sampled without error, and that sampling density is
sufficient to achieve reconstruction [Amenta, 1998 #16; Figueiredo, 1995 #8; Dey, 1999 #10]. Recent work by Giesen [Giesen, 1999 #6; Giesen, 2000 #11] and Althaus and Melhorn [Althaus, 2000 #9] has demonstrated that the requirement of smoothness can be relaxed somewhat; requiring only that curves be “benign semi-regular” (i.e. every point on the curve has left and right tangents and a turning angle less than \( \pi \)). Giesen was able to show that, given sufficiently dense sampling, the traveling salesman path (TSP) provides the correct reconstruction for curves in \( \mathbb{R}^d \). A similar result holds for the minimum spanning tree (MST) – asymptotically the minimum spanning tree is a unique path, and given sufficiently dense sampling a Euclidean MST correctly reconstructs differentiable arcs [Figueiredo, 1995 #8].

While the TSP or MST methods work well when applied to samples sets that are observed without error, the presence of noise introduces undesirable results to the polygonal reconstruction algorithms. For example, even when noise is distributed uniformly around \( F \) then polygonal reconstructions tend to zig zag back and forth through the data cloud.

The second major class of algorithms which have been applied to the curve reconstruction problem are variants of the Principal Curve algorithm first described by Hastie and Stuetzle [Hastie, 1989 #14]. Hastie and Stuetzle defined a principal curve as “self-consistent” smooth curve that passes through the middle of a \( d \)-dimensional data cloud. Principal Curves and related techniques explicitly assume that the observations in the sample represent points sampled with noise [Hastie, 1989 #14; Kégl, 2000 #15]. While the assumptions about the sampling noise are different between polygonal reconstruction and Principal Curve techniques, both classes of algorithms require a certain sampling density to insure good reconstructions.

The Ordering Problem in a Time Series

As stated above, we assume that there is a time-parameterized vector valued function, \( \bar{f}(t) \), which is sampled at a finite number of \emph{unknown} time points with error. We also allow for multiplicity in the sampling such that the collection may contain multiple samples of the same time point. Let \( E = \{\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_n\} \) be the collection of observed samples. For each observation, \( \bar{f}_i \), let \( s_i \) be the unknown time index of the \( i \)th observation. Then we have:

**Definition:** A permutation \( \pi \) of the index set \{1, 2, \ldots, n\} is a temporal ordering of the points, \( V = \{\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_n\} \), if \( \pi(i) \leq \pi(j) \Rightarrow s_i \leq s_j \) for all \( i, j \) in the index set.
The ordering problem is to find the temporal ordering permutation, $\pi$, given the data $V = \{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_n\}$. Note that this is a reduced problem from curve reconstruction. In this formulation we do not care about the details of the curve but only the temporal ordering of the observed points.

The time series $\vec{f}(t)$ is an one-dimensional curve embedded in the space of our measurements; e.g., gene expression values. Let the geometry of the measurements be determined by some quadratic form determining the notion of a distance, say the standard Euclidean inner product. Given two points, $\vec{f}_a$ and $\vec{f}_b$, if we were somehow able to measure the arc length, $l_{a,b} = \int_a^b \sqrt{1+[f''(t)]^2} \, dt$, of the curve between $\vec{f}_a$ and $\vec{f}_b$, then the ordering could be found by simple sorting of the arc length distances between the points. However, the only information we have is the embedded geometry and if the curve has significant curvature it will be difficult to approximate the arc length from the embedded geometry except for short segments (Fig X).

These observations motivate the following “rules” for estimating arc length distances from distribution of the observed data:

- If two points, $\vec{f}_a$ and $\vec{f}_b$, are relatively similar (compared to the dissimilarity of the other pairs of observations in the dataset), then the appropriate distance between them is the the standard “pairwise” dissimilarity measure.

- If $\vec{f}_a$ and $\vec{f}_b$ are relatively dissimilar, then the appropriate distance between them is best approximated by a “pathwise” dissimilarity measure – the sum of a series of short pairwise “hops” between points intervening $\vec{f}_a$ and $\vec{f}_b$.

A similar approach to estimating distances among data points was proposed by Tenebaum et al. [Tenenbaum, 2000 #5] who argued that the use of geodesic manifold distances between pairs of data points helps to preserve the intrinsic geometry of the data.

The key idea in the polygonal curve reconstruction problem is to use pair-wise distances from the embedded geometry to link up the points by short segments. But, as noted above, this does not take into account possible noise in the measurements, such that actual observed points are of the form $\vec{f}(s) + \delta(t)$.
where $\delta(t)$ represents a noise parameter. Figueiredo and Gomes [Figueiredo, 1995 #8] suggested that when noise is present, rather than finding a path, one should find a tree graph (namely the minimum spanning tree) and consider the diameter path (longest path) of the tree for the curve reconstruction. However, this ignores the fact that when the curve has high curvature relative to sampling density, segments of the actual path may be present in the branches off the diameter path.

Here we modify Figueiredo and Gomes’s idea by suggesting that the structure of the tree graph may be analyzed to delineate the noise component from the path component. Briefly, we assume that we are initially given a complete graph between the $n$ sampled points with edge weights given by the pair-wise distance function in the embedded geometry. A sub-graph is constructed that is the MST. The diameter path is considered an initial estimate of the path through points and then the non-diameter path branches of the MST are analyzed to determine whether they represent noise deviations or possible components of the actual path. We expect noise branches to be relatively short in length compared to the diameter path and uniformly distributed. Path component branches, in contrast, are expected to be relatively long. Any path component branches are incorporated into the diameter path by a “short cut” algorithm we describe below.

**MST as a Tree Graph Reduction**

As mentioned, the first step in the algorithm is to generate a tree graph reduction of the potential paths through the input weighted complete graph. We favor the use of minimum spanning trees as a basis for constructing orderings because MSTs provide useful “pairwise” and “pathwise” estimates of dissimilarity.

MST algorithms are based on locally optimal search criteria. A path on a MST between points that are separated by small intervals tends to approximate the standard pairwise distance between the points. In contrast, the MST shortest path distance between points separated by large intervals is the sum of a series of shorter intervals. Therefore, distances measured on the minimum spanning tree tend to provide estimates that are consistent with the distribution of the observed data. Some minor modifications of the MST may be required in the case of sparse sampling, as described below.

Other properties of minimum spanning trees make them attractive as a framework for ordering samples. If the mean dynamic process of interest is a curve in $\mathbb{R}^d$, and sampling is noiseless and sufficiently dense then the MST guarantees correct reconstruction [Giesen, 1999 #6]. Furthermore, as we describe below, the diameter path of a MST provides a means of evaluating sampling density, noise, and the distribution of noise. Finally, MSTs can be calculated quickly and efficiently even for very large datasets.
Assessing Noise and Sampling Density

Neither polygonal reconstruction nor principal curves techniques provide a means of assessing sampling density and/or sampling noise. For curves embedded in two or three dimensions, or for sampling protocols in which the density of sampling can be controlled, visual inspection may be sufficient for assessing such sample characteristics. For datasets embedded in higher dimensions, examining two or three dimensional projections of the data may or may not be adequate. Therefore, it is desirable to develop heuristics for assessing noise and sampling density.

[Figueiredo, 1995 #8] suggest that in the presence of noise, the diameter path of a MST can be used for reconstruction. The diameter path is the longest path in a tree (Fig. 2c). Figueiredo and Gomes [Figueiredo, 1995 #8] point out that the diameter path of a noisy sample will have numerous edges dangling from it. We will refer to these dangling non-diameter path nodes and edges as the branches of the diameter path.

We suggest that a useful measure of the relative noise of the sample is the ratio of the number of points off the diameter path to the number of points on the diameter path. We call this the diameter path noise ratio. A small diameter path noise ratio (< 0.05) suggests a fairly clean sample, while larger ratios indicate a greater amount of noise. The distribution of non-diameter path points can also be used as a heuristic tool to evaluate the quality of the diameter path as an estimator. If the non-diameter path points are truly noise, then the size of branches coming off the diameter path should be relatively uniform along the extent of the diameter path. However, if the noise is confined to a relatively small number of long branches, then this suggests that these branches may represent signal rather than noise. We refer to this distribution as the diameter path branch distribution.

The diameter path of the MST on a weighted graph can also be used to assess sampling density by calculating the ratio of the average segment length to the total length of the diameter path. For a dense sample, this ratio should be small (< 0.05), while in a sparse sample this ratio will be large. We will use the term diameter path sampling ratio to refer to this quantity.

Ordering Observations using Minimum Spanning Trees

We propose a general algorithm for establishing an ordering of sample elements based on modifications of the minimum spanning tree of a weighted graph. The algorithm can be described as follows:
1. Find the minimum spanning tree, $G_{mst} = \{V, E_{mst}\}$, of the weighted graph $G$, where $G$ is a complete graph of the sampled observations and the edge weights are pairwise dissimilarities. A variety of algorithms are available to construct the MST; Kruskal’s and Prim’s algorithms are among the easiest to implement. If $G_{mst}$ is a path, take this to be the best estimate of the ordering.

2. If $G_{mst}$ is not a path, assess the diameter path noise ratio, branch distribution, and sampling ratio as described above using the diameter path of $G_{mst}$.

3. If the sampling appears to be relatively dense and the noise appears to have a uniform distribution with respect to the diameter path, then the diameter path gives an estimate of the ordering. Sample elements contained in branches off the diameter path are assigned the same ordering index as the diameter path element to which they connect.

4. If the diameter path sampling ratio is large and the noise distribution appears to be highly non-uniform with a few long branches coming off the diameter path, then find a set of “short-cut” paths consistent with the minimum spanning tree, as described below.

**The Short Cut Algorithm**

The “short cut” algorithm is a heuristic algorithm for generating a set of orderings which are consistent with a MST derived from a sparsely sampled dataset. The inspiration for this algorithm is Christofides’ Algorithm for approximating the TSP [Christofides, 1976 #17; Bern, 1997 #19]. Christofides’ algorithm finds a tour of a set of points in a graph which is at most 1.5 times as long as the TSP [Bern, 1997 #19].

Rather than searching for a single near-optimal path, we enumerate a set of paths which are consistent with the MST. We call these paths “shortcut paths” because they shortcut indecisive vertices, as described below.

1. Calculate the minimum spanning tree, $G_{mst}$, as described above.

2. Compute the minimum weight matching $G_{match} = \{V_{\text{odd}}, E_{\text{match}}\}$ on the odd vertices of $G_{mst}$.

   Edmonds [Edmonds, 1965 #20] and Cook and Rohe [Cook, 1999 #18] provide algorithms for performing perfect matchings.

3. Create a multigraph by the union of $G_{mst}$ and it’s odd vertex matching; $G_{\text{multi}} = \{V, E_{mst} \cup E_{\text{match}}\}$.

4. Create a new graph, $G_{\text{sub}}$, by removing all vertices, $V_{\text{indecisive}}$, with degree > 2, and all edges, $E_{\text{indecisive}}$, containing those vertices. $G_{\text{sub}} = \{V_{\text{sub}}, E_{\text{sub}}\}$ where $V_{\text{sub}} = V \cap V_{\text{indecisive}}$ and
\[ E_{\text{sub}} = E_{\text{multi}} \cap E_{\text{indecisive}}^c \]  
\[ G_{\text{sub}} \] is composed of a set of subpaths (some possibly with degree one). Let 
\[ G_{\text{indecisive}} = \{V_{\text{indecisive}}, E_{\text{indecisive}}^c\}. \]

5. Build bridging edges, \( E_{\text{bridge}} \), from \( G_{\text{indecisive}} \). We say that there is a bridge \( \{E_i, E_j\} \) if there is a path in \( G_{\text{mst}} \) between \( E_i \) and \( E_j \) which, except for the beginning and ending vertices, only transverses vertices and edges in \( G_{\text{indecisive}} \).

6. Construct the annealed graph, \( G_{\text{anneal}} = \{V_{\text{sub}}, E_{\text{sub}} \cup E_{\text{bridge}}\}. \)

7. Find all complete circuits of \( G_{\text{anneal}} \), in which no vertex except the starting vertex, is visited more than once. Call this set of circuits, \( C_{\text{mst}}. \)

8. Reduce \( C_{\text{mst}} \) to a set of paths, \( P_{\text{mst}} \), by removing the longest edge in each circuit. Order these paths by their total length. \( P_{\text{mst}} \) is the set of paths consistent with \( G_{\text{mst}} \) which ‘shortcut’ the indecisive vertices.

Each path in \( P_{\text{mst}} \) is an ordering on \( V_{\text{sub}} \). It is fairly straightforward to generate a larger set of orderings on \( V \) by expanding paths in \( P_{\text{mst}} \) using the indecisive vertices. Nothing seems to be gained by doing this however, as this merely results in more paths with relatively small variations in ordering.

**Example Applications**

We demonstrate the efficacy of our approach by applying the methods described above to estimate orderings for two datasets – one artificial and the other derived from DNA microarray studies of gene expression in yeast.

**Artificial Dataset – The Jelly Roll**

An artificial two-dimensional dataset of 150 pts was generated which estimates a common spiraling curve (involute of a circle). This dataset is fairly noisy and has relatively dense sampling (Fig. 2a). The scatter of points resembles a jelly-roll in cross section.

The MST for this dataset is shown in figure 2b, and the diameter path for the MST is shown in figure 2c. The distribution of non-diameter path points is relatively evenly distributed along the extent of the diameter path, and the diameter path noise ratio for this sample is 29/120 = 0.24. About one quarter of
the points in the sample are not included in the diameter path and the diameter path noise ratio confirms our visual assessment that the sample is noisy.

Examining the distribution of diameter path branch sizes indicates that there is no strong pattern to the distribution of noise with respect to the diameter path. Most diameter path nodes have no branches or a branch of degree one.

Finally we estimate sampling density from the diameter path. The diameter path has a total length of 95.08 units, and the average edge length in the diameter path is 1.07 units. Therefore the diameter path sampling density ratio is 0.011, confirming that sampling is relatively dense.

Given the above considerations we use the diameter path of the MST (Fig. 2c) to estimate the ordering for this dataset. Each point off of the diameter path is assigned to the same ordering index as its corresponding root node on the diameter path.

**Biological Datasets – Yeast and Caulobacter Expression Data**

We now describe the application of our ordering algorithm to two biological datasets derived from DNA microarray studies.

**Caulobacter Microarray Study**

Laub et al. [Laub, 2000 #1] describe a microarray based analysis of gene transcription during a single cell cycle of the bacterium *Caulobacter crescentus*. A population of wild-type *Caulobacter* swarmer cells were isolated and allowed to proceed synchronously through a 150-minute cell cycle, with samples drawn at 15 minute intervals. The relative abundance of mRNAs from cells in each time sample was assessed via competitive microarray hybridizations of reverse transcribed cDNA from each sample point compared to cDNA from a mixed, unsynchronized reference population of *Caulobacter*.

We log transformed and mean-centered the published data on expression levels. We reduced the dataset to the 1594 ORFs for which expression measurements were available at all time points. We applied the ordering algorithm described above to this dataset of 1594 ORFs, as well as to subsets containing the 500 most variable and 300 most variable genes. The estimated ordering was the same regardless of the sample used and we describe the results for an analysis based on all 1594 ORFs.

With only 11 samples, the dataset is expected to be rather sparse. However, the diameter path of the MST for the sample elements has only a single branch of degree one (Fig. 3). Though sparse, the diameter path
appears to provide a relatively low-noise reconstruction, and we take the points on the diameter path as the best estimate of the ordering (red path, Fig. 3).

The known ordering of data points is given by the permutation \([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]\). The reconstructed ordering gives the permutation \([1, \{2,3\}, 4, 5, 6, 7, 8, 9, 10, 11]\). The reconstructed ordering therefore provides a very good estimate of the known temporal ordering.

Yeast Microarray Study
Spellman et al. [Spellman, 1998 #21] described a set of experiments done with the yeast *Saccharomyces cerevisiae* to quantify how gene expression patterns change during the cell cycle. One of these experiments involves synchronizing a population of yeast cells by treatment with the mating pheromone \(\alpha\)-factor. The original dataset based on 6177 ORFs was reduced to a set of 5541 genes as described in Rifkin et al. [Rifkin, 2000 #35], and the data were log-transformed and normalized by the centroid. The original dataset consists of 18 sequential measurements of gene expression, sampled at seven minute intervals for 140 minutes. As with the *Caulobacter* dataset the true ordering for the set of samples is known, but we wished to test the ability of our algorithm to estimate this ordering. This dataset is expected to be somewhat more difficult for the reconstruction algorithm to deal with than was the *Caulobacter* dataset because the data cover two cell-cycles (the data are known to evidence a pseudo-cycle which may confuse the ordering algorithm) and the sampling is sparser.

Of the approximately 5500 genes in the dataset, a great number do not exhibit appreciable variation in expression levels. For the purposes of ordering it seems reasonable to treat those variables with low variation as if they represented noise. The 500 genes exhibiting the most variation sample variation were used to construct a distance matrix among the 18 sample points. An ordination of these data in the space represented by the 3 largest principal coordinates of the distance matrix is depicted in figure 4a.

With only 18 samples, we expect that the sampling is rather sparse. The minimum spanning tree for this dataset, depicted in the space of a 3-D ordination of these points is shown in figure 4b. The diameter path is highlighted in red. The diameter path sampling ratio confirms that sampling is sparse, and the distribution of branch lengths suggests that the branches should be treated as “signal” rather than “noise.”

We constructed the shortcut paths which are consistent with the MST using the algorithm described above. Three sample points are inconsistent in the MST (blue data points in Fig. 4b). Short-cutting these points results in 12 unique shortcut paths (Table 1), the shortest of which is shown in figure 4c. Figure 4d
shows the known ordering for these samples. The highest ranked short cut path shares the major features of the true ordering. The true ordering appears to show a pseudo-cycle which makes the placement of several of the points difficult to recover in the ordering reconstruction, however the major geometric features of the true ordering are well approximated. Denser sampling would lead to improved reconstructions.

**Conclusions**

The approach we describe for ordering samples is well suited to handle datasets which utilize a variety of dissimilarity measures and sampling conditions. The demonstration that our heuristic algorithm works well for both artificial and experimental datasets suggests there is significant practical value in utilizing this approach to study dynamic biological processes when it is not possible to synchronize samples, homogenize rates, and/or time index samples.

Minimum spanning trees provide a natural geometric characterization for samples. This characterization is free of *a priori* distributional assumptions, and can be applied to datasets using a variety of dissimilarity measures. Modifications of the MST or the MST itself provide a useful basis for estimating orderings and reconstructing curves, and additional can be used as a tool to evaluate distributional properties such as noise and sampling density.

The ability of our method described above to incorporate high-dimensional suggests that it may be a particularly suitable approach for deriving time series for data derived from high-throughput experimental techniques. Since our approach can also be applied to non-Euclidean distance measures it may also be particularly appropriate to incorporating data from multiple sources such as simultaneous information on phenotype, the transcriptome, and the proteome. Data such as these will facilitate the construction of well characterized and informative time-series. Constructing such time-series is a necessary first step towards gaining a better understanding of the dynamical behavior of biological systems.
Figure 1. An illustration of the difference between “pairwise” and “pathwise” estimates of distance. a) When curvature is relatively low, the arclength between points a and b is well approximated by the pairwise distance. b) When curvature is low, the pairwise distance $d_{a,b}$ is a poor estimator of the arclength. The pathwise distance $d_{a,x_1,x_2,x_3,x_4,b}$ is a better approximations of the true distance with respect to the function $f$. 
Figure 2. Generated "jellyroll" dataset. a) Original data; b) minimum spanning tree; c) diameter path of the minimum spanning tree. We use the diameter path as the basis for estimating an ordering for this dataset. See text for further explanation.
Figure 3. Caulobacter microarray dataset. Though this dataset is relatively sparse, the diameter path has low noise, and hence provides a useful basis for estimating an ordering. See text for further explanation.
Figure 4. Alpha-factor data from Saccharomyces microarray dataset. a) Ordination of sample points in the space of the 3 largest principal coordinates; b) minimum spanning tree (and diameter path in red) for this data; c) Reconstructed "short-cut" ordering; d) Known ordering. See text for further explanation.