Slingshot:

Kelly Street

Abstract

Recently, single-cell RNA-Seq has afforded researchers an unprecedentedly detailed view of cellular transcription. Communities of heterogeneous cells that could previously only be interrogated collectively can now reveal multiple functionally distinct groups with complex relationships. One common target for these studies has been stem cells and their descendants. Mapping transcriptional progression from stem cell populations to specialized cell types has become crucial for properly understanding these systems and many statistical and computational methods have been proposed. Slingshot is a uniquely robust and flexible tool for inferring developmental lineages and ordering cells to reflect continuous differentiation processes. It constructs a differentiation tree using clusters of cells as nodes, which provides stability and reduces the complexity of the inferred lineages. This map is used to assign individual cells to one or more developmental lineages, which are represented by smooth curves in a reduced dimension space. These curves provide discerning power not found in methods based on piecewise linear trajectories while also adding stability over a range of possible dimensionality reduction and clustering techniques. [real and simulated data, compare to X, Y, and Z methods]

1 Introduction

Traditional transcription assays, such as bulk microarrays and RNA-Seq afford us a bird's-eye view of transcription. Because they rely on RNA from a relatively large number of cells as starting material, these methods are not ideal for examining heterogeneous populations of cells. Single-cell RNA-Seq can give us a much more detailed picture. Higher resolution means the ability to distinguish closely related populations of cells and characterize their relationships.

For some cell types, there may not be a clear distinction between states, but rather a smooth transition, with individual cells existing on a continuum between known states. In these cases, cells may undergo gradual transcriptional changes where the relationship between states can be modeled as a continuous lineage dependent on an underlying spatial or temporal variable. This modelling has been referred to as "pseudotemporal reconstruction" and it can help us understand how cells change and how cell fate decisions are made [Monocle,Wanderlust,Waterfall,TSCAN,Embeddr].

One of the primary difficulties of single-cell RNA-Seq is the high level of noise. Transcriptional bursting and drop-out effects, in addition to the host of biological and technical confounders that affect bulk RNA-Seq, combine to make single-cell data particularly opaque. Normalization is a key concern for any single-cell analysis and best practices may vary from lab to lab or even day to day. Downstream analyses such as clustering, marker gene identification, and lineage reconstruction are similarly sensitive and unlikely to have a perfect one-size-fits-all solution.

Several methods have been proposed for the task of pseudotemporal reconstruction, with a wide range of strengths and inherent assumptions [review in Bacher and Kendziorski (2016)]. One of the most well-known is Monocle, which constructs a minimum spanning tree (MST) on cells in a

reduced-dimensionality space created by independent component analysis (ICA) and orders cells along the longest path through this tree. As others have noted [Waterfall], this MST solution can be highly unstable or even non-unique. The directionality of this path and the number of branching events are left to the user, who may examine a known set of marker genes or potentially use time of sample collection as indications of intial and terminal cell states.

One way to increase the stability of the global lineage structure is to cluster cells before ordering them [Waterfall, TSCAN]. Drawing an MST on the set of clusters, rather than individual cells, can greatly reduce the complexity of the inferred lineages and safeguard against spurious effects caused by outliers. This also serves as a simple, unsupervised method for identifying branching events at any cluster with degree ≥ 3.

Cluster-based MST methods typically require a dimensionality reduction step and both [Waterfall] and [TSCAN] encourage the use of principal component analysis (PCA). In other contexts, different dimensionality reduction techniques have been deemed appropriate for visualization and summarization, including ICA [Monocle], t-SNE [Wanderlust,Petropoulos], and Laplacian eigenmaps [Embeddr]. Given this wide range of applicable techniques, each of which have given satisfatory results on different datasets, it is hard to conclude that any one method performs best.

Finally, we are often interested in representing a lineage as an object in this low-dimensional space. Methods based on an MST generally represent lineages as piecewise linear paths through the tree and extract orderings either by orthogonal projection [Waterfall,TSCAN], or a PQ tree [Monocle]. Other methods prefer to construct smooth curves to represent lineages [Embeddr,Petropoulos] and extract orderings based on orthogonal projection. The latter approach includes fewer inferred elements and is more directly based on the data. The principal curve method of [HastieStuetzle] is a common way to construct these curves, suitable for a single non-branching lineage. The principal tree method of [Mao] handles branched lineages by a similarly unsupervised process, but its stability is not well established and incorporating a priori knowledge can be difficult or impossible.

Here we introduce Slingshot, a novel trajectory analysis tool that combines the stability-improving techniques necessary for single-cell data with the flexibility to easily integrate with a range of normalization and dimensionality reduction methods. Slingshot's cluster-based approach allows for easy supervision when researchers have prior knowledge of their data while still being able to detect novel branching events and its multiple (simultaneous?) curve-fitting method can turn this information into smooth, stable trees.

2 Methods

[*** Add table summarizing options for each step and referring to other methods. Check terminology "Mapping", etc.]

2.1 Statistical Inference Framework

We start from a $P \times n$ matrix of expression measures for n single cells. The goals of our analysis are **1.** to identify the number of distinct terminal cell states and **2.** for each of these states, construct a one-dimensional variable representing transcriptional progression toward that state. The number of terminal cell states corresponds to the number of unique lineages to be constructed and we term this number L. For a given lineage l, we refer to its one-dimensional representation as $\hat{t}_l$, which will contain values for every cell plausibly derived from that lineage.

2.2 Normalization

After read mapping and expression quantification, a typical RNA-Seq analysis will perform some sort of normalization. This can include global scaling adjustments, batch correction, gene and sample filtering, or a number of other techniques. Raw gene expression profiles need to be normalized prior to lineage reconstruction to remove unwanted technical effects that bias expression measures. We used SCONE to compare various normalization methods and select the one that performs best on our data.

2.3 Clustering

Using unsupervised clustering to find biologically distinct classes of cells is another feature of many single-cell RNA-Seq data analysis pipelines. Clustering cells provides stability in downstream analysis by dramatically reducing the number of possible relationships between different cells and lessening the potential impact of outliers on the final ordering. We denote these clusters as non-overlapping subsets of cells, $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$.

Our data were clustered with the `clusterExperiments` package, which implements an ensemble clustering method using an iterative procedure to find tight, stable clusters. To increase the stability of the clustering process and lineage reconstruction, a small percentage of cells may be left unclustered. These cells can be ordered by the final set of lineages, but will not be used in their construction.

[*** ADD: Other clustering methods such as model-based clustering [TSCAN] could be used.]

2.4 Dimensionality Reduction

Before constructing lineageas, we recommend finding a suitable low-dimensional representation of the data. There are a wide variety of methods available for this purpose and some may be more appropriate than others for particular datasets. We note only that since Slingshot's default lineage reconstruction method is nonlinear, there is no reason to believe that linear dimensionality reduction methods cannot perform well. By way of example, we use PCA for our data [or for a particular dataset]. We will format this reduced-dimensionality dataset in an $n \times p$ matrix denoted by $\mathbf{X}$.

[*** From my notes: ICA/PCA/MDS – Check relationship (Trapnell et al. 14, Monocle): Linear, non-robust to outliers
- Laplacian eigenmaps, aka spectral embedding (Campbell et al. 15, Embeddr): Non-linear, more robust, but weird looking (cf. their choice of dist)
- tSNE]

2.5 Mapping Cell Types

Slingshot identifies relationships between clusters by treating them as nodes in a graph and drawing a minimum spanning tree (MST). Distances between clusters depend on both the Euclidean distance between their centers and their respective shapes, as measured by within-cluster covariance matrices. By default, Slingshot will use the full covariance matrix of both clusters in the reduced dimensionality space to calculate the distance:

$$d(\mathcal{C}_i, \mathcal{C}_j) = (\bar{x}_i - \bar{x}_j)^T (S_i + S_j)^{-1} (\bar{x}_i - \bar{x}_j)$$

where $\bar{x}_i$ represents the center (mean) of cluster i and S_i its empirical covariance matrix. In the presence of small clusters, the matrix $S_i + S_j$ may not be invertible and we instead use $\tilde{S}_i + \tilde{S}_j$, where $\tilde{S}_i$ is 0 for all off-diagonal elements involving dimensions higher than the minimum cluster size. [Subject to change]

This allows Slingshot to draw trees that are better covered by and representative of the cells in a dataset. For datasets in which there are outlying clusters or more than one tree is appropriate, Slingshot includes a granularity parameter that effectively limits the maximum edge length in the tree. [By default, we look for a large jump (more than 3 times the average jump) in the ordered edge lengths of the MST and if there is one, we set the limit at the halfway point of that jump].

2.6 Lineage Identification

Lineages are defined as ordered sets of clusters created by tracing paths through the MST. Given a pre-specified origin cluster, every direct path from this cluster to a leaf node will be returned as a lineage. In the simple case where the MST has only two leaf nodes, this results in one lineage if the origin cluster is a leaf and two lineages if it is an interior node. Clusters with more than two connections will create bifurcations and produce additional lineages. In the absence of a pre-specified origin, Slingshot will select an origin based on parsimony, producing a set of lineages with the maximal number of clusters shared between them.

2.7 Modelling and Pseudotime Calculation

The next step is to model each of these lineages with a smooth curve. This is achieved by an iterative procedure based on the principal curve algorithm of [HastieStuetzle]. Our goal is to construct a set of variables, $\{t_1, ..., t_L\}$, the pseudotime values for each lineage, and a set of functions, $\{c_1(t_1), ..., c_L(t_L)\}$, the curves associated with each lineage.

In the case where there is only a single lineage (ie. $L = 1$), Slingshot will draw a smooth curve through the center of the data. The primary distinction between our method and that of [HastieStuetzle] is to start with an initial curve based on the piecewise linear path through the cluster centers rather than the first principal component of all points. Given an initial curve, the algorithm proceeds iteratively:

1. Project all data points onto the curve and calculate the arclength from the beginning of the curve to each point's projection. Setting the lowest value to zero, this produces pseudotime values.

2. Update the curve by modelling the cells' coordinates in each dimension as a smooth function of pseudotime. This is accomplished using a smoothing spline.

3. Repeat until convergence.

Starting with the path through the cluster centers allows us to leverage the prior knowledge that went into lineage identification as well as helping to improve the speed and stability of the algorithm.

In the case of branching lineages with the same origin, a shrinkage step will be included at each iteration that forces similarity between the curves in the neighborhood of shared clusters. We accomplish this by constructing an average curve and lineage-specific weight curves determining how much we should shrink toward the average. The average curve, as with the the individual

lineage curves, is a function of pseudotime and consists of the mean of the individual lineage curves at each unique pseudotime value.

$$c_{\text{avg}}(\lambda) = \frac{1}{L}\sum_{i=1}^{L} c_i(\lambda)$$

.

The lineage-specific weight curves are similarly functions of pseudotime, based on two kernel density estimates: $D_i(\lambda)$, which uses all cells along the lineage and $d_i(\lambda)$ which uses only the cells common to the branching lineages. Additionally, d_i includes a scaling factor given by the proportion of common cells, ensuring that $d_i(\lambda) \leq D_i(\lambda)$ for all values of λ. We determine the weight curves by the ratio:

$$w_i(\lambda) = \frac{d_i(\lambda)}{D_i(\lambda)}$$

with the conventional rule that $0/0 = 0$. These weight functions allow us to shrink the individual lineage curves toward their shared average curve with the update step:

$$c_i^{\text{new}}(\lambda) = w_i(\lambda)c_{\text{avg}}(\lambda) + (1 - w_i(\lambda))c_i(\lambda)$$

This shrinkage step typically ensures that the final curves will form a tree structure. In both cases (branching or single-lineage), final pseudotime values are derived from the each points orthogonal projection onto the curves. Thus, cells belonging to clusters that are included in multiple lineages will have multiple pseudotime values, but these values will agree quite well for cells positioned before the lineage bifurcation.

2.8 DE Genes

After lineage reconstruction, the next step in many single-cell analyses is the identification of genes which vary their expression temporally. For each gene, we are interested in testing the relationship between expression and pseudotime. Since lineages may be arbitrarily complex and contain any number of biologically meaningful intermediate states, we recommend using nonlinear methods for finding these genes. One common choice is a General Additive Model (GAM) [TSCAN,others?,GAM paper?] which describes the relationship using a smoothing spline or loess curve. A Fused Lasso Additive Model (FLAM) [Peterson] provides more flexibility for dramatic changes in expression at the cost of treating expression as piecewise constant over pseudotime.

- Nir's approach: Impulse modeling.

2.9 Visualization

- lineage-specific heatmap, split heatmap (like Monocle 2) - gene-specific pseudotime-expression plots

2.10 Software Implementation

The method we have described is implemented in the `slingshot` package, which is available online at [github/bioconductor]. We focus on modularity and intend for the package to be used as part of a larger pipeline. As with the method itself, the `slingshot` package is designed to

be flexible and adaptable to a wide range of different workflows. The core functions [mention get_lineages/get_curves] can take inputs in multiple formats and return similar objects, so that new users are not required to learn new data types. [should probably say more]

3 Datasets

4 Results

Performance measures, cf. TSCAN.

5 Discussion