Slingshot: Inference of cell lineages and pseudotimes using single-cell RNA-Seq

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, with analyses focused on charting gradual transcriptional progression from multipotent cells to fully differentiated populations. These studies provide insight into cell fate decisions and many statistical and computational methods have been proposed for their analysis. 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 systems, there may be no clear distinctions between states, but rather a smooth transition with individual cells representing points along a continuum. 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 (Bendall et al., 2014; Campbell et al., 2015; Ji and Ji, 2016; Shin et al., 2015; Trapnell et al., 2014)[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 (Trapnell et al., 2014), 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 Shin et al. (2015), 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 (Ji and Ji, 2016; Shin et al., 2015). 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 Ji and Ji (2016) and Shin et al. (2015) 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 (Trapnell et al., 2014), t-distributed stochastic neighbor embedding (t-SNE) (Bendall et al., 2014; Petropoulos et al., 2016), and Laplacian eigenmaps (Campbell et al., 2015). 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 (Ji and Ji, 2016; Shin et al., 2015) or a PQ tree (Trapnell et al., 2014). Other methods will construct smooth curves to represent lineages (Campbell et al., 2015; Petropoulos et al., 2016) and extract orderings based on orthogonal projection. This latter approach provides stability and robustness to outliers as every point along the lineage is an average. The principal curve method of Hastie and Stuetzle (1989) is a common way to construct these curves, suitable for a single non-branching lineage. The principal tree method of [Mao] handles arbitrarily complex branching patterns by a similar process, but its stability is not well established and incorporating a priori knowledge can be difficult or impossible.

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

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 an $n \times J$ matrix of expression measures (typically read counts) for n single cells and J features. The goals of our analysis are to **1.** identify cell *lineages*, i.e., ordered subsets of cell clusters, where clusters correspond to cellular states and each lineage leads to a unique terminal state, **2.** for each of these lineages, infer cell *pseudotimes*, i.e., one-dimensional variables representing transcriptional progression toward the terminal state.

Let $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ denote K cell clusters, i.e., disjoint subsets of cells, obtained as in Section 2.3. We then define a lineage as an ordered set of clusters and let L denote the total number of lineages. For a particular lineage, $\mathcal{L}_l$, we denote its length as K_l and the k^{th} cluster as $\mathcal{C}^l_k$, for $l \in 1, \ldots, L$ and $k \in 1, \ldots, K_l$; in particular, $\mathcal{C}^l_1$ and $\mathcal{C}^l_{K_l}$ correspond, respectively, to the initial and terminal states for the l^{th} lineage. It is important to note that a cluster can belong to multiple lineages and that the ordering of the clusters within a lineage does not necessarily determine the final relative orderings of cells in those clusters.

As a given cluster can belong to multiple lineages, so can a cell. We therefore allow a given cell to have a distinct pseudotime for each of the lineages to which it belongs. The pseudotime value for cell i in lineage l is denoted by $t^l_i \in \mathbb{R}_{\geq 0}$; if cell i does not belong to lineage l, i.e., $i \notin \cup_{k=1}^{K_l} \mathcal{C}^l_k$, then set $t^l_i = \emptyset$. The vector of pseudotime values for lineage l is denoted by $t^l = (t^l_i : i = 1, \ldots, n)$.

2.2 Normalization

After read mapping and the computation of gene-level counts, a typical RNA-Seq analysis involves a number of pre-processing steps, such as gene and sample filtering, log-transformation, and normalization. Raw gene expression profiles need to be normalized prior to lineage reconstruction to remove unwanted technical effects (e.g., batch effects) that bias expression measures [paper with Marioni Group]. We used SCONE to compare various normalization methods and select the one that performs best on our data [scone].

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.

Our data were clustered with the `clusterExperiments` package, which implements a resampling-based sequential ensemble clustering (RSEC) method to find tight, stable clusters [clusterExperiments]. 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 lineages, 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 non-linear (Section 2.7), 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 format the reduced-dimensionality dataset in an $n \times p$ matrix $\mathbf{X} = (X_{ij})$, where X_{ij} denotes the expression measure of cell i in dimension j and $p \ll J$.

[*** ADD: Pros and cons of other dimensionality reduction methods. 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)
- t-SNE]

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). Constructing an MST involves specifying a distance matrix between clusters. Although in principle any type of distance matrix could be used (e.g., Euclidean, Manhattan), we found that a covariance-scaled Euclidean distance, that accounts for cluster shape, works well in practice. Specifically, the pairwise distance between clusters i and j is defined as

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

where $\bar{x}_i$ represents the center (mean) of cluster i and S_i its empirical covariance matrix in the reduced-dimensionality space. By default, Slingshot uses the full covariance matrix of both clusters, allowing us to draw trees that are better covered by and representative of the cells in a dataset. However, in the presence of small clusters, the matrix $S_i + S_j$ may not be invertible and we replace the full covariance matrix with the diagonal covariance matrix.

[*** Mention/try other standard bw-cluster distances, such as, max/min/avg. Cf. linkage method in dendrogram.]

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]. [*** Add details]

2.6 Lineage Identification

Lineages are defined as ordered sets of clusters created by tracing paths through the MST. Leaf nodes of the MST (ie. clusters with only one edge) are assumed to be either original or terminal states. If a root cluster (origin) is specified, every direct path from this root to a leaf node will be identified as a lineage. In the simple case where the MST has only two leaf nodes and one is specified as the origin, this results in a single lineage. If an interior node is specified as the origin, this results in two lineages, one terminating in each leaf node. Clusters with more than two edges will create bifurcations and produce additional lineages. In the absence of a pre-specified root node, Slingshot will select a root based on parsimony, producing a set of lineages with the maximal number of clusters shared between them.

[*** Add details on supervision for root and leaf nodes.]

2.7 Modelling and Pseudotime Calculation

The next step is to model each of these lineages with a smooth curve. Specifically, our goal is to infer, for each lineage l, a vector of pseudotimes, $t^l = (t^l_i : i = 1, \ldots, n)$, and a function $c_l : \mathbb{R}_{\geq 0} \to \mathbb{R}^p$ for the associated curve in the low-dimensional space. Since some cells may not belong to every lineage, we adopt the convention that $c_l(\emptyset) = \emptyset$. This curve-fitting step is achieved by an iterative procedure based on the principal curve algorithm of Hastie and Stuetzle (1989).

In the case where there is only a single lineage (i.e., $L = 1$), Slingshot draws a smooth curve through the center of the data. The primary distinction between our method and that of Hastie and Stuetzle (1989) 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. For each of the p dimensions, use the pseudotime values to model cells' coordinates. This is accomplished with a smoothing spline, producing a set of functions which collectively map $\mathbb{R}_{\geq 0}$ into $\mathbb{R}^p$, producing a smooth curve.

3. Repeat this process until convergence. The total sum of squared distances from cells to their projections onto the curves is used to determine convergence.

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

[Separate section?]

In the case of branching lineages with the same origin, a shrinkage step is 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 every pseudotime value

$$c_{\text{avg}}(t) = \frac{1}{L} \sum_{l=1}^{L} c_l(t),$$

where t ... *** Details on how individual lineage curves are obtained and on "unique pseudotime value".

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

$$w_l(t) = \frac{d_l(t)}{D_l(t)},$$

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:

$$c_l^{\text{new}}(t) = w_l(t)c_{\text{avg}}(t) + (1 - w_l(t))c_l(t).$$

This shrinkage step typically ensures that the final curves will form a tree structure with smooth branching events. 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 non-linear 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

We demonstrate the flexibility of Slingshot by applying it to two previously published single-cell RNA-Seq datasets: that of Trapnell et al. (2014) and Shin et al. (2015). [Talk about Monocle dataset here]

Shin et al. (2015) assayed 132 adult hippocampal quiescent neural stem cells (qNSCs) and their immediate progeny, cells known to be involved in adult neurogenesis. Their goal was to assess cellular heterogeneity among this population and analyze continuous-time developmental dynamics. After removing a few outliers, their analysis focuses on 101 cells believed to represent the development of qNSCs into intermediate progenitor cells (IPCs), a transitional state between qNSCs and mature neurons. They also note an additional cluster of 23 cells branching off of this lineage, potentially representing an alternative terminal cell type. Here we use the original cluster

labels based on a hierarchical clustering procedure and represent cells by their coordinates along the first two principal components.

4 Results

[Monocle results]

For the dataset of Shin et al. (2015), we compare the results obtained by running Waterfall separately on each lineage to those obtained by running Slingshot on the full dataset. Because the Waterfall analysis pipeline features a secondary clustering step using k-means, we follow the authors' choice of K for the main lineage and attempt to follow their logic to make a reasonable choice regarding the second lineage. In both cases, the two methods provide very similar results.

5 Discussion

References

Rhonda Bacher and Christina Kendziorski. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biology*, 17(63), 2016.

S. C. Bendall, K. L. Davis, E. D. Amir, M. D. Tadmor, E. F. Simonds, T. J. Chen, D. K. Shenfeld, G. P. Nolan, and D. Pe'er. Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell*, 157(3):714–725, 2014.

K. Campbell, C. P. Ponting, and C. Webber. Laplacian eigenmaps and principal curves for high resolution pseudotemporal ordering of single-cell RNA-seq profiles. Technical report, MRC Functional Genomics Unit, University of Oxford, UK, 2015. URL `biorxiv.org/content/early/2015/09/18/027219`.

T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84 (406):502–516, 1989.

Z. Ji and H. Ji. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Research*, 2016.

S. Petropoulos, D. Edsgärd, B. Reinius, Q. Deng, S. P. Panula, S. Codeluppi, A. Plaza Reyes, S. Linnarsson, R. Sandberg, and F. Lanner. Single-cell RNA-Seq reveals lineage and X chromosome dynamics in human preimplantation embryos. *Cell*, 165(In press):1–15, 2016.

J. Shin, D. A. Berg, Y. Zhu, J. Y. Shin, J. Song, M. A. Bonaguidi, G. Enikolopov, D. W. Nauen, K. M. Christian, G. Ming, and H. Song. Single-cell RNA-Seq with Waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell*, 17(3):360–372, 2015.

C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 4(32):381–391, 2014.