GPseudoRank: a permutation sampler for single cell orderings

Magdalena E Strauß[1,*], John E Reid[1,2], and Lorenz Wernisch[1]

[1] *MRC Biostatistics Unit, University of Cambridge, Cambridge CB2 0SR, UK*
[2] *Alan Turing Institute, London NW1 2DB, UK*
[*] *Corresponding author: magdalena.strauss@mrc-bsu.cam.ac.uk*

7th February 2018

Abstract

Motivation: A number of pseudotime methods have provided point estimates of the ordering of cells for scRNA-seq data. A still limited number of methods also model the uncertainty of the pseudotime estimate. However, there is still a need for a method to sample from complicated and multi-modal distributions of orders, and to estimate changes in the amount of the uncertainty of the order during the course of a biological development, as this can support the selection of suitable cells for the clustering of genes or for network inference.
Results: In an application to a microarray data set our proposed method, GPseudoRank, identifies two modes of the distribution, each of them corresponding to point estimates of orders obtained by a different established method. In an application to scRNA-seq data we demonstrate the potential of GPseudoRank to identify phases of lower and higher pseudotime uncertainty during a biological process. GPseudoRank also correctly identifies cells precocious in their antiviral response.
Availability and implementation: Our method is available on github: https://github.com/magStra/GPseudoRank.
Contact: magdalena.strauss@mrc-bsu.cam.ac.uk
Supplementary information: Supplementary materials are available.

1 Introduction

Providing mRNA expression levels of genes for individual cells, scRNA-seq has shown heterogeneity of gene expression across cells during various biological developments. While part of this results from technical noise, part is generally attributable to genuine cell heterogeneity. See, for instance, [5, 31]. Due to the destruction of the cells as a result of the measurement process, scRNA-seq only provides a single measurement per cell [28], never time series data following the development of the same single cell. However, individual cells progress through changes at different time scales [30]. Thus it is possible to obtain a form of time series data even from cross-sectional data by statistical means, an approach referred to as pseudotime ordering.

Most approaches to pseudotemporal ordering are based on representing cells as n_g-dimensional vectors, where n_g is a selected number of genes in a cell. Algorithms exploit the neighborhood structure of these vectors to find a pseudotemporal ordering, a linear ordering of all or most cells so that cells which are close in $\mathbb{R}^{n_g}$ are also close in the linear ordering.

Wanderlust [4] and SLICER [32, 33] are two examples of methods based on k nearest neighbours graphs. SLICER additionally first applies LLE (local linear embedding) [25] for dimensionality reduction. A number of methods are based on diffusion maps [3, 11, 12, 26]. TSCAN [16, 15] is based on the construction of a minimum spanning tree (MST) between centroids of clusters, with an intermediate clustering step. Another well-known method using MST and clustering is Monocle 2 [22], which applies graph structure learning [17].

The approaches mentioned above and a number of others provide singular pseudotime orderings without modelling uncertainty. Campbell and Yau [7] examined the stability of Monocle's pseudotime estimation when applied to random subsets of cells. They showed that the estimates can vary significantly. Thus quantification of uncertainty in pseudotime is crucial to avoid overconfidence. There are two existing methods for pseudotime estimation using MCMC to sample from a posterior distribution [7, 24], and a few others using variational methods [1, 24, 34]. They use Gaussian processes (GPs, see Section 2.1) to model the data. However, these methods sample from, or approximate, in the case of variational inference, posterior distributions of continuous pseudotime vectors in $\mathbb{R}^n$, rather than sampling the ordering as a permutation.

We propose GPseudoRank, an algorithm sampling from a posterior distribution of pseudo-orders instead of pseudotimes, avoiding the exploration of pseudotime assignments that all map to the same ordering. MCMC samplers (such as NUTS [14]) suitable for use in continuous pseudotime spaces make local moves that can have problems exploring bi-modal posteriors. GPseudoRank, by contrast, exploits a range of local and long-distance MCMC moves tailored to efficiently traverse the space of permutations. It also provides continuous pseudotime estimates by deriving a pseudotime vector from a fixed ordering through a deterministic transformation. This is based on the observation that most continuous pseudotime vectors with high likelihood are concentrated around pseudotime vectors derived from orderings through this transformation.

2 Methods

2.1 Single-cell trajectories as stochastic processes

We assume we have preprocessed logarithmised gene expression data in the form $y_g(c)$ of gene g, $g = 1, \ldots, n_g$, in cell c, $c = 1, \ldots, T$ (see section 2.6 for preprocessing steps). We start with a vector of time points $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_T)$ and define an ordering of cells as a permutation $\mathbf{o} = (o_1, \ldots, o_T)$, $o_i \in \{1, \ldots, T\}$, $o_i \neq o_j$ for $i \neq j$, where o_i is the index of the cell assigned to time τ_i. We model the gene expression trajectories $y_g = (y_g(o_1), \ldots, y_g(o_T))$ for each gene g by Gaussian processes (GPs) [23], conditional on an ordering $\mathbf{o}$ of the cells. A GP is a distribution over functions of time in terms of a mean function μ and a covariance function Σ. For an input vector $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_T)$ of time points, $\mu(\boldsymbol{\tau})$ returns a vector of T mean values of function evaluations at these time points and $\Sigma(\boldsymbol{\tau})$ a $T \times T$ matrix of covariances of function evaluations at the time points. The distribution of functions $f \sim GP(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is described by stating that, for any vector of time points $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_T)$, evaluations $f(\tau_i)$ follow a multivariate normal $(f(\tau_1), \ldots, f(\tau_T)) \sim \mathcal{N}_T(\boldsymbol{\mu}(\boldsymbol{\tau}), \boldsymbol{\Sigma}(\boldsymbol{\tau}))$. In this study we use a squared exponential covariance function for Σ.

$$[\boldsymbol{\Sigma}(\boldsymbol{\tau}, \sigma_w^2, l, \sigma_\epsilon^2)]_{i,j} = \sigma_w^2 \exp(-\frac{(\tau_j - \tau_i)^2}{2l^2}) + \delta_{ij}\, \sigma_\epsilon^2 \tag{1}$$

where σ_w^2 is a scale parameter, l a length scale and σ_ϵ^2 a term representing measurement noise.

Given an ordering $\mathbf{o}$, the expression data for gene g can be ordered accordingly: $y_g(\mathbf{o}) = (y_g(o_1), \ldots, y_g(o_T))$ and we model this trajectory as

$$y_g(\mathbf{o}) \sim \mathcal{N}_T(\boldsymbol{\mu}(\boldsymbol{\tau}), \boldsymbol{\Sigma}(\boldsymbol{\tau}, \sigma_w^2, l, \sigma_\epsilon^2)) \tag{2}$$

for each gene $g = 1, \ldots, n_g$, where $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_T)$ are time points. In practice, we assume a zero-mean GP, that is, $\boldsymbol{\mu} = 0$. To adjust the data for this assumption we subtract the overall mean across all genes and cells from each entry in the matrix of gene expression levels (see Sections 2.6.2 and 2.6.3).

2.2 Geodesic mapping

Pseudotime should not be confused with physical time in which cell development unfolds. In order to identify the latent time points $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_T)$, which we assume to be unknown, together with the smoothness parameters of the GP, we have to make additional assumptions. The overall scale can be fixed by assuming $\tau_i \in [0, 1]$ and each cell could be assigned some *rank time*, equidistant time points $((i - 0.5)/T \mid i = 1, \ldots, T)$. Rank time is similar to the concept of master time developed in [34]. However, rank time depends on the number of cells sampled per capture time, which could be rather arbitrary, and does not allow for any local change in scale. We therefore suggest a different route to identify latent time points. We assume the covariance structure, essentially the smoothness of the process, is independent of time, that the GP is *stationary*. Pseudotime can then be considered a latent variable measuring biological development rather than physical time [1, 7, 24, 34]. For periods of slower development, for example, pseudotime intervals will be shorter than physical time intervals and longer for faster development. In order to account for such change in scale over time we compute time points for any given ordering $\mathbf{o}$ as follows (recall o_j is the index of the cell in position j).

$$\tilde{\tau}_1(\mathbf{o}) = 0, \quad \tilde{\tau}_{j+1}(\mathbf{o}) = \tilde{\tau}_j(\mathbf{o}) + \|\mathbf{y}(o_j), \mathbf{y}(o_{j+1})\|_2, \quad j = 1, \ldots, T-1 \tag{3}$$

where $\mathbf{y}(o_j) = (y_1(o_j), \ldots, y_{n_g}(o_j))^T$ and $\|.\|_2$ is the Euclidean norm in $\mathbb{R}^{n_g}$. Then we set $\boldsymbol{\tau}(\mathbf{o}) = \tilde{\boldsymbol{\tau}}(\mathbf{o})/\max(\tilde{\boldsymbol{\tau}}(\mathbf{o}))$ to obtain pseudotimes $\boldsymbol{\tau}(\mathbf{o})$ in the interval $[0, 1]$. For cells next to each other in the order $\mathbf{o}$, this mapping puts them closer in pseudotime if they are similar in their expression profiles and further apart if they are less so. That is, the j-th time point τ_j is the geodesic distance of cell o_j from the first cell o_1, where we approximate the geodesic distance as the sum of the Euclidean distances between the cells ranked next to each other, similar to the dimensionality reduction method Isomap [29]. Geodesic distances have previously been used for pseudotime estimation, see for instance [22, 32]. The importance of allowing pseudotime to deviate from rank time for sampling from the correct posterior distribution is illustrated in more detail in Section 3.2 below and Section 2 of the supplementary materials.

2.3 Gaussian process priors

The correct ordering $\mathbf{o}$ of cells is distinguished by comparatively low measurement noise σ_ϵ^2 in (1), since most of the variation is captured by the trajectory whose variability is determined by the scale parameter σ_w^2. Therefore informative priors for the noise parameters are necessary to ensure the model concentrates probability mass around the correct order and to avoid that a sampling or estimation algorithm gets trapped in local modes. Furthermore, since total variability is a sum of measurement noise and signal variability, we sample only σ_w^2 and set $\sigma_\epsilon^2 = V - \sigma_w^2$, where V is the sample variance taken across the entire $n_g \times T$ matrix of gene expression levels of T cells for n_g genes. The priors are as follows:

$$\begin{aligned}
\log(\sigma_w) &\sim \mathcal{N}(\log(\sqrt{0.9 \cdot V}), 0.01) \\
\log(l) &\sim \mathcal{N}(\log(\tfrac{1}{2}), v) \\
\mathbf{o} &\sim \text{uniform(permutations of } \{1, \ldots, T\}) \\
y_g(\mathbf{o}) \mid \sigma_w^2, l &\sim \mathcal{N}_T(0, \boldsymbol{\Sigma}(\boldsymbol{\tau}(\mathbf{o}), \sigma_w^2, l, V - \sigma_w^2))
\end{aligned}$$

We set $v = 0.1$ for the microarray data set considered in [35] (see Section 2.6.2) and $v = 0.01$ for the scRNA-seq data set [27] (see Section 2.6.3).

2.4 MCMC sampling

Markov Chain Monte Carlo (MCMC) methods [10] have been widely used to sample from continuous posterior densities in Bayesian statistics. They construct Markov Chains with the posterior distribution as their equilibrium. After convergence, each sample from the MCMC is taken as a sample from the posterior distribution. Our proposed method uses the Metropolis-Hastings algorithm [13, 18] for the sampling. For each given state of the Markov Chain, a new state is proposed using a proposal distribution, and accepted if an acceptance ratio is less than a uniform random number. While the construction of proposal distributions is often straightforward in the continuous case, we developed novel proposal moves to sample from discrete distributions of orders (see Section 2.5). For the sampling of the GP parameters we use Gaussian proposal distributions, adapting their standard deviation during burn-in aiming at acceptance rates between 0.45 and 0.5.

2.5 Sampling orderings

In the following we propose a Metropolis-Hastings algorithm for the sampling of the orderings. Preliminary experience with a variety of combinatorial moves to sample permutations led to the following set of five core moves, each with probability p_j, $j = 1, \ldots, 5$:

1. Move 1, **iterated swapping of neighbouring cells**: draw the number r_1 of swaps to be applied uniformly from $1, \ldots, n_0$ and draw r_1 swap positions $P_1, \ldots, P_{r_1}$ from $1, \ldots, T-1$ with replacement. Then iterate for $j = 1, \ldots, r_1$: swap cell at position P_j with its neighbor at position $P_j + 1$.
2. Move 2, **swapping of cells with short L^1-distances**: select two positions i and j according to probability $p_{ij} \propto \exp(-d(c_i, c_j)^2/\gamma_1)$, where d refers to the L_1 distances of cells c_i and c_j (as n_g-dimensional vectors) in these positions. Move c_i to position j and c_j to position i.
3. Move 3, **reversing segments between cells with short L^1-distances**: obtain two positions i and j as in move 2 and reverse the ordering of all cells in between, including cells at i and j.
4. Move 4, **short random permutations**: draw a number r_2 of short permutations uniformly from $1, \ldots, n_3$. For each $j = 1, \ldots, r_2$, draw a number $r_{3,j}$ uniformly from $3, \ldots, \max(n_{3a}, 3))$ and a cell position k_j uniformly from $1, \ldots, T - r_{3,j}$. Randomly permute the cells at positions $k_j, \ldots, k_j + r_{3,j}$.
5. Move 5, **reversing the entire ordering**.

The rationale for moves 2 and 3 is that two cells which are positioned apart in the ordering should only be exchanged (move 2) or the segment between them reversed (move 3) if these cells have similar expression profiles and the smoothness of the trajectory remains intact after the move. For move 1 we use a default setting of $n_0 = \lfloor T/4 \rfloor$ for the simulation studies. For move 4 we set $n_3 = \lfloor T/20 \rfloor$, and $n_{3a} = \lfloor T/12 \rfloor$. The distributions for choosing moves 2 and 3 may be tempered, that is taken to the power of a factor $0 < \alpha < 1$, to lower acceptance rates if required.

For the simulation studies we apply all possible combinations of moves 1 to 4 with equal probabilities and move 5 with a probability of 0.002. For the microarray data we apply only move 3, as (as will be shown below) it is the best sampling strategy for multi-modal distributions. For the scRNA-seq data set, we use moves 1 to 4 with probability 0.2495, and move 5 with probability 0.002. For the microarray data set we use $\gamma = 1000$ in move 3 and an additional tempering factor $a = 0.1$. For the scRNA-seq data set we set $\gamma = 4000$, without any tempering factor for moves 2 or 3.

As our posterior distribution is a symmetric function of the order, each order and its reverse will be sampled with equal probability from the posterior distribution. We remove this symmetry in further analysis by reversing orders which are negatively correlated with the capture times.

2.6 Data sets

2.6.1 Simulated data

The efficacy of the individual moves and of combinations of different moves for different types of data is first assessed on simulated data. We simulate $n_g = 50$ genes for $T = 90$ cells. For each simulation study we generate 16 data sets. On each of these data sets we run MCMC chains using all the possible combinations of the four proposed moves (with equal probability for combinations of more than one move). Since in the simulations we are mostly interested in the assessment of ordering moves and not any parameter estimation, we fix them to their true values and fix time points to rank time.

Simulation 1: three capture times, low noise. Each of the 16 data sets is generated as follows. First 90 temporal input points are drawn uniformly from $[0, 1]$. Then for each of the 50 genes in each of the simulated data sets, a parameter set for a GP underlying the trajectory of the simulated gene is drawn from

$$\begin{aligned}\log(\sigma_w) &\sim \mathcal{N}(0, 0.1)\\ \log(l) &\sim \mathcal{N}(\log(0.4), 0.1)\\ \log(\sigma_\epsilon) &\sim \mathcal{N}(\log(1/\sqrt{2}), 0.1).\end{aligned}$$

The data are assumed to be obtained at three capture times with 30 cells each.

Simulation 2: two capture times, low noise. The setup is similar to simulation 1, but with two capture times, where 30 cells are assigned to the first capture time, and the remaining 60 to the second.

Simulation 3: three capture times, high noise. The setup is similar to simulation 1, but $\log(\sigma_\epsilon) \sim \mathcal{N}(0, 0.1)$.

2.6.2 Microarray data

Windram et al. [35] studied the response of Arabidopsis thaliana to infection by the fungal pathogen Botrytis cinerea, generating microarray time series data over 48 hours, with measurements at intervals of 2 hours. As in Reid and Wernisch [24], we assume 4 capture times with 6 cells each. We compare the result to estimates produced by two established pseudotime methods, TSCAN [16, 15] and SLICER [32, 33], using the standard settings for the latter algorithms. For SLICER setting the number of edges of the nearest neighbours graph in the low dimensional space to 4 and 5 resulted in orders closest to the true one. For all analyses, we use the 150 genes mentioned in the paper by Windram et al. [35].

2.6.3 Single cell RNA-seq data

Shalek et al. [27] examined the response of primary mouse bone-marrow-derived dendritic cells in three different conditions using single-cell RNA-seq. We apply GPseudoRank to the lipopolysaccharide stimulated (LPS) condition. Shalek et al. [27] identified four modules of genes. As in Reid and Wernisch [24], we use a total of 74 genes from the four modules with the highest temporal variance relative to their noise levels [24]. The number of cells is 307, with 49 unstimulated cells, 75 captured after 1h, 65 after 2h, 60 after 4h, and 58 after 6h. We use an adjustment for cell size developed by Anders and Huber [2], also used in Reid and Wernisch [24].

2.7 Convergence assessment

For thorough convergence assessment, we run 12 different chains for each of the real data sets, and 5 for each of the simulation set-ups. For the simulated and the microarray data sets we run 100,000 iterations per MCMC chain and apply a thinning factor of 10. For the scRNA-seq data we use the same thinning factor, but 500,000 iterations. In order to assess convergence and not to bias the sampler towards specific orderings, all chains are seeded with random starting orders and with random GP parameters sampled from the prior distribution. However, we do restrict starting orders to permutations of cells within, but not across capture times.

To check convergence, we use the Gelman-Rubin $\hat{R}$-statistic [8], corrected for sampling variability [6], implemented in the R-package coda [21]. The $\hat{R}$-statistic estimates the factor by which the pooled variance across all the chains is larger than the within-sample variance. For convergent chains, $\hat{R}$ approaches 1 as the number of samples tends to infinity. According to [6], convergence may be assumed to have been reached if $\hat{R} < 1.2$. We apply the stricter recommendation of $\hat{R} < 1.1$ [9]. We compute the $\hat{R}$ statistics for the following two quantities: first, the log-likelihood, and second the L^1-distances of the sampled cell positions from a fixed reference set of cell positions, for which we use the true order, if known, and $1, \dots, T$, where T is the number of cells, in case of scRNA-seq data. We compute the $\hat{R}$ statistics a number of times during sampling, each time discarding the first 50% [9]. We compare the speed of convergence for different combinations of proposal moves in the simulation studies. See Section 1 in the supplementary materials for details.

3 Results

3.1 Simulation studies

This section summarises the insights gained from the simulation studies. For details on thassessment criteria and results, see Section 1 in the supplementary materials.

Simulation 1. Any combination of moves leads to good convergence, and although there are differences in the speed and level of convergence, any combination of moves is recommended.

Simulation 2. There are only two capture times, hence there is more variety in the starting orders for each chain. The performance of the combinations of moves is different from simulation 1. Move 3 performs better than any other single move.

Move 3 generally traverses the space of permutations faster by reversing whole segments of an ordering and it is the only move for which all $\hat{R}$-statistics go below 1.1 within the first 10,000 thinned samples. The combination of moves ranked first according to the criteria described in Section 1 of the supplementary materials is the combination 1,2,3,4 of all the moves.

Simulation 3. All moves and combinations thereof perform well in this situation, though move 3, while still achieving reasonable levels of convergence, is now the comparatively less well performing single move. The combination of all four moves performs well.

3.2 Validation on microarray data

The experimental data set has been acquired at equidistant time points every 2 hours. However, to adjust for differences in the speed of biological development during the process, we apply GPseudoRank with irregular pseudotimes

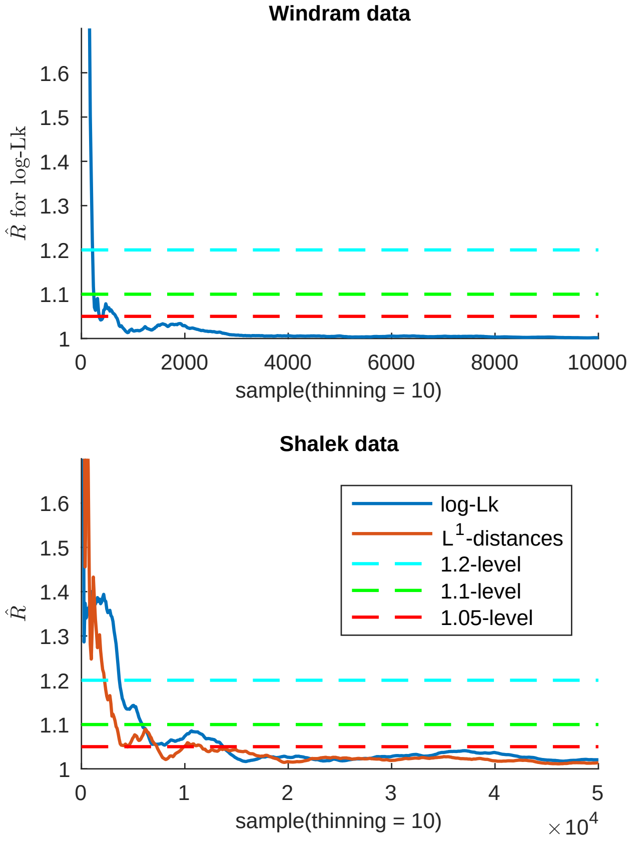

Figure 1: **Convergence analysis for GPseudoRank.** Gelman-Rubin statistics for the log-likelihood and for the L^1-distances of the sampled permutations of cell positions from the reference permutation (Shalek data).

(as explained above in Section 2.2). In fact, adjusting for the speed of biological development is needed, and an approximation with simple equidistant input points for the GP changes the posterior distribution significantly, as shown in Section 2 of the supplementary materials. Because of the bi-modality of the L^1-distances from the true cell positions (Figure 2), the Gelman-Rubin statistic for this distance is less useful and we show the statistic for the log-likelihood instead (Figure 1).

Figure 1 suggests a very fast convergence. However, if multi-modality is suspected, which might not show in the log likelihood trace, sampling beyond convergence for the log likelihood is recommended. Further plots illustrating convergence can be found in Section 1 of the supplementary materials.

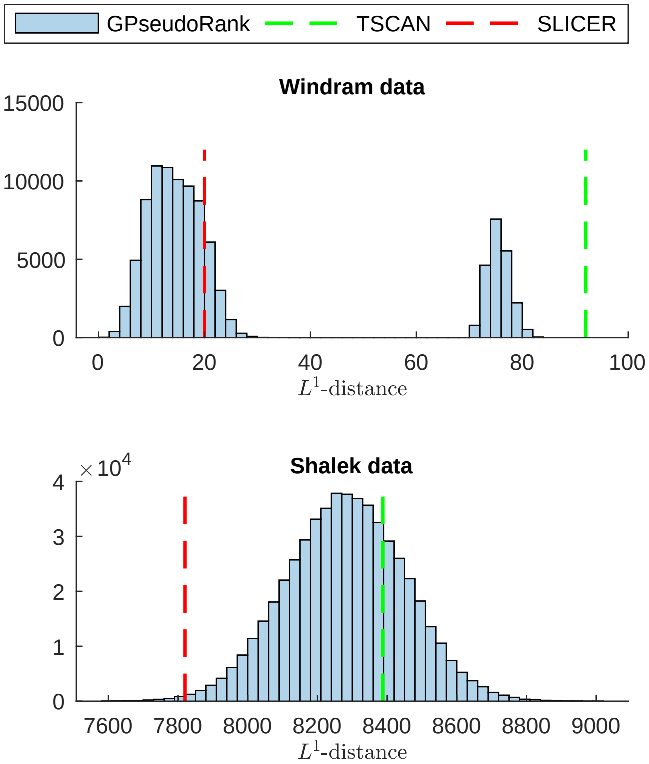

Figure 2: **Histogram of $\mathbf{L^1}$-distances from the reference permutation of cell positions.** Distribution sampled with GPseudoRank, point estimates with TSCAN and SLICER.

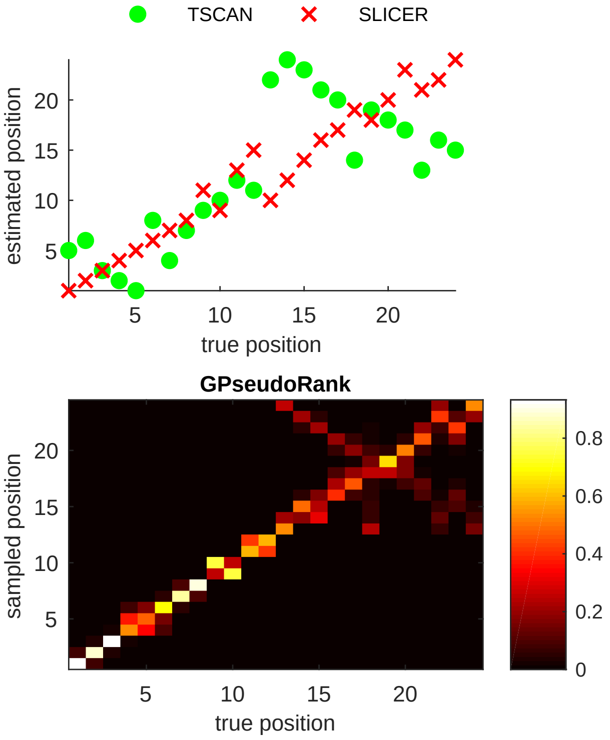

Figure 3: **Comparing bi-modal posterior by GPseudoRank to point estimates by TSCAN and SLICER: Windram data.** For GPSeudoRank, the matrix illustrates the posterior probabilities of the positions of the cells: the true cell position is along the x-axis, the posterior density is plotted along the y-axis. For TSCAN and SLICER, we plotted along the y-axis the estimated position.

As illustrated by Figure 2, the distribution of the L^1-distances of the sampled permutations of cell positions from the correct order is bi-modal. The estimates provided by SLICER [32, 33] and TSCAN [16, 15] fall in different modes illustrating the importance of sampling from the distribution of the orderings rather than just obtaining a single estimate. The DeLorean MCMC sampler, sampling continuous pseudotimes, is also unable to capture the multi-modality of the posterior [24, Figure 1]. Figure 3 illustrates that the posterior distribution sampled by GPseudoRank covers the two point estimates obtained from TSCAN and from SLICER. For GPseudoRank, it contains the samples from one randomly selected MCMC chain. For the corresponding plots of each of the 12 chains for convergence analysis, see Section 2 in the supplementary materials.

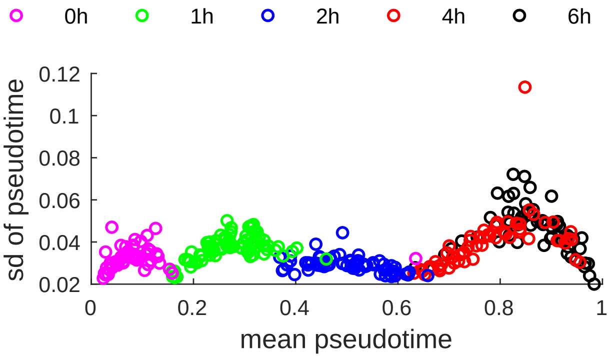

Figure 4: **Uncertainty of pseudotime as a function of mean pseudotime.** For each cell, the mean pseudotime is plotted along the x-axis, and the respective standard deviation along the y-axis. Cells are coloured by capture time.

3.3 Pseudotemporal uncertainty varies during response to infection

For the scRNA-seq data from Shalek et al. [27], collected at five different capture times, the true cell ordering is unknown. To check convergence of orders the $\hat{R}$-statistic is computed both on the log-likelihood and on the L^1-distances of the permutation of cell positions to an arbitrary reference permutation (Figure 1).

Figure 1 shows that a threshold for the $\hat{R}$-statistic of 1.1 has been reached after 10,000 thinned samples. We therefore discard a burn-in of 5,000 thinned samples at the beginning of each chain, as recommended by Gelman and Shirley [9]. Indeed, by the 1.1 threshold for the $\hat{R}$ statistic 10,000 thinned samples would have been sufficient for convergence.

Figure 2 demonstrates again the value of providing a posterior distribution for orders, rather than a single estimate: TSCAN and SLICER give different results. The TSCAN result is compatible with the sampled distribution, however, the SLICER result seems to be an outlier. Knowledge of the uncertainty can prevent over-confidence in the results.

Figure 4 illustrates the uncertainty of the pseudotime over the mean pseudotime. To ensure that the inverted U-shape in the amount of uncertainties of the first two capture times at 0h and 1h is not a sampling artifact, cells from these capture times were mixed together for initialising the sampler (that is, capture time information was discarded). On the other hand, despite being separated during initialisation of the sampler, cells from capture times 4h and 6h are completely merged, again indicating that the sample has reached convergence.

Overall uncertainty in the ordering of cells is markedly lower around capture time 2h, when the reaction to the infection has set in, but is not yet complete.

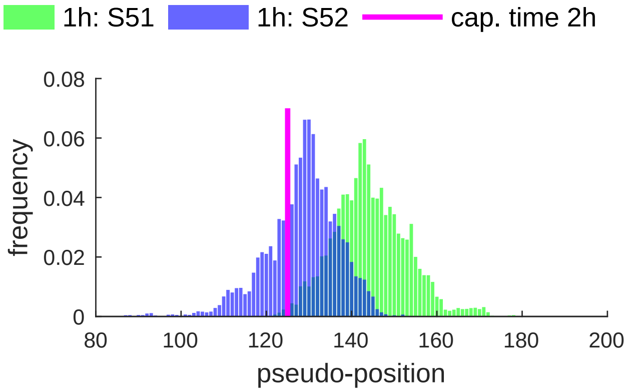

Figure 5: **Posterior distribution of cell positions of the precocious cells.** For each posterior position of the cells we plot the frequency at which this position occurs among all samples. One random MCMC chain was used. Both of the precocious cells have a high probability of being located within capture time 2h, with S51 likely to be ahead of S52.

The slight U-shape in the amount of uncertainty for capture times 0h, 1h, and 4h/6h seems to be an experimental batch effect of capturing multiple heterogeneous cells at different time points. Within a batch (or merged batches 4h and 6h) cells which are either lagging behind or slightly ahead in their development are assigned a more specific pseudotime with lower uncertainty behind or ahead of the bulk of cells whose pseudotimes are more interchangeable with higher uncertainty.

GPseudoRank identifies two precocious cells, pointed out in the original analysis by [27], ahead in terms of their response to the stimulus, see Figure 5. Shalek et al. identified a set of genes particularly associated with antiviral response. Ahmed et al. and Reid and Wernisch also used this score to demonstrate that their methods identify two cells at capture time 1h precocious in their antiviral response. Figure 6 shows the average expression of a set of genes associated with antiviral response for each cell. As expected, this antiviral score increases over pseudotime, confirming that the pseudotime assignment captures a biological phenomenon. In contrast to Figure 6, both DeLorean [24] and GrandPrix [1] show considerable edge effects in comparable plots [24, Fig. 4], [1, Fig. 2]. Such edge effects are not biologically motivated and presumably algorithmic artifacts which GPseudoRank is able to avoid by restricting pseudotimes to a finite interval and by using a geodesic mapping.

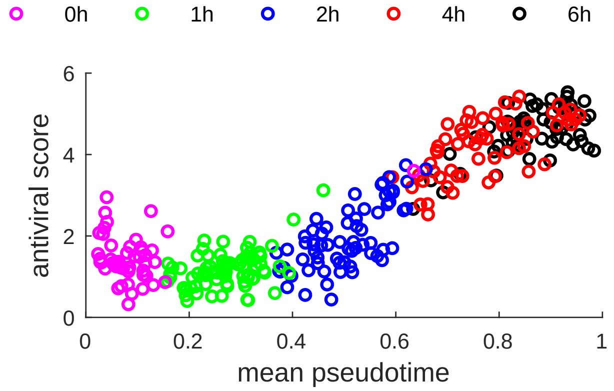

Figure 6: **Core antiviral score as a function of mean pseudotime**

4 Discussion

GPseudoRank is a new type of Gaussian process latent variable model for pseudotemporal ordering. It samples orderings instead of pseudotimes, with combinatorial proposal moves designed to allow the Metropolis-Hastings sampler to make large changes to permutations and still achieve a high acceptance rate. Figure 2 clearly illustrates the advantage of sampling from a posterior distribution of cell orderings over deriving a single estimate. Although for this data set the true ordering is known, our sampler shows that orders near a second distinct mode are still likely. In fact one such alternative order is returned by a popular algorithm. For data with unknown order, knowledge of the possibility of alternative modes is preferable to the return of just one arbitrary solution.

For the microarray data set we used move 3 only, which reverses whole segments of a permutation, because of its particular suitability for capturing multimodality. We therefore recommend to run two MCMC chains of GPseudoRank in parallel: one with move 3 only, which performs best in case of multi-modality, and a second one with all moves, for faster convergence for less complicated distributions of orders with higher noise levels, as our simulation studies show.

The application to an scRNA-seq data set illustrates another advantage of sampling from the posterior of orderings: the amount of uncertainty about the position of a cell can vary with time. In this case, the uncertainty is lowest in the middle of the process, where the heterogeneity of cells with regard to their progress through the response to the infection is highest. This identifies parts of the process with increased change and higher biological variability compared to technical noise.

The uncertainty of the orders is relevant to any further analysis that models scRNA-seq data in terms of time-series data. This applies, for instance, to any type of network inference where the order of the input time series is relevant, including GP models [20] and vector-autoregressive ones [19]. Alternatively, identifying the regions of the process where the uncertainty of a cell's position is low can support the selection of suitable cells for the clustering of genes, for example.

Variational inference, which avoids sampling altogether, is considered a computationally efficient if only approximate Bayesian inference alternative to MCMC sampling. However, it turns out that MCMC sampling from discrete permutations in GPseudoRank is efficient enough that its run time is comparable to that of a variational approach: 100,000 iterations for the Windram data take 7min 20s on a single Intel Xeon X5 2.0GHz CPU, compared to about 3 minutes for each initialisation of the variational sampler in DeLorean on one core of an AMD 6174 2.2 Ghz CPU. Similarly, for the scRNA-seq data set, sampling the 100,000 samples shown to be sufficient for convergence takes about 40 minutes, compared to 20 minutes for each initialisation of the variational sampler in DeLorean.

Overall, GPseudoRank offers new insights into biological phenomena and experimental artifacts. It quantifies the amount and variability of uncertainty in single-cell ordering (Figure 4). Assessing the degree of uncertainty enables spotting experimental batch effects created by sampling from a continuous spectrum of developmental stages at only a few capture times. Our approach is also able to identify precocious cells (Figure 5). By combining a geodesic pseudotime mapping with sampling permutations, GPseudoRank also avoids edge effects present in other GP methods for pseudotime ordering (Figure 6).

References

[1] S Ahmed, M Rattray, and A Boukouvalas. GrandPrix: Scaling up the Bayesian GPLVM for single-cell data. *bioRxiv*, 2017. doi: 10.1101/227843.

[2] S Anders and W Huber. Differential expression analysis for sequence count data. *Genome Biol*, 11(10):R106–R106, 2010. doi: 10.1186/gb-2010-11-10-r106.

[3] P Angerer, L Haghverdi, M Büttner, F J Theis, C Marr, and F Buettner. destiny: diffusion maps for large-scale single-cell data in R. *Bioinformatics*, 32(8):1241–1243, 2016. doi: 10.1093/bioinformatics/btv715.

[4] S C Bendall, K L Davis, el-A D Amir, M D Tadmor, E F Simonds, T J Chen, et al. Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. *Cell*, 157(3):714–725, 2014. doi: 10.1016/j.cell.2014.04.005.

[5] P Brennecke, S Anders, J K Kim, A A Kolodziejczyk, X Zhang,

V Proserpio, et al. Accounting for technical noise in single-cell RNA-seq experiments. *Nat Meth*, 10(11):1093–1095, 2013. doi: 10.1038/nmeth.2645.

[6] S P Brooks and A Gelman. General methods for monitoring convergence of iterative simulations. *J Comput Graph Stat*, 7(4):434–455, 1998. doi: 10.1080/10618600.1998.10474787.

[7] K R Campbell and C Yau. Order under uncertainty: robust differential expression analysis using probabilistic models for pseudotime inference. *PLOS Comput Biol*, 12(11):e1005212, 2016. doi: 10.1371/journal.pcbi.1005212.

[8] A Gelman and D B Rubin. Inference from iterative simulation using multiple sequences. *Stat Sci*, 7(4):457–472, 1992.

[9] A Gelman and K Shirley. Inference from simulations and monitoring convergence. In S Brooks, A Gelman, G L Jones, and X-L Meng, editors, *Handbook of Markov Chain Monte Carlo*, volume 6, pages 163–174. CRC, Boca Raton, 2011.

[10] W R Gilks, S Richardson, and D J Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 1996.

[11] L Haghverdi, F Buettner, and F J Theis. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics*, 31 (18):2989–2998, 2015. doi: 10.1093/bioinformatics/btv325.

[12] L Haghverdi, M Büttner, F A Wolf, F Buettner, and F J Theis. Diffusion pseudotime robustly reconstructs lineage branching. *Nat Meth*, 13(10): 845–848, 2016. doi: 10.1038/nmeth.3971.

[13] W K Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97.

[14] M D Hoffman and A Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J Mach Learn Res*, pages 1593–1623, 2014.

[15] Z Ji and H Ji. *TSCAN: Tools for Single- Cell ANalysis*, 2015. R package version 1.14.0.

[16] Z Ji and H Ji. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res*, 44(13):e117–e117, 2016. doi: 10.1093/nar/gkw430.

[17] Q Mao, L Wang, S Goodison, and Y Sun. Dimensionality reduction via graph structure learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 765–774, New York, 2015. ACM. doi: 10.1145/2783258.2783309.

[18] N Metropolis, A W Rosenbluth, M N Rosenbluth, A H Teller, and E Teller.

Equation of state calculations by fast computing machines. *J Chem Phys*, 21:1087–1092, 1953. doi: 10.1063/1.1699114.

[19] R Opgen-Rhein and K Strimmer. Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process. *BMC Bioinformatics*, 8(2):S3, 2007. doi: 10.1186/1471-2105-8-S2-S3.

[20] C A Penfold, V Buchanan-Wollaston, K J Denby, and D L Wild. Nonparametric Bayesian inference for perturbed and orthologous gene regulatory networks. *Bioinformatics*, 28(12):i233–i241, 2012. doi: 10.1093/bioinformatics/bts222.

[21] M Plummer, N Best, K Cowles, and K Vines. CODA: convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.

[22] X Qiu, Q Mao, Y Tang, L Wang, R Chawla, H A Pliner, and C Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nat Meth*, 14:979–982, 2017. doi: 10.1038/nmeth.4402.

[23] C E Rasmussen and C K I Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge, MA, 2006.

[24] J E Reid and L Wernisch. Pseudotime estimation: deconfounding single cell time series. *Bioinformatics*, 32(19):2973–2980, 2016. doi: 10.1093/bioinformatics/btw372.

[25] S T Roweis and L K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. doi: 10.1126/science.290.5500.2323.

[26] M Setty, M D Tadmor, S Reich-Zeliger, O Angel, T M Salame, P Kathail, et al. Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nat Biotech*, 34(6):637–645, 2016. doi: 10.1038/nbt.3569.

[27] A K Shalek, R Satija, J Shuga, J J Trombetta, D Gennert, D Lu, et al. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature*, 510:363–369, 2014. doi: 10.1038/nature13437.

[28] O Stegle, S A Teichmann, and J C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nat Rev Genet*, 16(3):133–145, 2015. doi: 10.1038/nrg3833.

[29] J B Tenenbaum, V de Silva, and J C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[30] C Trapnell, D Cacchiarelli, J Grimsby, P Pokharel, S Li, M Morse, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol*, 32(4):381–386, 2014. doi: 10.1038/nbt.2859.

[31] C A Vallejos, J C Marioni, and S Richardson. BASiCS: Bayesian analysis of single-cell sequencing data. *PLOS Comput Biol*, 11(6):1–18, 2015. doi: 10.1371/journal.pcbi.1004333.

[32] J D Welch. *SLICER: Selective Locally Linear Inference of Cellular Expression Relationships*, 2017. R package version 0.2.0.

[33] J D Welch, A J Hartemink, and J F Prins. SLICER: inferring branched, nonlinear cellular trajectories from single cell RNA-seq data. *Genome Biol*, 17(1):106, 2016. doi: 10.1186/s13059-016-0975-3.

[34] J D Welch, A J Hartemink, and J F Prins. MATCHER: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. *Genome Biol*, 18(1):138, 2017. doi: 10.1186/s13059-017-1269-0.

[35] O Windram et al. Arabidopsis defense against botrytis cinerea: Chronology and regulation deciphered by high-resolution temporal transcriptomic analysis. *The Plant Cell*, 24(9):3530–3557, 2012. doi: 10.1105/tpc.112.102046.