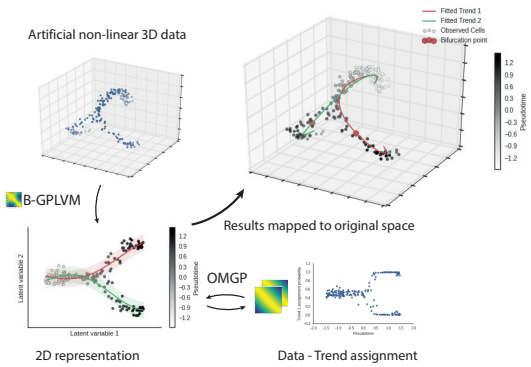
Supplementary Computational Methods - The GPfates model

1 Introduction

GPfates is based on a three-stage approach that first i) infers a low-dimensional representation of single-cell RNA-seq data, then ii) infers pseudotime to iii) model the temporal dynamics of gene expression profiles with a mixture model. These steps build on existing modeling components: The Gaussian Process Latent Variable Model (*16*), and the Overlapping Mixture of Gaussian Processes (*17*). For a graphical illustration of the major steps involved in this analysis, see Figure 2D of the main text (as well as Supp. Comp. Fig 1). In Sections 2 and 3 we describe the statistical models that underlie the components of GPfates. In Section 4 we describe downstream analysis methods for interpreting the fitted model followed by the integrated implementation and combination with existing workflows in Section 5. Section 6 and 7 include additional validation experiments using simulations, robustness analyses and analyses on multiple existing data sets. Finally, in Section 8, we compare the outcome of other published methods inferring pseudotime and branching events on the same data.



Sup. Comp. Fig. 1: Illustration of the analysis workflow. A low dimensional parametrization of the data is found using Bayesian GPLVM. The low-dimensional representation is viewed as a mixture problem, and solved by an Overlapping Mixture of Gaussian Processes. This allows us to represent our cells as members of different smooth processes. But also interpret in terms of the high-dimensional space parametrized by the GPLVM.

2 Pseudotime inference

2.1 Gaussian Process Regression

A main component of GPfates is to model temporal transitions. We use the Gaussian process (GP) framework, thereby casting this problem as non-parametric regression. Let us begin by assuming that the developmental time t for each cell we observe is known. Then, the output y_g (i.e. expression of gene g) is modelled as a continuous function of the input t (i.e. developmental

progression)

$$y_g = f(t) + \varepsilon, \tag{1}$$

where

$$p(\varepsilon) = \mathcal{N}(0, \sigma^2)$$

is Gaussian distributed residual noise and $f(t)$ denotes the unknown regression function. In this work y_g is considered to be an N-dimensional vector of N cells with observed expression of the gene g. We denote the expression of g in an individual cell n as $[y_g]_n$.

A GP can be interpreted as a function-valued prior on the elements of f, which is defined by a covariance function that in turn is parametrized by the input (developmental time) t:

$$\mathrm{cov}(f(t_{n_1}), f(t_{n_2})) = k(t_n, t_{n_2}).$$

The *covariance function* $k(t_{n_1}, t_{n_2})$ encodes prior assumptions on the smoothness and lengthscales of the function $f(t)$. The most widely used covariance function is the Squared Exponential (SE) covariance function,

$$k(t_{n_1}, t_{n_2}) = \sigma^2_{\mathrm{SE}} \exp\left(-\frac{|t_{n_1} - t_{n_2}|^2}{2l^2_{\mathrm{SE}}}\right), \tag{2}$$

and this is the covariance function we will generally be used in this work. This covariance has the hyperparameters $\theta = (\sigma^2_{\mathrm{SE}}, l^2_{\mathrm{SE}})$, which parametrize the amplitude (σ^2_{SE}) and the lengthscale (l^2_{SE}) of functions under the prior. Throughout the remainder of the text we will omit the hyperparameters from equations for the sake of brevity. Note that there is a whole compendium of valid covariance functions, which can also be combined using sum or multiplication; see (*50*) for an overview.

We write that a function f is *Gaussian Process distributed* by

$$f(t) \sim \mathcal{GP}(0, k(t_{n_1}, t_{n_2})).$$

This prior on the function f can be linked to the finite observed data using a Gaussian likelihood:

$$p(y_g|f) = \prod_{n=1}^{N} \mathcal{N}([y_g]_n | f_n, \sigma^2).$$

Together with the prior on the corresponding (finite) elements of f,

$$p(f) = \mathcal{N}(f|0, \boldsymbol{K}_t),$$

this results in the marginal likelihood

$$p(y_g|t) = \mathcal{N}(y_g|0, \boldsymbol{K}_t + \sigma^2 \cdot I).$$

Here $\boldsymbol{K}_t$ is an $N \times N$ matrix of pairwise evaluations of the covariance functions at the observed times t. I.e.

$$[\boldsymbol{K}_t]_{n,m} = k(t_n, t_m). \tag{3}$$

By considering the joint distribution of the observed data y_g and an unseen function value $f(t_\star)$, it is possible to derive the predictive distribution for $f(t_\star)$:

$$p(f(t_\star)|t, y_g, t_\star) = \mathcal{N}\left(\overline{f(t_\star)}, k(t_\star)\right),$$

where

$$\overline{f(t_\star)} = k(t_\star, t)[\boldsymbol{K}_t + \sigma^2 \cdot I]^{-1} y_g, \text{ and}$$
$$k(t_\star) = k(t_\star, t_\star) - k(t_\star, t)[\boldsymbol{K}_t + \sigma^2 \cdot I]^{-1} k(t, t_\star).$$

For a full review on Gaussian Processes, see (*50*).

So far, we have only described Gaussian Process Regression for expression y_g of a single gene g. If we consider a collection of G genes $\{1, \ldots, G\}$, their expression can be modelled together by

$$(y_1, \ldots, y_G) = (f^1(t), \ldots, f^G(t)) + (\varepsilon, \ldots, \varepsilon). \tag{4}$$

We use Y to compactly denote the $N \times G$ expression matrix of cells $\times$ genes, where

$$Y_{n,g} = [y_g]_n.$$

The assumption that all genes are governed by similar functional relationships with t means we place the same GP prior (with shared covariance function):

$$p(Y|t) = \prod_{g=1}^{G} p(y_g|t) = \prod_{g=1}^{G} \mathcal{N}(y_g|0, \boldsymbol{K}_t + \sigma^2 \cdot I). \tag{5}$$

In the next section we will see the usefulness of considering multiple genes at once.

2.2 Pseudotime inference by Bayesian GPLVM with *per-cell prior*

The Gaussian Process regression framework described above assumes we know the time t of each cell. While many scRNAseq experiments record data over some time course, these snapshots sample cells from a population where responses are asynchronous. Each cell has reached a certain stage in the differentiation process under investigation, which we do not observe directly. The progress in to this process is referred to as *pseudotime*. We can however infer this from the data. In the Gaussian Process Latent Variable Model (GPLVM) (*16*), we use the multiple output case of Gaussian Process regression (equation 4), but consider the values of t to be parameters which we wish to infer.

The joint probability of the GPLVM is

$$p(Y, t) = p(Y|t)p(t),$$

where $p(Y|t)$ is defined in equation 5, and the prior $p(t)$ is such that for cell n,

$$p(t_n) = \mathcal{N}(0, 1).$$

Following (*52*), we can also consider the prior $p(t)$ to be informed about the experimental ordering of collection times of the cells, putting the mean of t_n to correspond to the time point of cell n. If we use our Malaria time course as and example, we can put the prior on t so that

$$p(t_n) = \mathcal{N}(\text{day}_n, \sigma^2_{\text{prior}}),$$

where $\text{day}_n \in \{1, 2, 3, 4, 5\}$ correspond to the collection order of those cells. The parameter σ^2_{prior} alters the strength of the prior.

The objective of Bayesian GPLVM (*51*), is to find the posterior probability distribution $p(t|Y) \propto p(Y|t)p(t)$. This is intractable though, due to the t values appearing non-linearly in the matrix inverse $[\boldsymbol{K}_t + \sigma^2 \cdot I]^{-1}$.

In (*51*), a lower bound to the marginal likelihood is calculated by estimating the posterior $p(t|Y)$ by a variational distribution $q(t)$. The distribution

$$q(t) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mu_n, S_n)$$

is described in that paper, and Bayesian training of the model to maximize this lower bound. This is the method we use.

For a simple and fast implementation of *per-cell* priors, in practice we use a prior on the variation parameters μ_n rather than the latent t_n. Rather than a lower bound of the marginal likelihood the Bayesian GPLVM will optimize the lower bound marginal maximum a posteriori probability (MAP).

2.3 Dimensionality reduction

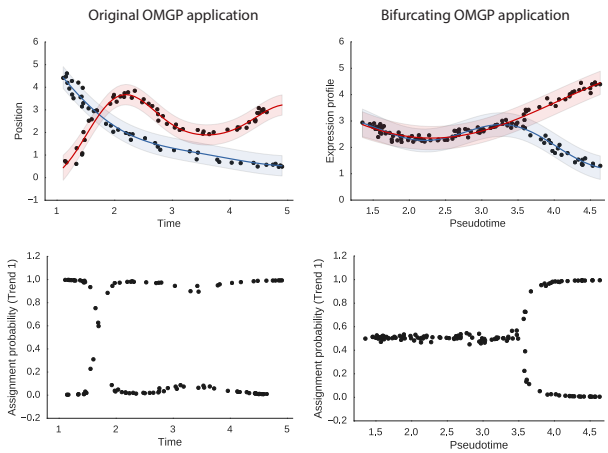
In many cases it is useful to work on a reduced representation of cellular expression profiles. For example, when modelling transcriptomic data, fitting a model to a low-dimensional representation can be preferable to fitting it to expression profiles of thousands of genes. Formally, the objective of *dimensionality reductions* is to find some M-dimensional representation of the G-dimensional expression measurements, where $M << G$. Typically M is 2 or 3, which aides visual interpretation. Analogous to the pseudotime inference, these latent cell states can also be inferred using the GPLVM. Say X is an $M \times N$ matrix so that each cell n correspond to an M-dimensional vector,

$$X_n = (x_n^1, \ldots, x_n^M).$$

We want to model the expression matrix Y so that

$$[y_g]_n = f^g(x_n^1, \ldots, x_n^M) + \varepsilon = f^g(X_n) + \varepsilon.$$

Note that now the covariance function is evaluated as $k(X_{n_1}, X_{n_2})$, where, in the Squared Exponential covariance function in equation 2, the operator $|\cdot|$ is evaluated as the Euclidean norm for vectors, rather than absolute value.



Sup. Comp. Fig. 2: Comparison of the original OMGP use case (left) and our use case (right), in both cases where the number of trends $K = 2$. In the original use case trends are expected to be independent throughout time, albeit with some ambiguity in some locations. In our application, we interpret ambiguous cell assignment to be in a common precursor state.

Just as the t_n values are inferred from data above, so can the X_n vectors be inferred from the data.

3 Bifurcation inference using overlapping mixtures of Gaussian processes

In a continuous setting, a bifurcating process can be seen as one function, splitting apart into two functions over time. One approach to model this could be to consider two functions throughout time, but before the bifurcation happens, the two functions are identical. With this in mind, we can use a mixture model to tease apart the shared and bifurcated functions.

3.1 Mixture model

Mixture models are hierarchical models where an observation is assumed to be generated from one of C *components*, each of which is described by its own model. The goal of mixture models is to infer which component an observation stems from, and at the same time model that component.

The Overlapping Mixture of Gaussian Processes (OMGP) model (*17*) assumes there are C different underlying latent functions producing the N observed cells. This model was originally developed for the application of missile tracking, and in that setting an observation is e.g. a radar based location at a given real time point. As such, the main focus of the definition of the model is for the case of C completely independent components. The approach presented here is based on the realisation that the model would also be able to handle the case of *branching* trajectories. There would simply be a time interval where it does not matter which mixture trajectory data is sampled from. In our setting, an observation is a single cell, and the analog to real time is pseudotime (Supp. Comp. Fig. 2). As an additional extension, we phrase a version of the OMGP model which is non-parametric in the number of trajectories.

In the original regression case described in equation 1, data is assumed to be generated by a single smooth unknown function. When modeling our gene expression data with the Overlapping Mixture of Gaussian Processes, data is considered to be generated by

$$X = f_c(t) + \varepsilon.$$

However, we are lacking information about which latent function f_c generated any given observation (t_n, X_n) of pseudotime and gene expression for the N observed cells. Here X correspond to some representation of the transcriptional state of the cells. It could be the expression of all genes ($X = Y$), a single gene ($X = y_g$), or an M-dimensional inferred representation as discussed above.

This is viewed as a mixture modelling problem, where each cell has a latent variable z_i specifying to which component f_c the cell should be allocated to. Write F for the collection of all latent

functions. The covariance functions k_c for each f_c can be different from each other, though for the applications we discuss here, we take them as Squared Exponential covariance functions with different hyperparameter values.

In the OMGP formulation, the likelihood is

$$p(X|F,T,Z) = \prod_{n=1}^{N}\prod_{c=1}^{C}\mathcal{N}(x_i|f_c(t_n),\sigma^2)^{z_{nc}}.$$

We specify a multinomial prior on the latent variables Z, namely

$$p(Z) = \prod_{n=1}^{N}\prod_{c=1}^{C}\Pi_{n,c}^{z_{n,c}},$$

$$\sum_{c=1}^{C}\Pi_{n,c} = 1.$$

Additionally, each of the latent functions f_c has an independent Gaussian process prior:

$$p(F|T) = \prod_{c=1}^{C}\mathcal{N}(f_c|0,\boldsymbol{K}_t^c).$$

The covariance matrices $\boldsymbol{K}_t^1,\ldots,\boldsymbol{K}_t^C$ for the latent functions $f_1,\ldots,f_C$ are generated from a covariance functions $k_1(t_{n_1},t_{n_2}),\ldots,k_C(t_{n_1},t_{n_2})$ like in equation 3.

Now we rephrase this as a Dirichlet Process Gaussian Process mixture model (*53*). Let every latent function f_c have an associated "stick-breaking length" v_c, based on the "stick-breaking" formulation of the Dirichlet Process. Here $V = [v_1,\cdots,v_\infty]$ is the collection of stick-breaking lengths for constructing the Dirichlet process for the assignment. The joint distribution of the OMGP model is

$$p(X,Z,V,F) = p(F|T)p(X|F,Z)p(Z|V)p(V|\alpha).$$

The value α is a parameter of the model which controls the expected concentrations of mixtures (which we in practice take as $\alpha = 1$, a common default), and

$$p(V|\alpha) = \prod_{c=1}^{\infty}Beta(v_c|1,\alpha),$$

where $Beta(\cdot,\cdot)$ is the beta distribution. The prior distribution over the collection of Gaussian Processes is

$$p(F|T) = \prod_{c=1}^{\infty}\mathcal{N}(f_c|0,\boldsymbol{K}_c).$$

Following the stick-breaking formulation,

$$p(Z|V) = \prod_{i=1}^{N}\prod_{c=1}^{\infty}\pi_c(V)^{z_{i,c}},$$

where $\pi_c(V) = v_c\prod_{j=1}^{c-1}(1-v_j)$.

The assignments between observations X and the latent functions F is given by a binary $N\times C$ matrix Z. The assignments to latent functions are considered as additional variational parameters. Let $\boldsymbol{\phi}$ be an $N\times C$ matrix where ϕ_{nc} is the approximate posterior probability of assigning the nth observation to the cth latent function. The $\boldsymbol{\phi}$ parameters are inferred by collapsed variational inference as described in (*54*). Overall, the likelihood of the model is

$$p(X|F,Z) = \prod_{n=1}^{N}\prod_{c=1}^{\infty}\mathcal{N}(x_n|f_c,\boldsymbol{K}_c)^{z_{n,c}}.$$

(It should be noted that everything described generalizes to the case where the latent functions f_c are vector valued, as long as all output dimensions of such a function share the same covariance function. In this case, probabilities factorize over output dimensions, but beyond that all calculations are the same.)

3.2 Parameter inference

In (*17*) the latent variables $\mathbf{Z}$ in the parametric version of OMGP were inferred using an expectation-maximization scheme. Here we describe how we perform variational inference for the $\boldsymbol{\phi}$-parameters in the non-parametric version of the model.

To make the inference problem tractable, the variational distribution $q(Z)$ is introduced with

variational parameters ϕ, at a given truncation level C such that

$$q(Z) = \prod_{n=1}^{N} \prod_{c=1}^{C} \phi_{n,c}^{z_{n,c}}.$$

with the objective of approximating $p(Z|F, X, T)$.

The lower bound of the log-likelihood of the OMGP model, which we write as $\mathcal{L}_{\mathrm{KL}}$, when approximating $p(Z)$ by $q(Z)$ can be split up in three terms as

$$\mathcal{L}_{\mathrm{KL}} = \mathcal{L}^{\mathrm{M}} + \mathcal{L}^{\mathrm{MP}} + \mathcal{L}^{\mathrm{H}}.$$

Here $\mathcal{L}^{\mathrm{M}} = \sum_{c=1}^{C} \mathcal{L}_c^{\mathrm{M}}$ is the log-likelihood of the latent functions as represented by Gaussian processes. For the cth latent function, the variational distribution of f_c which maximizes the lower bound was derived in (*17*) to be

$$q(f_c) = \mathcal{N}(f_c|\mu_c, \boldsymbol{\Sigma}_c)$$

where $\boldsymbol{\Sigma}_c = (\boldsymbol{K}_c^{-1} + \boldsymbol{B}_c)^{-1}$, and $\mu_c = \boldsymbol{\Sigma}_c \boldsymbol{B}_c y_c$. Here $\boldsymbol{B}_c$ is a diagonal matrix with entries $[\boldsymbol{B}_c]_{i,i} = \frac{\phi_{i,c}}{\sigma^2}$. Thus the log-likelihood for a particular latent function f_c, assuming we have optimal assignments ϕ, is

$$\mathcal{L}_c^{\mathrm{M}} = -\frac{1}{2} y^T \boldsymbol{\Sigma}_c^{-1} y - \frac{1}{2} \ln |\boldsymbol{\Sigma}_k| - \frac{N}{2} \ln 2\pi.$$

The second and third parts of $\mathcal{L}_{KL}$ were derived in [Hensman2014] as

$$\begin{aligned}\mathcal{L}^{\mathrm{MP}} &= \ln \int \exp\{\mathbb{E}_{q(Z)} [\ln p(Z|V)]\} p(V) dV \\ &= \ln \prod_{c=1}^{C} \left(\frac{\Gamma(\hat{\phi}_c + 1)\Gamma(\tilde{\phi}_c + \alpha)\alpha}{\Gamma(\hat{\phi}_c + \tilde{\phi}_c + \alpha + 1)} \right)\end{aligned}$$

and

$$\mathcal{L}^H = -\mathbb{E}_{q(Z)} [\ln q(Z)].$$

For optimizing variational mixture assignment parameters we follow (*54*), and use *natural gradient descent*. For hyperparameters of the kernels, as well as the variance parameter σ^2 of the model, we perform gradient descent.

If we know $\frac{\partial \mathcal{L}_{KL}}{\partial \phi}$ we can calculate the natural gradient by equation (22) in (*53*). The gradients $\frac{\partial \mathcal{L}^{MP}}{\partial \phi}$ and $\frac{\partial \mathcal{L}^{H}}{\partial \phi}$ were derived in (*53*), the only unknown part is $\frac{\partial \mathcal{L}^{M}}{\partial \phi}$.

We then use the identity $\frac{\partial \mathcal{L}_k^M}{\partial \phi_{n,i}} = \frac{1}{2} \operatorname{Tr}\left(\frac{\partial \mathcal{L}_c^M}{\partial \boldsymbol{B}_c^{-1}} \cdot \frac{\partial \boldsymbol{B}_c^{-1}}{\partial \phi_{n,i}} \right)$. Here $\frac{\partial \mathcal{L}_c^M}{\partial \boldsymbol{B}_c^{-1}} = \alpha\alpha^T - (\boldsymbol{K}_c + \boldsymbol{B}_c^{-1})^{-1}$, and the matrix $\frac{\partial \boldsymbol{B}_c^{-1}}{\partial \phi_{n,i}}$ will be zero everywhere, except in the diagonal element (n, n) where it will be $\frac{-\sigma^2}{\phi_{n,i}^2}$.

Using the chain rule, we can calculate log-likelihood gradients of the model hyperparameters for any covariance function, since we know $\frac{\partial \boldsymbol{K}_c}{\partial \theta}$, resulting in a very general and modular framework. We only need $\frac{\partial \mathcal{L}_{KL}}{\partial K_c} = \frac{\partial \mathcal{L}_c^M}{\partial B_c^{-1}} = \alpha_c \alpha_c^T - (K_c + B_c^{-1})^{-1}$. In the case of the model variance σ^2 we have $\frac{\partial \mathcal{L}^M}{\partial \sigma^2} = \sum_k \frac{1}{2} \operatorname{Tr}\left(\frac{\partial \mathcal{L}_c^M}{\partial B_c^{-1}} \cdot \frac{\partial B_c^{-1}}{\partial \sigma^2} \right)$ where $\frac{\partial B_c^{-1}}{\partial \sigma^2}$ will be a diagonal matrix with $\frac{1}{\phi_{i,c}}$ on element (i, i) for all i.

4 Downstream analysis

4.1 Ranking genes by bifurcation

Once the OMGP model has been fit, it can be used to investigate individual genes in terms of their bifurcating trajectory.

The log-likelihood of the OMGP model depends on the covariance matrices $\mathbf{K}_t = \{\boldsymbol{K}_t^c, c = 1, \ldots, C\}$, the variational mixture parameter matrix ϕ, and the N observations (t, X). Let us assume that we have mixture parameters ϕ_b which have been found to distinguish a bifurcating trend based on some X response variables. We can now keep the fitted parameters and evaluate the marginal likelihood of a model where the response variables X are replaced by gene expression values y_g. We call this new model $\mathcal{H}_{\text{bifurcating}}$. We wish to find genes which fit this bifurcating model better than a model where there is no bifurcation. To this end, we make a third model $\mathcal{H}_{\text{not bifurcating}}$ identical to the previous one, except we replace

ϕ_b with ambiguous assignments ϕ_a. To asses whether a given gene g is better described by the *bifurcating* or the *not bifurcating* model, we evaluate the Bayes factor:

$$BF_g = \log p(y_g|\mathcal{H}_{\text{bifurcating}}) - \log p(y_g|\mathcal{H}_{\text{not bifurcating}}).$$

We refer to this ratio as the *bifurcation statistic*.

To estimate p-values, we used a permutation approach where we perform the same analysis for every gene g, except with permuted t values to estimate a null distribution.

As a proxy for *effect size* of bifurcation, we consider how well the expression values of a gene correlate with the trend assignments to a latent function. Strong positive correlation will mean the gene is particularly up-regulated in the cells unambiguously belonging to the trend. Conversely, a strong negative correlation indicates the gene is down-regulated in the strongly assigned cells compared to all cells.

4.2 Inferring the bifurcation time point

It is possible to qualitatively appreciate from the GP assignment probability (ϕ_c) for each trajectory ($f_c(t)$) of the OMGP model, which cells are *ambiguous* and which cells are *exclusive* to individual GP's. In the case of two trends, ambiguous cells have assignment probability (ϕ) close to 0.5. A model where the data can be described by two trends, but not by one, will have a higher likelihood. Similarly, if only a *region* of the ϕ parameters over time are replaced by ambiguous cell assignment values, the new model will have a lower likelihood.

For the sake of clarity, we make the assumption that the OMGP will begin as ambiguous, and then become less ambiguous over time, splitting into two trends, in this special case. To investigate these cases, we pick a time-point t_b in an OMGP, then replace all ϕ values prior to t_b with 0.5. We define this new ϕ as $\phi_{>t_b}$:

$$\begin{aligned} [\phi_{>t_b}]_{i,c} &= 0.5 & t_i < t_b \\ [\phi_{>t_b}]_{i,c} &= \phi_{i,k} & t_i \geq t_b. \end{aligned}$$

Now we can evaluate the model likelihood for this particular t_b and define

$$\mathcal{L}_{t_b} = \mathcal{L}_{KL}(\phi_{>t_b}, \mathbf{K}_t, \sigma^2 | X, T).$$

This procedure is repeated for multiple ts over the predictor variable of the OMGP model. In our implementation, we consider 30 evenly spaced bins by default, which has given enough resolution for the data investigated (though the number of bins can easily be changed).

The likelihood has to decrease by definition. However, after the bifurcation the decrease is much more pronounced. We use a break-point heuristic to detect this elbow, which is indicative of the bifurcation time.

To identify the region at which the likelihood decreases more rapidly, we fit a piece-wise linear curve to the log-likelihoods, defined by

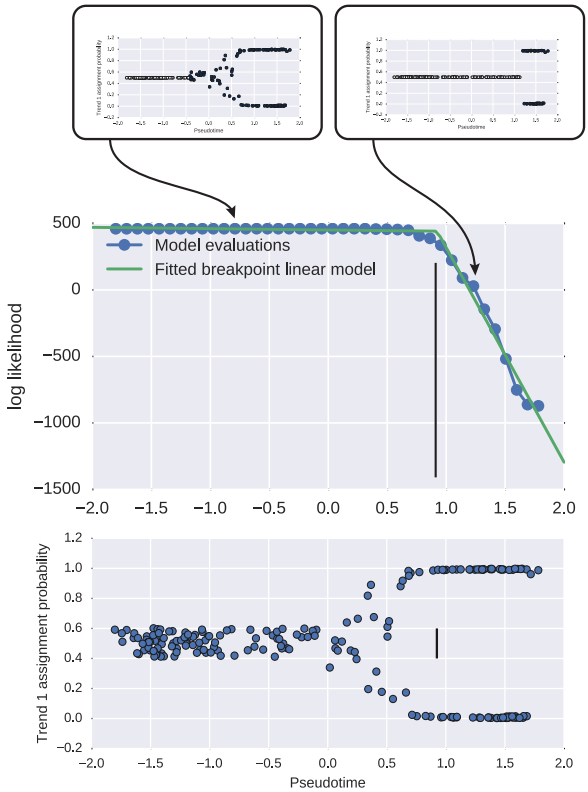
$$\begin{aligned} \mathcal{L}_{t_b} &= k_1 \cdot t + c_1 & t < t_b \\ \mathcal{L}_{t_b} &= k_2 \cdot t + (k_1 - k_2) \cdot p + c_1 & t \geq t_b \end{aligned}$$

This curve consists of two linear pieces, broken up at the point p. When the curve is fitted, we consider the break-point p to be the point after which we can be confident that a bifurcation has occurred, see Supp. Comp. Fig 3.

5 Limitations

The models we have described here are implemented in the aim of describing a cellular trajectory and decomposing parallel trends. Using the Gaussian Process framework we can ask the question we want while at the same time considering information we know about our experimental data, such as collection time.

We have found this strategy powerful, but it is not perfect, and there are man avenues for potential future improvement.



Sup. Comp. Fig. 3: Inferring bifurcation point. The plot illustrates how different points along the pseudotime are sampled. Ambiguous assignment probabilities replace trained assignment probabilities in the observations earlier than the sampled points. The breakpoint model identifies the points where a decrease in likelihood differences becomes more extreme.

A more biologically appropriate view of the problem of cellular bifurcation would potentially be a tree-based model, with an explicit branch time encoded in the model. The computational gain from the mixture model used here should not be overlooked though, an explicit branch point would require either sampling or global optimization strategies for inference which is much slower than gradient based optimization.

Throughout we are using regular Gaussian likelihoods for our models, though we empirically know this is not completely appropriate. In particular due to the so called "drop-out" problem in scRNA-seq measurements, where there is an inflation of zero-values at low to intermediate expression levels (*55*).

The likelihood is also limited in that we assume expression values in the $\log TPM + 1$ unit have Gaussian noise in the absence of dropouts. While this seem to empirically hold true, perhaps a count based likelihood such as Poisson would be more appropriate for scRNA-seq data?

With regards to the full method, we should point out that here we are doing two steps of inference. First we infer pseudotime, with uncertainty. Then we treat the pseudotime as known truth in the mixture model. This could be improved by either propagating the uncertainty in pseudotime in to the mixture model, or by setting up a joint model which learns pseudotime and mixture simultaneously. (An "Overlapping Mixture of GPLVMs" if you will).

When performing pseudotime inference, in this work we are using priors on the variational parameters rather than the latent parameters the variational distributions are estimating. A way to improve the model would be to implement the priors on the latent variables, and learn the variational posterior given these priors.

6 Implementation and combination with existing workflows

6.1 Practical use of GPfates

The basis principle of GPfates is the combination of pseudotime and mixture modelling.

Input to the GPLVM is an expression table consisting of log scaled relative abundance values Transcripts Per Million, TPM, with a value of 1 added to handle cases where expression is 0. As relative abundance follow a log-normal distribution, the Gaussian likelihood used for Gaussian Process regression should be appropriate.

In practice, the pseudotime should represent the biological process of interest. If this process is clear, the expression data should be usable without pre-processing. In single cell time course experiments where the process of interest is less immediate, a strategy highlighted in (*19*) is to select the gene set used could be to rank the genes by an ANOVA test over the time points, and select a larger number of significant genes.

Similarly, the low-dimensional representation of the transcriptomic cell state should represent the biological response of interest. It can be beneficial to select the parts of the representation which correspond to this. For example, in the analysis of CD4+ T cell time course, we use the second GPLVM latent variable as a representation of T cell response, and model this factor by the OMGP.

While pseudotime can be inferred directly from the expression matrix Y, in many cases it helps interpretation to perform an intermediate step of dimensionality reduction. This process could also be beneficial if the data has a very complex structure.

Another practical consideration is that single cell expression values can be quite noisy. This limits the time-scale at which we can expect to measure proper functional differences in expression levels. Due to this, we tend to put lower limits on the lengthscale l_{SE} of the squared exponential covariance function.

6.2 Integration of existing methods

We have presented use of the GPfates method when pseudotime or low-dimensional representations have been based on the GPLVM. This is because the OMGP follows from this framework, and the statistical assumptions are consistent between the models.

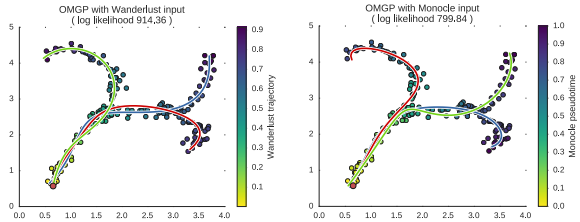
In practice, other methods for inferring pseudotime or low-dimensional representations could also be considered. Here we briefly outline possible strategies for applications of GPfates downstream of popular single-cell analysis methods.

Recall that to perform the GPfates inference, we need pseudotime t and some representation of transcriptomic state X. These variables can be set as the output from other methods.

In Monocle (*19*), the low-dimensional representation X is found by independent component analysis, and the pseudotime t for each cell is defined by the path distance to a starting cell through a minimum spanning tree in the coordinates of X.

In Wanderlust (*36*), a heuristic is used to build a stable k Nearest Neighbor (kNN) graph of the data in the high-dimensional space of protein measurements. The pseudotime t for a cell is then defined as the average shortest path from a known starting cell through the kNN graph. Note that for CyTOF data, which Wanderlust is designed for, only up to 40 analytes can be measured at once, so it could be feasible to take X to be the original expression matrix (Y in our notation).

Another dimensionality reduction technique which have been used for single cell RNA seq data is Diffusion Maps (*56*). Here X is a spectral embedding of the data manifold, based on the Laplace operator. It has been pointed out that these embeddings preserve branching structure in the data. Taking the pseudotime t as the Diffusion Pseudotime (*21*), a surprisingly effec-



Sup. Comp. Fig. 4: OMGP is compatible with e.g. Wanderlust and Monocle, as demonstrated with a toy data set

tive approximation of geodesic distance over the data manifold (from a known starting cell), based on the diffusion map, GPfates modelling could be used downstream to quantify the branching structure of the data.

We list alternative compatible pseudotime methods in table 1.

As a demonstration, we generated a toy data set with three branches, and extracted the pseudotime using both the Monocle method and the Wanderlust method. Then fitted and OMGP with $C = 3$ on the output. The results can be seen in Comp. Supp. Fig 4, which illustrates the correct identification of the branching processes for either input.

6.3 Software availability

We have made a software package for using the GPfates method, which is available at https://github.com/Teichlab/GPfates. It provides guidance and sensible defaults for the kind of analysis we have described here. It makes extensive use of the `GPy`[1] package, and the `GPclust`[2] package, where we implemented the non-parametric OMGP model.

[1] https://github.com/SheffieldML/GPy
[2] https://github.com/SheffieldML/GPclust

7 Assessment of GPfates on simulated and real data

7.1 Sample-size robustness analysis

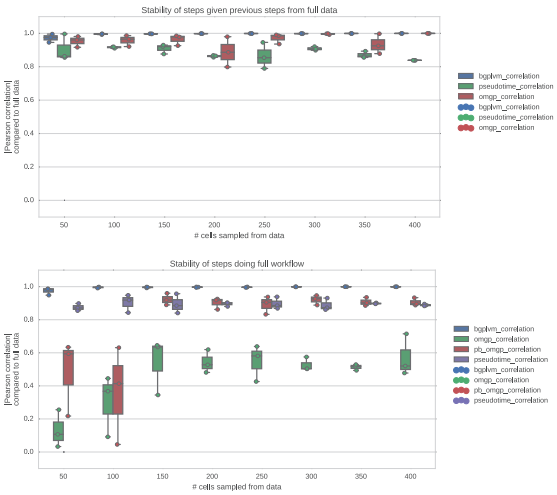
Our full analysis consists of several independent consecutive steps: first the GPfates method where we are i) finding a low-dimensional representation, ii) smoothing the data over a pseudotime, and iii) finding a trend mixture model. After this we perform downstream analysis where we are iv) dentifying the end states and bifurcation.

How much data do we need to accurately reconstruct trends from all four of the above steps, and how much data is needed for individual steps? We investigated both how stable the full procedure is, as well as the individual steps, by re-running it on sub-sampled datasets with fewer cells than the entire dataset.

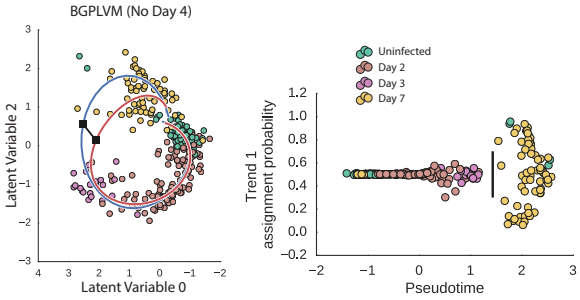
To measure the stability of the methods, we consider absolute Pearson correlation of the parameters inferred for sub-sampled data relative to the results obtained from performing the analysis on the full data set.

We found that recovering a low-dimensional representation is extremely stable with respect to the number of cells (Supp. Comp. Fig. 5), with almost perfect correlations between analysis of the sub-sampled data and the original GPLVM values. (For example, the lowest absolute Pearson correlation for a run with 50 cells was 0.96). Similarly, the pseudotime inference is also very stable to sub-sampling.

Finding the entire OMGP mixture model over pseudotime requires a larger number of cells. We do not see any higher degrees of consistency until we reach 150 sub-sampled cells, with correlations around 0.5. It is rare to see single cell studies with so few cells, and in the study accompanying this text we had many more cells (408). Identifying only the end states is rather robust (but in many cases might be best analyzed as a cluster problem



Sup. Comp. Fig. 5: Robustness of analysis steps by subsampling. Parameters inferred from a subsample of the data are compared to parameters inferred using the full data. The top panel indicates this analysis for independent steps assuming the previous step is known. The lower panel shows the result when running the workflow from start to end.



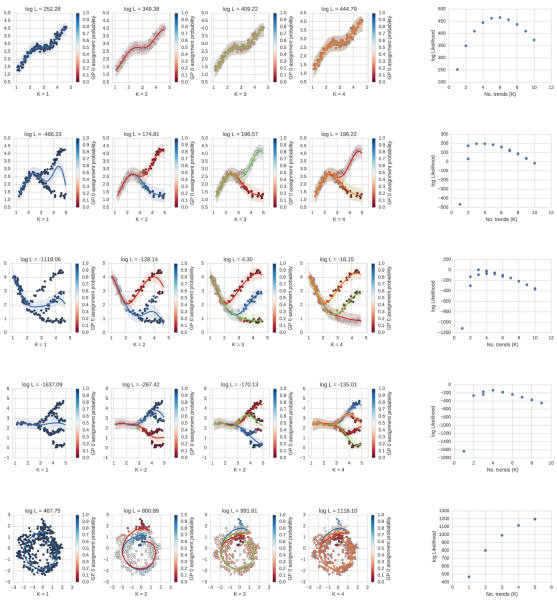
Sup. Comp. Fig. 6: Complete reanalysis of our T-cell data excluding cells cellected at day 4. The bifurcation point is identified as being between Day 3 and Day 7, and is not forced in to either of the days.

rather than a continuous value problem), where we start seeing a correlation of 0.9 at 150 cells.

The individual steps were in general very stable to sub-sampling, relative to the "gold standard" of using the full data set. When running the entire procedure, we see that smaller errors early on in the analysis will propagate and affect later steps.

7.2 Predicted bifurcation time is not biased by collection times

We consider the risk that the identified bifurcation point in the CD4+ T cell data potentially just reflects the time points at which we have collected data. We test the robustness of the prediction of the bifurcation as having happened at Day 4 by re-running the analysis after removing cells collected at Day 4. In this analysis we find that the bifurcation happens at some point between Day 3 and Day 7 where we do not have any observed cells. The alternate hypothesis would have been that the bifurcation would be found in either Day 3 or Day 7. This provides confidence both in the bifurcation point identification, and more generally in the meaningfulness of the low-dimensional GPLVM representation of the data (Supp. Comp. Fig. 6).



Sup. Comp. Fig. 7: Attempts detecting number of trends with OMGP. Simulated data with expected numbers of trends where fitted with OMGP, where the C cutoff was set to a range of values.

7.3 Assessment of the ability to select the number of trends in OMGP

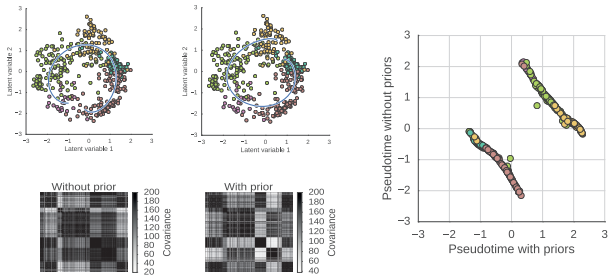
In principle, the marginal log likelihood of the OMGP model should let us select the C number of trends which optimally explain the data. We investigated this by generating four synthetic data sets with between 1 and 4 underlying trends. For each of the data sets, we optimized OMGP models with the number of trends C varying from 1 to 9 (three times per C value). We found that the marginal likelihood of the models corresponded to the correct number of trends in the cases of 3 and 4 ground truth trends, but not for the 1-trend and 2-trend synthetic data. For 1 trend, the likelihood was lowest for a larger number of trends, and for 2 trends, the likelihood was very similar for 2 and 3 trends. This suggests that the OMGP may have a tendency to overestimate the number of trends if there is a single progression. Supp. Comp. Fig. 7.

For our CD4+ T cell data, we found that the marginal likelihood continuously increased with the number C. We elected to keep the model simple and made the assumption that we could sufficiently explain the data with $C = 2$.

It is possible that the optimal likelihood for K is not well defined when we have trends branching off from a common trend. In the original application of the OMGP model, the assumption is that the trends will be completely independent of each other. As we are already to some extent failing to fit two models in the ambiguous case, this might cause the likelihood to reflect a poor fit. For quantitatively determining the number of trends in the data, further work is needed, probably with a model which explicitly considers branching from a common original trend. The marginal likelihood of the model is an indication, but the choice of C should also reflect the biological system under consideration.

7.4 Comparison of pseudotime inference with and without priors

For the 1-dimensional Bayesian GPLVM which we use to find the pseudotime of the data, we put priors on the cells based on their known time points. This is not strictly necessary, but helps to enhance interpretability as there is intrinsic invariance to the inferred values. If we do not use priors, qualitatively, the same trajectory is identified. Additionally, comparing the two versions of pseudotime against each other, we see that they correspond to a circular shift relative to each other. The covariance matrices inferred using either strategy have a very similar block structure (Frobenius norm ... of difference) indicating that neighbor relations are consistent. Supp. Comp. Fig. 8.



Sup. Comp. Fig. 8: Comparison of pseudotime with and without per-cell priors. The upper left shows the fit of the pseudotime predicted in to the 2D GPLVM with and without priors. Below are the corresponding inferred covariance matrices. The right plot shows the relations between the two versions of pseudotime, clearly indicating that they have an approximate one-to-one mapping.

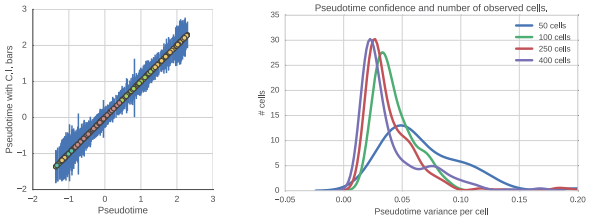
7.5 Pseudotime uncertainty

As pointed out in (*57*), we can use the posterior distribution of pseudotime from the Bayesian GPLVM to assess how meaningful the ordering is. By investigating the confidence intervals of the pseudotime for each cell compared to neighboring cells, we see that the ordering is quite meaningful (few cells overlap in confidence interval). (Supp. Comp. Fig. 9)

We also investigated how the confidence of the pseudotime depends on the number of cells observed. As the number of observed cells increases, the distribution of variance per cell decreases towards zero. (Supp. Comp. Fig. 9)

7.6 Stability of the circular shape of the GPLVM representation

We wanted to rule out the possibility that the latent variable representations of data which appear circular might be artifacts due to random noise, as suggested by (*58*). To make sure this was not the case for our CD4+ T cell data, we removed two 'slices' of cells from the circular

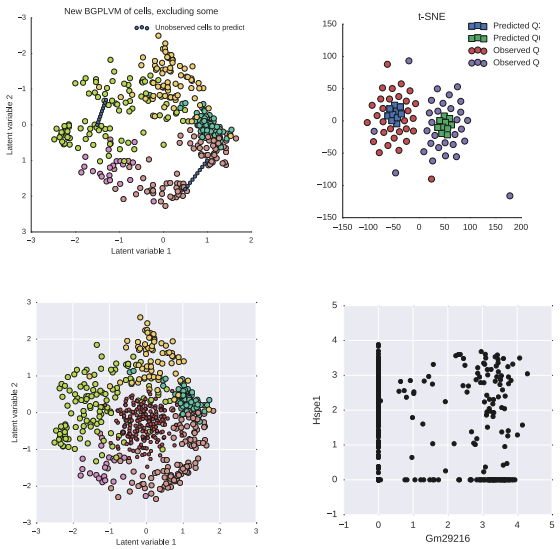


Sup. Comp. Fig. 9: Investigation of uncertainty of inferred pseudotimes. Left panel, since the Bayesian GPLVM fits the variance of the pseudotime for each cell, we can compare the assignments with each other. The bars correspond to 95% confidence intervals. On the right panel we see how the lengths of the confidence intervals globally decrease as the number of cells used increases.

2D GPLVM pattern. Following this, we fitted a new GPLVM with this reduced data set. After optimizing the GPLVM, a representation was found which was again missing the same slices, Supp. Comp. Fig. 10A. This control experiment strongly suggests that the GPLVM learns the actual topology of the data.

7.7 Assessing the accuracy of imputing virtual cells

Unlike many other dimensionality reduction techniques, the GPLVM creates a model which maps into the high dimensional observed space. It is, however, not clear how meaningful this representation is. We assessed this by taking the "sliceless" model described above, and in the empty areas corresponding to the removed cells, predicting "virtual cells" (Supp. Comp. Fig 10A). Using an independent clustering technique, t-SNE (*59*), on both the left out slices of cells and the predicted virtual cells, we find that single cell transcriptomes predicted from a given slice coincide with the real cells from the corresponding slice (Supp. Comp. Fig. 10). This indicates that GPLVM prediction in to high-dimensional spaces is not simply producing overfitted results.



Sup. Comp. Fig. 10: Stability of GPLVM representation, and prediction through GPLVM. Top row: Predicting cells from regions of higher similarity with left out real cells from corresponding regions than non-corresponding regions. Bottom row: Predicting cells from unobserved regions potentially identifies antagonizing gene combinations.

Following from this, we investigated the "hole" in our CD4+ T cell data. We create a number of virtual cells from the hole region and compare which genes would be expressed in these cells compared to genes expressed in all cells (Supp. Comp. Fig. 10C). The underlying reasons for data being non-linear is that particular combinations of gene expression patterns do not occur together. If we find genes which are highly expressed in the virtual cells but are not observed at the same time in actual cells, this might indicate that they are incompatible with each other. This might be a good complementary tool for generating hypotheses about regulation. For instance, we identified the genes Hspe1 and Gm29216 which would be co-expressed in the hole, but are generally not co-expressed in observed cells (Supp. Comp. Fig. 10D).

8 Validating the BGPLVM and OMGP approach by application to other data sets

In order to further corroborate our analysis approach, we considered two recently published single cell data sets produced to investigate progression of single cells in two developmental contexts: mouse fetal lung and human fetal primordial germ cells. In addition to that, we analyzed a third RNA-seq study that examines the development of frog embryos in a high temporal resolution.

8.1 Analysis of lung development data

We downloaded the data from (*23*) and quantified the expression using Salmon. To smooth the data over pseudotime, we found genes that vary over the *a priori* known time points by a likelihood ratio test of an ANOVA model of the time

points. The expression values for the top varying genes were run on a GPLVM. One of the factors of the optimized GPLVM was used as pseudotime, and the top two factors of the GPLVM were used to represent the entire data set. An OMGP was then optimized on this low-dimensional representation to identify the two trends corresponding to the AT1 and AT2 lung cell lineages without prior annotation. The bifurcation statistic for all expressed genes in these cells reconstituted many of the genes identified in a largely manual manner by (*23*).

8.2 Analysis of human primordial germ cell data

The data from (*24*) was downloaded and quantified with Salmon as with the other data, but with an index based on the human transcriptome: Ensembl 78 annotationa of GRCh38, together with ERCC sequences and human specific repeats from RepBase. To smooth the time course data, we used a likelihood ratio test to find the top genes which were described linearly along the time points in the data. The expression of these genes were then used to fit a GPLVM. This low-dimensional representation of the data was then used to fit an OMGP, taking one of the latent factors as pseudotime.

In this data set, the ground truth about the sex of the cells is known, and thus we could have used a supervised approach such as `GPTwoSample` (*60*) or `DETime` (*61*). Interestingly, the OMGP model identifies the split between male and female cells in an unsupervised fashion.

We applied the bifurcation statistic test to identify genes that follow the male and female development differently.

Unlike in the case of the lung development data, the majority of the genes we identify are not discussed in the original study. In the original study, the authors focused on genes specific to given conditions (e.g. Male PGC's from week 11 compared to all other cells). In our analysis, we consider the dynamics of gene expression over development. We find that in the male branch, the GAGE family is highly upregulated over development. Additionally we find the Y-linked gene ENSG00000279950. Also among the top male hits is RHOXF2, a gene linked to male reproduction (*62*). Further down the list we also interestingly find PIWIL4, a gene with function in development and maintenance of germline stem cells (*63*). On the female side, the top hit is MDK, a gene involved with fetal adrenal gland development (by similariry: UniProtKB P21741). Other top hits include MEIOB, a meiosis related gene, and the satellite repeat GSATII. Surprisingly, we also see upregulation of SPATA22, a gene associated with spermatogenesis.

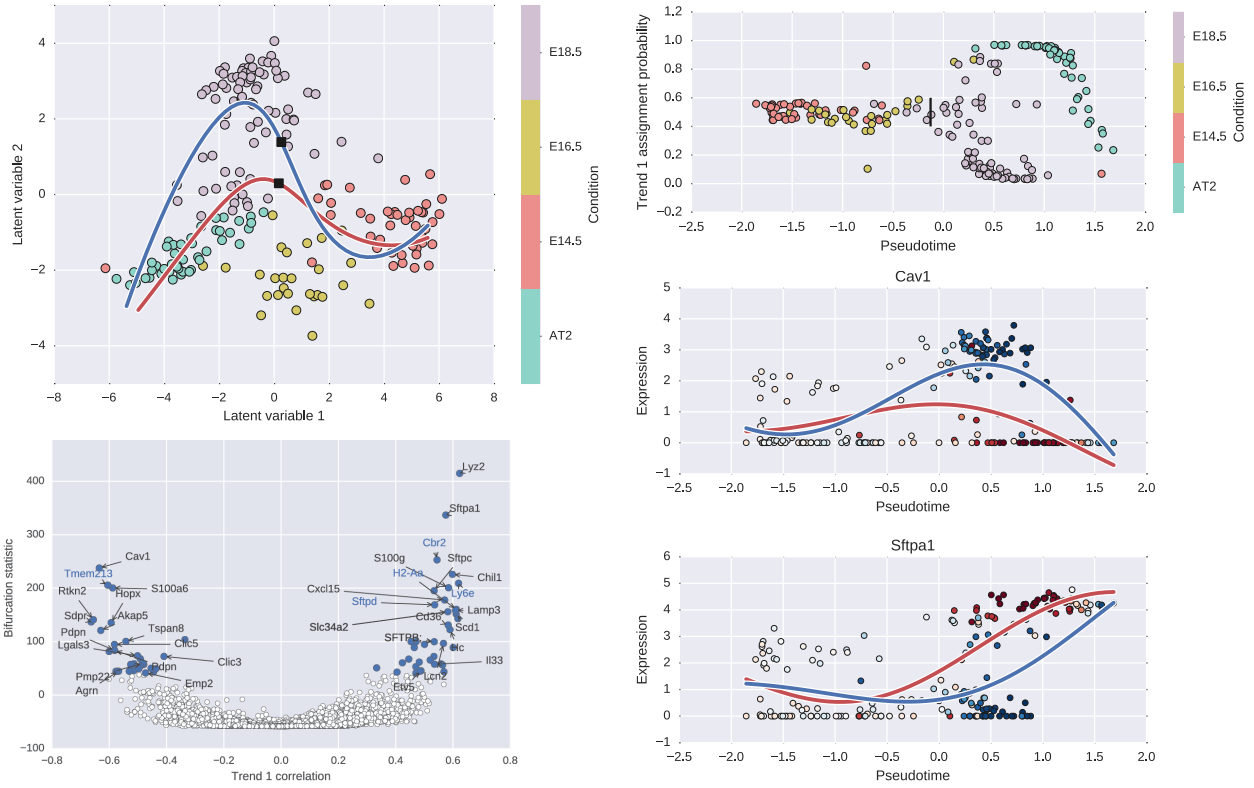
8.3 Analysis of frog development data

As a form of negative control, we subjected the data from (*64*) to our bifurcating analysis. This is data from a high resolution time course of developing frog embryos. RNA-sequencing is done using material from the entire embryo, so no biological bifurcation should occur during the development. It should be noted that frog embryos have much more RNA than single cells, so the data is less noisy than single-cell RNA-seq samples.

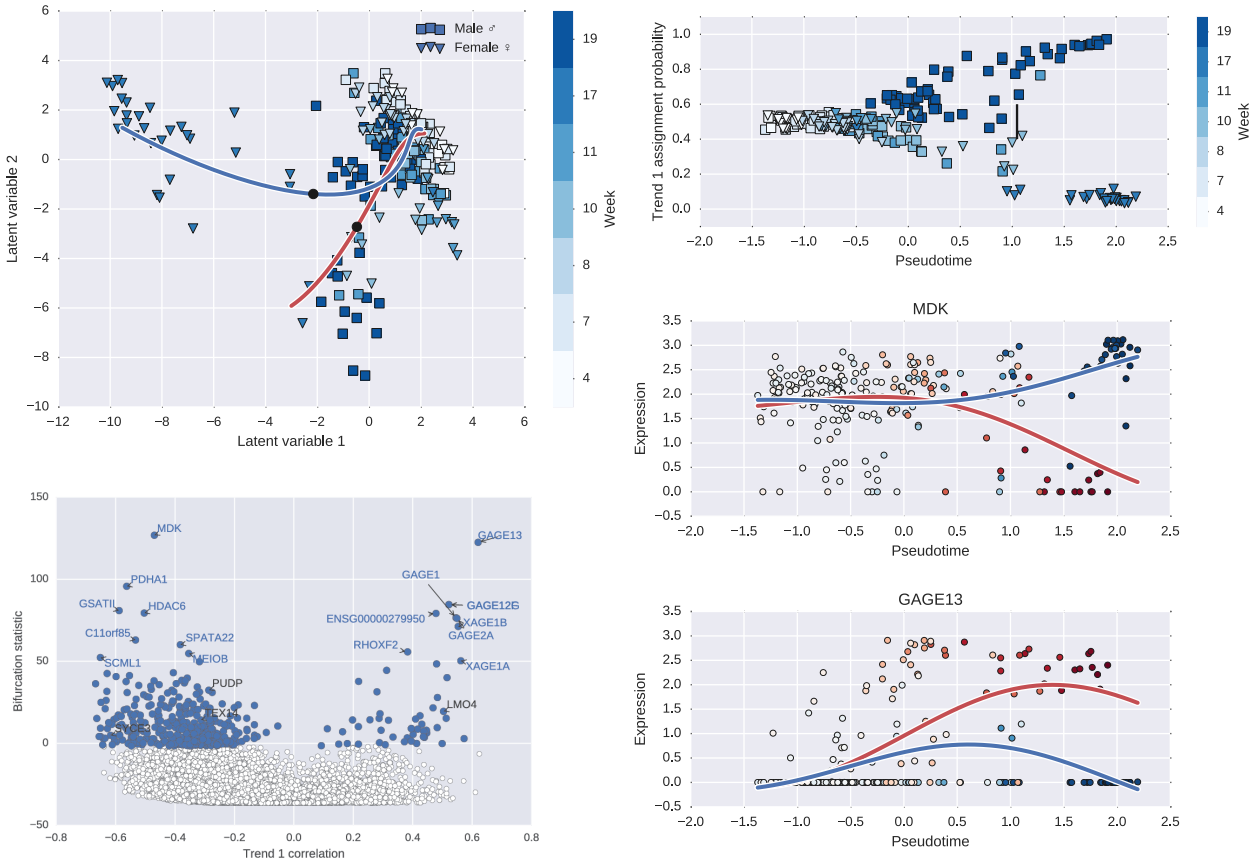
Expression was quantified with Salmon, and the expression table was read in with ERCC's removed. All expressed genes were used to find a low-dimensional representation, and a pseudotime was found using 8,000 genes which were linearly differentially expressed over the time course.

The pseudotime corresponded extremely well with the real time point ordering. And in this data set, almost all expression variability can be explained by the time course. Still we trained the OMGP model on the 2D Bayesian GPLVM representation of the data.

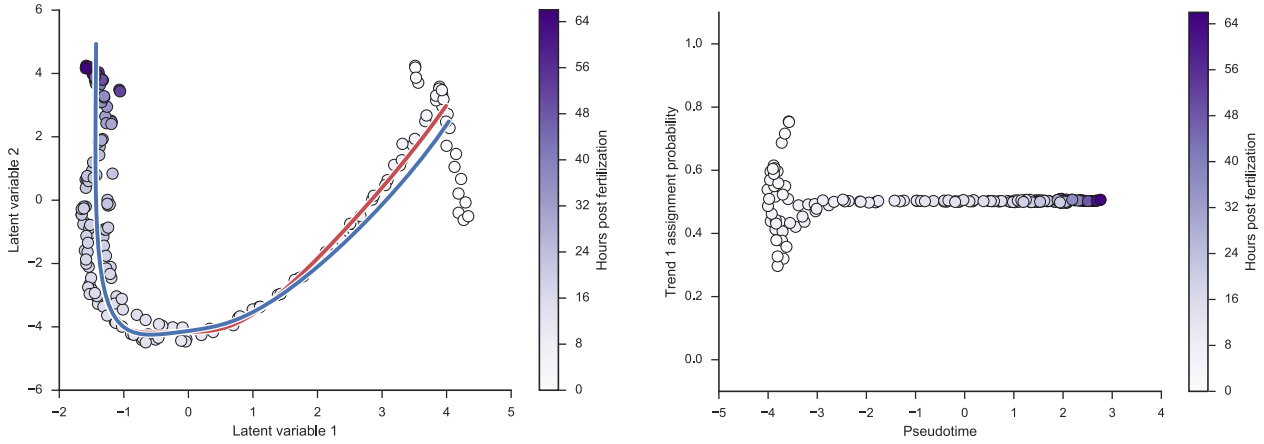
No strong bifurcation is detected, and thus we skipped gene bifurcation analysis. The single-trend model explains the data well. Some heterogeneity can be seen in the early part of the



Sup. Comp. Fig. 11: Summary of GPfates result of Treutlein et al developing lung data.



Sup. Comp. Fig. 12: Summary of GPfates result of Guo et al developing primordial germ cell data.



Sup. Comp. Fig. 13: Summary of GPfates result of Owens et al developing frog embryo data.

time course. This might suggest that expression is somewhat noisy in extremely early embryos, but not in a way that indicates discrete cell populations.

9 Comparison to other pseudotime and bifurcation methods

We compared GPfates with various methods inferring pseudotime and bifurcation events: Wishbone (*22*), Monocle2 [3], Diffusion Pseudotime (*21*), SCUBA (*18*) and Mpath (*20*). We applied the methods to our data and different public developmental data sets mentioned earlier. Please note, the developmental embryonic frog data was treated as a negative control to investigate if methods are able to detect false positives, i.e. identifying branches when they do not exit.

The results are summarized in Supp. Comp. Fig. 13 through 13. In order to validate the approaches, we counted the number of bifurcation events for each method in each data set and compared it to the expected number of bifurcations (Table 2).

Furthermore, we assessed the accuracy of a method by calculating Spearman's rank correlation between the inferred pseudotime and the real time in a given data set (Supp. Comp. Fig 14). For this analysis only Wishbone, Monocle2 and DPT could be considered as the SCUBA and Mpath tools do not report an inferred pseudotime which can be parsed.

Monocle2 and GPLVM perform similarly with a high accuracy (> abs(0.80) Spearman's correlation) on public data. However, Monocle2 as well as Wishbone and DPT failed to assign the correct temporal order with regard to the malaria infection data. Overall, applying Wishbone and DPT to the data sets they achieved poor to moderate accuracy, except for Wishbone performing well on the frog data, and DPT performing well on developmental lung data.

Concerning the bifurcation event in developing lung data, most of the methods cluster AT2 and E18.5 cells into one branch which has been confirmed in a previous study (*23*). However, in primordial germ cell data none of the published methods were able to detect branching events between male and female cells. With regard to the frog embryonic development study only SCUBA reflects the non-branching structure of the data. All other public methods report a branching point.

9.1 Preprocessing public RNA-seq data

We removed ERCCs from our expression data table and re-scaled the expression values to TPM. Furthermore, we eliminate cells containing NA's in the frog data.

Some of the used methods require a start or root cell. Therefore, We randomly picked a cell from an early collection time point: `1771-026-187-E6` (malaria), `SRP033209_E14.5_rep_1_cell_24` (lung), `2013600` (pgc) and `1795679` (frog).

9.1.1 Wishbone

The analysis with Wishbone version 0.4.1 was performed according to the tutorial using default or suggested parameters (*22*). We ran t-SNE with `n_components = 5` and `perplexity = 30`. To run wishbone the start cells were chosen as stated above with `k=15` or `k=50` for frog data, `components_list=[1,2]` and `num_waypoints = 150`.

9.1.2 Monocle

The Monocle analysis was performed with version 2.1.0 of the Monocle package, following the steps outlined in the original vignette (*19*). In brief, the analysis was performed using the size normalized

[3] http://cole-trapnell-lab.github.io/monocle-release/articles/v2.0.0/

data (TPM) including all genes expressed in $\geq$ 50 cells with default parameters. The genes used for the ordering of cells were defined by carrying out a differential expression analysis of the time points using the `differentialGeneTest` in the `Monocle` package. Following the original vignette, genes with q-value < 0.01 were selected. To reduce the dimension the `max_components` option was set to 2 and the `DDRTree` methods was used.

9.1.3 Diffusion Pseudotime (DPT)

DPT analysis was done using the R package version `0.6.0` and an additional package called `destiny` (version `1.3.4`) (*21*). In order to calculate the transition matrix DPT uses a Gaussian kernel with parameter `sigma`. The optimal `sigma` was chosen by using the function `find.sigmas()` of the `destiny` package. Given the transition matrix and root cell `dpt()` was executed with `branching=TRUE`.

9.1.4 SCUBA

In order to run SCUBA we used the python package `PySCUBA` version `0.1.1`[4] which provides a graphical user interface (*18*). Selecting the RNA-seq data set including temporal information we ran SCUBA with `cluster_mode = PCA2` and `pseudotime_mode = 0`.

9.1.5 Mpath

We performed analyses with Mpath using the package version `1.0` (*20*). Prior to the analysis, a quality check includes a removal of genes having TPM values < 1 in more than 95 percent of cells in each group. In order to find the number of optimal clusters the parameters `diversity_cut` and `size_cut` were set as suggested to 0.6 and 0.05, respectively, when calling the function `landmark_designation()`. Inspecting the resulting plots, the number of optimal clusters were chosen as 10 (malaria), 19 (lung) and 24 (pgc). Mpath failed to run on the frog data set. Using the landmark clusters we constructed the weighted neighborhood graph and trimmed it using the minimal spanning tree method.

[4]https://github.com/GGiecold/PySCUBA

10 Discussion

We have demonstrated the applicability of our GPfates method, where we use latent variable modeling to infer temporal expression dynamics, and Gaussian process mixture modeling for identifying diverging global trends. The method has been investigated in terms of robustness, and applied on several simulated and real data sets showing good results.

Of course there is no silver bullet for these sorts of problems, and it would not be surprising if other methods than the ones we have used work better for some biological systems. Nevertheless, we have illustrated that the main component, the Gaussian process mixture modeling, is compatible with other methods in these cases.

A benefit from the methods we use is that diagnostics such as marginal likelihood can be used to aide the user with regards to the models to use. Still, the user will need to keep the biological system in mind, and be critical of results.

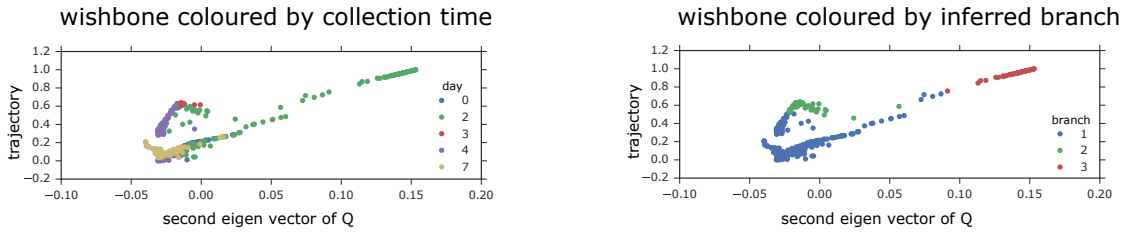
Pseudotime Method	**Strategy**	**OMGP Compatibility**
Wishbone (*22*)	Diffusion maps on reduced k-NN graph (using waypoints)	Yes
Monocle2 Pseudotime (*19*)	Minimum Spanning Tree path length in 2D DDRTree space	Yes
Diffusion maps (*56*)	Spectral embedding of data manifold	With postprocessing, e.g. DPT (*21*)
Wanderlust (*36*)	Heuristic k-NN graph geodesic distance	Yes
GPLVM	Latent data parametrization	Yes

Bifurcation Method	**Strategy**	**Dim. Reduction**	**Clus-tering**	**Diff. Expr. Analysis**
Wishbone (*22*)	Disagreements between shortest paths	Yes	No	No
Monocle2 states (*19*)	Create k PQ trees from a Minimum Spanning Tree	Yes	No	Yes
DPT (*21*)	Switch in correlation behavior	No	No	No
SCUBA (*18*)	Transitions between clusters in pseudo-time bins	Yes	Yes	No
Mpath (*20*)	Finding Minimum Spanning Tree in neighborhood graph of landmarks	Yes	Yes	Yes
OMGP	Model data as mixture of continuous processes	Yes	No	No

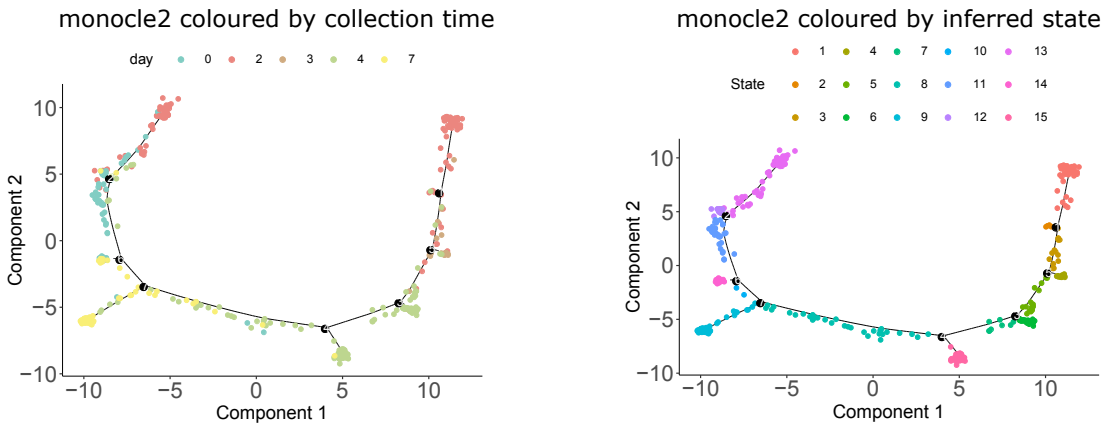
Table 1: Examples of common pseudotime- and bifurcation methods.

Application on malaria data

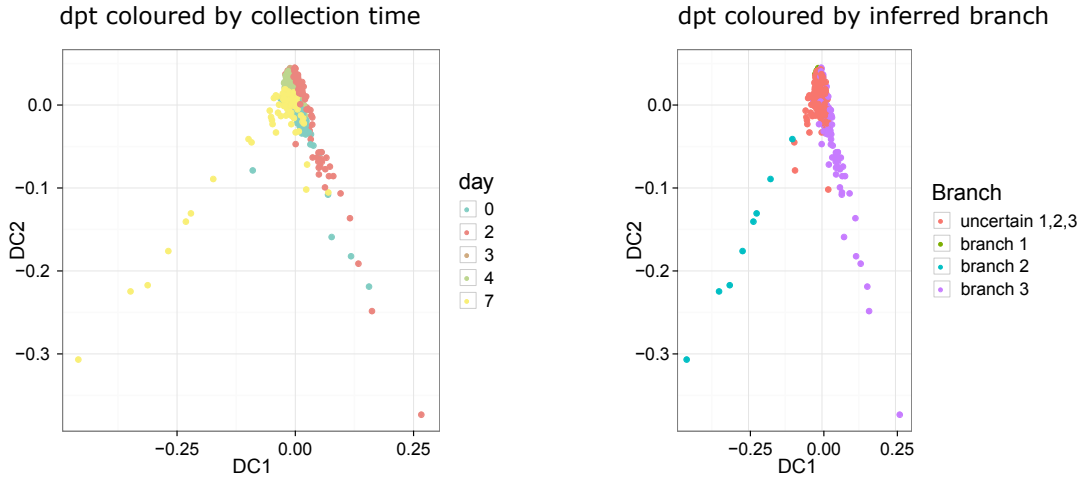
A



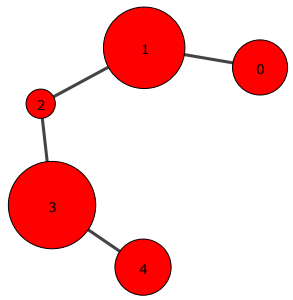
B



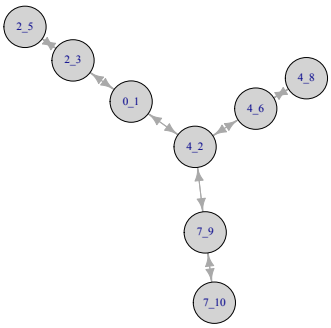
C



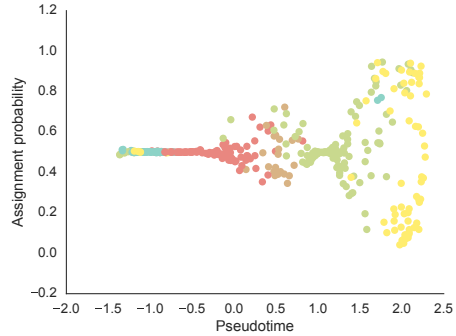
D



E



F



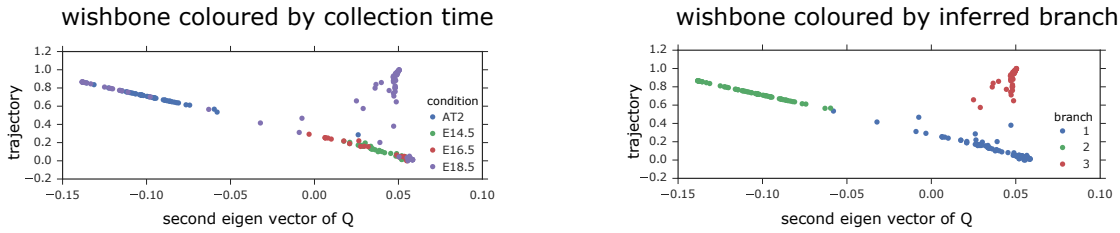
Sup. Comp. Fig. 13: Output of bifurcation methods applied to malaria data. **(A)** Wishbone results showing the branching structure colored by time points (left) and inferred branches (right). **(B)** Minimum spanning tree on cells generated by Monocle2. Cells are colored by time points (left) and inferred cell states (right). **(C)** Visualisation of diffusion maps in DPT colored by time points (left) and inferred branches (right). **(D)** Lineage tree by SCUBA reports no bifurcation. Sizes of bubbles are according to number of cells. **(E)** MPath's minimum spanning tree: First number corresponds to the collection time, second number corresponds to the landmark cluster. **(F)** GPfates trajectory, colored by time points.

	Malaria **this work**	**Lung** (*23*)	**PGC** (*24*)	**Frog** (*64*)
Wishbone	1 (1)	1 (1)	1 (1)	1 (0)
Monocle2	7 (1)	1 (1)	2 (1)	1 (0)
DPT	1 (1)	0 (1)	1 (1)	1 (0)
SCUBA	0 (1)	0 (1)	0 (1)	0 (0)
Mpath	1 (1)	2 (1)	1 (1)	NA (0)

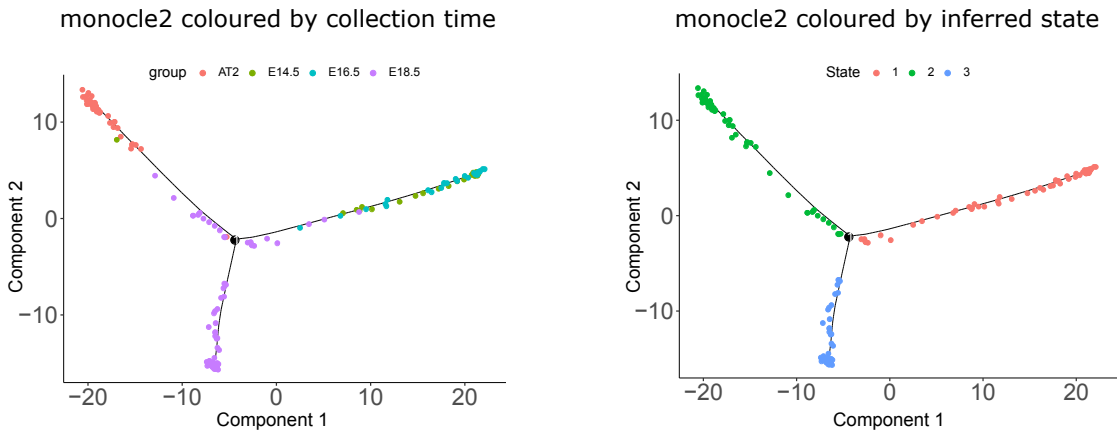
Table 2: Number of detected (and expected) bifurcations of other methods.

Application on lung development

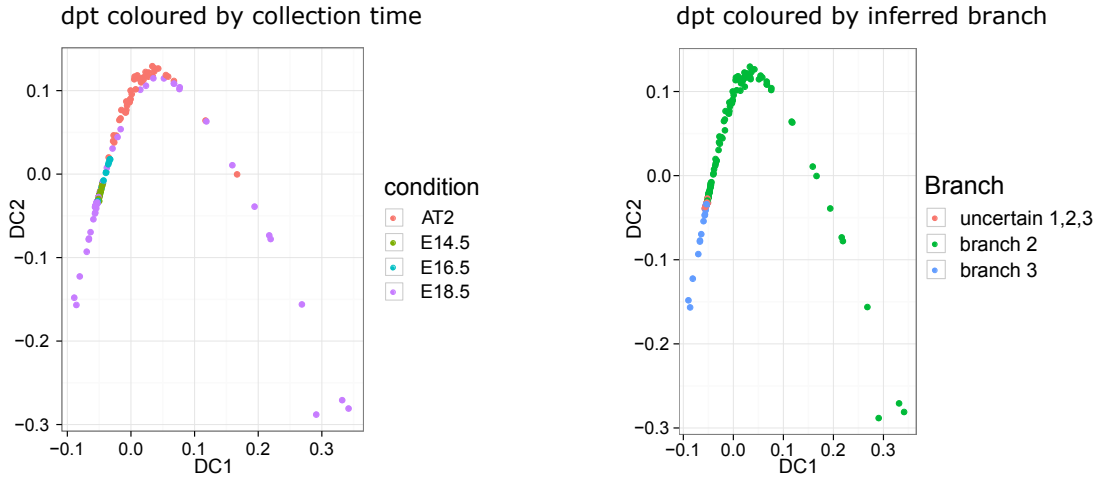
A



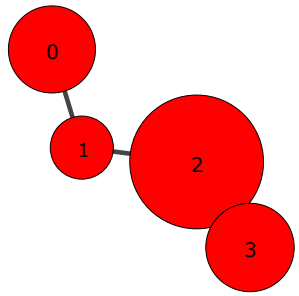
B



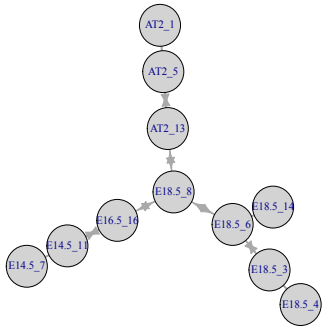
C



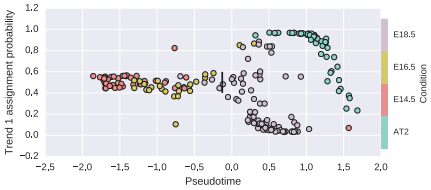
D



E

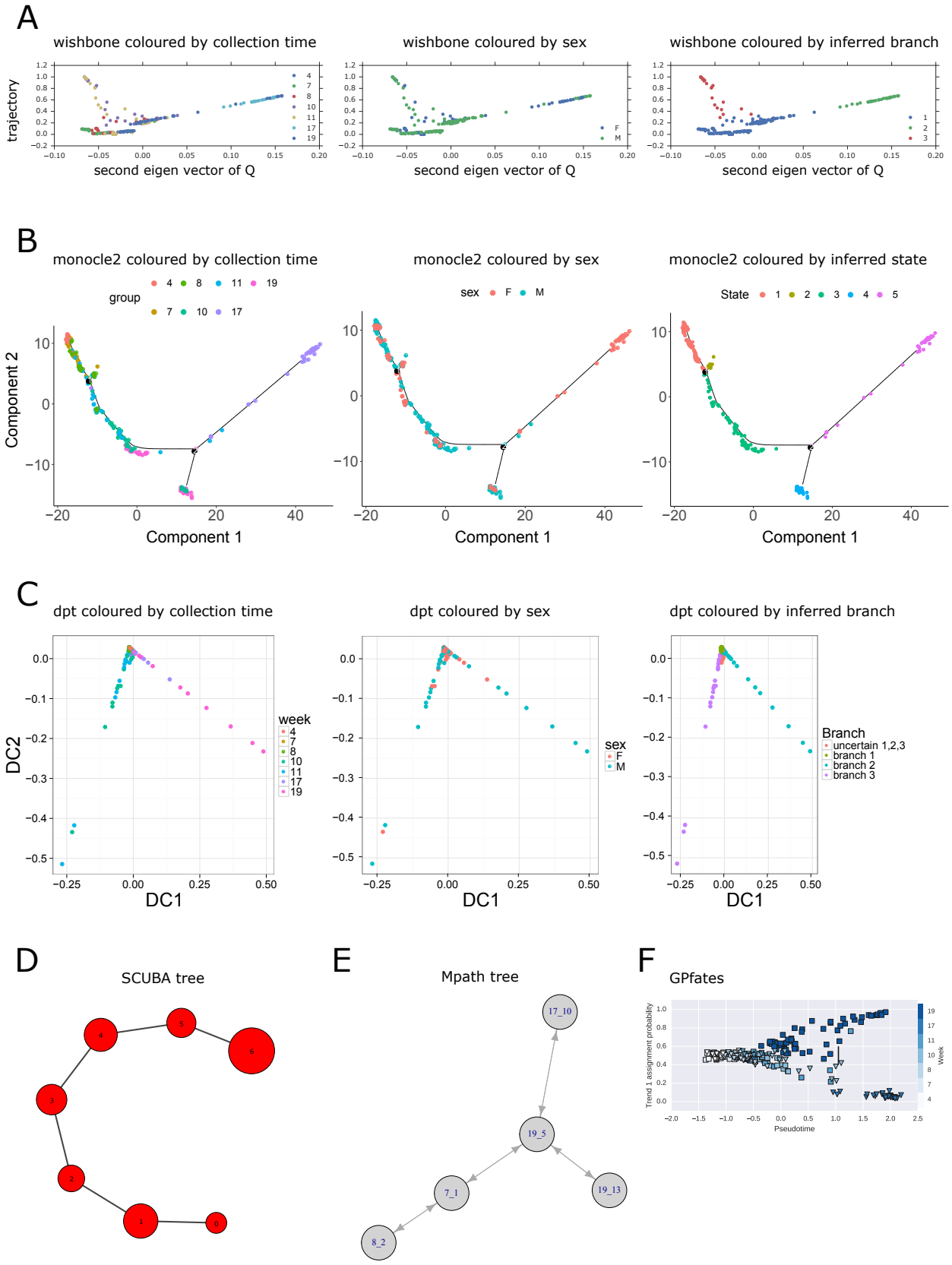


F



Sup. Comp. Fig. 13: Output of bifurcation methods applied to lung data (*23*). AT2 and E18.5 cells are expected to occur in one branch. **(A)** Wishbone's branching structure. Cells are colored by time points (left) and inferred branches (right). **(B)** Minimum spanning tree generated by Monocle2. Cells are colored by time points (left) and inferred cell states (right). **(C)** Visualization of diffusion maps in DPT colored by time points (left) and inferred branches (right). **(D)** Lineage tree by SCUBA: Sizes of bubbles are according to number of cells. **(E)** MPath's minimum spanning tree: First number corresponds to the collection time, second number corresponds to the landmark cluster. **(F)** GPfates trajectory, colored by time points.

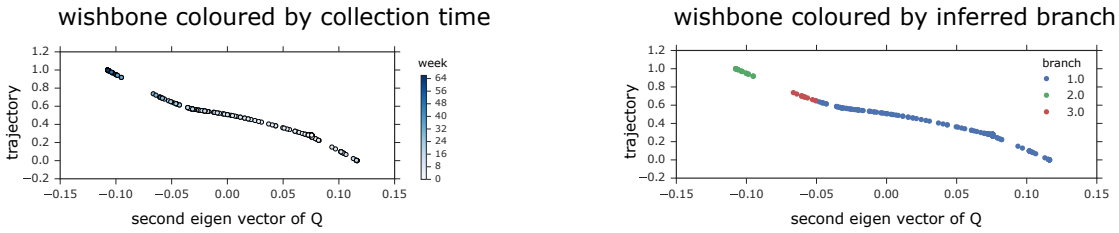
Application on primordial germ cells



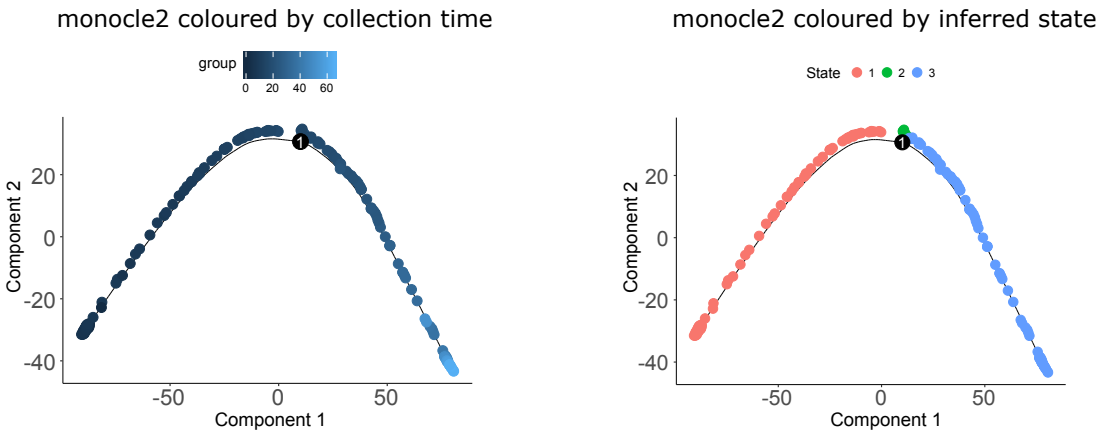
Sup. Comp. Fig. 13: Output of bifurcation methods applied to primordial germ cell data (*24*). Bifurcation event is expected to split female and male cells. **(A)** Wishbone results colored by time points (left), sex (middle) and inferred branches (right). **(B)** Monocle2 results colored by time points (left), sex (middle) and inferred cell states (right). **(C)** DPT results colored by time points (left), sex (middle) and inferred branches (right). **(D)** SCUBA result: Sizes of bubbles are according to number of cells. **(E)** MPath result: First number corresponds to the collection time, second number corresponds to the landmark cluster. **(F)** GPfates trajectory, colored by time points. (□) corresponds to male, (▽) to female cells.

Application on frog development

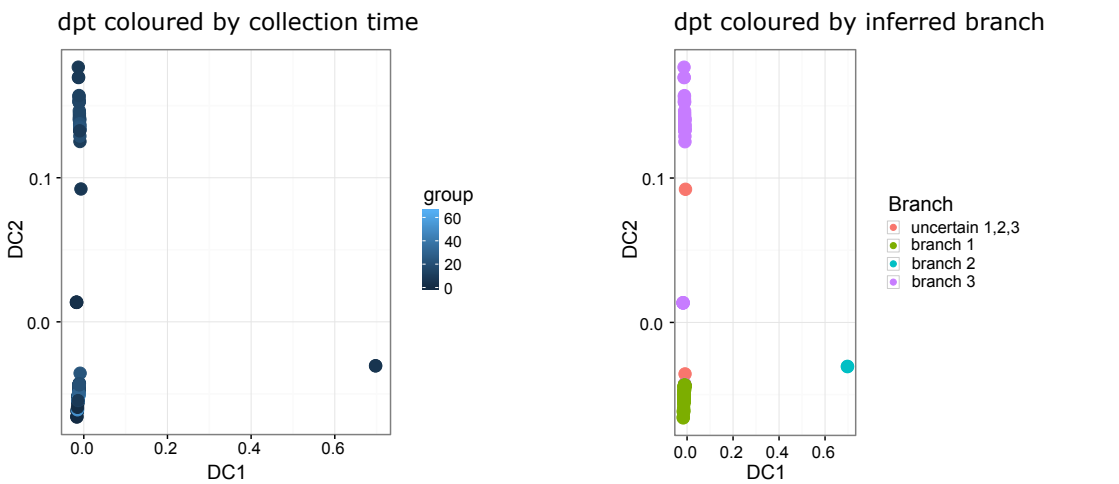
A



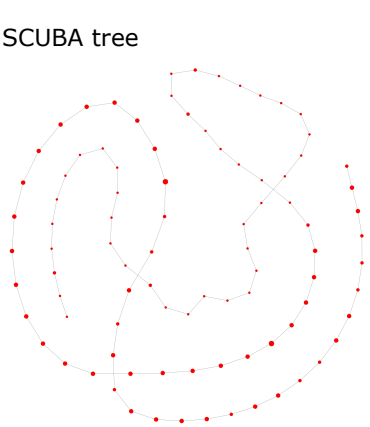
B



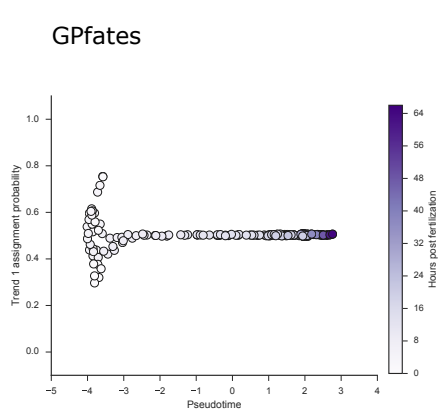
C



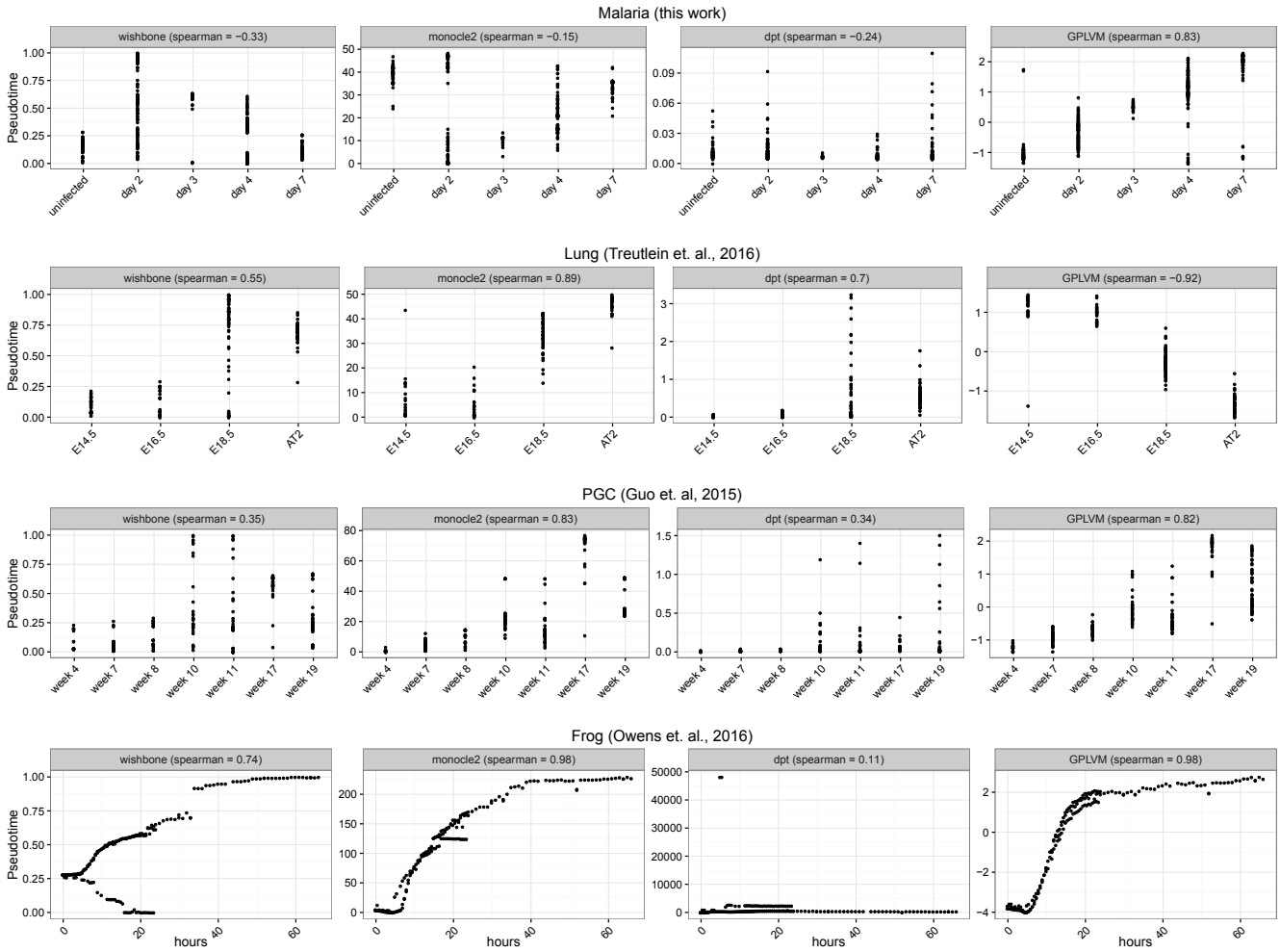
D



E



Sup. Comp. Fig. 13: Output of bifurcation methods applied to developing frog data (*64*). Treated as a negative control, no branching events should be reported. Please note, Mpath failed to model on developmental frog data. **(A)** Wishbone results colored by time points (left) and inferred branches (right). **(B)** Monocle2 results colored by time points (left) and inferred cell states (right). **(C)** DPT results colored by time points (left) and inferred branches (right). **(D)** SCUBA result: Sizes of bubbles are according to number of cells. **(E)** GPfates trajectory, colored by time points.



Sup. Comp. Fig. 14: Accuracy of bifurcation methods. Spearman's rank correlation was calculated by comparing the real time and the inferred pseudotime.