THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Inferring time-derivatives including cell growth rates using Gaussian processes

Citation for published version:
Swain, P, Stevenson, K, Leary, A, Montano-Gutierrez, LF, Clark, IBN, Vogel, J & Pilizota, T 2016, 'Inferring time-derivatives including cell growth rates using Gaussian processes', *Nature Communications*, vol. 7, 13766. https://doi.org/10.1038/ncomms13766

Digital Object Identifier (DOI):
10.1038/ncomms13766

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Nature Communications

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 25. Aug. 2022

Inferring time-derivatives including cell growth rates using Gaussian processes

Peter S. Swain*† Keiran Stevenson* Allen Leary‡
Luis F. Montano-Gutierrez* Ivan B. N. Clark* Jackie Vogel‡
Teuta Pilizota*

Often the time-derivative of a measured variable is of as much interest as the variable itself. For a growing population of biological cells, for example, the population's growth rate is typically more important than its size. Here we introduce a non-parametric method to infer first and second time-derivatives as a function of time from time-series data. Our approach is based on Gaussian processes and applies to a wide range of data. In tests, the method is at least as accurate as others, but has several advantages: it estimates errors both in the inference and in any summary statistics, such as lag times, and allows interpolation with the corresponding error estimation. As illustrations, we infer growth rates of microbial cells, the rate of assembly of an amyloid fibril, and both the speed and acceleration of two separating spindle pole bodies. Our algorithm should thus be broadly applicable.

Introduction

Estimating the time-derivatives of a signal is a common task in science. A well-known example is the growth rate of a population of cells, which is defined as the time-derivative of the logarithm of the population size [1] and is used extensively in both the life sciences and biotechnology.

A common approach to estimate such derivatives is to fit a mathematical equation that, say, describes cellular growth and so determine the maximum growth rate from the best-fit value of a parameter in the equation [2]. Such parametric approaches rely, however, on the mathematical model being a suitable description of the underlying biological or physical process, and, at least for cellular growth, it is common to find examples where the standard models are not appropriate [3].

The alternative is to use a non-parameteric method and so estimate time-derivatives directly from the data. Examples include taking numerical derivatives [4] or using local polynomial or spline estimators [5]. Although these approaches do not require knowledge of the underlying process, it can be difficult to determine the error in their estimation [5] and to incorporate experimental replicates, which with wide access to high throughput technologies, are now the norm.

*: SynthSys – Synthetic and Systems Biology, School of Biological Sciences, University of Edinburgh, Mayfield Road, Edinburgh EH9 3BF, U.K.; ‡: Department of Biology, McGill University, Montreal, Quebec, Canada H3G 0B1; †: Corresponding author (peter.swain@ed.ac.uk)

Here we develop a methodology that uses Gaussian processes to infer both the first and second time-derivatives from time-series data. One advantage of using Gaussian processes over parametric approaches is that we can fit a wider variety of data. Rather than assuming that a particular function characterises the data (a particular mathematical equation), we instead make assumptions about the family of functions that can describe the data. An infinite number of functions exist in this family, and the family can capture many more temporal trends in the data than any one equation. The advantages over existing non-parametric methods are that we can straightforwardly and systematically combine data from replicate experiments (by simply pooling all datasets) and predict errors both in the estimations of derivatives and in any summary statistics. A potential disadvantage because we use Gaussian processes is that we must assume that the measurement noise has a normal or log-normal distribution (as do many other methods), but we can relax this assumption if there are multiple experimental replicates.

To illustrate how our approach predicts errors and can combine information from experimental replicates, we first focus on inferring growth rate from measurements of the optical density of a growing population of biological cells. Plate readers, which are now wide-spread, make such data easy to obtain, typically with hundreds of measurements and often at least 3-10 replicates. We will also, though, show other examples: estimating the rate of *in vitro* assembly of an amyloid fibril and inferring the speed and acceleration of two separating spindle pole bodies in a single yeast cell.

Results

An overview of Gaussian processes

A Gaussian process is a collection of random variables for which any subset has a joint Gaussian distribution [6]. This joint distribution is characterised by its mean and its covariance.

To use a Gaussian process for inference on time-series, we assume that the data can be described by an underlying, or latent, function, and we wish to infer this latent function given the observed data. For each time point of interest, we add a random variable to the Gaussian process. With n time points, there are therefore n corresponding random variables in the Gaussian process. The latent function is given by the values taken by these random variables (Fig. 1A). Without losing any generality, we set the mean of each random variable to be zero [6].

How each random variable in the Gaussian process depends on the other random variables both at earlier and later times, i.e. how the variables covary, determines the shape of the latent function. For example, if each random variable does not covary with any other (the covariance matrix of the Gaussian process is the identity matrix), then the latent function will randomly jump back and forth around zero. If each random variable covaries equally with every other random variable (all the entries of the covariance matrix are one), then the functions sampled will be straight horizontal lines starting at the value of the random variable associated with the first time point. More pertinently, if the covariance for each random variable is positive for those random variables close in time and tending to zero for random variables far away in time, then the functions generated vary, but do so smoothly.

To proceed, we therefore must chose the type of covariance function for the Gaussian process and in doing so we necessarily make some assumptions about the latent functions

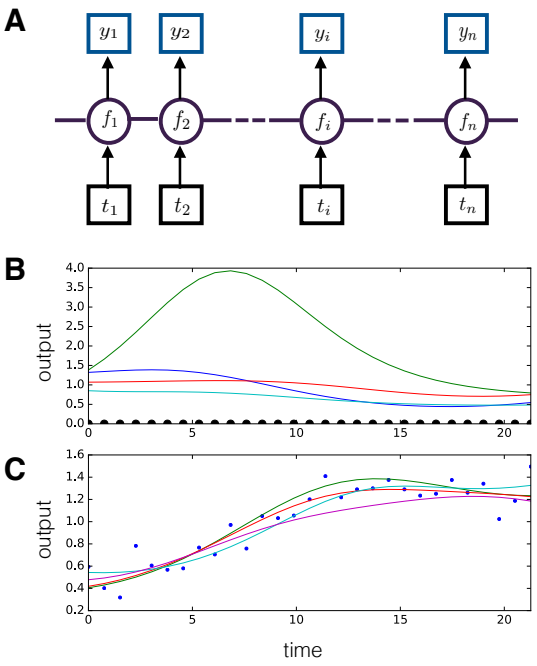

Figure 1. An overview of inference with Gaussian processes. A) A graphical model of a Gaussian process [6]. Squares denote known variables (times, t_i, and data points, y_i); circles denote unknown variables (the underlying, latent function, f_i). We associate a variable in the Gaussian process to each time point and the value of this variable gives the value of the latent function. Each observed data point, y_i, depends only on the corresponding latent variable, f_i. Each f variable, however, depends on all the other f variables (they covary). B) Four examples of latent functions with a squared exponential covariance function. The functions are strictly only defined at the time points of the observations (shown with black semi-circles on the x-axis) but are drawn with a continuous line for clarity. C) Four examples of latent functions after conditioning on the data (data are shown as blue dots). Although each individual function is smooth, there is more variation between functions where the data is more spread. Averaging many latent functions gives the best-fit. The hyperparameters of the covariance matrix are the same as those in B.

that underlie the data. We will often use a squared exponential covariance, which imposes little restriction on the shape of the latent function other than to assume that it is smooth (infinitely differentiable) (Fig. 1B). The Matern covariance function (paramaterised with $\nu = 5/2$) relaxes this smoothness assumption and imposes that the latent function is only twice differentiable [6]. Another alternative is the neural network covariance, which tends to generate sigmoid-shaped latent functions [6]. We emphasise that choosing a covariance function to describe the latent function puts much less restriction on its shape than the more traditional choice of selecting a particular mathematical equation to model the latent function.

Each covariance function is parameterised in its own way, and we must find the appropriate values of these parameters given the data. More correctly, the parameters are called hyperparameters (Methods), and determining the hyperparameters is the computationally intensive part of the inference.

Once the hyperparameters are optimised, we can sample from the Gaussian process to generate a latent function that is consistent with our data (Fig. 1C). The mean latent

function, which we could find from averaging many samples but can also calculate directly, gives the 'best-fit', and the variance of the latent function provides an estimation of the error in the inference.

Following on from earlier work [7], we adapt this standard inference procedure to also allow the inference of the first two time-derivatives of the latent function because these time-derivatives are, in many applications, of more interest than the latent function itself. Errors in inference of the latent function are automatically carried through to the errors in inferring time-derivatives.

Verification of the algorithm

To verify our algorithm's inference of first and second time-derivatives, we followed the tests of De Brabanter *et al.* [8]. Gaussian measurement noise was added to the same analytic functions chosen by De Brabanter *et al.* for which time-derivatives can be found exactly, and the mean absolute difference between the inferred derivative and the exact derivative was used to score the inference (see [8] for details – the end points are not included). We show the distribution of scores for 100 different datasets each with a different sample of the measurement noise (Fig. 2).

For these tests, our method outperforms established alternatives. For illustration, we show results for both the squared exponential covariance function and the neural network covariance function. Independent of the choice, the method performs at least as well as alternatives (Fig. 2).

Estimating cellular growth rates

We now consider the inference of microbial growth rates (strictly, we infer the specific growth rate: the time derivative of the logarithm of the population size). The population size as a function of time is commonly fit to a parametric equation [2], although these equations are restrictive and describe only a particular type of growth [3]. Therefore to provide a further test of our algorithm, we considered a linear sum of two growth equations – the Gompertz [9] and Richards [10] models – to generate a growth curve that cannot in principle be fit by either, but where an exact expression for the first derivative can still be found. We compare our results with smoothing splines, an established non-parametric alternative [3].

For these data sets and this magnitude of measurement noise, both methods perform equally well, but the inference using Gaussian processes becomes more robust as the number of data points increase (Fig. 3). We note that we have artificially favoured the smoothing spline because the smoothing parameter for the spline is set with the variance used to generate the synthetic measurement noise. The Gaussian process methodology, in contrast, infers this variance. Despite the advantage of the spline-based inference, its median error is approximately 45% higher when $n = 1000$.

Turning to experimental measurements, we fit optical densities, which are proportional to the number of cells if properly calibrated [11, 12], and show that we can infer growth rates for two cases that cannot be easily described by parametric approaches [3]. The first exhibits a diauxic shift with two distinct phases of growth and the second shows an exceptionally long lag (Fig. 4). We infer the growth rate and the estimated errors in our inference as a function of time using all experimental replicates. Data from replicate measurements are pooled together, and the algorithm applied as for a single replicate.

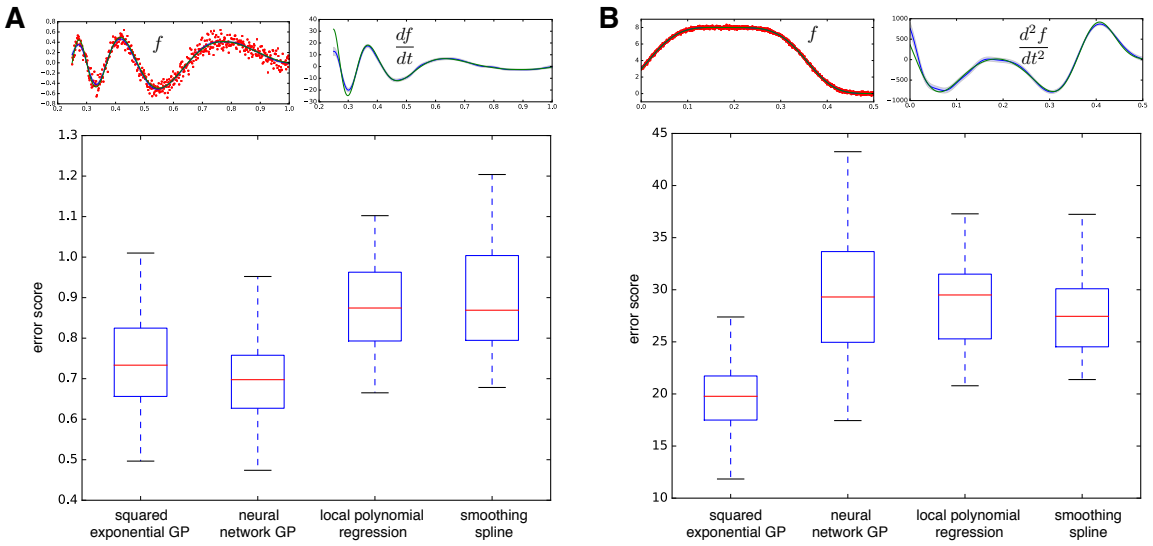

Figure 2. The inference method can perform better than alternatives. A) Inference of the first derivative. A box plot of error scores (related to the mean absolute difference between the inferred and exact derivative) for inference of the first derivative. We use either a squared exponential covariance function or a neural network covariance function for the Gaussian process (GP) and compare with local polynomial regression (with $p = 3$) and a quintic penalised smoothing spline (data for both from [8]). Top left shows one sample data set (in red with 500 data points), the true underlying function (in green), and the inferred latent function using a neural network covariance function – the best fit (in blue); top right shows the corresponding first derivative (with here an error score of 0.64): exact (in green) and inferred (in blue). Equivalent plots for the alternative inference methods are given by De Brabanter *et al.* [8]. Errors (in light blue) are standard deviations. B) Inference of the second derivative. A box plot of scores for inference of the second derivative. The two alternatives are local polynomial regression (with $p = 5$) and a septic penalised smoothing spline (data for both from [8]). Top right shows one sample data set (in red with 1500 data points), the underlying function (in green), and the inferred latent function using a neural network covariance function (in blue); top left shows the corresponding second derivative (with here an error score of 26.2): exact (in green) and inferred (in blue).

Having the inferred growth rate over time can make identifying different stages of the growth curve substantially easier than making this identification from the optical density data alone. For example, the local minimum in the growth rate of Fig. 4A is expected to indicate a shift from cells using glucose to using galactose. Inferring a time-dependent growth rate should increase the robustness of high throughput automated studies, which usually focus on identifying exponential growth [13, 14].

Often summary statistics are used to describe a growth curve, such as the maximum growth rate and the lag time [2], and we can estimate such statistics and their associated errors. From our inference, we can sample latent functions that are consistent with the data. Each sample provides an example of a latent function that 'fits' the data. To estimate errors in statistics, we generate say 100 samples of the latent function and its time-derivatives (Fig. 4A inset). For each sample, we calculate the statistic for that sample, such as the maximum growth rate. We therefore obtain a probability distribution for the statistic and report the mean and standard deviation of this distribution as the

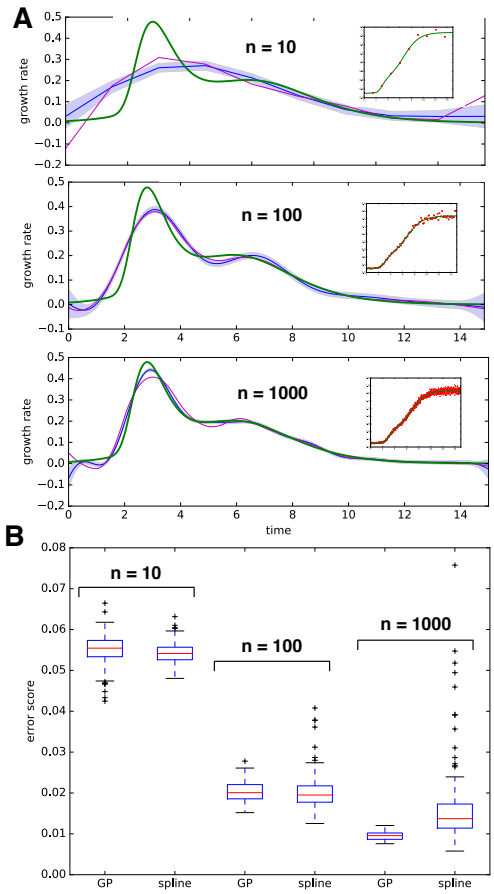

Figure 3. Inference of growth rates with Gaussian processes becomes more robust with increasing numbers of data points. A) We show the inferred growth rate (the best-fit with a squared exponential Gaussian process in blue with errors in light blue and from a cubic smoothing spline in purple) and the exact growth rate (in green) for synthetic data sets with either 10, 100, or 1000 time points (insets, with the underlying growth curve in green). Even though we favour smoothing splines by setting the smoothing parameter to be proportional to the exact variance of the measurement noise (whereas the Gaussian process infers this parameter), both methods perform similarly. B) A box plot of errors scores for inference of the growth rate from 100 data sets with randomly generated log-normal measurement noise. For the Gaussian process, the distribution of error scores tightens with increasing number of data points.

best-fit value and the estimated error (0.16 ± 0.002 hr^{-1} for the maximum growth rate for the data in Fig. 4A). A similar approach applies for any statistic that can be calculated from a single growth curve (Methods).

The data for Fig. 4B are considerably noisier than the data for Fig. 4A, and the spread of data is larger at short times than at long times. The magnitude of the measurement noise changes with time. More correctly, we typically assume that the measurement noise can be described by a Gaussian distribution with zero mean and a constant standard deviation. The magnitude of the measurement noise is this standard deviation, and the standard deviation here, for the data of Fig. 4B, appears to be time-dependent (it is largest at early times). To empirically estimate the relative scale of this change, we calculate the variance across replicates at each time point. We assume that the magnitude of the measurement noise is a time-independent constant multiplied by this time-dependent

relative scale, and we fit that constant (Methods).

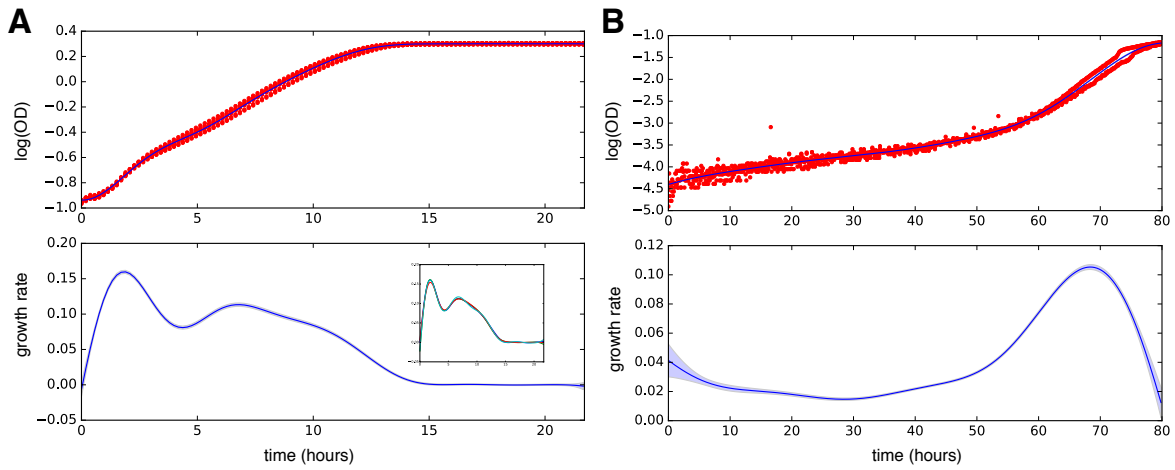

Figure 4. Microbial growth rates can be inferred as a function of time. A) A growth curve of *Saccharomyces cerevisiae* in a mixture of 0.4% glucose and 1% galactose showing a diauxic shift (7 replicates, each with $n = 115$). The best-fit (mean) latent function is shown in dark blue and the inferred growth rate is shown below. All error bars (light blue) are standard deviations. The inset shows, as an example, 4 sample estimates of the growth rate as a function of time (samples of the first derivative of the latent function – the corresponding samples of the latent function itself are not shown). B) Growth of *Escherichia coli* in hyperosmotic conditions with an unusually long lag and short growth period (2 replicates, each with $n = 646$) and the inferred growth rate. The magnitude of the measurement noise is here allowed to vary with time and empirically estimated across the replicates (Methods). Error bars (light blue) are standard deviations.

Further applications

As additional examples, we first infer the rate of assembly of an amyloid fibril as a function of time from *in vitro* data (Fig. 5A) [15]. Despite each replicate having high measurement noise compared to the microbial data, the rate of fibril assembly can be inferred accurately because of the many replicates. The second example is one where both the first and the second derivative are useful: estimating the speed of separation of the spindle poles during anaphase in a single cell of budding yeast (Fig. 5B). We demonstrate that we can infer both time-derivatives and their errors from a single replicate. As expected, the size of the estimated error increases for the first derivative relative to the error in the regression and increases again for the second derivative. Changes in the speed of separation (extrema in the second derivative) are used to characterise anaphase [16] into the fast, pause and slow elongation phases [17]. We chose a Gaussian process with a neural network covariance function for this data rather than the squared exponential covariance function used for the others: a difference that is important here because we only have a single replicate with few data points. The latent functions generated then tend to be flatter either side of the increase in separation, which leads to smoother inferences of the acceleration.

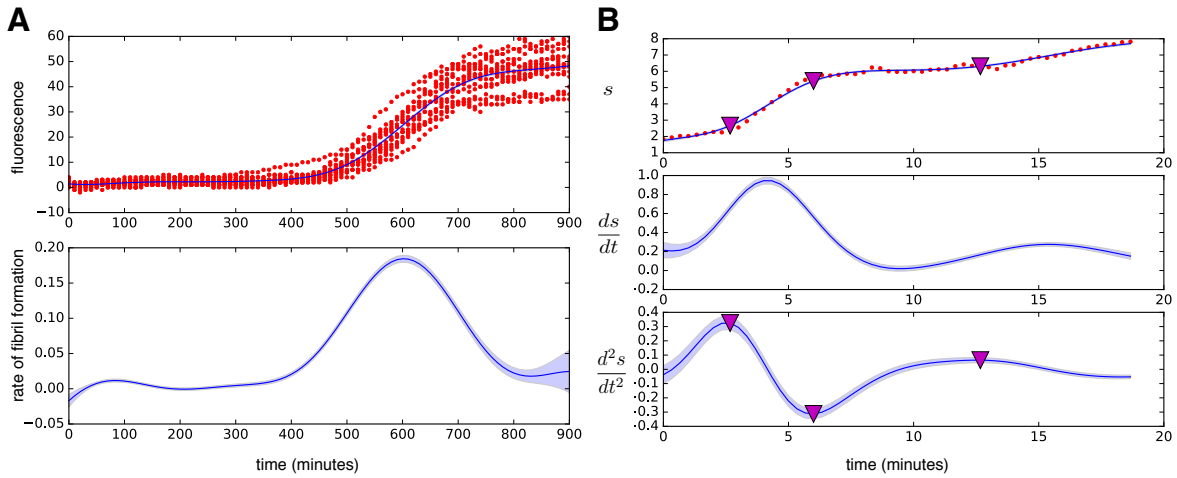

Figure 5. The algorithm has wide application. A) Inferring the *in vitro* rate of assembly of an amyloid fibril. Fluorescence data reporting the formation of fibrils in bovine insulin (at a concentration of 0.1 mg ml^{-1}) by the binding of the dye Thioflavin T are shown (red dots) with 15 replicates (each with $n = 91$) [15]. The best-fit (top) and the inferred rate of fibril assembly (bottom) are shown in dark blue. We empirically estimate the magnitude of the measurement noise across the replicates. Errors (in light blue) are standard deviations. B) Inferring the speed and acceleration of separation of the spindle poles in *S. cerevisiae*. The distance, s, between the two spindles in a single cell is plotted in microns as a function of time (red dots with $n = 57$). The best-fit and the inferred speed (middle) and acceleration (bottom) are shown in dark blue. The triangles denote turning points in the acceleration and separate anaphase into stages with fast and slow elongation separated by a pause [16]. Errors are standard deviations.

Discussion

To conclude, we have introduced a non-parametric method that uses Gaussian processes to infer first and second derivatives from time-series data. In tests, our approach is at least as accurate as others (Figs. 2 and 3), but has several advantages: it systematically estimates errors, both for the regression and the inferred derivatives; it allows interpolation with the corresponding error estimation (Gaussian processes were developed for interpolation [6]); and it allows sampling of the latent function underlying the data and so can be used to estimate errors in any statistic of that function by calculating the statistic for the samples.

For fitting growth curves, several alternatives exist [3, 18, 19, 20], which, although mostly focusing on parametric approaches, do allow spline fitting [3] and polynomial regression [18, 20]. Both approaches have been criticised, being sensitive to outliers and potentially having systematic biases [5], and at least in the case of splines appear less robust (Fig. 3). Further, our software performs inference using all replicates, can infer second derivatives, and rigorously estimates errors. Where error estimation in summary statistics has been addressed [3], bootstrapping of the data is used. This approach is perhaps less suited for time-series data than our approach of sampling latent functions because it leads to some randomly chosen data points being weighted more than others when generating sample fits.

Of the three we considered, we find that the squared exponential function is generally the best choice of covariance function when estimating time-derivatives because it typically results in the inference of first- and second-derivatives with a smoothness that is consistent

with *a priori* expectations of the nature of the underlying dynamics. Although the Matern covariance is not as restrictive because it constrains the smoothness of the latent functions less, it can lead to the inference of rough, fluctuating derivatives, particularly for the second-derivative and if the magnitude of the measurement noise is high. For example, using the Matern covariance gives poor results for the data in Fig. 2A (with median error scores that are approximately 60% higher than those for the squared exponential covariance), but performs slightly better (medians within 10%) for the less noisy data in Fig. 3. Finally, the neural network covariance, although perhaps the least prone to the inference of rough time-derivatives, can be more sensitive to prior information: the hyperparameter controlling the flexibility of the latent function is optimised to its upper bound more often than for the other covariance functions. All three covariance functions are implemented in our code and can be tested for a new type of data.

Like any Bayesian method, prior information on bounds for the hyperparameters of the covariance function can affect the inference, although these bounds can typically be set so that the best-fit values are far from the bounds. In particular, how closely the latent function follows the data depends both on its flexibility and on the size of the measurement noise. An outlier can be followed if the flexibility is high or if the measurement noise is low. When there is not sufficient data, the algorithm, rightly in our opinion, requires prior information to make this choice. Alternative methods also require prior specification, such as the degree of smoothness in fitting with either splines or a local polynomial method, like LOESS. For a particular type of data, the bounds typically need to be set once allowing high throughput analyses.

Measuring cellular growth rates is a daily task in many laboratories, but, if using a non-parameteric approach, researchers often follow the method developed in the early days of molecular biology: finding the gradient of a line fit to the portion of the growth curve that appears most straight on a semi-log plot. Our methodology takes advantages of advances in machine learning to allow inference not only of the maximum growth rate but of the growth rate as a function of time. Time-dependent growth rates must capture more of the underlying biology, such as the time of the diauxic shift in Fig. 4A, but they have been little exploited. We believe that using more advanced inference techniques, such as the one based on Gaussian processes that we present here, in combination with developments in high throughput technologies will transform our understanding of cellular growth and the factors that control it.

Methods

Using a Gaussian process to fit time-series data

In the following, we will denote a Gaussian distribution with mean μ and covariance matrix Σ as $\mathcal{N}(\mu, \Sigma)$ and use the notation of Rasmussen and Williams [6] as much as possible.

Prior probability

For n data points y_i at inputs x_i (each x_i is a time for a growth curve), we denote the underlying latent function as $f(x)$. We define a covariance matrix $k(x, x')$, which has an

explicit (although here not written) dependence on hyperparameters θ, and obeys

$$\begin{aligned} \mathrm{Cov}[f(x), f(x')] &= E\Big[\Big(f(x) - E[f(x)]\Big)\Big(f(x') - E[f(x')]\Big)\Big] \\ &= k(x, x'; \theta), \end{aligned} \tag{1}$$

where the expectations are taken over distribution of latent functions (samples of $f(x)$) and θ denotes the covariance function's hyperparameters.

We interpret Eq. 1 as giving the prior probability distribution of the latent functions $f(X)$, where were we use X to denote the inputs x_i, such that

$$\Big[f(x_1), ..., f(x_n)\Big]^T \sim \mathcal{N}(\mathbf{0}, K(X, X)) \tag{2}$$

where $K(X, X)$ is the $n \times n$ matrix with components $k(x_i, x_j)$. With $\mathbf{f}$ denoting $[f(x_1), ..., f(x_n)]$, this prior probability can be written as

$$P(\mathbf{f}|X, \theta) \sim \mathcal{N}(\mathbf{0}, K(X, X)) \tag{3}$$

noting the dependence of $k(x, x'; \theta)$ on the hyperparameters θ.

Marginal likelihood

After choosing a covariance function, to use Gaussian processes in regression, we must optimise the covariance function's hyperparameters given the observed data. We will do so by maximising the marginal likelihood, where the marginalisation is made by integrating over all possible latent functions [6]. Once the parameters of the covariance function have been determined, we can sample latent functions given the data. We consider the squared exponential, Matern (with $\nu = 5/2$), and neural network covariance functions.

To optimise the hyperparameters given the data, we therefore consider the likelihood $P(\mathbf{y}|\theta, X)$, which, more correctly, is a marginal likelihood

$$\begin{aligned} P(\mathbf{y}|\theta, X) &= \int d\mathbf{f} P(\mathbf{y}, \mathbf{f}|\theta, X) \\ &= \int d\mathbf{f} P(\mathbf{y}|\mathbf{f}, X, \theta) P(\mathbf{f}|X, \theta) \end{aligned} \tag{4}$$

where the marginalisation is over all choices of the latent function $\mathbf{f}$ evaluated at X.

If we assume that for all y_i, $y_i = f(x_i) + \epsilon_i$ where each ϵ_i is an independent Gaussian variable with zero mean and a standard deviation of $\sigma_i = \sigma$ for simplicity, then

$$P(\mathbf{y}|\mathbf{f}, X, \theta) \sim \mathcal{N}(\mathbf{f}, \sigma^2 I) \tag{5}$$

where I is the $n \times n$ identity matrix. Eqs. 3 and 5 imply that the marginal likelihood is also Gaussian:

$$P(\mathbf{y}|\theta, X) \sim \mathcal{N}(\mathbf{0}, K(X, X) + \sigma^2 I). \tag{6}$$

We use a maximum-likelihood method to find the hyperparameters and maximise the marginal likelihood (Eq. 6). We have two hyperparameters for the squared exponential covariance function and the parameter, σ, which characterises the measurement noise. We assume a bounded, uniform prior probability for each of these hyperparameters and use the Broyden-Fletcher-Goldfarb-Shanno algorithm [4] to find their optimum values. Although one optimisation run from random initial choices of the hyperparameters is usually sufficient, choosing the best from multiple runs can prevent the algorithm finding local maxima.

Making predictions

Given the optimum choice of the hyperparameters, we would like to generate sample latent functions at points X^*, which to include the possibility of interpolation need not be the same as X, by sampling from $P(\mathbf{f}^*|X, \mathbf{y}, \theta, X^*)$. Using Eq. 6 and that the distribution of the latent function evaluated at X^* is also Gaussian, we can write the joint probability of $\mathbf{y}$ and $\mathbf{f}^*$ as [6]:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \end{pmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X,X^*) \\ K^T(X,X^*) & K(X^*,X^*) \end{bmatrix}\right) \tag{7}$$

where $K(X, X^*)$ is the $n \times n^*$ matrix with components $k(x_i, x_j^*)$.

Conditioning Eq. 7 on the data $\mathbf{y}$, standard results for Gaussian distributions [6] give that the probability distribution $P(\mathbf{f}^*|X, \mathbf{y}, \theta, X^*)$ is also Gaussian with mean

$$E[\mathbf{f}^*] = K(X^*, X)\Big[K(X,X) + \sigma^2 I\Big]^{-1}\mathbf{y} \tag{8}$$

and covariance matrix

$$\mathrm{Cov}[\mathbf{f}^*] = K(X^*,X^*) - K(X^*,X)\Big[K(X,X) + \sigma^2 I\Big]^{-1}K^T(X^*,X). \tag{9}$$

We use Eqs. 8 and 9 to sample $\mathbf{f}^*$.

Inferring the first- and second time-derivatives

To determine the time-derivative of the data, we use that the derivative of a Gaussian process is another Gaussian process [6]. We can therefore adapt standard techniques for Gaussian process to allow time-derivatives to be sampled too.

Building on the work of Boyle [7], we let $g(x)$ and $h(x)$ be the first and second derivatives with respect to x of the latent function $f(x)$. If $f(x)$ is a Gaussian process then so are both $g(x)$ and $h(x)$. Writing ∂_1 and ∂_2 for the partial derivatives with respect to the first and second arguments of a bivariate function, we have

$$C_{gf}(x_i,x_j) = \partial_1 k(x_i,x_j) \quad ; \quad C_{fg}(x_i,x_j) = \partial_2 k(x_i,x_j) \quad ; \quad C_{gg}(x_i,x_j) = \partial_1\partial_2 k(x_i,x_j) \tag{10}$$

and that

$$C_{hf}(x_i,x_j) = \partial_1^2 k(x_i,x_j) \quad ; \quad C_{fh}(x_i,x_j) = \partial_2^2 k(x_i,x_j) \tag{11}$$

as well as

$$C_{hg}(x_i,x_j) = \partial_1^2\partial_2 k(x_i,x_j) \quad ; \quad C_{gh}(x_i,x_j) = \partial_1\partial_2^2 k(x_i,x_j) \quad ; \quad C_{hh}(x_i,x_j) = \partial_1^2\partial_2^2 k(x_i,x_j) \tag{12}$$

following [21].

Consequently the joint probability distribution for $\mathbf{y}$ and $\mathbf{f}^*$, $\mathbf{g}^*$, and $\mathbf{h}^*$ evaluated at points X^* is again Gaussian (cf. Eq. 7):

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \\ \mathbf{g}^* \\ \mathbf{h}^* \end{pmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K + \sigma^2 I & K(X,X^*) & \partial_2 K(X,X^*) & \partial_2^2 K(X,X^*) \\ K(X^*,X) & K^* & \partial_2 K^* & \partial_2^2 K^* \\ \partial_1 K(X^*,X) & \partial_1 K^* & \partial_1\partial_2 K^* & \partial_1\partial_2^2 K^* \\ \partial_1^2 K(X^*,X) & \partial_1^2 K^* & \partial_1^2\partial_2 K^* & \partial_1^2\partial_2^2 K^* \end{bmatrix}\right) \tag{13}$$

where we write $K = K(X, X)$ and $K^* = K(X^*, X^*)$ for clarity.

The covariance function is by definition symmetric: $k(x_i, x_j) = k(x_j, x_i)$ (Eq. 1). Therefore $\partial_1^k \partial_2^\ell k(x_i, x_j) = \partial_2^k \partial_1^\ell k(x_j, x_i)$ and so

$$\partial_1^k \partial_2^\ell K(X^*, X) = [\partial_1^\ell \partial_2^k K(X, X^*)]^T \tag{14}$$

for all positive integers k and ℓ. Consequently, the covariance matrix in Eq. 13 is also symmetric.

Conditioning on $\mathbf{y}$ now gives that the distribution $P(\mathbf{f}^*, \mathbf{g}^*, \mathbf{h}^* | X, \mathbf{y}, \theta, X^*)$ is Gaussian with mean

$$E\left[\begin{pmatrix} \mathbf{f}^* \\ \mathbf{g}^* \\ \mathbf{h}^* \end{pmatrix}\right] = \begin{pmatrix} K(X^*, X) \\ \partial_1 K(X^*, X) \\ \partial_1^2 K(X^*, X) \end{pmatrix} \Big[K + \sigma^2 I\Big]^{-1} \mathbf{y} \tag{15}$$

and covariance matrix

$$\begin{aligned} \mathrm{Cov}\left[\begin{pmatrix} \mathbf{f}^* \\ \mathbf{g}^* \\ \mathbf{h}^* \end{pmatrix}\right] &= \begin{pmatrix} K^* & [\partial_1 K^*]^T & [\partial_1^2 K^*]^T \\ \partial_1 K^* & \partial_1 \partial_2 K^* & [\partial_1^2 \partial_2 K^*]^T \\ \partial_1^2 K^* & \partial_1^2 \partial_2 K^* & \partial_1^2 \partial_2^2 K^* \end{pmatrix} \\ &- \begin{pmatrix} K(X^*, X) \\ \partial_1 K(X^*, X) \\ \partial_1^2 K(X^*, X) \end{pmatrix} \Big[K + \sigma^2 I\Big]^{-1} \begin{pmatrix} K^T(X^*, X) \\ [\partial_1 K(X^*, X)]^T \\ [\partial_1^2 K(X^*, X)]^T \end{pmatrix}. \end{aligned} \tag{16}$$

Eq. 16 includes Eq. 9 and shows that

$$\mathrm{Cov}[\mathbf{g}^*] = \partial_1 \partial_2 K^* - \partial_1 K(X^*, X) \Big[K + \sigma^2 I\Big]^{-1} [\partial_1 K(X^*, X)]^T \tag{17}$$

which gives the error in the estimate of the first derivative [7]. Similarly,

$$\mathrm{Cov}[\mathbf{h}^*] = \partial_1^2 \partial_2^2 K^* - \partial_1^2 K(X^*, X) \Big[K + \sigma^2 I\Big]^{-1} [\partial_1^2 K(X^*, X)]^T \tag{18}$$

is the error in estimating the second derivative.

Using an empirically estimated measurement noise

Although our derivation is given for a Gaussian process where the measurement errors in the data are independent and identically distributed with a Gaussian distribution of mean zero, the derivations are unchanged if the measurement noise has a different standard deviation for each time point [6].

When the magnitude of the measurement noise appears to change with time, we first empirically estimate the relative magnitude of the measurement noise by the variance across all replicates at each time point. We then smooth this estimate over time (with a Gaussian filter with a width of 10% of the total time of the experiment, but the exact choice in not important) and replace the identity matrix, I, in Eqs. 6, 15, and 16 by a diagonal matrix with the relative measurement noise on the diagonal in order to make predictions.

Estimating the growth characteristics

From the growth curve, we estimate the maximum growth rate as the maximum time-derivative of the logarithm of the growth curve [2]:

$$\text{growth rate} = \max_t \frac{y'(t)}{y(t)} \tag{19}$$

where we denote the growth curve as $y(t)$. The doubling time is $\ln(2)$ times the inverse of the growth rate. We define the lag time as the intercept of the line parallel to the time axis that passes through the initial OD, $y(0)$, and the tangent to the logarithm of the growth curve from the point on the growth curve with maximum growth rate (a standard choice [2]). If this point of maximum growth rate is at $t = t^*$, then

$$\text{lag time} = t^* - \frac{y(t^*)}{y'(t^*)} \ln \frac{y(t^*)}{y(0)}. \tag{20}$$

For each characteristic, we can estimate measurement error through calculating the characteristic for 100s of sampled latent growth curves.

Implementation and GUI

The code for our algorithm is freely available and written in Python 3 using NumPy [22], SciPy (Jones, Oliphant, Peterson, *et al.*), Matplotlib [23], and the Pandas data analysis library (all available via the free Anaconda package) and is compatible with Microsoft's Excel. We give an example script and data set and have written a GUI that runs on Windows, OS X, and Linux.

Code availability

The software and instructions for its use are at
`http://swainlab.bio.ed.ac.uk/software/fitderiv`

Generating synthetic data for Figure 3

To generate the synthetic data shown in Fig. 3, we use a weighted sum of a Gompertz model (parameters: $A = 1.1$, $\mu_m = 0.6$, and $\lambda = 2.3$; weight: 0.3) and a Richards model (parameters: $A = 1.5$, $\mu_m = 0.3$, $\lambda = 4.3$, and $\nu = 0.8$; weight: 0.7) using the notation of [2]. We added log-normal measurement noise with zero mean and a standard deviation of $\sigma_m = 0.03$ and used the SciPy implementation of a cubic spline and its time-derivative, setting the smoothing parameter to be $n\sigma_m^2$ where n is the number of data points.

The error score in Fig. 3 is the mean absolute deviation of the inferred growth rate from the exact growth rate ignored 5% of the data points both at the beginning and end of the time-series in order to avoid end-point effects potentially dominating the error [8].

Experimental methods

Data for Fig. 4A was gathered using Tecan Infinity M200 plate reader and a BY4741 strain of *S. cerevisiae* growing in synthetic complete media supplemented with 0.4% glucose and 1% galactose at 30^{o}C, following an established protocol [24]. OD was measured at an absorbance wavelength of 595 nm every 11.4 minutes.

Figure	Covariance function	Hyperparameter	Lower bound	Upper bound
1A	squared exponential	0	10^{-5}	10^{5}
		1	10^{-3}	10^{2}
		2	10^{-5}	10^{2}
	neural network	0	10^{-1}	10^{5}
		1	10^{3}	10^{3}
		2	10^{-6}	10^{2}
1B	squared exponential	0	10^{-5}	10^{5}
		1	10^{-3}	10^{4}
		2	10^{-5}	10^{2}
	neural network	0	10^{-1}	10^{5}
		1	10^{1}	$10^{2.5}$
		2	10^{-6}	10^{2}
2A & 2B	squared exponential	0	10^{-5}	10^{5}
		1	10^{-6}	10^{2}
		2	10^{-5}	10^{2}
3A	squared exponential	0	10^{-5}	10^{5}
		1	10^{-6}	10^{2}
		2	10^{-5}	10^{0}
3B	neural network	0	10^{-1}	10^{5}
		1	10^{-4}	10^{-1}
		2	10^{-6}	10^{2}

Table 1. Ranges of hyperparameters used for the examples. For the squared exponential covariance function, the hyperparameters determine the amplitude of the variation in the latent function, its flexibility, and the magnitude of the measurement noise; for the neural network covariance function, the hyperparameters determine the initial y-value of the latent function, its flexibility, and the magnitude of the measurement noise.

Data for Fig. 4B was gathered using a Spectrostar Omega microplate reader and a BW25113 strain of *E. coli* growing in MM9 (sodium-sodium instead of sodium-potassium) media with 0.1% glucose and 1106 mOsm sucrose at 37^{o}C. OD was measured at an absorbance wavelength of 600 nm every 7.5 minutes.

Data for Fig. 5A is from [15].

Data for Fig. 5B was gathered using a custom spinning disk confocal microscope for 20mins in 20s time steps with 50ms exposure time per focal plane. Spindle pole bodies were labelled with Spc42-Cerulean. An image stack of 30 z-planes with 300nm step size was gathered for each time point to allow the position of the spindle poles to be fitted to 3-D Gaussian distributions and tracked in time. Imaging, fitting and tracking followed an established protocol [16].

Data availability

Data generated in this work is available at `http://dx.doi.org/10.7488/ds/1405`.

References

[1] Monod, J. The growth of bacterial cultures. *Ann. Rev. Microbiol.* **3**, 371–394 (1949).

[2] Zwietering, M. H., Jongenburger, I., Rombouts, F. M., and van 't Riet, K. Modeling of the bacterial growth curve. *App Environ Microbiol* **56**, 1875–1881, June (1990).

[3] Kahm, M., Hasenbrink, G., Lichtenberg-Frate, H., Ludwig, J., and Kschischo, M. grofit: Fitting biological growth curves with R. *J Stat Softw* **33**, 1–21 (2010).

[4] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, U.K., (2007).

[5] Newell, J. and Einbeck, J. A comparative study of nonparametric derivative estimators. In *Proc. of the 22nd International Workshop on Statistical Modelling*, (2007).

[6] Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. MIT Press, Cambridge, Massachusetts, (2006).

[7] Boyle, P. *Gaussian processes for regression and optimization*. PhD thesis, Victoria University of Wellington, (2007).

[8] De Brabanter, K., De Brabanter, J., De Moor, B., Gijbels, I., De Brabanter, K., De Brabanter, J., and Gijbels, I. Derivative estimation with local polynomial fitting. *J Mach Learn Res* **14**, 281–301, January (2013).

[9] Gompertz, B. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. *Philos Trans R Soc London* **115**, 513–585 (1825).

[10] Richards, F. J. A flexible growth function for empirical use. *J Exp Bot* **10**, 290–300 (1959).

[11] Warringer, J. and Blomberg, A. Automated screening in environmental arrays allows analysis of quantitative phenotypic profiles in Saccharomyces cerevisiae. *Yeast* **20**, 53–67 (2003).

[12] Stevenson, K., McVey, A. F., Clark, I. B. N., Swain, P. S., and Pilizota, T. General calibration of microbial growth in microplate readers. Preprint at `http://dx.doi.org/10.1101/061861` (2016).

[13] Zeevi, D., Sharon, E., Lotan-Pompan, M., Lubling, Y., Shipony, Z., Raveh-Sadka, T., Keren, L., Levo, M., Weinberger, A., and Segal, E. Compensation for differences in gene copy number among yeast ribosomal proteins is encoded within their promoters. *Genome Res* **21**, 2114–2128 (2011).

[14] Chevereau, G., Dravecká, M., Batur, T., Guvenek, A., Ayhan, D. H., Toprak, E., and Bollenbach, T. Quantifying the determinants of evolutionary dynamics leading to drug resistance. *PLoS Biol* **13**, e1002299 (2015).

[15] Morris, R. J., Eden, K., Yarwood, R., Jourdain, L., Allen, R. J., and Macphee, C. E. Mechanistic and environmental control of the prevalence and lifetime of amyloid oligomers. *Nat Commun* **4**, 1891 (2013).

[16] Nazarova, E., O'Toole, E., Kaitna, S., Francois, P., Winey, M., and Vogel, J. Distinct roles for antiparallel microtubule pairing and overlap during early spindle assembly. *Mol Biol Cell* **24**, 3238–3250 (2013).

[17] Kahana, J. A., Schnapp, B. J., and Silver, P. A. Kinetics of spindle pole body separation in budding yeast. *Proc Nat Acad Sci USA* **92**, 9707–9711 (1995).

[18] Verissimo, A., Paixao, L., Neves, A. R., and Vinga, S. BGFit: management and automated fitting of biological growth curves. *BMC Bioinformatics* **14**, 283 (2013).

[19] Huang, L. IPMP 2013 – a comprehensive data analysis tool for predictive microbiology. *Int J Food Microbiol* **171**, 100–107 (2014).

[20] Bukhman, Y. V., DiPiazza, N. W., Piotrowski, J., Shao, J., Halstead, A. G. W., D., B. M., Xie, E., and Sato, T. K. Modeling microbial growth curves with GCAT. *Bioenerg Res* **8**, 1022–1030 (2015).

[21] Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. Derivative observations in Gaussian process models of dynamic systems. *Adv Neural Inf Process Syst* **15**, 1033–1040 (2003).

[22] Oliphant, T. E. Python for scientific computing. *Comput Sci Eng* **9**, 10 (2007).

[23] Hunter, J. D. Matplotlib: A 2D graphics environment. *Comput Sci Eng* **9**, 90 (2007).

[24] Lichten, C. A., White, R., Clark, I. B. N., and Swain, P. S. Unmixing of fluorescence spectra to resolve quantitative time-series measurements of gene expression in plate readers. *BMC Biotechnol.* **14**, 11 (2014).

Acknowledgements

We thank Guido Sanguinetti and Nacho Molina for advice on Gaussian processes, Ryan Morris and Cait MacPhee for providing the data in Fig. 5A, Alejandro Granados for technical assistance, and our funders: a BBSRC iCASE award (KS), the Wellcome Trust and Conacyt (LFMG), and the CIHR (AL & JV). PSS, IBNC, and TP gratefully acknowledge the support of the UK Research Councils Synthetic Biology for Growth programme and are members of a BBSRC/EPSRC/MRC funded Synthetic Biology Research Centre. The authors have no competing financial interests.

Author contributions

PSS developed the algorithm and, with KS & AL, the computer code; KS, AL, LFM, IBNC, & JV generated the data; TP & PSS wrote the manuscript.