

RESEARCH ARTICLE

Open Access

# A Simple Approach to Ranking Differentially Expressed Gene Expression Time Courses through Gaussian Process Regression

Alfredo A Kalaitzis\* and Neil D Lawrence\*

## Abstract

**Background:** The analysis of gene expression from time series underpins many biological studies. Two basic forms of analysis recur for data of this type: removing inactive (quiet) genes from the study and determining which genes are differentially expressed. Often these analysis stages are applied disregarding the fact that the data is drawn from a time series. In this paper we propose a simple model for accounting for the underlying temporal nature of the data based on a Gaussian process.

**Results:** We review Gaussian process (GP) regression for estimating the continuous trajectories underlying in gene expression time-series. We present a simple approach which can be used to filter quiet genes, or for the case of time series in the form of expression ratios, quantify differential expression. We assess via ROC curves the rankings produced by our regression framework and compare them to a recently proposed hierarchical Bayesian model for the analysis of gene expression time-series (BATS). We compare on both simulated and experimental data showing that the proposed approach considerably outperforms the current state of the art.

**Conclusions:** Gaussian processes offer an attractive trade-off between efficiency and usability for the analysis of microarray time series. The Gaussian process framework offers a natural way of handling biological replicates and missing values and provides confidence intervals along the estimated curves of gene expression. Therefore, we believe Gaussian processes should be a standard tool in the analysis of gene expression time series.

## Background

Gene expression profiles give a snapshot of mRNA concentration levels as encoded by the genes of an organism under given experimental conditions. Early studies of this data often focused on a single point in time which biologists assumed to be critical along the gene regulation process after the perturbation. However, the *static* nature of such experiments severely restricts the inferences that can be made about the underlying dynamical system.

With the decreasing cost of gene expression microarrays time series experiments have become commonplace giving a far broader picture of the gene regulation process. Such time series are often irregularly sampled and may involve differing numbers of replicates at each time point [1]. The experimental conditions under which

gene expression measurements are taken cannot be perfectly controlled leading the signals of interest to be corrupted by noise, either of biological origin or arising through the measurement process.

Primary analysis of gene expression profiles is often dominated by methods targeted at *static* experiments, i. e. gene expression measured on a single time-point, that treat time as an additional experimental factor [1-6]. However, were possible, it would seem sensible to consider methods that can account for the special nature of time course data. Such methods can take advantage of the particular statistical constraints that are imposed on data that is naturally ordered [7-12].

The analysis of gene expression microarray time-series has been a stepping stone to important problems in systems biology such as the genome-wide identification of direct targets of transcription factors [13,14] and the full reconstruction of gene regulatory networks [15,16]. A more comprehensive review on the motivations and

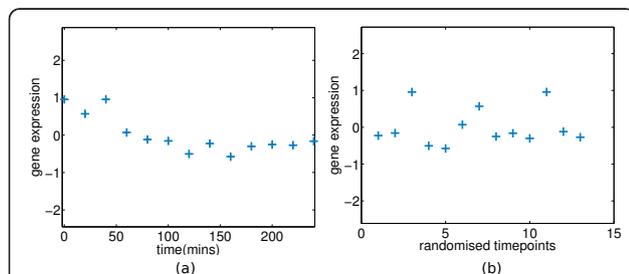
\* Correspondence: A.Kalaitzis@sheffield.ac.uk; N.Lawrence@dcs.shef.ac.uk  
The Sheffield Institute for Translational Neuroscience, 385A Glossop Road, Sheffield, S10 2HQ, UK

methods of analysis of time-course gene expression data can be found in [17].

### Testing for Expression

A primary stage of analysis is to characterize the activity of each gene in an experiment. Removing inactive or *quiet* genes (genes which show negligible changes in mRNA concentration levels in response to treatments/perturbations) allows the focus to dwell on genes that have responded to treatment. We can consider two experimental set ups. Firstly, we may be attempting to measure the absolute level of gene expression (for example using Affymetrix GeneChip microarrays). In this case a quiet gene would be one whose expression level is indistinguishable from noise. Alternatively, we might be hybridizing two samples to the same array and quantifying the ratio of the expression levels. Here a quiet gene would be one which is showing a similar response in both hybridized samples. In either case we consider such expression profiles will consist principally of *noise*. Removing such genes will often have benign effects later in the processing pipeline. However, mistaken removal of profiles can clearly compromise any further downstream analysis. If the temporal nature of the data is ignored, our ability to detect such phenomena can be severely compromised. An example can be seen in Figure 1, where the temporal information is removed from an experimental profile by randomly reordering its expression samples. Disregarding the temporal correlation between measurements, hinders our ability to assess the profile due to critical inherent traits of the signal being lost such as the speed and scale of variation.

Failure to capture the signal in a profile, irrespective of the amount of embedded noise, may be partially due to *temporal aggregation* effects, meaning that the coarse sampling of gene expression or the sampling rates do not match the natural rates of change in mRNA



**Figure 1** Temporal information removed from the profile of gene *Cyp1b1* in the experimental mouse data. **(a)** The centred profile of the gene *Cyp1b1* (probeID 1416612\_at in the *GSE10562* dataset). The blue crosses represent zero-mean hybridised gene expression in time of measurement (log2 ratios between treatment and control). **(b)** The same profile with its timepoints randomised.

concentrations [18]. For these reasons, the classification scheme of differential expression in this paper is focused on reaching a high *true positive rate* (TPR, *sensitivity* or *recall*) and is to serve as a pre-processing tool prior to more involved analysis of time-course microarray data. In this work we distinguish between *two-sample* testing and experiments where *control* and *treated* cases are directly-hybridized on the microarray (For brevity, we shall refer to experiments with such setups as *one-sample testing*). The *two-sample* setup is a common experimental setup in which two groups of sample replicates are used [13,19]; one being under the treatment effect of interest and the other being the control group, so to recover the most active genes under a treatment one may be interested in testing for the statistical significance of a treated profile being differentially expressed with respect to its control counterpart. Other studies use data from a *one-sample* setup [11,12], in which the *control* and *treated* cases are directly hybridized on a microarray and the measurements are normalized log fold-changes between the two output channels of the microarray [20], so the analogous goal is to test for the statistical significance of having a non-zero signal.

A recent significant contribution in regards to the estimation and ranking of differential expression of time-series in a *one-sample* setup is a hierarchical Bayesian model for the analysis of gene expression time-series (BATS) [11,12] which offers fast computations through exact equations of Bayesian inference, but makes a considerable number of prior biological assumptions to accomplish this (cf. **Simulated data**).

### Gene Expression Analysis with Gaussian Processes

*Gaussian processes* (GP) [21,22] offer an easy to implement approach to quantifying the true signal and noise embedded in a gene expression time-series, and thus allow us to rank the differential expression of the gene profile. A Gaussian process is the natural generalisation of a multivariate Gaussian distribution to a Gaussian distribution over a *specific family of functions* – a family defined by a *covariance function* or *kernel*, i.e. a metric of similarity between data-points (Roughly speaking, if we also view a function as a vector with an infinite number of components, then that function can be represented as a point in an infinite-dimensional *space of a specific family of functions* and a Gaussian process as an infinite-dimensional Gaussian distribution over that space).

In the context of expression trajectory estimation, a Gaussian process coupled with the *squared-exponential* covariance function (or *radial basis function*, RBF) – a standard covariance function used in regression tasks – makes the reasonable assumption that the underlying true signal in a profile is a *smooth* function [23], i.e. a

function with an infinite degree of differentiability. This property endows the GP with a large degree of flexibility in capturing the underlying signals without imposing strong modeling assumptions (e.g. number of basis functions) but may also erroneously pick up spurious patterns (false positives) should the time-course profiles suffer from temporal aggregation. From a generative viewpoint, the profiles are assumed to have been corrupted by additive white Gaussian noise. This property makes the GP an attractive tool for bootstrapping simulated biological replicates [24].

In a different context, Gaussian process priors have been used for modeling transcriptional regulation. For example in [25], while using the time-course expression of a-priori known direct targets (genes) of a transcription-factor, the authors went one step further and inferred the concentration rates of the transcription-factor protein itself and [26] extended the same model for the case of regulatory repression. The ever-lingering issue of outliers in time series is still critical, but is not addressed here as there is significant literature on this issue in the context of GP regression, which is complementary to this work.

For example [19,27] developed a probabilistic model using Gaussian processes with a robust noise model specialised for two-sample testing to detect *intervals* of differential expression, whereas the present work optionally focuses on *one-sample* testing, to rank the differential expression and ultimately detect *quiet/active* genes. Other examples can also be easily applied; [28] use a Student-*t* distribution as the robust noise model in the regression framework along with variational approximations to make inference tractable, and [29] employ a Student-*t* observation model with Laplace approximations for inference. The standard GP regression framework is straightforward to use here with a limited need for manual tweaking of a few hyper-parameters. We describe the GP framework, as used here for regression, in more detail in the **Methods** section.

## Results and Discussion

We apply standard Gaussian process (GP) regression and the Bayesian hierarchical model for the analysis of time-series (BATS) on two in-silico datasets simulated by BATS and GPs, and on one experimental dataset coming from a study on primary mouse keratinocytes with an induced activation of the TRP63 transcription factor, for which a reverse-engineering algorithm was developed (TSNI: time-series network identification) to infer the direct targets of TRP63 [13].

We assume that each gene expression profile can be categorized as either quiet or differentially expressed. We consider algorithms that provide a rank ordering of the profiles according to which is most likely to be non-

quiet (or differentially expressed). Given ground truth we can then evaluate the quality of such a ranking and compare different algorithms. We make use of *receiver operating characteristic* curves (ROC curves) to evaluate the algorithms. These curves plot the *false positive rate* on the horizontal axis, versus the *true positive rate* on the vertical axis; i.e. the percentage of the total negatives (non-differentially expressed profiles) erroneously classified as positives (differentially expressed) versus the percentage of the total positives correctly classified as positives.

From the output of each model a ranking of differential expression is produced and assessed with ROC curves to quantify how well in accordance to each of the three ground truths (BATS-sampled, GP-sampled, TSNI-experimental) the method performs. The BATS model can employ three different noise models, where the marginal distribution of the error is assumed to be either Gaussian, Student-*t* or double exponential respectively. For the following comparisons we plot four ROC curves, one for each noise model of BATS and one for the GP. We demonstrate that the ranking of the GP framework outperforms that of BATS with respect to the TSNI ranking on the experimental data and on GP-sampled profiles.

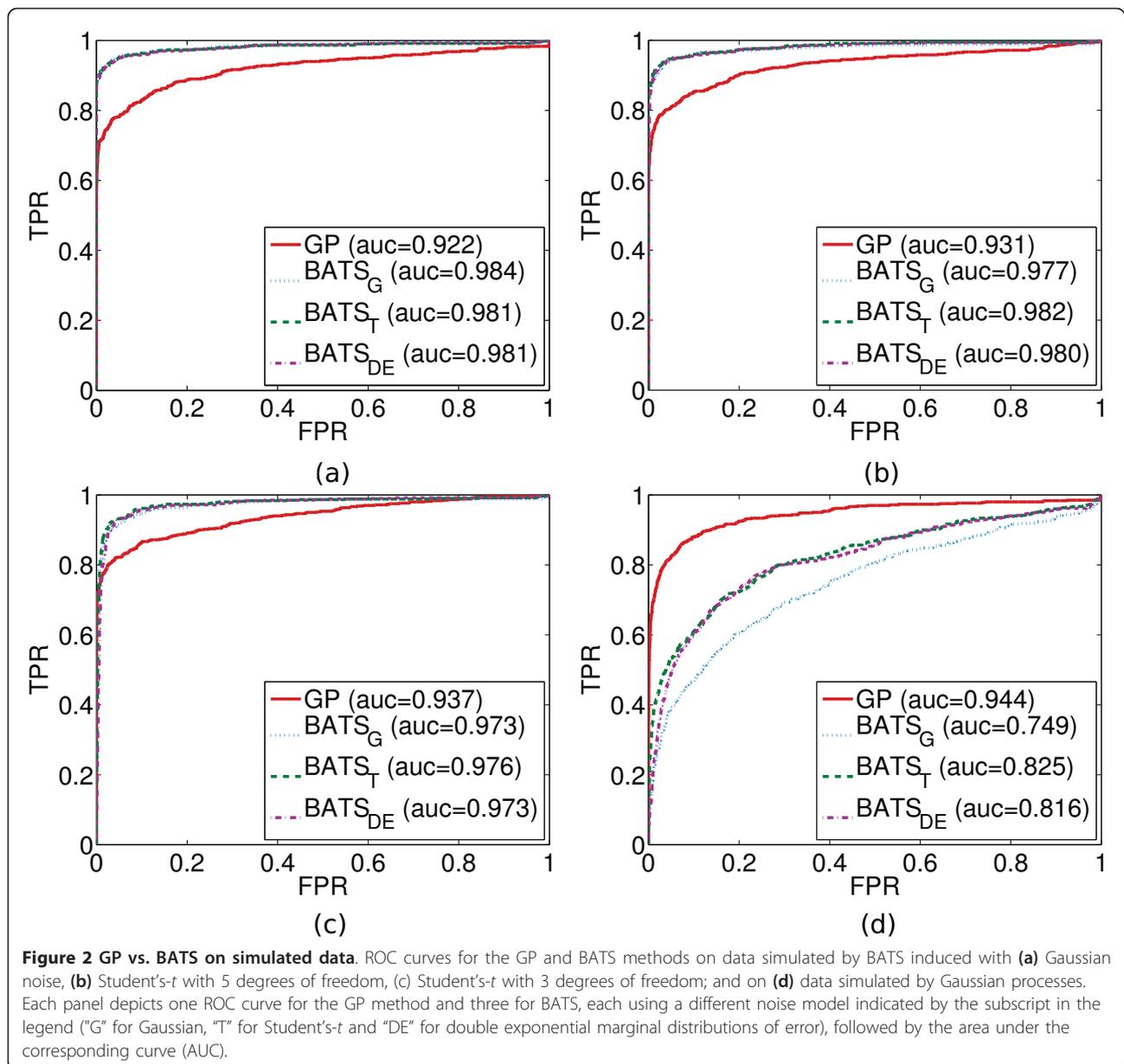
### Simulated data

The first set of in-silico profiles are simulated by the BATS software <http://www.na.iac.cnr.it/bats/> in accordance to the guidelines given in [12]. In BATS [11] each time-course profile is assumed to be generated by a function expanded in an orthonormal basis (Legendre or Fourier) plus noise. The number of bases and their coefficients, are estimated with analytic computations in a fully Bayesian manner. Thus the global estimand for every gene expression trajectory is the linear combination of some number of bases whose coefficients are estimated by a posterior distribution. In addition, the BATS framework allows various types of non-Gaussian noise models.

### BATS simulation

We reproduce one instantiation of the simulations performed in [11]; specifically, three sets of  $N = 8000$  profiles, of  $n = 11$  timepoints and  $k_i^j = 2$  replicates, for  $i = 1; \dots, N, j = 1, \dots, n$  except  $k_i^{2,5,7} = 3$ , according to the model defined in [11, sec. 2.2]. In each of the three sets of profiles, 600 out of 8000 are randomly chosen to be differentially expressed (labeled as “1” in the ground truth) and simulated as a sum of an orthonormal basis of Legendre polynomials with additive i.i.d.(identically and independently distributed) noise.

The other 7400 non-differentially expressed profiles (labeled as “0” in the ground truth) are essentially zero



functions with additive i.i.d. noise. The three simulated datasets are induced with different kinds of i.i.d. noise; respectively, Gaussian  $N(0, \sigma^2)$ , Student-*t* distributed with 5(T(5)) and 3 (T(3)) degrees of freedom. Figure 2 (a, b, c) illustrates the comparison on the BATS-sampled data with all three kinds of induced noise.

#### GP simulation

In a similar setup, the second in-silico dataset consists of 8000 profiles sampled from Gaussian processes, with the same number of replicates and time-points, among which 600 were setup as differentially expressed. To generate a differentially expressed profile, each of the *hyperparameters* of the RBF covariance function, namely the *characteristic lengthscale*, *signal variance* and *noise*

*variance* (cf. **Methods**) is sampled from separate Gamma distributions. The three Gamma distributions are fitted to sets of their corresponding hyperparameters, which are observed for the true positive profiles under a near zero FPR during the first test on BATS-generated profiles. In this way, we attempt to resemble the behaviour of the BATS-sampled profiles. Table 1 lists the parameters of the three fitted Gamma distributions.

The other 7400 non-differentially expressed profiles are simply zero functions with additive white Gaussian noise of variance equal to the sum of two samples from the Gamma distribution for the *signal variance* and the *noise variance*. This addition serves to create a non-

**Table 1 Parameters of the Gamma distributions for sampling the RBF-hyperparameters**

		Sampling Gamma distribution $\Gamma(a, b)$	
		$a$ (scale)	$b$ (shape)
Sampled RBF-Hyperparameters	$\ell^2$ (characteristic lengthscale)	1.4	5.7
	$\sigma_f^2$ (signal variance)	2.76	0.2
	$\sigma_n^2$ (noise variance)	23	0.008

These are the parameters of the Gamma distributions from which we sample the RBF- hyperparameters. For example, the characteristic lengthscale is sampled from a Gamma with scale 1.4 and shape 5.7. The hyperparameters are then used in the RBF covariance function to sample/simulate a profile from the Gaussian process.

differentiated profile of comparative scale to the differentiated ones, but nonetheless of completely random nature. Figure 2(d) illustrates the comparison on the GP-sampled data.

### Experimental data

We apply the standard GP regression framework and BATS on an experimental dataset coming from a study on primary mouse keratinocytes with an induced activation of the TRP63 transcription factor (GEO-accession number [GEOdataset:GSE10562]), where a reverse-engineering algorithm was developed (TSNI: time-series network identification) to infer the direct targets of TRP63 [13]. In that study, 786 out of 22690 gene reporters were chosen based on the area under their curves, and ranked by TSNI according to the probability of belonging to direct targets of TRP63. The ranking list was published in a supplementary file available for download

(genome.cshlp.org/content/suppl/2008/05/05/gr.073601.107.DC1/DellaGatta\_SupTable1.xls) and used here as a *noisy ground truth*. We pre-process the data with the robust multi-array average (RMA) expression measure [30], implemented in the “affy” R-package.

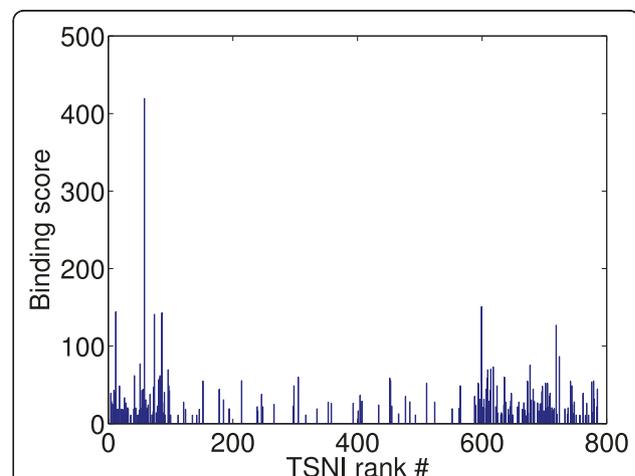
We label the top 100 position of the TSNI ranking as “1” in the ground truth as they are the most likely to be direct targets of the TRP63 transcription factor and because the *binding scores* (computed as the sum of  $-\log_2$  of  $p$ -values of all TRP63-binding regions identified by ChIP-chip experiments) are most densely distributed amongst the first 100 positions, see Figure 3. Furthermore, in [13] these 100 positions were further validated by gene set enrichment analysis (GSEA) [31] to check if their up/down regulation patterns were correlated to genes that respond to TRP63 knock-downs in general. In summary, “the top 100 TSNI ranked transcripts are significantly enriched for the strongest binding sites” [13]. Figure 4 illustrates the comparison on the experimental data.

### Discussion

On BATS-sampled data, Figure 2(a, b, c), we observe that the change in the induced noise is barely noticeable

in regards to the performances of both methods and that BATS maintains its stable supremacy over the GP framework. This performance gap is partially due to the lack of a robust noise model for the GP (cf. **Conclusions**). Furthermore, there is a modeling bias in the underlying functions of the simulated profiles, which contain a finite small degree of differentiability (maximum degree of Legendre polynomial is 6). This puts the GP in a disadvantaged position as it models for (smooth) infinitely differentiable functions when its covariance function is a *squared exponential*. Consequently, for this simulated dataset the GP is more susceptible to capturing spurious patterns as they are more likely to lie within its modeling range, whereas for BATS modeling the polynomials with a limited degree acts as a safeguard against spurious patterns, most of which vary rapidly in time.

On GP-sampled data, Figure 2(d), we observe the reversal of the performance gap in favor of the GP framework while its performance is almost unaffected. The



**Figure 3 Distribution of binding scores along the TSNI ranking.**

By inspection, the distribution of the binding scores is mostly dense along the first 100 positions of the TSNI ranking. The authors in [13] only selected the top 100 genes and the bottom 200 genes to search for binding sites and thus showed that the top 100 genes have more binding sites than the bottom 200 genes.

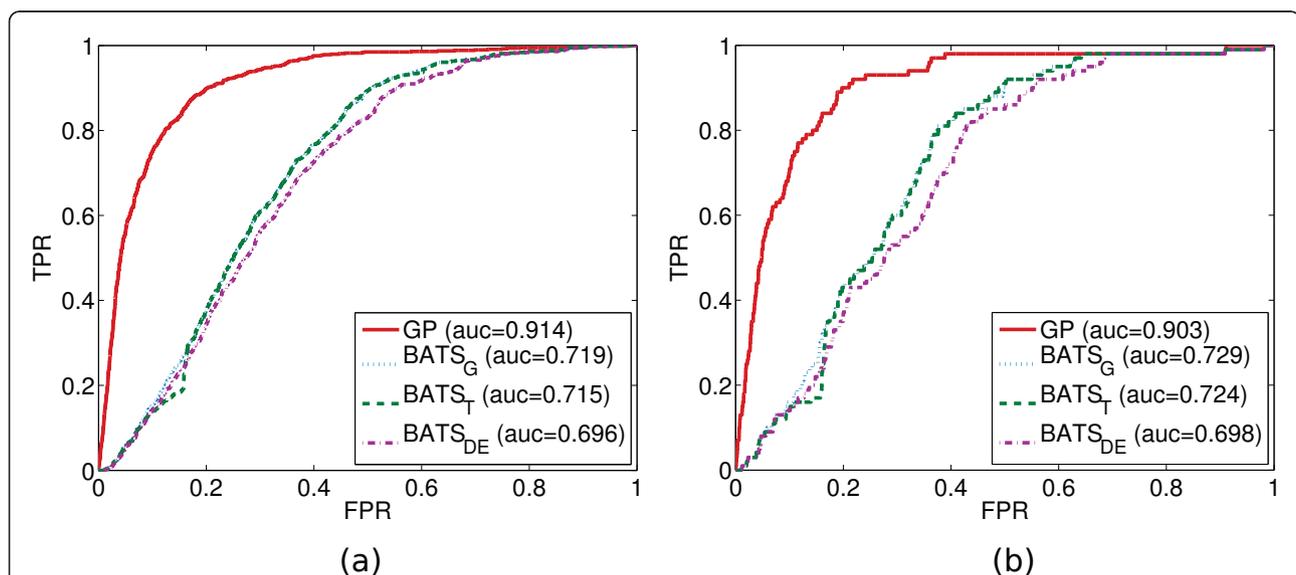
GP is still prone to non-differentially expressed profiles with spurious patterns and differentially expressed profiles with excessive noise. However, the limited polynomial degree of BATS proves to be inadequate for many of the GP-sampled functions and the two BATS variants with robust noise models ( $BATS_T$ ,  $BATS_{DE}$ ) only alleviate the problem slightly. In Figure 4 we observe the GP outperforming the Gaussian noise variant of BATS ( $BATS_G$ ) by a similar degree as in Figure 2(d). The experimental data are much more complex and apparently the robust BATS variants now offer no increase in performance. Since the ground truth focuses on the 100 most differentially expressed genes with respect to the induction of the TRP63 transcription factor, then these results indicate that the GP method of ranking presented here indeed highlights differentially expressed genes and that it naturally features an attractive degree of robustness against different kinds of noise.

### Conclusions

We presented an approach to estimating the continuous trajectory of gene expression time-series from microarray data through *Gaussian process* (GP) regression and ranking the differential expression of each profile via a log-ratio of marginal likelihoods of two GPs, each one representing the hypothesis of differential and non-differential expression respectively. We compared our method to a recent Bayesian hierarchical model (BATS) via ROC curves on data simulated by BATS and GPs and experimental data. Each evaluation was made on

the basis of matched percentages to a ground truth - a binary vector which labeled the profiles in a dataset as differentially expressed or not. The experimental data were taken from a previous study on primary mouse keratinocytes and the top 100 genes of its ranking were used here as the noisy ground truth for the purposes of assessment. The GP framework significantly outperformed BATS on experimental and GP-sampled data and the results showed that standard GP regression can be regarded as a serious competitor in evaluating the continuous trajectories of gene expression and ranking its differential expression.

This ranking scheme presented here is reminiscent of the work in [19] on *two-sample* data (separate time-course profiles for each treatment), where the two competing hypotheses are represented in a graphical model of two different generative models connected with a *gating* scheme; one where the two profiles of the gene reporter are assumed to be generated by two different GPs, and thus the gene is *differentially expressed* across the two treatments, and one where the two profiles are assumed to be generated by the same GP, and thus the gene is *non-differentially expressed*. The gating network serves to *switch* between the two generative models, in time, to detect *intervals* of differential expression and thus allow biologists to draw conclusions about the propagation of a perturbation in a gene regulatory network. Instead, the issue presented in this paper is more basic and so is the methodology to deal with it. However, we note that the robust mechanisms against outliers used



**Figure 4 GP vs. BATS on experimental data.** ROC curves for the GP and BATS methods on experimental data from [13]. As in Figure 2, one ROC curve and the area under it (AUC) are depicted for the GP method and three for BATS, each using a different noise model indicated by the subscript in the legend. **(a)** Ground truth consists of 22690 labels among which only the 786 profiles chosen to be ranked by TSNI (based on the area under their curves) are labeled as "1", cf. **Experimental data**. **(b)** Same number of labels; here only the top 100 profiles ranked by TSNI are labeled as "1".

in [19,28,29] are complementary to this work and one should not hesitate to incorporate one into a framework similar to ours. Practicalities aside, this paper also introduces additional proof that Gaussian processes, naturally and without much engineering, fit to the analysis of gene expression time-series and that simplicity can still be preferred over the ever-increasing – but sometimes necessary – complexity of hierarchical Bayesian frameworks.

### Future work

A natural next step would be to add a robust noise mechanism in our framework. In this regard, fine examples can be found in [19,28,29]. Finally, an interesting biological question is about the potential periodicity of the underlying signal in a gene expression profile. In this regard a different of kind stationary covariance function, the *periodic* covariance function [22], can fit a time-series generated by an periodic process and thus its lengthscale hyperparameter can be interpreted as its cycle.

### Methods

As we mentioned earlier, analysing time-course microarray data by means of Gaussian process (GP) regression is not a new idea (cf. **Background**). In this section we review the methodology to estimating the continuous trajectory of a gene expression by GP regression and subsequently describe a likelihood-ratio approach to ranking the differential expression of its profile. The following content is based on the key components of GP theory as described in [21,22].

#### The Gaussian process model

The idea is to treat trajectory estimation given the observations (gene expression time-series) as an interpolation problem on functions of one dimension. By assuming the observations have Gaussian-distributed noise, the computations for prediction become tractable and involve only the manipulation of linear algebra rules.

#### A finite parametric model

We begin the derivation of the GP regression model by defining a standard *linear regression* model (a more concrete example of such a model is for  $\varphi = (1, x, x^2)^\top$ , i.e. a line mapped to a quadratic curve)

$$f(x) = \phi^\top \mathbf{w}, \quad y = f(x) + \varepsilon, \quad (1)$$

where gene expression measurements in time  $\mathbf{y} = \{y_n\}_{n=1..N}$  are contaminated with white Gaussian noise and the inputs (of time) are mapped to a feature space  $\Phi = \{\varphi(x_n)\}_{n=1..N}$ . Furthermore, if we assume the noise to be i.i.d. (identically and independently distributed) as a

Gaussian with zero mean and variance  $\sigma_n^2$

$$\sim \mathcal{N}(0, \sigma_n^2), \quad (2)$$

then the probability density of the observations given the inputs and parameters (*data likelihood*) is Gaussian-distributed

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \mathbf{w}) &= \prod_{i=1}^n p(y_i|x_i, \mathbf{w}) \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(y_i - \phi_i^\top \mathbf{w})^2}{2\sigma_n^2}\right) \\ &= \frac{1}{(2\pi)^{n/2} |\Sigma_n|^{1/2}} \times \\ &\quad \exp\left(-\frac{1}{2}(\mathbf{y} - \Phi\mathbf{w})^\top \Sigma^{-1}(\mathbf{y} - \Phi\mathbf{w})\right) \\ &= \mathcal{N}(\Phi\mathbf{w}, \Sigma_n), \end{aligned} \quad (3)$$

Where  $\Sigma_n = \sigma_n^2 \mathbf{I}$ .

#### Introducing Bayesian methodology

Now turning to *Bayesian linear regression*, we wish to encode our initial beliefs about the parameters  $\mathbf{w}$  by specifying a zero mean, isotropic Gaussian distribution as a *prior* over the parameters

$$\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2). \quad (4)$$

By integrating the product of the *likelihood*  $\times$  *prior* with respect to the parameters, we get the *marginal likelihood*

$$p(\mathbf{y}|\mathbf{x}) = \int d\mathbf{w} p(\mathbf{y}|\mathbf{x}, \mathbf{w}) p(\mathbf{w}), \quad (5)$$

which is jointly Gaussian. Hence the *mean* and *covariance* of the *marginal* are

$$\langle \mathbf{y} \rangle = \langle \Phi\mathbf{w} \rangle + \langle \varepsilon \rangle = \mathbf{0} \quad (6)$$

$$\begin{aligned} \langle \mathbf{y}\mathbf{y}^\top \rangle &= \Phi \langle \mathbf{w}\mathbf{w}^\top \rangle \Phi^\top + \langle \varepsilon\varepsilon^\top \rangle \\ &= \sigma_w^2 \Phi \Phi^\top + \sigma_n^2 \mathbf{I} \\ &= \mathbf{K}_f + \sigma_n^2 \mathbf{I} = \mathbf{K}_y \end{aligned} \quad (7)$$

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \frac{1}{(2\pi)^{n/2} |\mathbf{K}_y|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{y}^\top \mathbf{K}_y^{-1} \mathbf{y}\right) \\ &= \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K}_y). \end{aligned} \quad (8)$$

By computing the *marginal likelihood* in eq. (8), we can *compare* or *rank* different models, without fear of *overfitting* on the data, or having to explicitly apply a *regulariser* to the *likelihood*; the *marginal likelihood* implicitly *penalises* too complex models [21, sec. 5.4].

Notice in eq. (7) how the structure of the covariance implies that choosing a different feature space  $\Phi$  results in a different  $\mathbf{K}_y$ . Whatever  $\mathbf{K}_y$  is, it must satisfy the following requirements to be a valid covariance matrix of the GP:

- **Kolmogorov consistency**, which is satisfied when  $K_{ij} = K(x_i, x_j)$  for some *covariance function*  $K$ , such that all possible  $\mathbf{K}$  are *positive semidefinite* ( $\mathbf{y}^\top \mathbf{K} \mathbf{y} \geq 0$ ).
- **Exchangeability**, which is satisfied when the data are i.i.d.. It means that the order in which the data become available has no impact on the *marginal distribution*, hence there is no need to hold out data from the training set for *validation* purposes (for measuring generalisation errors, etc.).

**Definition of a Gaussian process**

More formally, a *Gaussian process* is a stochastic process (or collection of random variables) over a feature space, such that the distribution  $p(y(x_1), y(x_2), \dots, y(x_n))$  of a function  $y(x)$ , for any finite set of points  $\{x_1, x_2, \dots, x_n\}$  mapped to that space, is Gaussian, and such that any of these Gaussian distributions is Kolmogorov consistent.

If we remove the noise term  $\sigma_n^2 \mathbf{I}$  from  $\mathbf{K}_y$  in eq. (7) we can have noiseless predictions of  $f(x)$  rather than  $y(x) = f(x) + \varepsilon$ . However, when dealing with finite parameter spaces  $\mathbf{K}_f$  may be *ill-conditioned* (cf. sec. SE derivation), so the noise term guarantees that  $\mathbf{K}_y$  will have full rank (and an inverse). Having said that, we now formulate the GP prior over the latent function values  $f$  by rewriting eq. (8) as

$$p(\mathbf{f}|\mathbf{x}) = \frac{1}{(2\pi)^{n/2} |\mathbf{K}_f|^{1/2}} \times \exp\left(-\frac{1}{2}(\mathbf{f} - \mathbf{m})^\top \mathbf{K}_f^{-1}(\mathbf{f} - \mathbf{m})\right), \quad (9)$$

or  $\mathbf{f}|\mathbf{x} \sim \mathcal{GP}(\mathbf{f}; m(x), K_f(x_i, x_j))$ ,

where the *mean function* (usually defined as the zero function) and the *covariance function* respectively are

$$m(x) = \langle f(x) \rangle, \quad (10)$$

$$K_f(x_i, x_j) = \langle (f(x_i) - m(x_i))(f(x_j) - m(x_j)) \rangle. \quad (11)$$

**The squared-exponential kernel**

In this paper we only use the univariate version of the squared-exponential (SE) kernel. But before embarking on its analysis, the reader should be aware of the existing wide variety of kernel families, and potential combinations of them. A comprehensive review of the

literature on covariance functions is found in [21, chap. 4].

**Derivation and interpretation of the SE kernel**

In the GP definition section we mentioned the possibility of an *ill-conditioned* covariance matrix. In the case of a finite parametric model (as in eq. (1)),  $\mathbf{K}_f$  can have at most as many non-zero eigenvalues as the number of parameters in the model. Hence for any problem of any given size, the matrix is non-invertible. Ensuring  $\mathbf{K}_f$  is not ill-conditioned, involves adding the diagonal noise term to the covariance. In an infinite-dimensional feature space, one would not have to worry about this issue as the features are integrated out and the covariance between datapoints is no longer expressed in terms of the features but by a *covariance function*. As demonstrated in [22, sec.45.3] and [21, sec.4.2.1], with an example of a one-dimensional dataset, we express the covariance matrix  $\mathbf{K}_f$  in terms of the features  $\Phi$

$$K_{ij} = \sigma_w^2 \sum_h \phi_h(x_i) \phi_h(x_j), \quad (12)$$

then by considering a feature space defined by *radial basis functions* and integrating with respect to their centers  $h$ , eq. (12) becomes

$$\begin{aligned} K(x_i, x_j) &= \lim_{N \rightarrow \infty} \frac{\sigma_w^2}{N} \sum_{h=1}^N \phi_h(x_i) \phi_h(x_j) \\ &= S \int_{-\infty}^{\infty} dh \phi_h(x_i) \phi_h(x_j) \\ &= S \int_{-\infty}^{\infty} dh \exp\left(-\frac{(x_i - h)^2}{2r^2}\right) \exp\left(-\frac{(x_j - h)^2}{2r^2}\right) \\ &= \sqrt{\pi r^2} S \exp\left(-\frac{(x_i - x_j)^2}{4r^2}\right), \end{aligned} \quad (13)$$

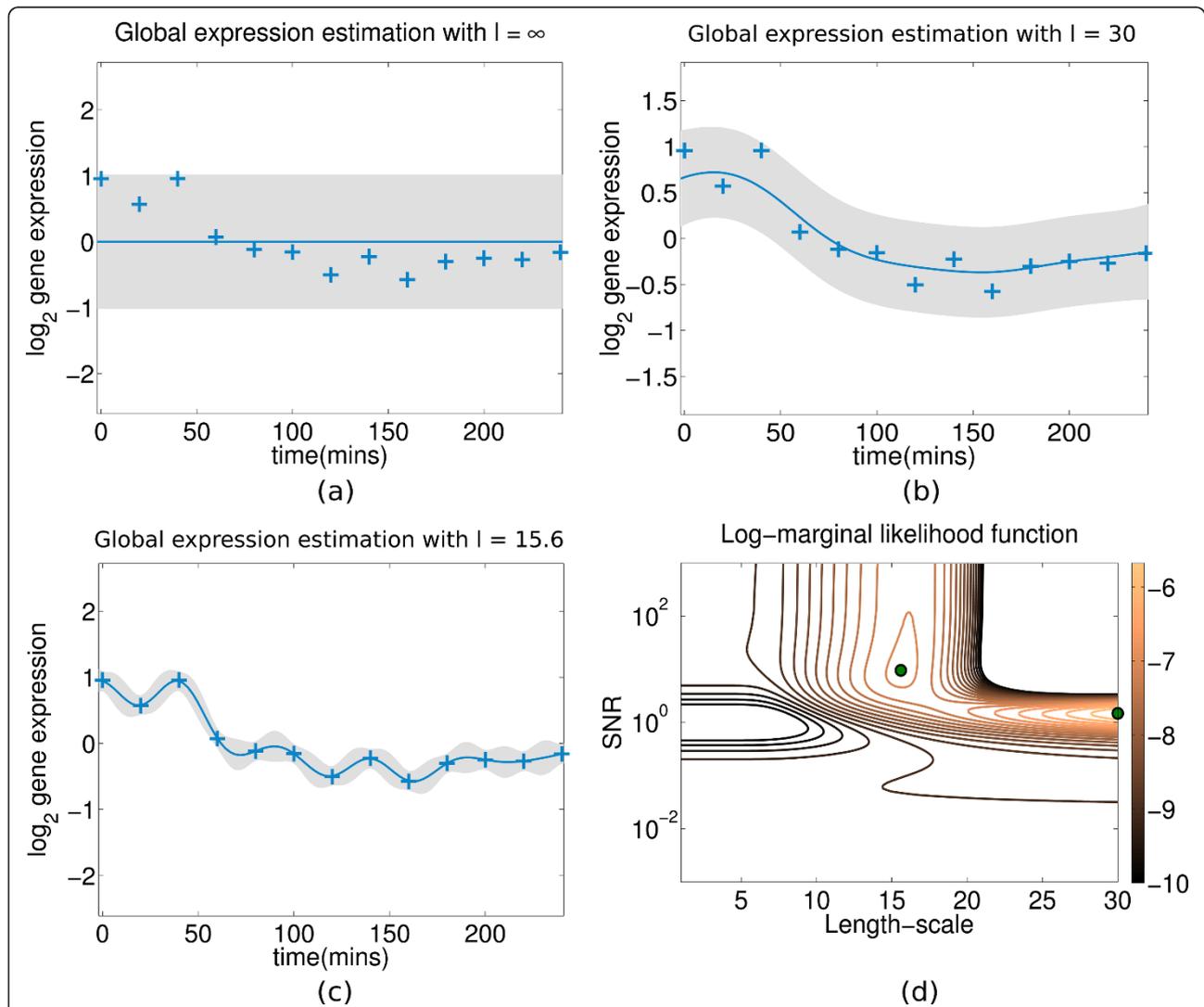
where one ends up with a smooth (infinitely differentiable) function on an infinite-dimensional space of (radial basis function) features. Taking the constant out front as a *signal variance*  $\sigma_f^2$  and squaring the exponential gives rise to the standard form of the *univariate squared-exponential* (SE) covariance function. The *noisy univariate* SE kernel – the one used in this paper is

$$K_y(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^2}{2\ell^2}\right) + \sigma_n^2 \delta_{ij}. \quad (14)$$

The SE is a *stationary* kernel, i.e. it is a function of  $d = x_i - x_j$  which makes it *translation invariant* in time.  $\delta_{ij}$  is the *Kronecker delta* function which is unity when  $i = j$  and zero otherwise and  $\ell^2$  is the *characteristic lengthscale* which specifies the distance beyond which any two inputs  $(x_i, x_j)$  become uncorrelated. In effect,

the lengthscale  $\ell^2$  governs the amount that  $f$  varies along the input (time). A small lengthscale means that  $f$  varies rapidly along time, and a very large lengthscale means that  $f$  behaves almost as a constant function, see Figure 5. This parameterisation of the SE kernel becomes very powerful when combined with *hyperparameter adaptation*, as described in a following section.

Other adapted hyperparameters include the *signal variance*  $\sigma_f^2$  which is a vertical scale of function variation and the *noise variance*  $\sigma_n^2$  (introduced in eq. (2)) which is not a hyperparameter of the SE itself, but unless we consider it as a constant in the *noisy* case, its adaptation can give different explanations about the latent function that generates the data.



**Figure 5 Gaussian process fit on expression profile of gene Cyp1b1 in the experimental mouse data.** Figure 5: A GP fitted on the centred profile of the gene *Cyp1b1* (probeID 1416612\_at in the *GSE10562* dataset) with different settings of the lengthscale hyperparameter  $\ell^2$ . The blue crosses represent zero-mean hybridised gene expression in time ( $\log_2$  ratios between treatment and control) and the shaded area indicates the point-wise mean plus/minus two times the standard deviation (95% confidence region). **(a)** Mean function is constant as  $\ell^2 \rightarrow \infty$  (0 inverse lengthscale in eq. (14)) and all of the observed data variance is attributed to noise ( $\sigma_n^2$ ). **(b)** The lengthscale is manually set to a local-optimum large value ( $\ell^2 = 30$ ) and thus the mean function roughly fits the data-points. The observed data variance is equally attributed to signal ( $\sigma_f^2$ ) and noise ( $\sigma_n^2$ ). Consequently, the GP features high uncertainty in its predictive curve. **(c)** The lengthscale is manually set to a local-optimum small value ( $\ell^2 = 15.6$ ) and thus the mean function tightly fits the data-points with high certainty. The interpretation from the covariance function in this case is that the profile contains a minimal amount of noise and that most of the observed data variance is attributed to the underlying signal ( $\sigma_f^2$ ). **(d)** The contour of the corresponding LML function plotted by an exhaustive search of  $\ell^2$  and SNR values. The two main local-optima are indicated by the green dots and a third optimum that corresponds to the 1st panel appears almost as flat in the contour and its vicinity encompasses the whole lengthscale axis for very small values of SNR (i.e. given that  $\text{SNR} \approx 0$ , the lengthscale is trivial).

One can also combine covariance functions as long as they are *positive-definite*. Examples of valid combined covariance functions include the *sum* and *convolution* of two covariance functions. In fact, eq. (14) is a combined sum of the SE kernel with the covariance function of *isotropic Gaussian noise*.

### Gaussian process prediction

To interpolate the trajectory of gene expression at non-sampled time-points, as illustrated in Figure 5, we infer a function value  $f_*$  at a new input (non-sampled time-point)  $x_*$ , given the knowledge of function estimates  $\mathbf{f}$  at known time-points  $\mathbf{x}$ . The joint distribution  $p(f_*, \mathbf{f})$  is Gaussian, hence the conditional distribution  $p(f_* | \mathbf{f})$  is also Gaussian. In this section we consider predictions using noisy observations; we know the noise is Gaussian too, so the noisy conditional distribution does not differ. By Bayes' rule

$$p(f_* | \mathbf{y}) = \frac{p(f_*, \mathbf{y})}{p(\mathbf{y})}, \quad (15)$$

where the Gaussian process prior over the noisy observations is

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{0}, \text{cov}(\mathbf{y})) = \mathcal{N}(\mathbf{0}, K_f(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I}) \quad (16)$$

### Predictive equations for GP regression

We start by defining the *mean function* and the *covariance* between a new time-point  $x_*$  and each of the  $i^{\text{th}}$  known time-points, where  $i = 1..N$

$$\mathbf{m}_i = m(\mathbf{x}_i), \quad (17)$$

$$(\mathbf{k}_*)_i = K_f(\mathbf{x}_i, \mathbf{x}_*). \quad (18)$$

For every new time-point a new vector  $\mathbf{k}_*$  is *concatenated* as an additional row and column to the covariance matrix  $\mathbf{K}_C$  to give

$$\mathbf{K}_{C+1} = \begin{bmatrix} [\mathbf{K}_C] & [\mathbf{k}_*] \\ [\mathbf{k}_*^\top] & [\kappa] \end{bmatrix}, \quad (19)$$

where  $C = N..N_*$  is incremented with every new  $\mathbf{k}_*$  added to  $\mathbf{K}_C$ . By considering a zero mean function and eq. (19), the joint distribution  $p(f_*, \mathbf{y})$  from eq. (15) can be computed

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} K_f(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} & K_f(\mathbf{X}, \mathbf{X}_*) \\ K_f(\mathbf{X}_*, \mathbf{X}) & K_f(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \right). \quad (20)$$

Finally, to derive the *predictive mean* and *covariance* of the posterior distribution from eq. (15) we use the Gaussian identities presented in [21, sec.A.2]. These are

the predictive equations for *GP regression* of a single new time-point

$$p(f_* | \mathbf{y}) \sim \mathcal{N}(f_*; m_*, \text{var}(f_*)), \quad \text{where} \quad (21)$$

$$m_* = \mathbf{k}_*^\top (K_f + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \quad (22)$$

$$\text{var}(f_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (K_f + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_*, \quad (23)$$

and  $K_f = K_f(\mathbf{x}, \mathbf{x})$ . These equations can be generalised easily for the prediction of function values at multiple new time-points by augmenting  $\mathbf{k}_*$  with more columns and  $k(\mathbf{x}^*, \mathbf{x}^*)$  with more components, one for each new time-point  $\mathbf{x}^*$ .

### Hyperparameter learning

Given the SE covariance function, one can learn the hyperparameters from the data by optimising the log-marginal likelihood function of the GP. In general, a non-parametric model such as the GP can employ a variety of kernel families whose hyperparameters can be adapted with respect to the underlying intensity and frequency of the local signal structure, and interpolate it in a probabilistic fashion (i.e. while quantifying the uncertainty of prediction). The SE kernel allows one to give intuitive interpretations of the adapted hyperparameters, especially for one-dimensional data such as a gene expression time-series, see Figure 5 for interpretations of various local-optima.

### Optimising the marginal likelihood

In the context of GP models the marginal likelihood results from the marginalisation over function values  $\mathbf{f}$

$$p(\mathbf{y} | \mathbf{x}) = \int d\mathbf{f} p(\mathbf{y} | \mathbf{f}, \mathbf{x}) p(\mathbf{f} | \mathbf{x}), \quad (24)$$

where the *GP prior*  $p(\mathbf{f} | \mathbf{x})$  is given in eq. (9) and the likelihood is a factorised Gaussian  $\mathbf{y} | \mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma_n^2 \mathbf{I})$ . The integral can be evaluated analytically [21, sec. A.2] to give the *log-marginal likelihood* (LML, it is common practice to take the log of the antiderivative for the sake of numerical stability, as it yields a sum instead of a product)

$$\ln p(\mathbf{y} | \mathbf{x}, \theta) = -\frac{1}{2} \mathbf{y}^\top \mathbf{K}_y^{-1} \mathbf{y} - \frac{1}{2} \ln |\mathbf{K}_y| - \frac{n}{2} \ln 2\pi, \quad (25)$$

where  $\mathbf{K}_y = \mathbf{K}_f + \sigma_n^2 \mathbf{I}$

We notice that the marginal here is explicitly conditioned on  $\theta$  (*hyperparameters*) to emphasise that it is a function of the hyperparameters through  $\mathbf{K}_f$ . To optimise the marginal likelihood we take the partial derivatives of the LML with respect to the hyperparameters

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\theta}} \ln p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) &= \frac{1}{2} \boldsymbol{\alpha}^\top \frac{\partial \mathbf{K}_y}{\partial \boldsymbol{\theta}} \boldsymbol{\alpha} - \frac{1}{2} \text{tr} \left( \mathbf{K}_y^{-1} \frac{\partial \mathbf{K}_y}{\partial \boldsymbol{\theta}} \right) \\ &= \frac{1}{2} \text{tr} \left( (\boldsymbol{\alpha} \boldsymbol{\alpha}^\top - \mathbf{K}_y^{-1}) \frac{\partial \mathbf{K}_y}{\partial \boldsymbol{\theta}} \right), \\ \text{where } \boldsymbol{\alpha} &= \mathbf{K}_y^{-1} \mathbf{y} \text{ and} \\ \frac{\partial \mathbf{K}_y}{\partial \boldsymbol{\theta}} &= \left( \frac{\partial \mathbf{K}_y}{\partial \ell}, \frac{\partial \mathbf{K}_y}{\partial \sigma_f^2}, \frac{\partial \mathbf{K}_y}{\partial \sigma_n^2} \right)^\top \\ &= \left( \mathbf{K}_f \frac{(x_i - x_j)^2}{\ell^3}, \exp \left( -\frac{(x_i - x_j)^2}{2\ell^2} \right), \delta_{ij} \right)^\top. \end{aligned} \tag{26}$$

We use *scaled conjugate gradients* [32] – a standard optimisation scheme – to maximise the LML.

### Ranking with likelihood-ratios

Alternatively, one may choose to go “fully Bayesian” by placing a *hyper-prior* over the hyperparameters  $p(\boldsymbol{\theta} | \mathcal{H})$ , where  $\mathcal{H}$  represents some type of model, and compute a posterior over hyperparameters

$$p(\boldsymbol{\theta} | \mathbf{y}, \mathbf{x}, \mathcal{H}) = \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{H})p(\boldsymbol{\theta}|\mathcal{H})}{\int d\boldsymbol{\theta} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{H})p(\boldsymbol{\theta}|\mathcal{H})}, \tag{27}$$

based on some initial beliefs, such as the functions having large lengthscales, and optimise the marginal likelihood so that the optimum lengthscale tends to a large value, unless there is evidence to the contrary. Depending on the model  $\mathcal{H}$ , the integral in eq. (27) may be analytically intractable and thus one has to resort to approximating this quantity [33] (e.g. Laplace approximation) or using *Markov Chain Monte Carlo* (MCMC) methods to sample from the posterior distribution [34].

In the case where one is using different types of models (e.g. with different number of hyperparameters), a Bayesian-standard way of comparing between such two models is through Bayes factors [11,19,23] – ratios of the *integral* quantities in eq. (27)

$$K = \frac{\int d\boldsymbol{\theta}_1 p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_1, \mathcal{H}_1)p(\boldsymbol{\theta}_1|\mathcal{H}_1)}{\int d\boldsymbol{\theta}_2 p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_2, \mathcal{H}_2)p(\boldsymbol{\theta}_2|\mathcal{H}_2)}, \tag{28}$$

where the models  $\mathcal{H}$  usually represent two different hypotheses, namely  $\mathcal{H}_1$  - the profile has a significant underlying signal and thus it is truly differentially expressed and  $\mathcal{H}_2$  - there is no underlying signal in the profile and the observed gene expression is just the effect of random noise. The ranking is based on how likely  $\mathcal{H}_1$  in comparison to  $\mathcal{H}_2$ , given a profile.

In this paper we present a much simpler – but effective to the task – approach to ranking the differential expression of a profile. Instead of integrating out the hyperparameters, we approximate the Bayes factor with a log-ratio of marginal likelihoods (cf. eq. (25))

$$\ln \left( \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_2)}{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_1)} \right), \tag{29}$$

with each LML being a function of different instantiations of  $\boldsymbol{\theta}$ . We still maintain hypotheses  $\mathcal{H}_1$  and  $\mathcal{H}_2$  that represent the same notions explained above, but in our case they differ simply by configurations of  $\boldsymbol{\theta}$ . Specifically, on  $\mathcal{H}_1$  the hyperparameters are **fixed** to  $\boldsymbol{\theta}_1 = (\infty, 0; \text{var}(\mathbf{y}))^\top$  to encode a function constant in time ( $\ell^2 \rightarrow \infty$ ), with no underlying signal ( $\sigma_f^2 = 0$ ), which generates a time-series with a variance that can be solely explained by noise ( $\sigma_n^2 = \text{var}(\mathbf{y})$ ). Analogously, on  $\mathcal{H}_2$  the hyperparameters  $\boldsymbol{\theta}_2$  are initialised to encode a function that fluctuates in accordance to a typical significant profile (e.g.  $\ell^2 = 20$ ), with a distinct signal variance that solely explains the observed time-series variance ( $\sigma_f^2 = \text{var}(\mathbf{y})$ ) and with no noise ( $\sigma_n^2 = 0$ ).

### Local optima of the log-marginal likelihood (LML) function

These two configurations correspond to two points in the three-dimensional function that is the LML, both of which usually lie close to local-optimum solutions. This assumption can be verified, empirically, by exhaustively plotting the LML function for a number of profiles, see Figure 5. In case the LML contour differs for some profiles, more initialisation points should be used to ensure convergence to the maximum-likelihood solution. Because the configuration of the second hypothesis (no noise,  $\sigma_n^2 = 0$ ) is an extremely unlikely scenario, we let  $\boldsymbol{\theta}_2$  adapt to a given profile by optimising the LML function, as opposed to keeping it fixed like  $\boldsymbol{\theta}_1$ .

In most cases the LML (eq. (25)) is not convex. Multiple optima do not necessarily pose a threat here; depending on the data and as long as they have similar function values, multiple optima present alternative interpretations on the observations. To alleviate the problem of spurious local optimum solutions however, we make the following observation: when we explicitly restrict the signal variance hyperparameter ( $\sigma_f^2$ ) to low values during optimisation, we also implicitly restrict the noise variance hyperparameter ( $\sigma_n^2$ ) to large values. This occurs as the explanation of the observed data variance ( $\text{var}(\mathbf{y})$ ) is shared between the signal and noise variance hyperparameters, i.e.  $\text{var}(\mathbf{y}) \simeq \sigma_f^2 + \sigma_n^2$ . This dependency allows us to treat the three-dimension optimisation problem as a two-dimension problem, one of lengthscale  $\ell^2$  and one of signal-to-noise ratio  $\text{SNR} = \frac{\sigma_f^2}{\sigma_n^2}$  without fear of missing out an optima.

Figure 5 illustrates the marginal likelihood as a function of the characteristic lengthscale  $\ell^2$  and the SNR. It features two local optima, one for a small lengthscale and a high SNR, where the observed data are explained

with a relatively complex function and a small noise variance, and one optimum for a large lengthscale and a low SNR, where the data are explained by a simpler function with high noise variance. We also notice that the first optimum has a lower LML. This relates to the algebraic structure of the LML (eq. (25)); the first term (dot product) promotes data fitness and the second term (determinant) penalizes the complexity of the model [21, sec.5.4]. Overall, the LML function of the Gaussian process offers a good fitness-complexity trade-off without the need for additional regularisation. Optionally, one can use multiple initialisation points focusing on different non-infinite lengthscales to deal with the multiple local optima along the lengthscale axis, and pick the best solution (max LML) to represent the  $\mathcal{H}_1$  hypothesis in the likelihood-ratio during the ranking stage.

#### Source code

The source code for the GP regression framework is available in MATLAB code <http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/gp/> and as a package for the R statistical computing language <http://cran.r-project.org/web/packages/gptk/>. The routines for the estimation and ranking of the gene expression time-series are available upon request for both languages. The time needed to analyse the 22690 profiles in the experimental data, with only the basic two initialisation points of hyperparameters, is about 30 minutes on a desktop running Ubuntu 10.04 with a dual-core CPU at 2.8 GHz and 3.2 GiB of memory.

#### Acknowledgements

The authors would like to thank Diego di Bernardo for his useful feedback on the experimental data. Research was partially supported by a EPSRC Doctoral Training Award, the Department of Neuroscience, University of Sheffield and BBSRC (grant BB/H018123/2).

#### Authors' contributions

AAK designed and implemented the computational analysis and ranking scheme presented here, assessed the various methods and drafted the manuscript. NDL pre-processed the experimental data and wrote the original Gaussian process toolkit for MATLAB and AAK rewrote it for the R statistical language. Both AAK and NDL participated in interpreting the results and revising the manuscript. All authors read and approved the final manuscript.

Received: 18 January 2011 Accepted: 20 May 2011

Published: 20 May 2011

#### References

1. Lönnstedt I, Speed TP: **Replicated microarray data.** *Statistica Sinica* 2002, **12**:31-46.
2. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B: **Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization.** *Molecular biology of the cell* 1998, **9**(12):3273.
3. Friedman N, Lital M, Nachman I, Pe'er D: **Using Bayesian networks to analyze expression data.** *Journal of computational biology* 2000, **7**(3-4):601-620.
4. Dudoit S, Yang YH, Callow MJ, Speed TP: **Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments.** *Statistica sinica* 2002, **12**:111-140.
5. Kerr MK, Martin M, Churchill GA: **Analysis of variance for gene expression microarray data.** *Journal of Computational Biology* 2000, **7**(6):819-837.
6. Efron B, Tibshirani R, Storey JD, Tusher V: **Empirical Bayes analysis of a microarray experiment.** *Journal of the American Statistical Association* 2001, **96**(456):1151-1160.
7. Bar-Joseph Z, Gerber G, Simon I, Gifford DK, Jaakkola TS: **Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes.** *Proceedings of the National Academy of Sciences of the United States of America* 2003, **100**(18):10146.
8. Ernst J, Nau G, Bar-Joseph Z: **Clustering short time series gene expression data.** *Bioinformatics* 2005, **21**(Suppl 1):i159.
9. Storey JD, Xiao W, Leek JT, Tompkins RG, Davis RW: **Significance analysis of time course microarray experiments.** *Proceedings of the National Academy of Sciences of the United States of America* 2005, **102**(36):12837.
10. Tai YC, Speed TP: **A multivariate empirical Bayes statistic for replicated microarray time course data.** *The Annals of Statistics* 2006, **34**(5):2387-2412.
11. Angelini C, De Canditiis D, Mutarelli M, Pensky M: **A Bayesian approach to estimation and testing in time-course microarray experiments.** *Stat Appl Genet Mol Biol* 2007, **6**:24.
12. Angelini C, Cuttillo L, De Canditiis D, Mutarelli M, Pensky M: **BATS: a Bayesian user-friendly software for Analyzing Time Series microarray experiments.** *BMC bioinformatics* 2008, **9**:415.
13. Della Gatta G, Bansal M, Ambesi-Impiombato A, Antonini D, Missero C, di Bernardo D: **Direct targets of the TRP63 transcription factor revealed by a combination of gene expression profiling and reverse engineering.** *Genome research* 2008, **18**(6):939.
14. Honkela A, Girardot C, Gustafson EH, Liu YH, Furlong EEM, Lawrence ND, Rattray M: **Model-based method for transcription factor target identification with limited data.** *Proceedings of the National Academy of Sciences* 2010, **107**(17):7793.
15. Bansal M, Gatta GD, Di Bernardo D: **Inference of gene regulatory networks and compound mode of action from time course gene expression profiles.** *Bioinformatics* 2006, **22**(7):815.
16. Finkenstadt B, Heron EA, Komorowski M, Edwards K, Tang S, Harper CV, Davis JRE, White MRH, Millar AJ, Rand DA: **Reconstruction of transcriptional dynamics from gene reporter data using differential equations.** *Bioinformatics* 2008, **24**(24):2901.
17. Bar-Joseph Z: **Analyzing time series gene expression data.** *Bioinformatics* 2004, **20**(16):2493.
18. Bay SD, Chrisman L, Pohorille A, Shrager J: **Temporal aggregation bias and inference of causal regulatory networks.** *Journal of Computational Biology* 2004, **11**(5):971-985.
19. Stegle O, Denby KJ, Cooke EJ, Wild DL, Ghahramani Z, Borgwardt KM: **A robust Bayesian two-sample test for detecting intervals of differential gene expression in microarray time series.** *Journal of Computational Biology* 2010, **17**(3):355-367.
20. Schena M, Shalon D, Davis RW, Brown PO: **Quantitative monitoring of gene expression patterns with a complementary DNA microarray.** *Science* 1995, **270**(5235):467.
21. Rasmussen CE, Williams CKI: *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* The MIT Press; 2005.
22. MacKay DJC: *Gaussian Processes. Information theory, inference, and learning algorithms* Cambridge University Press; 2003, 535-548.
23. Yuan M: **Flexible temporal expression profile modelling using the Gaussian process.** *Computational statistics & data analysis* 2006, **51**(3):1754-1764.
24. Kirk PDW, Stumpf MPH: **Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data.** *Bioinformatics* 2009, **25**(10):1300.
25. Lawrence ND, Sanguinetti G, Rattray M: **Modelling transcriptional regulation using Gaussian processes.** *Advances in Neural Information Processing Systems* 2007, **19**:785.
26. Gao P, Honkela A, Rattray M, Lawrence ND: **Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities.** *Bioinformatics* 2008, **24**(16):i70.
27. Stegle O, Denby KJ, Wild L, McHattie S, Meade A, Ghahramani Z, Borgwardt KM: **Discovering temporal patterns of differential gene expression in microarray time series.** *In GCB* 2009, 133-142.

28. Tipping ME, Lawrence ND: **Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis.** *Neurocomputing* 2005, **69**(1-3):123-141.
29. Vanhatalo J, Jylänki P, Vehtari A: **Gaussian process regression with Student-t likelihood.** *Neural Information Processing System, Citeseer* 2009.
30. Irizarry RA, Hobbs B, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, Speed TP: **Exploration, normalization, and summaries of high density oligonucleotide array probe level data.** *Biostatistics* 2003, **4**(2):249.
31. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP: **Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles.** *Proceedings of the National Academy of Sciences* 2005, **102**(43):15545.
32. Möller MF: **A scaled conjugate gradient algorithm for fast supervised learning.** *Neural networks* 1993, **6**(4):525-533.
33. MacKay DJC: **Comparison of approximate methods for handling hyperparameters.** *Neural Computation* 1999, **11**(5):1035-1068.
34. Neal RM: **Monte Carlo implementation of Gaussian process models for Bayesian regression and classification.** *Arxiv preprint physics/9701026* 1997.

doi:10.1186/1471-2105-12-180

**Cite this article as:** Kalaitzis and Lawrence: **A Simple Approach to Ranking Differentially Expressed Gene Expression Time Courses through Gaussian Process Regression.** *BMC Bioinformatics* 2011 **12**:180.

**Submit your next manuscript to BioMed Central  
and take full advantage of:**

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at  
[www.biomedcentral.com/submit](http://www.biomedcentral.com/submit)

