BIOINFORMATICS ORIGINAL PAPER Vol. 27 no. 7 2011, pages 946–952
doi:10.1093/bioinformatics/btr037

Gene expression

Advance Access publication January 25, 2011

Classifying short gene expression time-courses with Bayesian estimation of piecewise constant functions

Christoph Hafemeister[1,2,*,†], Ivan G. Costa[3,*,†], Alexander Schönhuth[4] and Alexander Schliep[2,*]

[1]Department of Computational Molecular Biology, Max Planck Institute for Molecular Genetics, Berlin, Germany, [2]Department of Computer Science and BioMaPS Institute for Quantitative Biology, Rutgers, The State University of New Jersey, Piscataway, NJ 08854, USA, [3]Center of Informatics, Federal University of Pernambuco, Recife, Brazil and [4]Centrum Wiskunde & Informatica, 1098 XG Amsterdam, Netherlands

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Analyzing short time-courses is a frequent and relevant problem in molecular biology, as, for example, 90% of gene expression time-course experiments span at most nine time-points. The biological or clinical questions addressed are elucidating gene regulation by identification of co-expressed genes, predicting response to treatment in clinical, trial-like settings or classifying novel toxic compounds based on similarity of gene expression time-courses to those of known toxic compounds. The latter problem is characterized by irregular and infrequent sample times and a total lack of prior assumptions about the incoming query, which comes in stark contrast to clinical settings and requires to implicitly perform a local, gapped alignment of time series. The current state-of-the-art method (SCOW) uses a variant of dynamic time warping and models time series as higher order polynomials (splines).

Results: We suggest to model time-courses monitoring response to toxins by piecewise constant functions, which are modeled as left–right Hidden Markov Models. A Bayesian approach to parameter estimation and inference helps to cope with the short, but highly multivariate time-courses. We improve prediction accuracy by 7% and 4%, respectively, when classifying toxicology and stress response data. We also reduce running times by at least a factor of 140; note that reasonable running times are crucial when classifying response to toxins. In conclusion, we have demonstrated that appropriate reduction of model complexity can result in substantial improvements both in classification performance and running time.

Availability: A Python package implementing the methods described is freely available under the GPL from http://bioinformatics.rutgers.edu/Software/MVQueries/.

Contact: hafemeis@molgen.mpg.de; igcf@cin.ufpe.br; schliep@cs.rutgers.edu;

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 8, 2010; revised on December 26, 2010; accepted on January 19, 2011

1 INTRODUCTION

Time-course experiments reflecting the dynamics of transcription are among the most intriguing, but at the same time, one of the most challenging sources of data in molecular biology. Intriguing, because they offer views of the temporal dynamics of life at the molecular level. We see, for example, gene expression change over time in the cell cycle or, on a larger time scale, during development as part of the normal regulatory process. We also see changes as a response to an external stimulus leading to a quick response as in the case of plants exposed to sunlight or a slower response as in disease progression or an exposure to toxins. These dynamic changes and their intricate regulatory control mechanisms are of fundamental interest in the study of biological systems with inference of regulatory mechanisms as a main goal.

However, these time-course experiments are also very challenging. One of the reasons is rather simple. Time-courses data in other disciplines such as finance, astronomy or climate literally span years at second interval resolution. As a consequence, it is reasonable to consider the data, for all practical purposes, *continuous*. Under weak assumptions, such *continuous* representations have their advantages when filling in missing values through interpolation or obtaining future predictions through extrapolation. However, in gene expression, time-courses are often sampled at very low frequencies and at irregular intervals. Indeed a query in the Gene Expression Omnibus (GEO) database (Edgar *et al.*, 2002) reveals that 90% of all time-courses have less than 9 time points implying that the name short time-courses (Ernst *et al.*, 2005) is well deserved. As a consequence, in such cases a *discrete* time view and a *discrete* representation might be more appropriate. This dichotomy between discrete and continuous view has implications with respect to imputing missing values, taking derivatives (or measuring slopes) and choosing appropriate functional classes for fitting and interpolation. More generally, it determines how appropriate the use of 'classical' tools from time-courses analysis for each specific biological dataset is. Interpolation might sometimes overstate the case the data make, see Figure 1. Simply plotting the time-courses joined by lines might lead to a false impression of continuous time and a false sense of security in choosing methods.

Undersampling gives biological reasons to question smooth interpolations between gene expression levels at sparsely sampled time-points. Additionally, sampling frequencies often do not allow

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

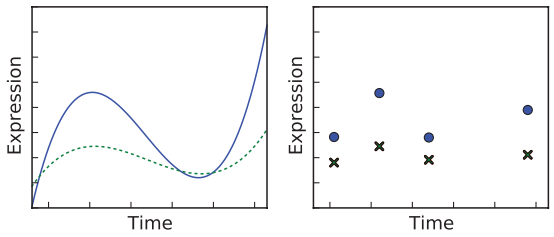

Fig. 1. We show the gene expression time-courses of two genes, on the left joined by a Bspline of order three and as points on the right. Depicted here are expression measurements for genes 5 and 13 from the data used by Smith *et al.* (2009) after treatment with 250 mg/kg ketoconazole.

to monitor the very processes one attempts to understand. For example, transcriptional on and/or offset happens often within 1–2 h or less (Gasch *et al.*, 2000; Hager *et al.*, 2009; Zaslaver *et al.*, 2004) in particular as a response to stress and treatment signals (Chechik and Koller, 2009). While this is sometimes accounted for in the experimental design, see e.g. yeast cell cycle data due to Spellman *et al.* (1998), it is not economical nor feasible, particularly in a clinical setting, to collect samples in minute intervals over several hours. Missing the transient part of such gene expression curves—the rapid rise and overshoot (Chechik and Koller, 2009)—and sampling the steady state instead, does indeed lead to gene expression curves, which are approximately piecewise constant (also called step functions) instead of polynomials of higher degrees.

1.1 Prior work

The particular challenges of short irregularly sampled time-courses have been addressed with a wide range of robust methods, mostly rooted in statistics (Ernst *et al.*, 2005). As the review (Bar-Joseph, 2004) pointed out, methods which did not rely on a continuous time representation performed better for time-courses with a reasonable number of time-points, about a dozen or larger. Querying gene expression datasets to indicate genes with a similar pattern of temporal co-expression was proposed for interactive analysis by Schliep *et al.* (2003, 2004, 2005) and implemented in the GQL tool (Costa *et al.*, 2005).

1.1.1 Clinical time series A substantial body of recent work was concerned with classifying response to treatment in clinical, trial-like settings. The characteristics of such settings are *first* the availability of prior knowledge about the disease in question, which allows to substantially reduce feature space dimension by discarding irrelevant features; *second* that samples are drawn at regular intervals, which sometimes span up to a few years and *third* that runtime is not an issue; up to, say 10 days for analysis are usually well acceptable. Most recent related work based on splines (Kaminski and Bar-Joseph, 2007) or Hidden Markov Models (HMMs) (Costa *et al.*, 2009; Lin *et al.*, 2008), for example, was concerned with the classification of response to interferon-β in multiple sclerosis patients.

1.1.2 Immediate response to external stimuli The questions addressed in our study relate to classifying response patterns to an *unknown* stimulus. The particulars of the corresponding classification task are *first* that no prior information about the

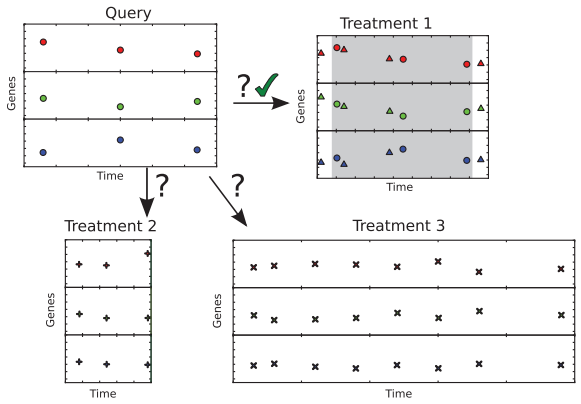

Fig. 2. The multivariate time-course classification problem with discretely represented time-courses. The task is to locally align a query with the expression of three genes, which might result from a process for which the regulatory program is executed at a different speed or phase shifted, to the most similar treatment in order to transfer toxicity information from treatment to query. The shaded area marks the optimal classification and local alignment of the query with Treatment 1.

incoming query time-course is available such that a preselection of features is not applicable; *second* and most importantly, that the query time-course is usually shorter and not in alignment with the time-courses already available and *third* that runtime is an issue—when classifying response to toxins, for example, hours, sometimes even minutes count. For toxicity assays, a classifier can indicate the toxicity level of new compounds based on similarities in the gene expression changes over time (Hayes *et al.*, 2005). For the latter application, Smith *et al.* (2009) modified the correlation optimized warping (COW) method (Nielsen *et al.*, 1998), a variant of dynamic time warping (DTW) (Sakoe and Chiba, 1978), to allow local warpings (i.e. local alignments).

The method, which the authors call shorting correlation optimized warping (`SCOW`), outperformed COW, a previously proposed model-based approach using generative splines by the same authors (Smith *et al.*, 2008) and two further methods, in particular dynamic time warping on absolute values. Moreover, to most sensibly address that incoming queries consist of infrequent and irregular samples, the authors suggested to *subsample* queries from the existing data and to construct classifiers from the remaining time-courses.

Other relevant applications more distantly related to the ones considered here are to find time shifts in stress/chemical treatments in order to detect relations between genes and transcription factors (Redestig *et al.*, 2007; Shi *et al.*, 2007).

1.2 Our approach

We propose the use of stochastic piecewise constant functions for modeling genome-wide expression time-courses, which reflect responses to toxic compounds. As explained in Figure 2, the classification task is to assign an unclassified time-course (*'Query'*) to a time-course which reflects a response to a known toxin (*'Treatment'*). Since Queries and Treatment can be different in terms of sampling times and frequencies, the technical challenge is to determine a local, gapped alignment between the Query and the Treatment time-course.

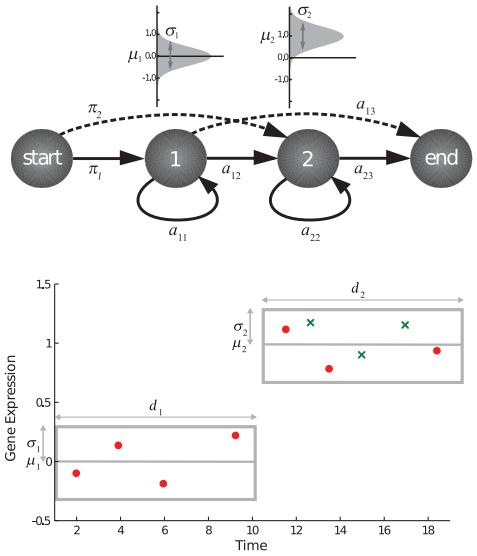

Fig. 3. A left–right HMM with two states (top) which can be viewed as a stochastic piecewise constant function (bottom). In the `1HMM`, per-state mean and variance determine expression value and the self-transition probability segment length. This is equivalent to a piecewise constant function (bottom), where observations fluctuate according to a Gaussian $N(\mu_m, \sigma_m)$ and with expected length d_m. The left to right transitions represented by dashed arrows allow the classification to be based on local alignments. For example, the `1HMM`, which has been trained with the red time-course (red circles), will give a high likelihood to a query sequence (green crosses), where observations only fit the latter upregulation expression pattern.

The main contributions of our work are the following:

- We propose the use of stochastic piecewise constant functions, which can be viewed as HMMs with left–right topology (see Fig. 3 for an example). The most substantial differences between this and our prior works (Costa *et al.*, 2009; Schliep *et al.*, 2003, 2005) is to account for the high dimensionality of the data per time-point (>1000) and the shortness as well as irregularity of the time-courses <9. We do this by Bayesian estimation of parameters and an efficient log-scale implementation. To allow local alignments including gaps, thereby addressing that query time-courses have been sampled irregularly, we estimate probabilities associated with left-to-right transitions by means of Bayesian regularization.
- As a result, we improve the mean classification accuracy obtained by the previous method [`SCOW` (Smith *et al.*, 2009)] by more than 7% (EDGE data) and 4% (Arabidopsis data) while being at least 140 times faster than `SCOW` (0.27 versus 38.5 h on the EDGE data. Clearly, both accuracy and running time are relevant in particular when classifying response to toxins (EDGE) since this translates to choosing the right remedy both accurately and timely.
- We provide a novel way for assigning times to states and observations based on assigning time-courses to HMMs with left–right topology.
- We also demonstrate that modeling time-courses with one-piece constant functions (or one-nearest neighbor) still leads to a small, but significant ($p = 1.1 \cdot 10^{-3}$) improvement in classification accuracy over `SCOW` on EDGE data and no significant difference on Arabidopsis data, while it is over 20 000 times faster to compute.

2 METHODS

2.1 Notation

In the following, let

- $i \in \{1,...,N\}$ denote the treatments/queries,
- $t \in \{1,...,T\}$ denote the time points,
- $g \in \{1,...,G\}$ be a running index for the genes,
- $O_{i,g} \in \mathbb{R}^T$: expression time-course of gene g of treatment/query i.
- $O_{i,t} \in \mathbb{R}^G$: expression of all genes at time t of treatment/query i.
- $O_{i,g,t} \in \mathbb{R}$: expression value of gene g of treatment/query i at time t.
- $O_i \in \mathbb{R}^{T \times G}$: collection of all expression values for treatment/query i for all genes g across all time-points t.
- Θ: Parameterization describing a HMM.

2.2 Modeling gene expression time-courses with left–right HMMs

2.2.1 Left–right HMMs definition An HMM gives rise to a stochastic process by acting on a Markov chain of M states from which real values are emitted as described by emission probability density functions (pdf) attached to the states [e.g. Durbin *et al.* (1998); Rabiner (1989)]. Since the sequence of states is not observed, states are referred to as *hidden*. Here we use HMMs with left–right topologies, that is the Markov chains can be sequentially ordered and are exclusively visited in that order. More formally, let $A = (a_{ml})_{1 \le m,l \le M}$ be the transition probability matrix of the Markov chain, where a_{ml} is the probability of going from state m to state l. A left–right HMM is defined by $a_{ml} = 0$ for $m > l$. As is well known, duration times for the single states m follow a geometric distribution where the expected duration time d_m is computed as $d_m = 1/(1-a_{mm})$ where a_{mm} is the self-transition probability for state m. The last state M is also referred to as 'End' state and, in our case, does not emit values. The initial probability distribution $\Pi = (\pi_1,...,\pi_m)$ over the states which reflects from which state the generation procedure is started will be modeled as a special 'Start' state. We refer to left–right HMMs as `1HMM`s in the following. For example, a `1HMM` with two emitting states, the first state with a mean emission of zero and the second one with a mean emission of one, models time-courses displaying an upregulation expression pattern (Fig. 3).

2.2.2 Emission pdfs We use a mixture of two multivariate Gaussians as emission pdfs. Their first component is a multivariate Gaussian with diagonal covariance matrix modeling multivariate expression values, where the different dimensions correspond to the different genes. The second component models observation due to noise as suggested by Fraley and Raftery (1998). More formally, the pdf $\mathbb{P}_m^k$ for emitting a (multivariate) expression value $O_{i,t} \in \mathbb{R}^G$ from state m of `1HMM` Θ^k is

$$\mathbb{P}_m^k(\cdot) = (1-\phi_{\text{noi}}) \cdot N(\cdot|\vec{\mu}_m^k, \Sigma_m^k) + \phi_{\text{noi}} \cdot N(\cdot|\vec{\mu}_{\text{noi}}, \Sigma_{\text{noi}}),$$

where $N(\cdot|\vec{\mu}, \Sigma)$ are G-dimensional Gaussians as parameterized by a G-dimensional mean $\vec{\mu}$ and a $G \times G$ covariance matrix Σ and ϕ_{noi} is the proportion of noise observations. In our experiments, $\vec{\mu}_{\text{noi}} \in \mathbb{R}^G$ is a vector containing the average expression values of the genes across all treatments and time-points and $\Sigma_{\text{noi}} \in \mathbb{R}^{G \times G}$ has diagonal entries set to a high value, e.g. $\sigma_{gg} = 2.00$, and all other entries are set to zero. In the experiments, we set $\phi_{\text{noi}} = 0.05$. All parameters from the noise component are fixed during parameter estimation.

More formally, each left–right HMM has parameterization

$$\Theta^k = (A^k, B^k, \Pi^k) \quad \text{with emission pdf parameterizations}$$

$$B^k = (\vec{\mu}_1^k,...,\vec{\mu}_M^k, \Sigma_1^k,...,\Sigma_M^k, \vec{\mu}_{\text{noi}}, \Sigma_{\text{noi}}, \phi_{\text{noi}})$$

Note that only $\vec{\mu}_1^k, ..., \vec{\mu}_M^k, \Sigma_1^k, ..., \Sigma_M^k$ vary among different HMMs Θ^k in the following.

2.2.3 Stochastic piecewise constant functions Considering only univariate time-courses for simplicity, we can interpret the temporal behavior modeled by an `1HMM` in the following way: each state represents a particular level of expression of a gene specified by μ_m, where a certain level of error (encoded by σ_m^2) is allowed, and the time-course has an expected length of d_m of staying at this particular expression level. This can be interpreted as a 'bounding box' specifying the expected expression of the time-course as depicted in Figure 3 (bottom). Another view helping to understand why our models perform so well is that of regression with *stochastic piecewise constant functions*. The per-state mean values and the expected state durations define location and length of segments of a piecewise constant function. If variances are homogeneous, that is identical in all states, then the probability of observations under our model are the inverse of the regression errors. Heterogeneous variances correspond to a time-specific weighting of regression errors in that model. While not exploited in this article, there is one very important aspect of temporal asynchronicity built into the models. If one estimates one `1HMM` from multiple observations, then the actual segment lengths, the length of the constant pieces, are the state durations of the state path in the `1HMM` and are chosen per *observation* and will differ between different observations, to account for phase shift and distinct frequencies.

2.2.4 Classification with left–right HMMs For performing classification, we train one left–right HMM Θ^i for each treatment response time-course. For a new incoming query time-courses O, classification is performed by finding the `1HMM` with maximum likelihood, or

$$\text{argmax}_i \mathbb{P}(O|\Theta^i)$$

where $\mathbb{P}(O|\Theta^i)$ is the HMM likelihood function as computed with the forward–backward algorithm (Baum *et al.*, 1970).

The estimation of the `1HMM`, which is based on the Baum–Welch algorithm (Baum *et al.*, 1970), has to be performed on a single short time-courses. Therefore, we need to perform regularization of the parameters to avoid overfitting. We use maximum likelihood estimates (MLEs) for μ_m^i. The covariance matrix Σ_m^i is diagonal and diagonal entries σ_{jj} are equal to $\max(\hat{\sigma}_{jj}, \sigma_{\min})$, where $\hat{\sigma}_{jj}$ is the MLE and $\sigma_{\min} = 2$. For the transition matrix and initial state probabilities, we assume the parameters come from a Dirichlet distribution.

More formally, let O_i be the observation sequence and $X = \{X_1, ..., X_T\}$ be the sequence of states visits of observation O_i such that $X_t \in \{1, ..., M\}$. The Baum–Welch algorithm is based on estimating the forward $\mathrm{f}_t(m) = \mathbb{P}(O_{i,0}, ..., O_{i,t}, X_t = m)$ and backward $\mathrm{b}_t(m) = \mathbb{P}(O_{i,t+1}, ..., O_{i,T}|X_t = m)$ variables [see (Durbin *et al.*, 1998) for estimates]. We use a Dirichlet distribution

$$\Pi^i \sim \text{Dir}(\cdot|\alpha^\pi) \quad \text{resp.} \quad (a_{11}^i, ..., a_{ml}^i, ..., a_{MM}^i) \sim \text{Dir}(\cdot|\alpha^a)$$

as prior for the initial transition probabilities, respectively, for transition probabilities $a_{ml}^i, m \le l$ where α^a and α^π are the hyper-parameters. Hence, the MAP estimates of these parameters are as follows:

$$\hat{a}_{ml}^i = \frac{\frac{a_{ml}\sum_{t=0}^{T-1}\alpha_t(m)\mathbb{P}_m^i(O_{i,t})\beta_{t+1}(l)}{\mathbb{P}(O_i|\Theta^i)} + \alpha^a - 1}{\frac{\sum_{t=0}^{T-1}\alpha_t(m)\beta_{t+1}(l)}{\mathbb{P}(O_i|\Theta^i)} + M*(\alpha^a - 1)}$$

and

$$\hat{\pi}_m^i = \frac{\frac{\alpha_0(m)\beta_0(m)}{\mathbb{P}(O_i|\Theta^i)} + \alpha^\pi - 1}{\sum_{l=1}^M \left(\frac{\alpha_0(l)\beta_0(l)}{\mathbb{P}(O_i|\Theta^i)}\right) + M*(\alpha^\pi - 1)}.$$

We choose $\alpha^\pi = \alpha^a = 1.1$. Thus, we enforce small probabilities for all transitions π_m and $a_m l$ with $m \le l$. This regularization is important for the support of local, gapped alignments, as it gives a small probability to all transitions regardless of the data support.

2.2.5 Locally aligning time-courses and estimating times In order to make a query time-course comparable with an existing time-course, we need to align the two. That is, we need to map each observation in the query to a specific time-point in the treatment time-course.

Let Θ^i be the HMM with the highest likelihood for a given query. We can then infer the time mapping from the Viterbi path of the query and the training data with Θ^i. The Viterbi path (Rabiner, 1989) is the most likely sequence of hidden states given the observation and, in this setting, can be interpreted as the sequence of the constant pieces defined by the HMM states which best describes the data. As our training data comes with time- point information, we can assign time to each state by evaluating which observations visit this state in the Viterbi path. We record the times and obtain collection of time-points assigned to each state, spanning a time interval. We expand the parameterization of a HMM Θ^k by S^k, where S_m^k is the time interval of the training observations which visit state m in their Viterbi path. Consider the example in Figure 3. The seven training observations result in a Viterbi path of {1,1,1,1,2,2,2}. This assigns the times 2, 4, 6, 9 h to state 1, which yields $S_1 = [2, 9]$ and 12, 14, 18 h to state 2, translating to $S_2 = [12, 18]$. After classification, query observations are aligned to time-points based on the query Viterbi path with Θ^i and the corresponding time interval in S^i. We assume that if several query time-points are generated in the same state, they will be assigned equal slices of that state's time interval. That is, all query observations visiting the same state m are assigned time-points equally distributed in S_m^i. Using our previous example with the query in Figure 3, all query observations visit state 2, S_2 is $[12, 18]$ and thus the observations will be assigned the times 12, 15 and 18 h.

2.3 Constant functions: one-nearest-neighbor classifier

K-nearest neighbor classifiers (Cover and Hart, 1967) are a widely used simple classification method from machine learning and a baseline for method comparison. The main idea is to classify an unlabeled object based on the most common label among its k nearest labeled neighbors. We use $k = 1$, or a one-nearest-neighbor classifier, which Cover and Hart (1967) found to outperform k-NN classifiers for larger k under natural assumptions on the underlying sample space and in an asymptotical sense.

For a query i and multiple treatments $j_1, ..., j_k$, we have observation matrices O_i, respectively, O_j measuring expression values per gene and per time-point with multiple observation for replicates. For both queries and treatments, we compute a genewise, vector-valued average over the time-points (and replicates), which we denote as $\vec{q}$ and $\vec{m}_{(j)}$, respectively. The classification consists of computing Euclidean distances between the vector $\vec{q}$ and all vectors $\vec{m}_{(j)}$ and assigning the treatment of minimal distance to the query.

The Euclidean distance between the query mean vector $\vec{q}$ and the treatment mean vector $\vec{m}$ can also be viewed as a regression error for fitting the query with the constant vector-valued regression function $\vec{f}(t) = \vec{m}$. The `1NN` rule—assign the class label from the closest treatment mean vector—is consequently interpreted as assigning the class label for the treatment with the minimal regression error.

2.4 Datasets

We use the same dataset used in Smith and Craven (2008); Smith *et al.* (2008, 2009) from the EDGE toxicology database (Hayes *et al.*, 2005). Moreover, we analyze Arabidopsis stress response data (Kilian *et al.*, 2007). The EDGE toxicology contains measurements of gene expression values in mice after treatment with different toxins at several dosage levels. The dataset consists of 216 unique observations of expression levels of 1600 genes spanning 11 treatments. Each observation is associated with a treatment and a time-point, where the times range from 2 h up to 192 h and a treatment might have replicates for one or more time-points. The number of observed times for a treatment ranges from 3 to 9 with an average of 5. The Arabidopsis dataset (Kilian *et al.*, 2007) contains 18 time-courses measured over 9 stress conditions and 2 distinct tissues. The data spans over 20 000 genes over 8

time points in duplicates (0.25, 0.5, 1, 3, 4, 6, 12 and 24 h). To reduce the number of genes a 2-fold-change filter was applied, which resulted in 2075 genes. An overview of the datasets is given in the Supplementary Material.

Additionally, to further investigate respective strengths and weaknesses of the methods, we use *simulated data*. We choose relevant smooth functions, note that the functional choices do not reflect underlying assumptions of any method used in the comparison, but rather reflect prototypical biological behavior: we modeled cell cycle signals by the (periodic) sine function and upregulation by the (monotonically increasing) sigmoid function. We further opted for the impulse model proposed by Chechik and Koller (2009), refer to the Supplementary Material for details. Additionally, we add mirror images (mirroring along the x-axis) of the functions and add a constant function. We obtain a total of 13 functions and sample data from these functions at regular intervals. Every such sampled observation spans 1000 genes, and we add independent noise from a normal distribution to each datum. We repeat this procedure three times such that we obtain three repetitions for each function/time-point.

2.5 Experimental design

As the EDGE toxicology and Arabidopsis datasets have no independent measurements of time-courses to serve as queries, we use the same procedure proposed in Smith *et al.* (2009). For a particular compound treatment, we generate a query (test data) time-course by (i) randomly selecting its number of time-points, (ii) randomly choosing which time points to represent and (iii) randomly picking observations at these time points. All remaining observations together with all other treatments' observations constitute the training data. The classification is performed by building a model for each time-course in the training data, then aligning the query to all models and returning the one with highest similarity. Classification is correct if the most similar model derived from the same treatment as the query.

Ten such train/test datasets for each treatment (110 total for EDGE and 180 for Arabidopsis) are used to evaluate performance. We repeat this procedure 100 times (EDGE) and 30 times (Arabidopsis). The first 50/15 performance tests are used to choose an adequate number of states for `1HMM`. We varied this one free parameter from 2 to 9 and selected the model with highest classification accuracy: 4 states for SCOW and 6 states for Arabidopsis (see Supplementary Material for all accuracies). We set the number of states to be equal to the time-course size, whenever its size is smaller than the predefined number of states. The latter 50/15 performance tests are applied to `SCOW`, `1NN` and `1HMM` and results of the test classifications are shown in Figure 4.

For the simulated data, we generate 10 train/test datasets for each function and use the same model parameters as with the EDGE data. Mean classification accuracies are shown in Figure 5.

2.6 Implementations

`SCOW` is implemented in the Curve Analysis Tool. For our experiments, we used the parameters with which Smith *et al.* (2009) obtained the best results: the number of segments was set to three, and $\sigma_s = \sigma_a = 10$. Splines of order two were used to interpolate the time-courses and queries and to reconstruct pseudo-observations for every 4 h. Note that Smith *et al.* (2009) did not segment the datasets for parameter selection, therefore this selection can include a positive bias favoring SCOW.

Our methods are implemented in Python http://www.python.org using the NumPy http://numpy.scipy.org/ package, using transition matrices and state density parameters as the core data structures. All computations are performed with log-probabilities to avoid numerical problems.

All experiments were done on a Linux machine using a single CPU core with 2.8 GHz. The running times in minutes for 50 EDGE experiments, consisting of 110 alignments/classifications each or 5500 alignments total, are as follows: 0.11 `1NN`, 16.2 `1HMM`, 2308 `SCOW`. Running times in minutes for the Arabidopsis experiments were as follows: 0.06 `1NN`, 18 `1HMM` and 10800 `SCOW`. That is, `1NN` achieves a speedup exceeding 20 000 and `1HMM` a speedup exceeding 140 over `SCOW`.

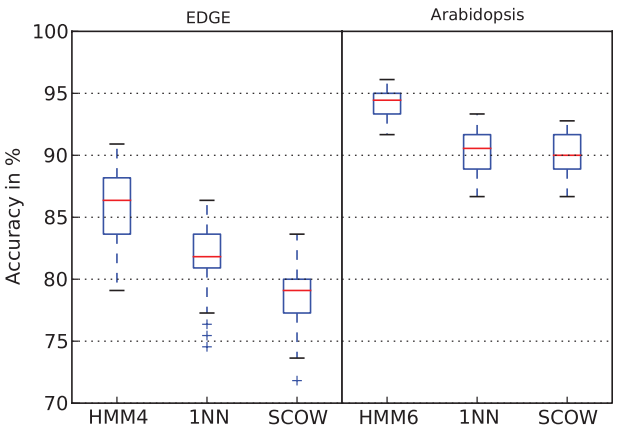

Fig. 4. Accuracy for classifying queries to treatments using the different methods on EDGE (left) and Arabidopsis (right).

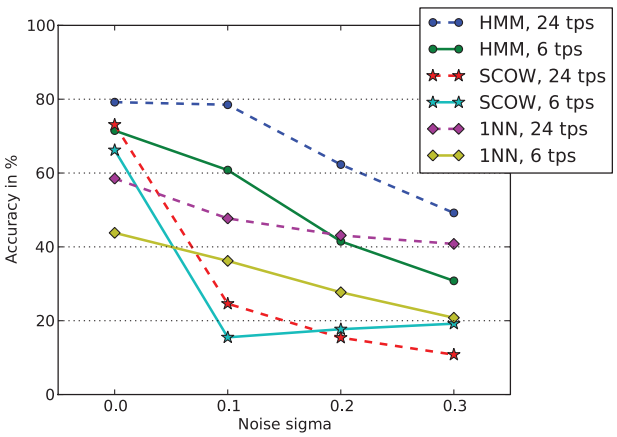

Fig. 5. Accuracies of the methods for the simulated dataset. Each method is tested with data consisting of 6 and 24 time-points (tps).

3 DISCUSSION

3.1 Results on biological data

The results for the EDGE and Arabidopsis data are given in a condensed form as box plots in Figure 4 (see Supplementary Material for stratified accuracy by treatment for EDGE). Figures comparing the alignment accuracies are in the Supplementary Material. Note that both `1NN` and `1HMM` do not consider query times in their computation. Consequently, the classification accuracy for those two methods is invariant under distortions as suggested and performed by Smith *et al.* (2009). For the EDGE data, their additional clustering of alignments only improves results for queries with more than three time-points, less than 43% of all queries, and by less than 10% in simulations with five equal-sized clusters with different warpings. As Table 4 from Supplementary Material shows, on such queries `1HMM` has an over 19% higher accuracy than `SCOW`. Given the running time requirements of `SCOW`, we refrained from

using either distortions, which would not influence our results as actual query times are not used as input or clustering of alignments. Note that the lHMM is robust to distortions in the unclustered case, because only the relative order of query observations is taken into account rather than relying on absolute time values.

SCOW: mean classification accuracy of SCOW is 78.8% (SD 2.5) on EDGE and 89.8% (SD 1.7) on Arabidopsis. Adding the requirements of an average time error in the alignment smaller than 24 and 12 h, mean accuracy declines to 69.4 and 59.6% (SD 3.4 and 3.0) on EDGE and 86.1 and 80.7% on Arabidopsis (see Supplementary Material). These results are slightly lower than what Smith *et al.* (2009) report. Our explanation is that we perform 50 repetitions of the experiment and report average results. It is interesting to see that SCOW performs very well for 7 of the 11 treatments of EDGE, even reaching 100% accuracy for six treatments, but at the same time makes between 22% and at most 40% correct decisions for three other treatments. This may indicate a lack of robustness or lack in generalization with respect to the treatments the queries are chosen from.

One nearest neighbor: mean classification accuracy is 81.7% (SD 2.5) on EDGE data and 90.3% (SD 1.8) on Arabidopsis data. This is a significant improvement over SCOW ($p = 1.1 \cdot 10^{-3}$ with McNemar's χ^2 test) on EDGE data, while there is no significant difference on Arabidopsis data.

That the method development of Smith *et al.* (2009) leads to a classification performance worse or equivalent than that of a one-nearest-neighbor classifier, often the base case for classification performance on the genewise averaged expression levels has the following implications. First, it indicates clearly that absolute gene expression values do matter. This has been previously reported in the biological literature (Ellis *et al.*, 2008; Pegg, 2008) in a comparable setting to the data used here. Second, for such short time-courses alignments seem to be of little informative value. Note that on one hand a dynamic time warping algorithm on absolute levels is reported to perform worse than SCOW; on the other hand, unconstrained optimization of correlation also results in spurious results (Smith *et al.*, 2009). An analogy might be found in the uncertainty in alignment scores for short biological sequences; profile HMMs in contrast perform better for short sequences. Of course our findings do not invalidate the method development per se, but rather indicate that biological problems with such low numbers of time-points, cf. Supplementary Material, should be tackled with a different approach.

lHMM: mean classification accuracy is 85.9% (SD 2.8) for lHMM with four states on EDGE and 94.0% (SD 1.3) for lHMM with six states for Arabidopsis. This is a significant improvement over 1NN; $p = 1.1 \cdot 10^{-15}$ (EDGE) and $p = 1.1 \cdot 10^{-10}$ (Arabidopsis) and over SCOW; $p = 1.0 \cdot 10^{-17}$ (EDGE) and $p = 2.5 \cdot 10^{-15}$ (Arabidopsis) (McNemar's χ^2 test).

Adding the criteria that the average error in the alignment is not more than 24 and 12 h, respectively, on EDGE data, mean accuracy decreases to 73.1 and 55.5% (SD 3.8 and 3.2) and to 85.6 and 69.8% on Arabidopsis data. While only the 12 h accuracy value is lower for lHMM than SCOW, the mean values of the temporal error give an incomplete picture (see Supplementary Material). In fact, lHMM aligns 386 queries more than SCOW without any temporal alignment error on EDGE data. Indeed, temporal errors are rarer, but if errors are made, they can be larger, which leads to the slightly worse performance in alignments for the range from 10 to 24 h for which SCOW performs better (see Supplementary Material). Moreover, while alignment is important it is only a secondary goal to the fundamental application problem of correct classification of the compounds.

3.2 Results on simulated data

Our hypothesis was that the inferior performance of SCOW was due to that time-courses are extremely short. In fact, COW (Nielsen *et al.*, 1998), the method that SCOW is based on, was developed for chromatographic time-course data with 3300 time-points translating to more than 1100 as many time-points as the average query of length 2.84 for the EDGE data. One observes an improvement as the number of time-points increases. However, SCOW also shows a larger susceptibility to noise, see Figure 5. Indeed the accuracy of SCOW is higher for a less frequently subsampled time-course.

4 CONCLUSION

Short time-courses are the predominant form of temporal gene expression information. However, their analysis poses particular challenges as short usually translates to infrequent samples, often drawn at irregular time intervals. The inherent noise is another central concern.

Here, we focused on toxicity assays and responses to stress. The challenge was to classify incoming short time-courses reflecting a response to toxins/stress in changes in gene expression over time by assigning them to existing, labeled time-courses. Incoming queries possibly have been sampled at intervals which cannot be straightforwardly aligned with the labeled time-course. In contrast to prior approaches, we opted to have a 'discrete' view on the problem, by modeling these time-courses as stochastic piecewise constant functions, implemented as HMM with a left–right topology. Building on an established statistical framework, we estimate parameters with a Bayesian approach and use maximum likelihood for classification.

On the EDGE experiments, we outperform the best known prior approach, SCOW, with an increase in accuracy exceeding 7% overall, an increase of over 19% over SCOW for queries with three or more time-points; clustered alignments only have an advantage of about 10% over SCOW for such queries. On the Arabidopsis experiments, the increase in accuracy was 4%. At the same time, our method is at least 140 times faster to compute. We additionally show that a one-nearest neighbor classifier on the genewise time averages of the expression time-courses outperforms or matches SCOW while it is 20 000 times faster to compute. One reason for our superior performance is likely the coarse time resolution of queries; indeed the method of correlation optimized warpings (Nielsen *et al.*, 1998), on which SCOW is built, was developed for 1100 as many time-points as the average query of length 2.84 we are dealing with. The time-courses of the transcriptional response rather reflect sequences of steady states while missing the very dynamics behind them. Another reason is that SCOW relies on continuous, polynomial time-course representations and interpolation. In particular, the results of 1NN also indicate that correlation might not be the right objective function to optimize; indeed Smith *et al.* (2009) spend some effort on trying to avoid alignments of high correlation but with large difference

in value. What remains somewhat inconclusive is the inconsistent performance of SCOW on the various treatments and its apparent inability to make good use of more frequently sampled data in the presence of noise, as the results on simulated data show.

Another issue, and a more general criticism is that DTW can be rephrased in the language of HMM and roughly translates to a pair HMM with continuous emissions (see the Supplementary Materials for a more detailed discussion). Just like profile HMMs are superior in detecting remote homolog proteins due to their position-dependent gap and substitution parameters, the HMMs with left–right topology offer equivalent flexibility for continuous-valued sequences. Combining these insights with the stochastic piecewise constant functions is, in terms of model complexity, simpler than splines that yield the usual explanations for why our method is more accurate, robust and less prone to overfitting.

In future applications, we are planning to explore approaches which put emphasis on optimal discrimination. Feature selection and/or a Bayesian significance analysis will also be addressed. Last but not least, given the maturing state of the field, a collection of benchmark biological and simulated time-course datasets and a comparison of the wide range of methods would be both achievable and very worthwhile.

ACKNOWLEDGEMENTS

Thanks to Adam A. Smith for providing the code and datasets used by Smith *et al.* (2009).

Funding: Fundacão de Amparo a Pesquisa de Pernambuco (to I.G.C.); Conselho de Desenvolvimento Científico e Tecnológico (Brazil) (to I.G.C.) in part.

Conflict of Interest: none declared.

REFERENCES

Bar-Joseph,Z. (2004) Analyzing time series gene expression data. *Bioinformatics*, **20**, 2493–2503.

Baum,L.E. *et al.* (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, **41**, 164–171.

Chechik,G. and Koller,D. (2009) Timing of gene expression responses to environmental changes. *J. Comput. Biol.*, **16**, 279–290.

Costa,I.G. *et al.* (2005) The graphical query language: a tool for analysis of gene expression time-courses. *Bioinformatics*, **21**, 2544–2545.

Costa,I.G. *et al.* (2009) Constrained mixture estimation for analysis and robust classification of clinical time series. *Bioinformatics*, **25**, i6–i14.

Cover,T. and Hart,P. (1967) Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, **13**, 21–27.

Durbin,R. *et al.* (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, New York, NY.

Edgar,R. *et al.* (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.*, **30**, 207–210.

Ellis,L. *et al.* (2008) Histone deacetylase inhibitor panobinostat induces clinical responses with associated alterations in gene expression profiles in cutaneous T-cell lymphoma. *Clin. Cancer Res.*, **14**, 4500–4510.

Ernst,J. *et al.* (2005) Clustering short time series gene expression data. *Bioinformatics*, **21** (Suppl. 1), i159–i168.

Fraley,C. and Raftery,A.E. (1998) How many clusters? which clustering method? answers via model-based cluster analysis. *Comput. J.*, **41**, 578–588.

Gasch,A.P. *et al.* (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, **11**, 4241–4257.

Hager,G.L. *et al.* (2009) Transcription dynamics. *Mol. Cell*, **35**, 741–753.

Hayes,K.R. *et al.* (2005) Edge: a centralized resource for the comparison, analysis, and distribution of toxicogenomic information. *Mol. Pharmacol.*, **67**, 1360–1368.

Kaminski,N. and Bar-Joseph,Z. (2007) A patient-gene model for temporal expression profiles in clinical studies. *J. Comput. Biol.*, **14**, 324–338.

Kilian,J. *et al.* (2007) The atgenexpress global stress expression data set: protocols, evaluation and model data analysis of uv-b light, drought and cold stress responses. *Plant J.*, **50**, 347–363.

Lin,T.-H. *et al.* (2008) Alignment and classification of time series gene expression in clinical studies. *Bioinformatics*, **24**, i147–i155.

Nielsen,N.V. *et al.* (1998) Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J. Chromatogr. A*, **805**, 17–35.

Pegg,A.E. (2008) Spermidine/spermine-N1-acetyltransferase: a key metabolic regulator. *Am. J. Physiol. Endocrinol. Metab.*, **294**, E995–E1010.

Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–285.

Redestig,H. *et al.* (2007) Transcription factor target prediction using multiple short expression time series from arabidopsis thaliana. *BMC Bioinformatics*, **8**, 454.

Sakoe,H. and Chiba,S. (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Sig. Proces.*, **26**, 43–49.

Schliep,A. *et al.* (2003) Using Hidden Markov Models to analyze gene expression time course data. *Bioinformatics*, **19**, (Suppl. 1), i255–i263.

Schliep,A. *et al.* (2004) Robust inference of groups in gene expression time-courses using mixtures of HMMs. *Bioinformatics*, **20**, (Suppl. 1), i283–i289.

Schliep,A. *et al.* (2005) Analyzing gene expression time-courses. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **2**, 179–193.

Shi,Y. *et al.* (2007) Inferring pairwise regulatory relationships from multiple time series datasets. *Bioinformatics*, **23**, 755–763.

Smith,A.A. and Craven,M. (2008) Fast multisegment alignments for temporal expression profiles. *Comput. Syst. Bioinformatics Conf.*, **7**, 315–326.

Smith,A.A. *et al.* (2008) Similarity queries for temporal toxicogenomic expression profiles. *PLoS Comput. Biol.*, **4**, e1000116.

Smith,A.A. *et al.* (2009) Clustered alignments of gene-expression time series data. *Bioinformatics*, **25**, i119–i127.

Spellman *et al.* (1998) Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.

Zaslaver,A. *et al.* (2004) Just-in-time transcription program in metabolic pathways. *Nat. Genet.*, **36**, 486–491.