A patient-gene model for temporal expression profiles in clinical studies

Naftali Kaminski[1] and Ziv Bar-Joseph[2]*

[1] Simmons Center for Interstitial Lung Disease, University of Pittsburgh Medical School, Pittsburgh, PA 15213, USA

[2] School of Computer Science and Department of Biology, Carnegie Mellon University 5000 Forbes Ave. Pittsburgh, PA, 15213, USA

Abstract. Pharmacogenomics and clinical studies that measure the temporal expression levels of patients can identify important pathways and biomarkers that are activated during disease progression or in response to treatment. However, researchers face a number of challenges when trying to combine expression profiles from these patients. Unlike studies that rely on lab animals or cell lines, individuals vary in their baseline expression and in their response rate. In this paper we present a generative model for such data. Our model represents patient expression data using two levels, a gene level which corresponds to a common response pattern and a patient level which accounts for the patient specific expression patterns and response rate. Using an EM algorithm we infer the parameters of the model. We used our algorithm to analyze multiple sclerosis patient response to Interferon-β. As we show, our algorithm was able to improve upon prior methods for combining patients data. In addition, our algorithm was able to correctly identify patient specific response patterns.

1 Introduction

Time series expression experiments have been used to study many aspects of biological systems in model organisms and cell lines [1, 2]. More recently, these experiments are playing an important role in several pharmacogenomics and clinical studies. For example, the Inflammation and the Host Response to Injury research program [3], a consortium of several leading research hospitals, studies the response of over a hundred trauma and burn patients using time series expression data. Table 1 lists a number of other examples of such studies.

Time series expression experiments present a number of computational problems [4]. These include handling noise, the lack of repeats and the fact that only a small number of points are measured (which is particularly true in clinical experiments since tissue or blood needs to be extracted from the patient for each time point). Clinical experiments, while promising, suffer from all these issues and also raise a number of new computational challenges. In many cases the major goal of these experiments is to combine results from multiple patients to

[*] To whom correspondence should be addressed: zivbj@cs.cmu.edu

identify common response genes. However, unlike lab animals which are raised under identical conditions, individuals responses may vary greatly. First, individuals may have different baseline expression profiles [5]. These differences result in some genes being expressed very differently from the common response. Second, the *response rate* or patients dynamics varies greatly among individuals [6, 7]. This leads to profiles which, while representing the same response, may appear different.

Previous attempts to address some of these problems have each focused on only one of the two aspects mentioned above. For example, many papers analyzing such data use the average response [6] to overcome individual (baseline) patterns. Such methods ignore the response rate problem, resulting in inaccurate description of the common response. Alignment methods where suggested to overcome response rate problems in time series expression data, especially in yeast [8, 9]. However, these methods rely on *pairwise alignment*, which is not appropriate for large datasets with tens of patients. In addition, it is not clear how to use these methods to remove patient specific response genes.

Table 1: Examples of time series clinical studies

Reference	**Treatment / condition**	**Num. patients**	**Num. time points**	**Combining method**
Inflammation and the Response to Injury[3]	Trauma and burn	over a hundred	varying	not described
Sterrenburg *et al*[7]	Skeletal myoblast differentiation	3	5	averaging and individual analysis
Weinstock-Guttman *et al*[6]	IFN-β for multiple sclerosis	8	8	averaging
Calvano *et al*[10]	bacterial endotoxin	8	6	assumed repeats

Table 1. Examples of a number of time series clinical studies. Note that in all cases only a few time points are sampled for each patient. In addition, in most cases researchers assume that the different patients represent repeats of the same experiment, even though most of these papers acknowledge that this is not the case (see citation above). Our algorithm, which does not make such assumption and is still able to recover a consensus response pattern is of importance to such studies.

In this paper we solve the above problems by introducing a model that consists of two levels: The gene level and the patient level. The *gene level* represents the consensus response of genes to the treatment or the disease being studied. The questions we ask at this level are similar to issues that are addressed when analyzing single datasets experiments including overcoming noise, continuous representation of the measured expression values and clustering genes to identify common response patterns [16, 17]. The *patient level* deals with the instances of these genes in specific patients. Here we assume that patient genes follow a mixture model. Some of these genes represent patient specific response, or baseline expression differences. The rest of the genes come from the consensus expression response for the treatment studied. However, even if genes agree with this consensus response, their measured values still depend on the patient unique response rate and starting point. Using the consensus curve from the gene level, we associate with each patient a starting point and speed. These values correspond to difference in the timing of the first sample from that patient (for example, if some patients were admitted later than the others) and the patient dynamics (some patients respond to treatment faster than others [6, 7]).

For the gene level we use a spline based model from [9]. The main focus of this paper is on the patient level and on the relationship between the patient level and the gene level. We describe a detailed generative model for clinical expression experiments and present an EM algorithm for inferring the parameters of this model.

There are many potential uses for our algorithm. For example, researchers comparing two groups of patients with different outcomes can use our algorithm to extract consensus expression curves for each group and use comparison algorithms to identify genes that differ between the two groups. Another use, which we discuss in the results section is in an experiment measuring a single treatment. In such experiment researchers are interested in identifying clusters of genes that respond in a specific way to the treatment. As we show, using our method we obtain results that are superior to other methods for combining patient datasets. Finally, we also show that our algorithm can be used to extract patient specific response genes. These genes may be useful for determining individualized response to treatment and disease course and outcome.

1.1 Related work

Time series expression experiments account for over a third of all published microarray datasets and has thus received a lot of attention [4]. However, we are not aware of any computational work on the analysis of time series data from clinical studies. Most previous papers describing such data have relied on simple techniques such as averaging. See Table 1 for some examples.

As mentioned above, there have been a number of methods suggested for aligning two time series expression datasets. Aach *et al* [8] used dynamic programming to align two yeast cell cycle expression datasets based on the measured expression values. Such method can be extended to multiple datasets, but the complexity is exponential in the number of datasets combined, making it impractical for clinical studies. Bar-Joseph et al [9] aligned two datasets by minimizing the area between continuous curves representing expression patterns for genes. It is not immediately clear how this methods can be extended to multiple datasets. In addition, the probabilistic nature of our algorithm allows it to distinguish between patient genes that result from a common response pattern and genes with an expression pattern unique to this patient. Again, it is not clear how an area minimization algorithm could have been extended for this goal.

Gaffney *et al* [11] presented an algorithm that extends the splines framework of Bar-Joseph *et al* discussed above to perform joint clustering and alignment. Unlike our goal of combining multiple expression experiments, their goal was to apply alignment to recover similar patterns in a single expression experiment. In addition, because of the fact that each gene was assumed to have a different response rate, regularization of the translation parameters was required in their approach. In contrast, because we assume one set of translation parameters for all genes in a single patient, such regularization is unnecessary. Finally, their method did not allow for identifying patient specific response patterns.

A number of methods have been suggested to identify differentially expressed genes in time series expression data. These include DiffExp [12] and more recently Edge [13]. It is not clear if and how these could be used to combine large sets of time series data. If we simply treat the different patients as repeats, we falsely identify differentially expressed genes due to differences in patient dynamics.

2 A generative model for expression profiles in clinical experiments

We assume that expression profiles in clinical experiments can be represented using a generative model. Here we discuss the details of this model. In Section 3 we present an algorithm for inferring the parameters of this model. Following the execution of this algorithm we can retrieve consensus expression patterns as well as unique, patient specific, responses.

2.1 Continuous representation of time series expression data

In previous work we described a method for representing expression profiles with continuous curves using cubic splines [9]. Here, we extend this model so that we can combine multiple time series expression datasets. We first briefly review the splines based model and then discuss extensions required for combining multiple time series datasets in the next subsection.

Cubic splines are a set of piecewise cubic polynomials, and are frequently used for fitting time-series and other noisy data. Specifically, we use B-splines, which can be described as a linear combination of a set of basis polynomials [18]. By knowing the value of these splines at a set of control points, one can generate the entire set of polynomials from these basis functions. For a single time series experiment, we assume that a gene can be represented by a spline curve and additional noise using the following equation:

$$Y_i = S(t)F_i + \epsilon_i$$

where Y_i is the expression profile for gene i in this experiment, F_i is a vector of spline control points for gene i and S is a matrix of spline coefficients evaluated at the sampling points of the experiment (t). ϵ_i is a vector of the noise terms, which is assumed to be normally distributed with mean 0. Due to noise and missing values, determining the parameters of the above equation (F_i and ϵ_i) for each gene separately may lead to overfitting. Instead, we constrain the control point values of genes in the same class (co-expressed genes) to co-vary, and thus we use other co-expressed genes to overcome noise and missing values in a single gene. In previous work [9], we showed that this method provides a superior fit for time series expression data when compared to previously used methods.

2.2 Extending continuous representation to multiple experiment

Unlike the above model, which assumes a single measurement for each gene, in clinical experiments we have multiple measurements from different individuals. As mentioned in the introduction, there are two issues that should be addressed. First, we need to allow for patient genes that represent individual response rather than the common response for that gene. Second, we should allow for a patient specific response rate.

To address the first issue, we assume that a patient expression data represents a mixture that includes genes from a patient specific response and genes whose expression follows the common response. This mixture model can be parametrized using a new distribution, w_q, which controls the fraction of genes from patient q that have a unique expression pattern (that is, genes that do not agree with the common response). To constrain the model, all such genes are assumed to have expression values that are sampled from the same Gaussian distribution. Thus, unless a gene instance deviates significantly from its common response it will be assigned to the consensus curve for that gene. To address the second issue, we introduce two new parameters for each patient q, a_q and b_q. These parameters control the stretch (b_q) and shift (a_q) of expression profiles from q w.r.t. the consensus expression curve. In other words, we assume that expression profiles for genes in individual patient lie on different (overlapping) segments of the consensus expression curve. The start and end points of these patient segments are controlled by the time points in which q was sampled and by a_q and b_q. Figure 1 presents the hierarchical graphical model for the generative process we assume in this paper. We denote the vector of values measured for gene g in patient q by $Y_{g,q}$. Below, we summarize the different steps for generating $Y_{g,q}$ according to our model.

1. The first step corresponds to the gene level (left side of Figure 1) and follows the assumptions discussed in Section 2.1. We assume that in order to generate a common (or consensus) expression curve for a gene g we sample a class z according to the class distribution α. Given a class z we use the class mean (μ_z) and sample a gene specific correction term ($\gamma_{g,z}$) using the class covariance matrix ($\varGamma_z$) as discussed in Section 2.1. Summing the vectors μ and γ we obtain the spline control points for the consensus curve.
2. We now turn to the patient level (right side of Figure 1). To generate an instance of gene g in patient q we sample a binary value $w_{g,q}$ according to the *patient specific* individual response distribution, β. If $w_{g,q}$ is 0 then the

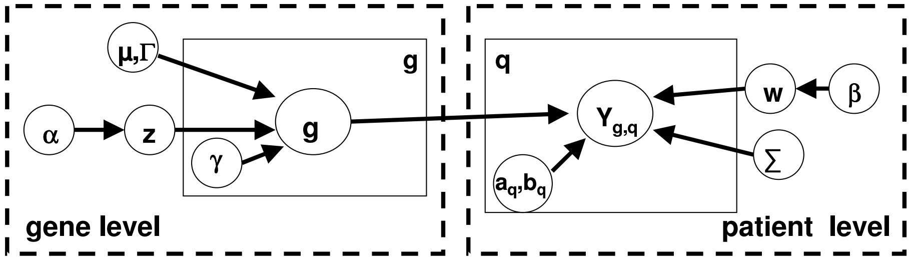


Fig. 1. Graphical representation of our generative model. The left part corresponds to the gene level and the right to the patient level. As can be seen, the two levels are connected. The node annotated with g represents the common response for a gene. An instance of this gene at a certain patients is derived from this curve using the patients response rate parameters (a_q, b_q). Some of the patient genes are not similar to their common response. These genes are assumed to come from separate distribution parameterized by a covariance matrix Σ. See text for complete details on the parameters shown in the figure.

expression of g in q is unique to q and does not come from the consensus expression pattern for g. We thus sample values for entries in $Y_{g,q}$ according to a normal (Gaussian) distribution with a mean of 0 and diagonal covariance matrix Σ. If $w_{g,q}$ is 1 we continue to step 3.

3. When $w_{g,q}$ is 1, we assume that the expression of g in q lies on the consensus expression curve for g, perhaps with some added noise. Recall that we already generated the control points for this curve (in step 1). Since the *response rate* of patient q determines where on the consensus expression curve for g the values of $Y_{g,q}$ will lie, we use a_q and b_q to construct the basis function matrix, and denote it by $S(at + b)$.
4. Finally, the expression values are generated by adding random noise to the segment of the curve that was extracted in step 3.

3 Inferring the parameters of our model

As described in the previous section, our model is probabilistic. The log likelihood of the data given the model is:

$$L(D|\Theta) = \sum_q \sum_g \delta(w_{g,q} = 0) \log \beta_{q,0} P_0(Y_{g,q}) \tag{1}$$

$$+ \sum_q \sum_g \delta(w_{g,q} = 1) \log \beta_{q,1} \tag{2}$$

$$+ \sum_q \sum_g \sum_i \delta(w_{g,q} = 1)\delta(z_g = i) \log \alpha_i P_{i,q}(Y_{g,q}) \tag{3}$$

$$+\sum_{g}\sum_{i}\log P_i(\gamma_g) \tag{4}$$

δ is the Delta function which is 1 if the condition holds and 0 otherwise. $\beta_{0,q}$ ($\beta_{1,q}$) is the fraction of genes that are patient specific (common) for patient q. α_i is the fractions of genes in class i.

The first row corresponds to the patient specific genes which are not a part of the common response. These are modeled using the P_0 distribution. The second and third rows correspond to patient genes that are expressed as the common response profile. These genes are modeled using the parameters of the class and patient they belong to, $P_{i,q}$ (the distribution differs between patients because of their shift and stretch parameters). The last row is from the gene level and involves the likelihood of the gene specific correction term γ.

To infer the parameters of the model we look for parameter assignment that maximize the likelihood of the data given the model. There are a number of normally distributed parameters in our model, including the noise term ϵ and the gene specific parameters γ. For such a model determining the maximum likelihood estimates is a non convex optimization problem (see [19]). We thus turn the the EM algorithm, which iterates between two steps. In the E step we compute the posterior probabilities (or soft assignments) of the indicator variables. In the M step we use these posteriors to find the parameter assignments that maximize the likelihood of our model. Below we discuss these two steps in detail, focusing mainly on the new parameter that were introduced for the patient level.

E Step: The posterior of our missing data (indicators) is $p(w_{g,q}, z_g|Y)$. As can be seen from the third row of Equation 1 the two indicators are coupled and thus $w_{g,q}$ and z_g are *not* independent given Y. However, we can factorize this posterior if either of these indicators were observed or estimated. Specifically, we can write:

$$p(w_{g,q}, z_g|Y) = p(w_{g,q}|Y, z_g)p(z_g|Y)$$

Which can be further expanded to compute $w_{g,q}$ using Bayes rule by setting:

$$p(w_{g,q}|Y, z_g) = \frac{P_{z_g}(Y_{g,q})\beta_{1,q}}{P_{z_g}(Y_{g,q})\beta_{1,q} + P_0(Y_{g,q})\beta_{0,q}}$$

Alternatively we can factor the posterior by conditioning on $w_{g,q}$ which leads to:

$$p(w_{g,q}, z_g|Y) = p(z_g|Y, w_{g,q})p(w_{g,q}|Y)$$

Again, using Bayes rule we can further derive $p(z_g|Y, w_{g,q})$:

$$p(z_g|Y, w_{g,q}) = \frac{\sum_q w_{g,q}\alpha_{z_g}P_{z_g}(Y_{g,q})}{\sum_q\sum_c w_{g,q}\alpha_i P_i(Y_{g,q})}$$

This observation leads to a message passing algorithm which relies on variational methods to compute a joint posterior [20]. We start with an initial guess for z_g (or the values retained in the previous EM iteration). Next, we compute the posterior for $w_{g,q}$ conditioned on Y and z_g. We now send a 'message' along the edge that connects the gene level and the patient level. This message contains our new estimate of $w_{g,q}$ and is used to compute z_g and so forth. This process is repeated until convergence. In [20] it is shown that for graphical models in which nodes can be separated into disjoint 'clusters' (as is the case for the parameters associated with the gene level and the patient level in our model) such message passing algorithm converges to a lower bound estimate of the joint posterior. The estimates we obtain are used to compute the expected log likelihood which is maximized in the M step.

M step: In the M step we maximize the parameters of the algorithm. There are two types of parameters. The first are the gene level parameters $\gamma, \mu, \Gamma, \alpha$ and σ^2. These parameters are the same as the parameters in the original model (section 2.1), that assumed a single dataset. The only difference when maximizing these parameters between the single experiment and multiple experiments settings is the weighting by the patient specific posterior value. For example, the mixture fractions in the *single* experiment case are computed by setting:

$$\alpha_i = \frac{\sum_g z_{i,g}}{\sum_g \sum_j z_{j,g}}$$

where $z_{g,i}$ is the posterior computed in the E step for the indicator z_i for gene g. In the multiple experiment setting these sums are weighted by the patient indicator:

$$\alpha_i = \frac{\sum_q \sum_g w_{g,q} z_{i,g}}{\sum_q \sum_g \sum_j w_{g,q} z_{j,g}}$$

where $w_{g,q}$ is the posterior for the patient common response indicator. Due to lack of space we do not include the complete derivation for the rest of the parameters. The interested reader is referred to [21] for these update rules.

The second type of parameters are the patient specific parameters:β, Σ, a_q, b_q. Taking the derivative w.r.t β we arrive at

$$\beta_{p,1} = \frac{\sum_g w_{g,q}}{n}$$

where n is the total number of genes.

Similarly, because we assume a Gaussian distribution with mean 0 for patient specific parameters, Σ can be computed by setting

$$\Sigma = \frac{\sum_q \sum_g w_{g,q} (Y_{p,g})^T (Y_{p,g})}{\sum_q \sum_g w_{g,q}}$$

And zeroing out not diagonal values (Σ is assumed to be diagonal).

Things are more complicated for the stretch and shift parameters. Changing these parameters results in change to the parameterization of the spline basis functions. Fortunately, splines are continuous in the first derivative and so we can compute $\frac{\partial}{\partial a_q} S(a_q + b_q t)$ and similarly for b_q. This leads to a polynomial function in a and b and an optimum can be computed using gradient descent.

In practice, we have found that it is better to maximize these parameters using line search. For example, in an experiment with a total time of one hour we can limit our interest to a scale of 1 minute for a. This leads to a search for a between 0 and 60 with increments of 1. Similarly, we usually can place upper and lower bounds on b and use a search in increments of .05. Denote by $|a|$ the total number of a value we are looking for and similarly for b. Note that the search for a and b is done independently for each patient and so the running time for of the a, b search is $|q||a||b|n$ where $|q|$ is the number of patients. This is linear in $n|q|$ when $|a|$ and $|b|$ are small.

The total running time of the M step is linear in $|q|nTC$ where T is the number of time points measured and C is the number of clusters. Each iteration in the E step involves updates to the $w_{g,q}$ (there are $n|q|$ such parameters) and z_g (n) parameters. These updates take TC time for the $w_{g,q}$ and $TC|q|$ for the z_g parameters. The E step usually converges within a few iteration and thus the total running time of the algorithm is $|q|nTC$ which is linear in n when the number of time points, clusters and patient is small.

4 Results

We have tested our algorithm on simulated data and on two biological datasets. The first biological dataset is from a non clinical experiment in yeast and the second is from a clinical multiple sclerosis (MS) experiment. While the yeast dataset is not the target of this method, since it was studied before using pairwise alignment methods it is an appropriate dataset for comparing the results of the two approaches.

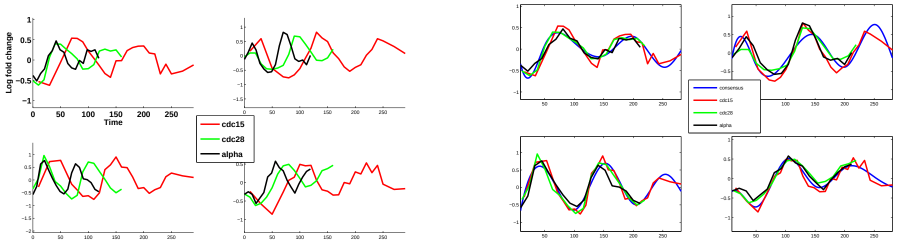
4.1 Simulated data

We generated three datasets. The first contained 100 rows of sines and 100 rows of cosine measured between 0 and 4π every $\pi/4$. The other two also contained sines and cosines, however, both were measured starting at a different point $(\pi/4)$ and with different rates (one slower by 0.8 and the other faster by 1.2). We added normally distributed random noise to each value in each dataset and used our algorithm to cluster and combine the three dataset.

Our algorithm was able to correctly retrieve both the shift and translation parameters for the second and third datasets. In addition, the algorithm correctly clustered the rows, perfectly separating the (shifted and stretched) sines and cosines. See website [22] for full details and figures.

4.2 Yeast cell cycle data

As mentioned above, a number of previous methods were suggested for aligning the three yeast cell cycle expression experiments from Spellman *et al* [1]. In that paper, yeast cells were arrested using three different methods (cdc15, cdc28 and alpha) and results were combined to identify cell cycle genes. However, as noted in that paper and subsequent papers, each of these methods arrests cells at different points along the cycle. In addition, different arrest methods result in different cell cycle duration (ranging from 60 minutes to almost two hours). This has led to a number of suggestion's for aligning these datasets. Some of the suggested methods use the peak expression time for alignment. Such methods are only appropriate for cell cycle data and are thus not appropriate for the more general setting we consider in this paper. In [9] we presented a method for pairwise alignment of time series data by minimizing the squared integral between expression profiles of genes was presented, and it was shown that this method outperform previously suggested methods.



(a) Measured values

(b) Aligned curves

Fig. 2. Aligning yeast cell cycle datasets. (a) Curves plotted based on measured time. (b) Curves aligned using our algorithm. Note that by aligning the different experiments we can clearly see the agreement between the three repeats.

We have used the algorithm discussed in this paper to combine the the three datasets. Figure 2 (a) presents the average expression value for genes in four clusters using the measured time points. Figure 2 (b) presents the same curves, this time aligned according to the results of our algorithm. The clusters were determined according to the gene level parameter assignments. The blue curve represents the consensus curve computed for this cluster. As can be seen, our algorithm was successful in aligning the three datasets. The alignment shows the inherent agreement between the three experiments, indicating that the data truly represents cyclic behavior.

Next ,we compared our results to the results in [9]. While the value for the stretch and shift parameters where slightly different, overall the results where very similar (both shift parameters where within 10 minutes and stretch parameters varied less than 10%). To quantify the differences we have computed the area between the gene curves in different arrest methods following alignment using the stretch and shift parameters determined by our algorithm and the corresponding parameters from [9]. While this quantity (area difference) is the target function to be minimized by that the algorithm, our algorithm has a slightly different objective function (based on the noise distribution). Based on the parameters in [9] we obtained an average absolute difference of 0.335, between cdc15 and alpha, 0.304 between cdc15 and cdc28 and 0.305 between cdc28 and alpha. The corresponding numbers for our algorithm were 0.330, 0.308 and 0.310. Thus, the values obtained by our algorithm were either very close or in one case better than the values obtained by the pairwise alignment method. As mentioned in the introduction, such pairwise alignment may not be appropriate for larger datasets. Thus, our algorithm, which is designed for larger datasets, allows us to enjoy both worlds, multiple datasets alignment with high quality results.

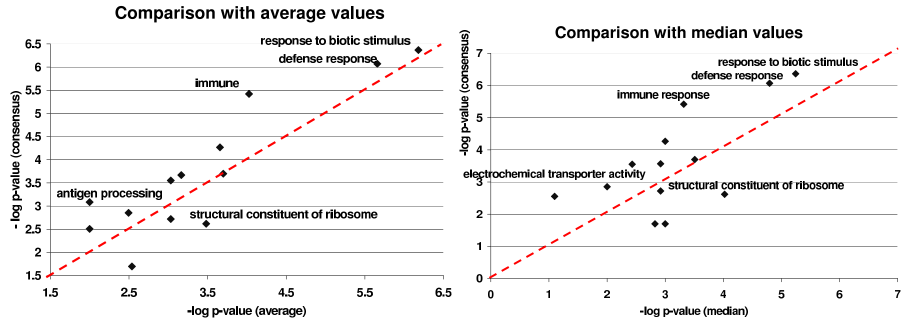
4.3 Clinical data from multiple sclerosis patients

To test our algorithm on clinical expression data, we used data from experiments that were carried out to test the effects of interferon-β (IFN-β) on multiple sclerosis (MS) patients [6]. In that experiment, eight patients with active relapsing MS were given IFN-β therapy. Peripheral blood was obtained from these patients at 9 time points (0, 1, 2, 4, 8, 24, 48, 120 and 168 hours).

In the original paper, the authors analyzed a small number of pre-selected genes using their average expression values. However, they also noted that such a method may be problematic. Based on figures for a number of genes in five of the patients they conclude that: “The dynamics of the three patients ... are different from those of the other two patients”. We have applied our algorithm to cluster this data, to infer patient dynamics and to extract the consensus response curve for each of the genes. In the results we discuss below we focus on six of the eight patients, mainly for presentation reasons.

Our algorithm determined that all patients have the same shift value ($a_p = 0$). This is reasonable, since unlike clinical trials that measure response with an unknown start time (for example, disease progression) these experiments measured response to treatment which was given at the same time in all patients. In contrast, our algorithm found that the dynamics of two of the patients where different from the other four. While the first four patients had a stretch parameter of 1 or .96, the other two had a stretch value of 0.84 indicating they they responded slower the the other four patients. These results are in good agreement with the (anecdotal) observations in the original paper mentioned above.

As mentioned before (Table 1), previous attempts to combine patient expression data used the average expression value for each time point. To compare our results with these methods we used k-means to cluster the average and median expression values from all six patients. Following several previous papers we used the Gene Ontology (GO) annotations to compare these sets of clusters. Figure 3 presents the results of these comparison using the negative log p-value. The set of categories that were retrieved are reasonable for this experiment. For example our analysis reveals an enrichment of inflammatory related GO annotations (including immune and defense responses, Figure 3). These results agree well with previous studies [23]. In both case (average and median), our algorithm



(a) Measured value (b) Aligned curves profile.

Fig. 3. Using GO to compare our results to other methods that have been suggested for combining patient data. (a) Comparison of the clustering achieved by our algorithm (y axis) with clustering of the average expression values using k-means (x axis). The same number of clusters (3) were used in both cases. P-value enrichments for GO categories were computed using the hypergeometric distribution. As can be seen, in most cases our algorithm was able to better separate the clusters, obtaining higher enrichments for relevant GO categories. (b) A similar comparison when using the median patient value instead of the average. See supporting website [22] for complete lists of categories and p-values.

was able to obtain better enrichment for 9 categories, whereas the other method was better at only 3 (average) or 4 (median). Note also that for the most enriched categories (with a p-value $< 10^{-5}$) our algorithm was always better than the other methods.

One of the main reasons for our improved results in this case is the ability of our algorithm to exclude certain expression instances (genes in specific patients) if it believes that these were not coming from the common response curve. Thus, the curves that are clustered are more coherent and the algorithm is able to infer better groupings for sets of genes. To examine this and to test the ability of our algorithm to identify specific patient responses we compared two sets of genes with different posteriors. The first set (top row of Figure 4) contains genes for which the posterior ($w_{g,q}$) in all patients were above 0.9. The second set (bottom row of Figure 4) contains genes for which 5 of the 6 patient instances had a posterior greater then 0.9 and the six patient had a posterior less than 0.1. Many of the genes in the first (always common) case are known interferon induced genes, including the protein kinase, interferon-inducible PRKR and the interferon regulatory factors 5 and 7 (IRF5, IRF7). However, a number of known response genes are also included in the second set, indicating that they may have different function in different patients. These include the tyrosine kinase TXK that functions as a Th1 cell-specific transcription factor that regulates IFN-gamma gene transcription [24] and the immunoglobulin lambda-like polypeptide 1 (IGLL1) which participates in immune response. While the meaning of this failure to induce a subset of genes in one patient is unclear, tools like ours that allow individual analysis of temporal changes in gene expression may lead to better diagnosis and treatment based on individual responses. See supporting website [22] for complete lists of genes in each of these sets.

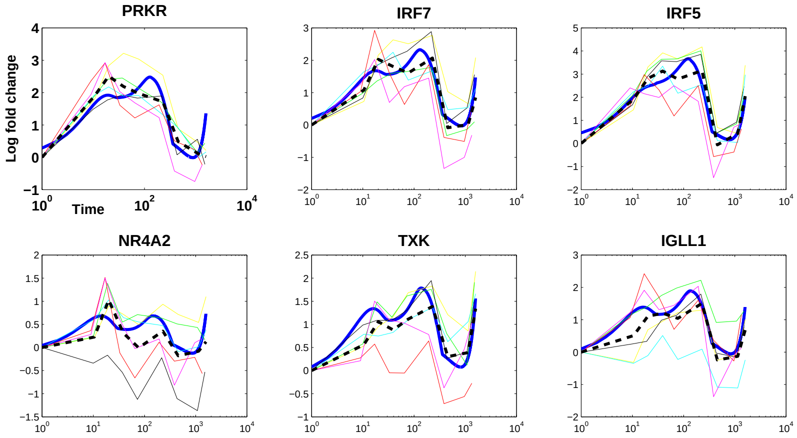


Fig. 4. Top row: Three genes expressed in a similar way in all six patients according to the posterior computed by our algorithm. Bottom row: Three genes that were expressed similarly in five of the patients, but differently in the six. Time is on a log scale due to the sampling rate. Note that the average computed in the bottom row (dotted line) is affected by the outlier, while the consensus computed by our algorithm (blue line) is not.

5 Conclusions and future work

We have presented the first algorithm for combining time series expression data in clinical experiments. In these experiments researchers face a number of challenges including problems related to different response rates in individuals and the differences in baseline expression values for many genes. Using a hierarchical generative model, we were able to provide a probabilistic algorithm that can solve the above problems. Our algorithm generates a consensus expression curve for each gene. We then learn the patient response rates, which in turn determine where on the curves the values for this patient lies. Our algorithm can also identify genes that represent patient specific response and remove them when computing the consensus expression curve.

There are a number of future directions we wish to explore. First, our assumption that patients have a single response rate may be too strict. As an alternative, we can assume that different pathways (or clusters) have unique response rates, and that these may be different even in a single patient. Another possible extension is to cluster the patients as well as the genes. In such application we will assume that groups of patients can be represented by consensus expression curves. We will then employ an algorithm to identify the different patients clusters, and for each cluster to determine its expression profiles.

References

1. Spellman, P. T. ,Sherlock, G. , Zhang, M.Q. , Iyer, V.R., *et al*: Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisia* by microarray hybridization. Mol. Biol. of the Cell. **9** (1998) 3273–3297
2. Whitfield, M.L., Sherlock, G., Saldanha, A.J., Murray, J.I., *et al*: Identification of genes periodically expressed in the human cell cycle and their expression in tumors. Mol. Biol. Cell. **13(6)** (2002) 1977–2000
3. Inflammation and the Host Response to Injury. URL: www.gluegrant.org
4. Bar-Joseph, Z.: Analyzing time series gene expression data. Bioinformatics. **20(16)** (2004) 2493–2503
5. Nau, G.J., Richmond, J.F.L., Schlesinger, A., Jennings, E.G., *et al*: Human Macrophage Activation Programs Induced by Bacterial Pathogens. PNAS. **99** (2002) 1503–1508
6. Weinstock-Guttman, B., Badgett, D., Patrick, K., Hartrich, L., *et al*: Genomic effects of IFN-beta in multiple sclerosis patients. J Immunol. **171(5)** (2002) 1503–1508
7. Sterrenburg, E., Turk, R., Peter, A.C., Hoen, P.A., van Deutekom, J.C., *et al*: Large-scale gene expression analysis of human skeletal myoblast differentiation. Neuromuscul Disord. **14(8-9)** (2004) 507–518
8. Aach, J., Church, G. M.: Aligning gene expression time series with time warping algorithms. Bioinformatics. **17** (2001) 495–508
9. Bar-Joseph, Z., Gerber, G., Jaakkola, T.S., Gifford, D.K., Simon, I.: Continuous Representations of Time Series Gene Expression Data. Journal of Computational Biology. **3-4** (2003) 39–48
10. Calvano, S.E., Xiao, W., Richards, D.R., Felciano, R.M., *et al*: A network-based analysis of systemic inflammation in humans. Nature. **437** (2005) 1032–7
11. S. Gaffney, P. Smyth: Joint Probabilistic Curve Clustering and Alignmen. Proceedings of The Eighteenth Annual Conference on Neural Information Processing Systems (NIPS). (2004)
12. Bar-Joseph, Z., Gerber, G., Jaakkola, T.S., Gifford, D.K., Simon, I.: Comparing the Continuous Representation of Time Series Expression Profiles to Identify Differentially Expressed Genes. PNAS. **100(18)** (2003) 10146–51
13. Storey, J.D., Xiao, W., Leek, J.T., Tompkins, R.G., Davis, R.W.: Significance analysis of time course microarray experiments. PNAS. **102(36)** (2005) 12837–42
14. Michalek, R., Tarantello, G.: Subharmonic solutions with prescribed minimal period for nonautonomous Hamiltonian systems. J. Diff. Eq. **72** (1988) 28–55
15. Tarantello, G.: Subharmonic solutions for Hamiltonian systems via a $\mathbb{Z}_p$ pseudoindex theory. Annali di Matematica Pura (to appear)

16. Troyanskaya, O., Cantor, M., *et al*: Missing value estimation methods for DNA microarrays. Bioinformatics. **17** (2001) 520–525
17. Sharan R, Shamir R.: Algorithmic Approaches to Clustering Gene Expression Data. Current Topics in Computational Biology. (2002) 269–300
18. Piegl, L., Tiller, W.: The NURBS Book. Springer-Verlag. New York (1997)
19. James, G., Hastie, T.: Functional Linear Discriminant Analysis for Irregularly Sampled Curves. Journal of the Royal Statistical Society, Series B. **63** (2001) 533–550
20. Xing, E.P., Jordan, M.I., Russell, S.: A generalized mean field algorithm for variational inference in exponential families. Proceedings of Uncertainty in Artificial Intelligence (UAI). (2003) 583-591
21. Bar-Joseph, Z.,Gerber, G.,Jaakkola, T.S., Gifford, D.K., Simon, I.: Continuous Representations of Time Series Gene Expression Data. Journal of Computational Biology. **3-4** (2003) 341–356
22. Supporting website: URL: www.cs.cmu.edu/∼zivbj/comb/combpatient.html
23. Achiron, A., *et al*: Blood transcriptional signatures of multiple sclerosis: unique gene expression of disease activity. Ann Neurol. **55(3)** (2004) 410–17
24. Takeba, Y., *et al*: Txk, a member of nonreceptor tyrosine kinase of Tec family, acts as a Th1 cell-specific transcription factor and regulates IFN-gamma gene transcription. J Immunol. **168(5)** (2002) 2365–70

This article was processed using the LaTeX macro package with LLNCS style