PairGP: Gaussian process modeling of longitudinal data from paired multi-condition studies

Michele Vantini *[1], Henrik Mannerström[1], Sini Rautio[1], Helena Ahlfors[2,3], Brigitta Stockinger[2], and Harri Lähdesmäki[1]

[1]Department of Computer Science, Aalto University, Finland
[2]The Francis Crick Institute, London, UK
[3]Autolus Ltd, London, UK

August 11, 2020

Abstract

We propose PairGP, a non-stationary Gaussian process method to compare gene expression time-series across several conditions that can account for paired longitudinal study designs and can identify groups of conditions that have different gene expression dynamics. We demonstrate the method on both simulated data and previously unpublished RNA-seq time-series with five conditions. The results show the advantage of modeling the pairing effect to better identify groups of conditions with different dynamics. The implementations is available at https://github.com/michelevantini/PairGP

1 Introduction

Gene expression time-series studies have become popular as they can reveal dynamics of transcriptional processes. These studies typically use longitudinal experimental designs where repeated measurements (over time) of each cell sample are collected. A common study design involves comparisons between treatments, or conditions, and the goal is to identify groups of conditions that have different gene expression dynamics. Further, to reduce variability between conditions and to increase statistical power, biological samples in different conditions are typically matched, resulting in paired longitudinal designs. Thus, it is important to take the paired design into account in the data analysis in order to reveal the true differences between different treatments.

Standard methods for longitudinal data analysis include linear mixed effect (LME) models. A number of non-linear, non-stationary and non-parametric methods for gene expression time-series have been proposed using Gaussian processes (GP) (see Supplementary materials for related research). Recently, we have developed GP based methods to implement Bayesian non-parametrics for longitudinal studies [1, 2] that can also be applied to data from paired longitudinal designs. However, posterior sampling for such models has a higher computational cost and does not scale efficiently to genome-wide studies. We propose PairGP, a GP method for paired, multi-condition longitudinal designs. The method is tested on simulated data, longitudinal gene expression data involving two treatments, and a previously unpublished longitudinal RNA-seq data from five treatments.

2 Methods

Each measured gene expression time-series is modeled as a combination of three components; 1) the response model, 2) the pairing model, and 3) uncorrelated random noise fluctuations. The response model is inferred from the data, so that all treatments that produce similar responses share a common response model. The pairing model is shared by all measurements coming from the same biolog-

*corresponding author

ical replicate or batch, and models the deviation from the response model. To enforce that the pairing model does not confound the response model, the sum of all the pairing model components is constrained to zero, as explained below. The model considers each gene separately. The measured gene expression x is transformed as $y = \log(x + 1)$ so that it can be more accurately modeled by a normal distribution. Most gene expression experiments are "hit-and-run", where the changes are rapid in the beginning and then slow down, thus, making it a non-stationary process. To model the non-stationarity, the user is given the choice to transform the wall-clock time $\tilde{t}$ as $t = \omega(\tilde{t}) = \log(1 + \tilde{t})$. This transformation was used in all analyses reported below.

The standardized measurements of treatment (condition) $c \in \{1, \ldots, C\}$ and pairing $p \in \{1, \ldots, P\}$ is modeled as

$$y_{cp}(t) = f_r(t) + f_p(t) + \epsilon,$$

where $\epsilon \sim N(0, \sigma_\epsilon^2)$. Each response effect f_r is a GP with the exponentiated quadratic (EQ) kernel $k_r(t, t') = \sigma_r^2 \exp[-\frac{1}{2}\ell_r^{-2}(t - t')^2]$, where σ_r^2 is the variance and ℓ_r the length scale of response effect r. For each pairing p, the pairing effect f_p is modeled with a centered EQ kernel $k_\rho((p, t), (p, t')) = \sigma_\rho^2 \exp[-\frac{1}{2}\ell_\rho^{-2}(t - t')^2]$, where σ_ρ^2 is the common variance and ℓ_ρ the common length scale of the pairing effect. The centered EQ kernel has negative covariance between the pairing effects f_p and $f_{p'}$ $(p \neq p')$ to force their sum to zero, i.e, $k_\rho((p, t), (p', t')) = -\frac{1}{P-1}\sigma_\rho^2 \exp[-\frac{1}{2}\ell_\rho^{-2}(t - t')^2]$ when $p \neq p'$ [2]. Note that the response and pairing GPs are non-stationary as the logarithmic time transformation corresponds to input-warped GPs with kernel $k(\tilde{t}, \tilde{t}') = \sigma^2 \exp[-\frac{1}{2}\ell^{-2}(\omega(\tilde{t}) - \omega(\tilde{t}'))^2]$. Prior distributions of hyperparameters used to analyze real data are described in Suppl. Material.

For each gene, all the partitionings of the treatments are modeled, and the one with the largest marginal likelihood (type-II) is selected as the correct response model. For example, an experiment with three treatments c_1, c_2 and c_3 evaluates five different partitionings (models) for each gene: 1) all the three treatments have a similar response, and there is only one response model: $r_1 = \{c_1, c_2, c_3\}$; 2) treatment c_1 has a different response compared to c_2 and c_3, and the two response models are $r_1 = \{c_1\}$ and $r_2 = \{c_2, c_3\}$; 3) same as (2) but with treatment c_2 singled out, $r_1 = \{c_2\}$ and $r_2 = \{c_1, c_3\}$; 4) same as (2) but with treatment c_3 singled out, $r_1 = \{c_3\}$ and $r_2 = \{c_1, c_2\}$; and 5) all three treatments produce different responses, $r_1 = \{c_1\}$, $r_2 = \{c_2\}$, and $r_3 = \{c_3\}$.

The above method is implemented using the GPy package [3]. Instructions for the usage are available on the github page.

3 Results

We first tested our method on simulated time-series data with different number of treatments and a varying amount of pairing effect size (see Suppl. Material for simulation details). Comparing our model to an otherwise equal GP model but without the pairing effect shows that modeling the pairing component improves the identification of correct partitioning (Fig. 1, Suppl. Fig. 2).

Next, we applied our method to microarray-based longitudinal gene expression data measured from activated CD4+ human T cells (Th0) and cells differentiated towards T helper 2 (Th2) cell type with three paired replicates [4]. We identified genes that respond differentially between Th0 and Th2 during the first 72 hours of differentiation (Suppl. Figs. 3 and 4, Suppl. Table 1).

We also applied our method to previously unpublished, longitudinal RNA-seq data measured from CD4+ mouse T cells that were either activated or differentiating towards Th17 lineage. Experiments include six cell cultures and five different treatments: two treatments (Th0, Th17) applied for the first three cultures and three treatments (Th17+IL1b, Th17+IL21, Th17+IL1b+IL21) for the last three cultures, resulting in two groups of three paired replicates (see Suppl. Material). Our model identifies genes that have different dynamics in different subsets of the five treatments. One example gene (Fasl) is shown in Fig. 1 and more examples are shown in Suppl. Figs. 5-6. Suppl. Table 2 summarizes how the pairing effect affects the proportion of genes detected for each partition.

4 Conclusions

We have implemented a GP-based model for analysis of longitudinal gene expression data that accounts for paired multi-condition study designs. Results demonstrate that our model improves the detection of correct partitioning of different conditions.

Acknowledgements and Funding

This work has been supported by the Academy of Finland grants no. 292660 and 313271.

References

[1] Lu Cheng, Siddharth Ramchandran, Tommi Vatanen, Niina Lietzen, Riitta Lahesmaa, Aki Vehtari, and Harri Lähdesmäki. An additive gaussian process regression model for interpretable non-parametric analysis of longitudinal data. *Nature Communications*, 10(1798), 2019.

[2] Juho Timonen, Henrik Mannerström, Aki Vehtari, and Harri Lähdesmäki. An interpretable probabilistic machine learning method for heterogeneous longitudinal studies. *arXiv preprint arXiv:1912.03549*, 2019. https://arxiv.org/abs/1912.03549.

[3] GPy. GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy, since 2012.

[4] Laura L Elo, Henna Järvenpää, Soile Tuomela, Sunil Raghav, Helena Ahlfors, Kirsti Laurila, Bhawna Gupta, Riikka J Lund, Johanna Tahvanainen, R David Hawkins, et al. Genome-wide profiling of interleukin-4 and stat6 transcription factor regulation of human th2 cell programming. *Immunity*, 32(6):852–862, 2010.

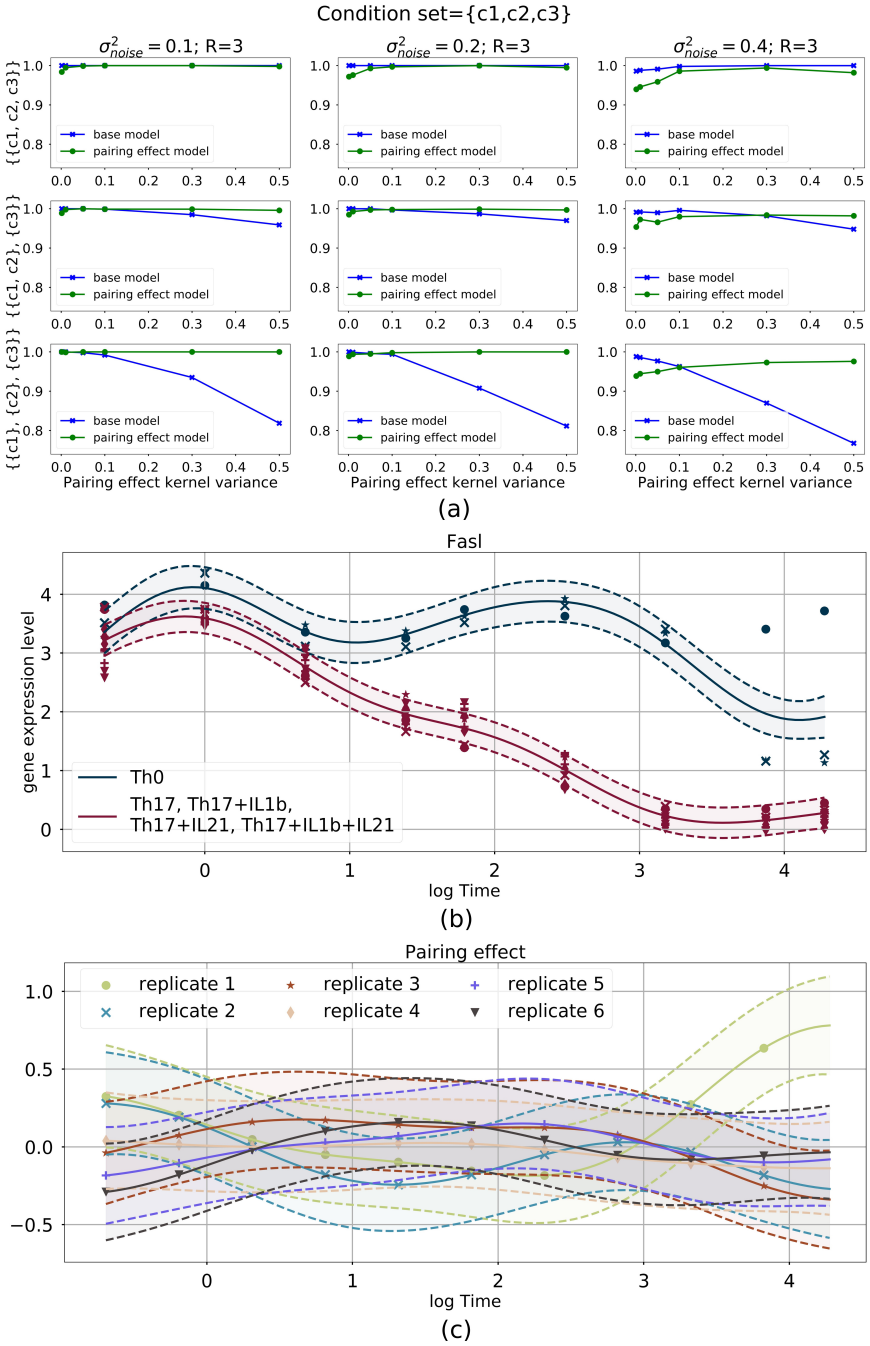

Figure 1: (a) Accuracy of the model inference for three treatments without pairing effect (base model) and our model with pairing effect. The accuracies (y-axis) are computed with simulated data where pairing effect (x-axis) ranges from zero (no pairing effect) to substantial (0.5). The correct partitioning of treatments for each row is indicated on left. (b) Result of the pairing effect model on the gene *Fasl*. Different colors indicate different subsets of the identified optimal partitioning and different markers represent data points coming from different replicates. (c) The pairing effect learnt from the data.

PairGP: Gaussian process modeling of longitudinal data from paired multi-condition studies

Supplementary material

Michele Vantini *[1], Henrik Mannerström[1], Sini Rautio[1], Helena Ahlfors[2,3], Brigitta Stockinger[2], and Harri Lähdesmäki[1]

[1]Department of Computer Science, Aalto University, Finland
[2]The Francis Crick Institute, London, UK
[3]Autolus Ltd, London, UK

August 11, 2020

Contents

*corresponding author

In this supplementary material we present the data sets that have been used in this study, details of the data simulation, some implementation details of our Gaussian process method and supplementary results.

1 Data

The methods have been developed generically and can be applied to any data set that has the following structure: the data set is made of N genes, C conditions (or treatments), P replicates and T time points for each gene. Thus, we have $C \times P$ time-series of length T for each gene. We used simulated data and two gene expression time-series data sets.

1.1 Simulated data

To simulate the data, we simulated one GP for each response and one GP for each replicate pair with a fixed set of hyperparameters. To simulate the data for a specific condition c and for a specific replicate pair p we use the following formulation

$$y_{cp}(t) = f_r(t) + f_p(t) + \epsilon,$$

where f_r and f_p are GPs

$$f_r \sim \mathcal{GP}(\mathbf{0}, k_r(x, x'))$$
$$f_p \sim \mathcal{GP}(\mathbf{0}, k_p(x, x'))$$

and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is a random noise term. Recall that the C treatments (or conditions) result in R different responses, depending on the partitioning (or model; see below), and each treatment c belong to one of the R responses. We used the same $T = 9$ time points 0.5h, 1h, 2h, 4h, 6h, 12h, 24h, 48h, 72h, as with the real data (see below). Once the kernel hyperparameters are fixed, then to simulate the data for a gene with C conditions and P replicates we simulate one realization from f_r for each response effect, and one realization from f_p for each replicate pair and we combine them together with additive noise as shown above. As a result, we obtain $C \times R \times T$ time points for each simulated gene.

In our settings, we used the EQ kernels with lengthscale $l_r = 1.0$ and variance $\sigma_r^2 = 1.0$ for the response kernel k_r, and $l_p = 1.0$ and $\sigma_p^2 \in [0.001, 0.01, 0.05, 0.1, 0.3, 0.5]$ for the pairing effect kernel k_p. If the individuals that are participating to the study are, for example, studied in a controlled environment, such as laboratory mice, then the variation between individuals is expected to be smaller compared to studies done with humans. Therefore, we decided to simulate data with different levels of pairing effect variance, aiming to cover several possible values. Additionally, for the random noise variance σ_ϵ^2 we used the set of values $[0.1, 0.2, 0.4]$. Similarly as for the pairing effect variance, the noise variance can also vary depending on the experiments. Thus, we want to assess the performance of the pairing effect model on simulated data as a function of the pairing effect variance and the noise variance, and compare the results to those obtained with the base model on the same data. We simulated gene expression time-series data with 3 and 4 conditions and 3 replicates with all the combinations of parameter values described above.

To evaluate the accuracy on simulated data we generated several simulated data sets, which is composed of genes generated according to each of the possible partition of the conditions. In particular, we used all the combinations of pairing effect variance, noise variance and number of conditions mentioned above to analyse how the pairing effect model behaves under several different settings. A total of 1000 genes for each partition have been simulated in each data set. However, some partitions of the conditions are *de facto* the same, for example, simulating data as $\{\{c_1\}, \{c_2, c_3\}\}$ or as $\{\{c_1, c_2\}, \{c_3\}\}$ is equivalent. Therefore, we simulate data for 3 partitions for a data set with 3 conditions, and 5 partitions for a data set with 4 conditions. Importantly, however, the model during the inference process can still choose between all the possible partitions of the condition set.

1.2 Human T-helper cell differentiation data

The first data set contains gene expression time-series data from human CD4+ T cells measured using microarrays originally published in [1]. We use data from two treatments measured at time points 0.5h, 1h, 2h, 4h, 6h, 12h, 24h, 48h, 72h. Th0 condition (or treatment) corresponds to activation of naive CD4+ T cells, and Th2 corresponds to activation and differentiation of naive CD4+ cells towards T helper 2 (Th2) lineage. Both conditions (across all timepoints) are measured from three cell cultures that correspond to three biological replicates that are paired across the conditions. Microarray data is RMA preprocessed as in [1] and further standardized.

1.3 Mouse T-helper cell differentiation data

The second data set has been collected from laboratory mice, and it has a total of five treatments and six cell cultures (i.e., biological replicates). The experimental details are as in [2]. Th0 treatment corresponds to activation of naive T cells. The other four treatments are Th17, Th17+IL1b, Th17+IL21, Th17+IL1b+IL21. Th17 corresponds to activation and differentiation of naive CD4+ cells towards T helper 17 (Th17) lineage. Th17+IL1b, Th17+IL21, Th17+IL1b+IL21 treatments corresponds to simultaneous activation and differentiation of naive CD4+ cells towards Th17 lineage and treatment with interleukin 1 beta (IL-1β), interleukin 21 (IL-21) and combination of IL-1β and IL-21 (with concentration 20 ng/ml) (R&D Systems), respectively. Experimental data for the treatments Th0 and Th17 have been measured from the first three replicates (cell cultures), using a paired design. Experimental data for the other three treatments have been measured from the other three replicates (cell cultures), again using a paired design. Cells are sampled for gene expression analysis at nine time points: 0.5h, 1h, 2h, 4h, 6h, 12h, 24h, 48h, 72h. Sequence reads were mapped with TopHat to mouse mm9 genome as well as to Ensembl transcriptome. After the alignment, the number of reads that mapped to each gene were summarized using HTSEQ-count tool. The raw RNA-seq data used in this manuscript will be made available upon publication via Gene Expression Omnibus (GEO).

1.4 Data standardization

After quantification of gene expression count data from the RNA-seq data, the expression data is further log-transformed. Microarray and RNA-seq data are standardized before analysis.

2 Methods

2.1 Previous methods

Gene expression microarray and RNA-seq techniques allow quantitative, genome-wide analysis of gene expression levels. A number of software tools are available for statistical analysis of gene expression data measured by microarrays (e.g. LIMMA [3]) and RNA-seq (e.g. DEseq [4] and edgeR [5]). These tools rely on linear and generalized linear models, use empirical Bayes to share information between genes, allow modeling complex experimental designs, and support testing a variety of hypothesis, but are not designed for longitudinal studies that involve repeated measurements of individuals over time. Standard methods for longitudinal data analysis include linear and generalized linear mixed effect (LME) models, as implemented in e.g. lme4 package [6]. Bayesian alternatives for modeling gene expression time-series data have been proposed e.g. in [7, 8] that also support non-Gaussian likelihood models. Methods of gene expression time-series data analysis include also *lmms* [9] and ImpulseDE2 [10]. The former is based on linear mixed models and ANOVA log likelihood ratio tests, while the latter is based on an impulse model as a continuous representation of temporal responses.

A number of non-linear, non-stationary and non-parametric methods for gene expression time series have been proposed using Gaussian processes (GP). Yuan was among the first who used GPs to model gene expression time course data [11]. A number of improved methods have been proposed, such as methods that can account for outliers [12], a method for analyzing multiple conditions [13], methods that identify

time intervals of differential expression [12, 14], and methods for accounting time delays between replicates and non-Gaussian likelihood models [15]. However, none of these tools can account for paired experimental designs that are commonly used in biological studies. Similar ideas have been proposed in the context of GP-based clustering of time-series data [16], where authors propose a hierarchical GP regression model. Nonetheless, the effects in [16] are not across replicate pairs but, instead, a different replicate effect is learned for each individual condition. To that end, Spies et al. [17] provide an extensive review of a large selection of methods proposed in the literature for time course data.

Recently, we have developed GP based methods to implement Bayesian non-parametrics for longitudinal studies [18, 19] that can also be applied to data from paired longitudinal designs. However, posterior sampling for such models has high computational cost. We propose a non-stationary GP method for paired, multi-condition longitudinal designs that provides efficient analysis for genome-wide studies.

2.2 Model selection

Given that an experiment contains C treatments, they can be partitioned into B_C different partitionings (or models), where

$$B_C = \sum_{k=0}^{C-1} \binom{C-1}{k} B_k \tag{1}$$

is the Bell number. For example, Bell number for 3, 4 and 5 treatments are $B_3 = 5$, $B_4 = 15$, and $B_5 = 52$. For each partitioning, we evaluate the marginal likelihood

$$\log p(\boldsymbol{y}|X, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^T(\boldsymbol{K}_{X,X} + \sigma_\epsilon^2 I)^{-1}\boldsymbol{y} - \frac{1}{2}\log\left|(\boldsymbol{K}_{X,X} + \sigma_\epsilon^2 I)\right| - \frac{n}{2}\log 2\pi, \tag{2}$$

where $\boldsymbol{y} \in \mathbb{R}^{C \cdot P \cdot T}$ contains the standardized gene expression data for a gene from all C treatments, P replicates and T time points, $X = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{C \cdot P \cdot T})$ contains the explanatory covariates (treatment c, replicate p and time point t) for each measurement, $\boldsymbol{\theta}$ is a vector containing all the kernel hyperparameters, $\boldsymbol{K}_{X,X}$ is the sum of the response covariance matrix and the pairing covariance matrix defined by the centered EQ kernels, σ_ϵ^2 is the Gaussian random noise variance, and $n = CPT$. An example of the covariance matrix $\boldsymbol{K}_{X,X}$ and its components $\boldsymbol{K}_r$ and $\boldsymbol{K}_p$ are shown in Figures 1.

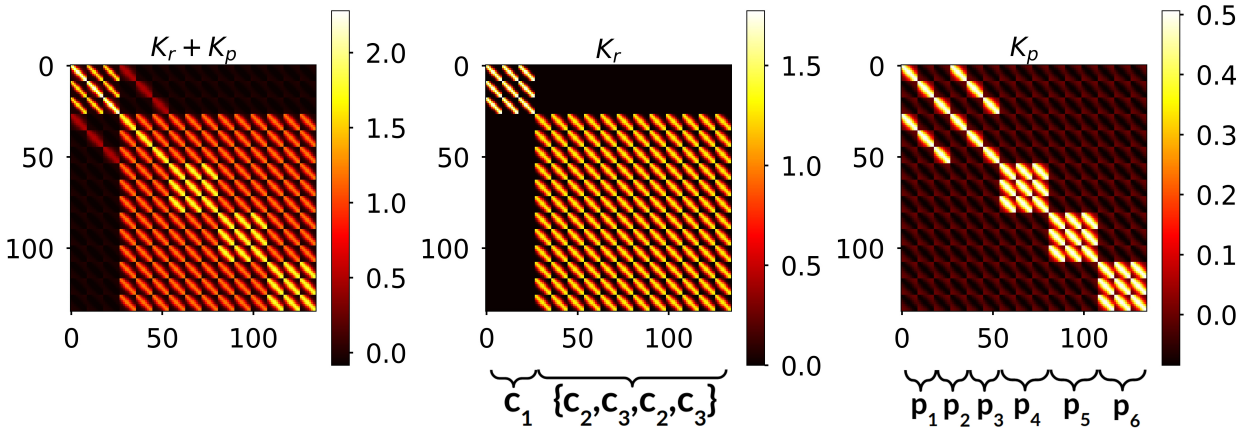

Figure 1: An example of the covariance matrices. (left) the combination of covariance matrices $\boldsymbol{K}_{X,X} = \boldsymbol{K}_r + \boldsymbol{K}_p$ without centering. (middle) the response covariance matrix $\boldsymbol{K}_r$. (right) the pairing covariance matrix $\boldsymbol{K}_p$ without centering. In this example, there are data from 5 different conditions and conditions are assumed to be partitioned into 3 response models $r_1 = \{c_1\}$, $r_2 = \{c_2, c_3\}$ and $r_3 = \{c_4, c_5\}$.

We call the model presented above the pairing effect model. To assess the performance of this model, we compare it against the base model. The base model is obtained by optimizing one GP regression model

for each possible subset of the condition set, and then combining the score of these models to have a score for each partitioning of the condition set. In other words, the log marginal likelihood $\log p(\boldsymbol{y}|X, \boldsymbol{\theta})$ of the models of different subsets is summed up to obtain the score for the partitioning that corresponds to the set of considered subsets. In the base model, we use EQ kernels which model the response functions, but not the pairing effect. In the pairing effect model we standardize the data of all the conditions together, while in the base model we standardize separately the data of the sets of conditions that corresponds to the different response functions.

2.3 Prior distribution for kernel hyperparameters

Typically, the hyperparameter optimization is done by maximizing the log marginal likelihood of the model. If one has prior information about the hyperparameters, then in a hierarchical structure one can also impose prior distribution on the hyperparameter, also called hyperprior. The kernel choice for the GP regression models is the exponentiated quadratic (EQ) kernel. This means that we can define hyperpriors for the variance σ^2 and the lengthscale ℓ. The assumption that we have on the data are essentially two:

- the lengthscale parameters for the response effect kernels should be relatively high as higher lengthscale parameters imply smoother functions.
- The variance parameter for the pairing effect kernel must be relatively small; the magnitude of the pairing effect cannot be as high as the response effect, but it should just represent slight variation around condition mean that is associated with the different replicates. The only exception to this is when a gene is silent. In this case, the variation in the gene expression over time is almost 0, thus the variation due to the effect introduced by the different replicate can be potentially higher.

We use the log-Gaussian distribution log-Normal(μ, σ^2) with $\mu = 0.5$, $\sigma^2 = 0.5$ as hyperprior distribution for the lengthscale of the condition effect, exponential distribution Exp(λ) with $\lambda = 2$ for the pairing effect variance and log-Gaussian distribution log-Normal(μ, σ^2) with $\mu = 0$, $\sigma^2 = 0.5$ for the pairing effect lengthscale. We do not use here any hyperprior distribution on the noise variance σ^2_ϵ.

The optimization is done w.r.t. to the following objective function

$$\arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{y}|X, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}), \tag{3}$$

where $p(\boldsymbol{\theta})$ represents the hyperpriors. We use the above prior distributions for kernel hyperparameters when analyzing real microarray or RNA-seq data and optimize the above objective function. For simulated data we ignore the hyperpriors and optimize the standard marginal likelihood, i.e., $\log p(\boldsymbol{y}|X, \boldsymbol{\theta})$. We use the gradient-based method L-BFGS-B [20] for the optimization. The optimizer is run for a maximum of 1000 iterations with tolerance for deciding convergence equals to $1e^{-5}$.

3 Results

3.1 Simulated data

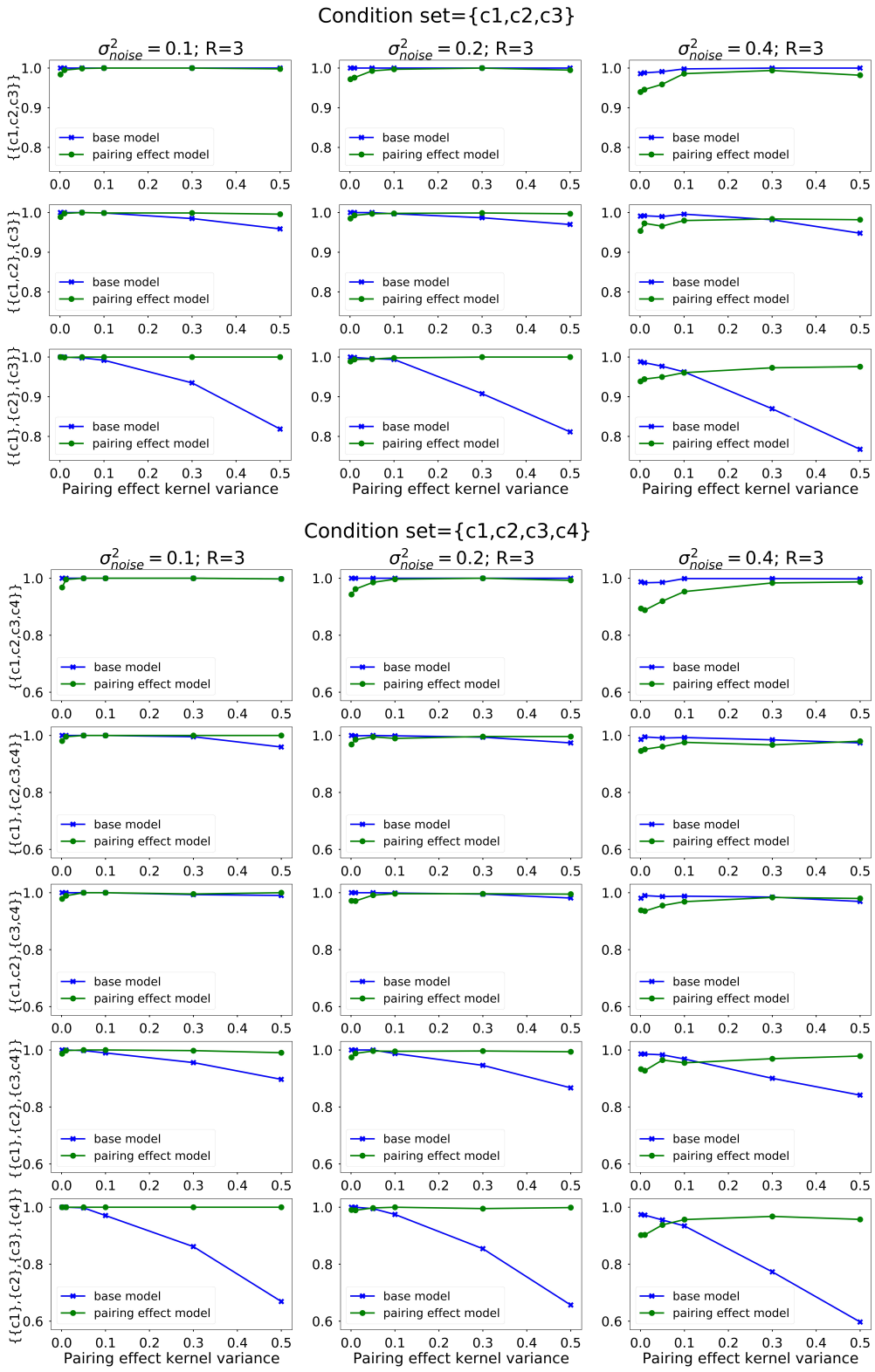

Figure 2: Accuracy of inferring the correct partitioning of conditions as a function of the pairing effect variance and the noise variance, obtained through simulated data using (top) 3 conditions (bottom) 4 conditions.

3.2 Microarray data

Partition	Base	Pairing
{{Th0, Th2}}	75.1	69.6
{{Th0}, {Th2}}	24.9	30.4

Table 1: Proportion of partitions obtained by fitting the base model and the pairing effect model on the genes from the human T-helper cell gene expression data set. Thus, when taking into account the paired design of the experiment, 30.4% of the genes were found to be differentially expressed between Th2 and Th0 cells, whereas 24.9% of genes were differentially expressed when only the response effect was modelled.

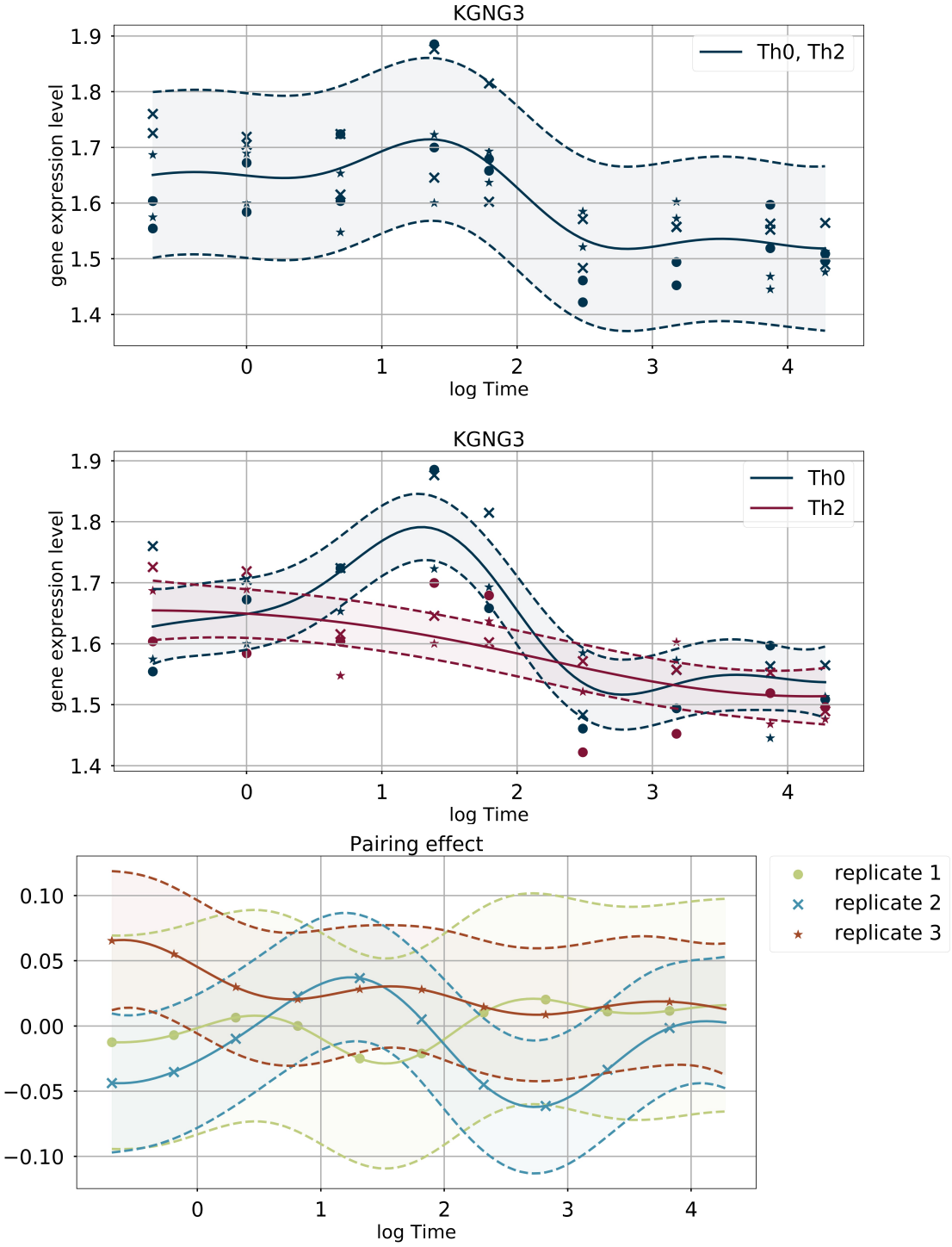

Figure 3: (top) The result of the base model on the gene *KCNG3* (probe set *1552897_a_at*). (middle) The result of the pairing effect model on the gene *KCNG3* and (bottom) the relative pairing effect learned from the data.

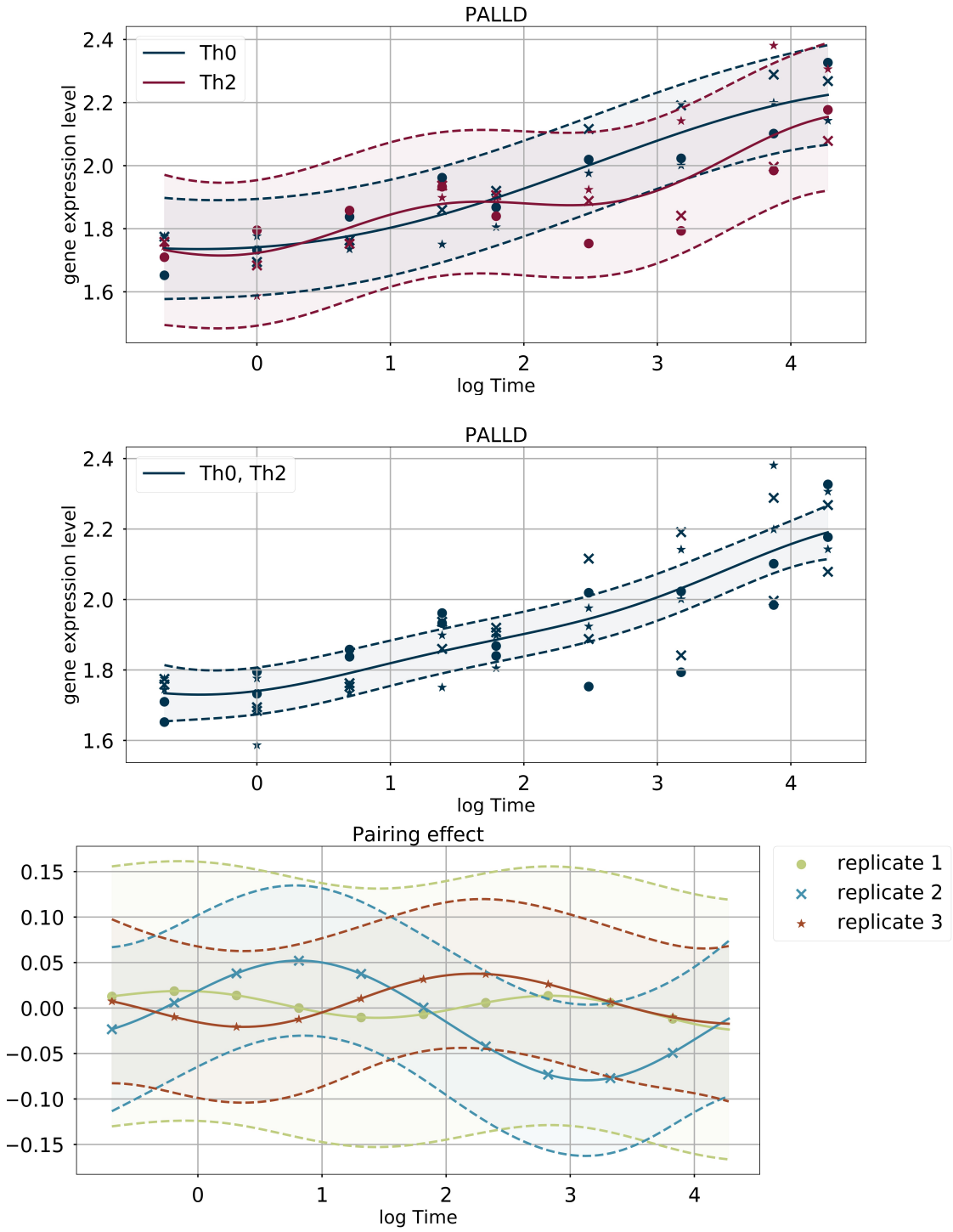

Figure 4: (top) The result of the base model on the gene *PALLD* (probe set *200897_s_at*). (middle) The result of the pairing effect model on the gene *PALLD* and (bottom) the relative pairing effect learned from the data.

3.3 RNA-seq data

Partition	Base	Pairing
{{Th0}, {Th17, Th17+IL1b, Th17+IL21, Th17+IL1b+IL21}}	47.9	22.8
{{Th0, Th17, Th17+IL1b, Th17+IL21, Th17+IL1b+IL21}}	9.6	19.4
{{Th17}, {Th0, Th17+IL1b, Th17+IL21, Th17+IL1b+IL21}}	0.5	6.6
{{Th0}, {Th17}, {Th17+IL1b, Th17+IL21, Th17+IL1b+IL21}}	19.7	5.6
{{Th0, Th17}, {Th17+IL1b, Th17+IL21, Th17+IL1b+IL21}}	17.4	3.9
{{Th0}, {Th17, Th17+IL21}, {Th17+IL1b, Th17+IL1b+IL21}}	2.3	3.8
others	2.6	37.9

Table 2: Most frequent partitions obtained by fitting the base model and the pairing effect model on all the genes from the T-helper cell RNA-seq data set. The percentage of the total amount of genes is reported.

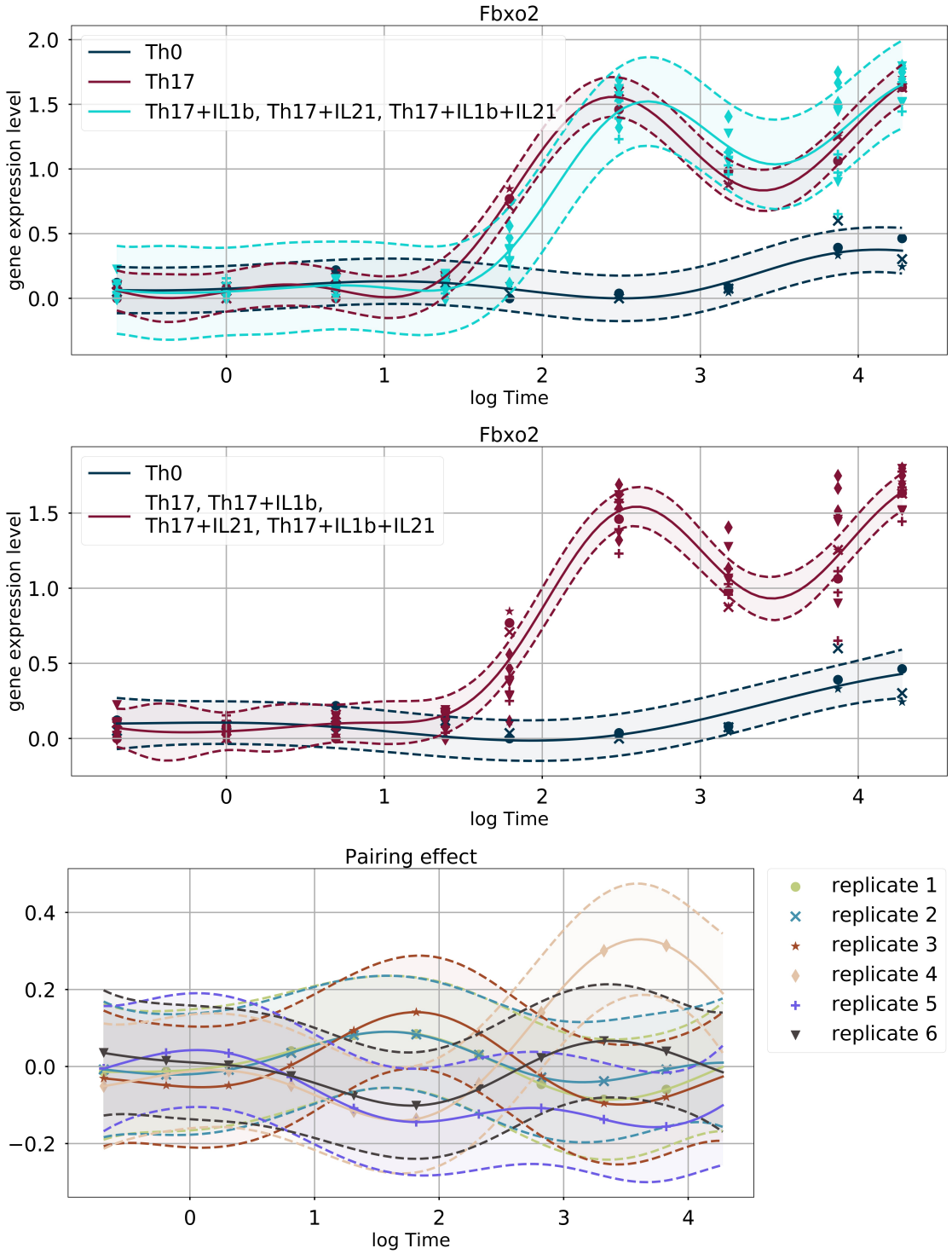

Figure 5: (top) The result of the base model on the gene *Fbxo2*. (middle) The result of the pairing effect model on the gene *Fbxo2* and (bottom) the relative pairing effect learnt from the data.

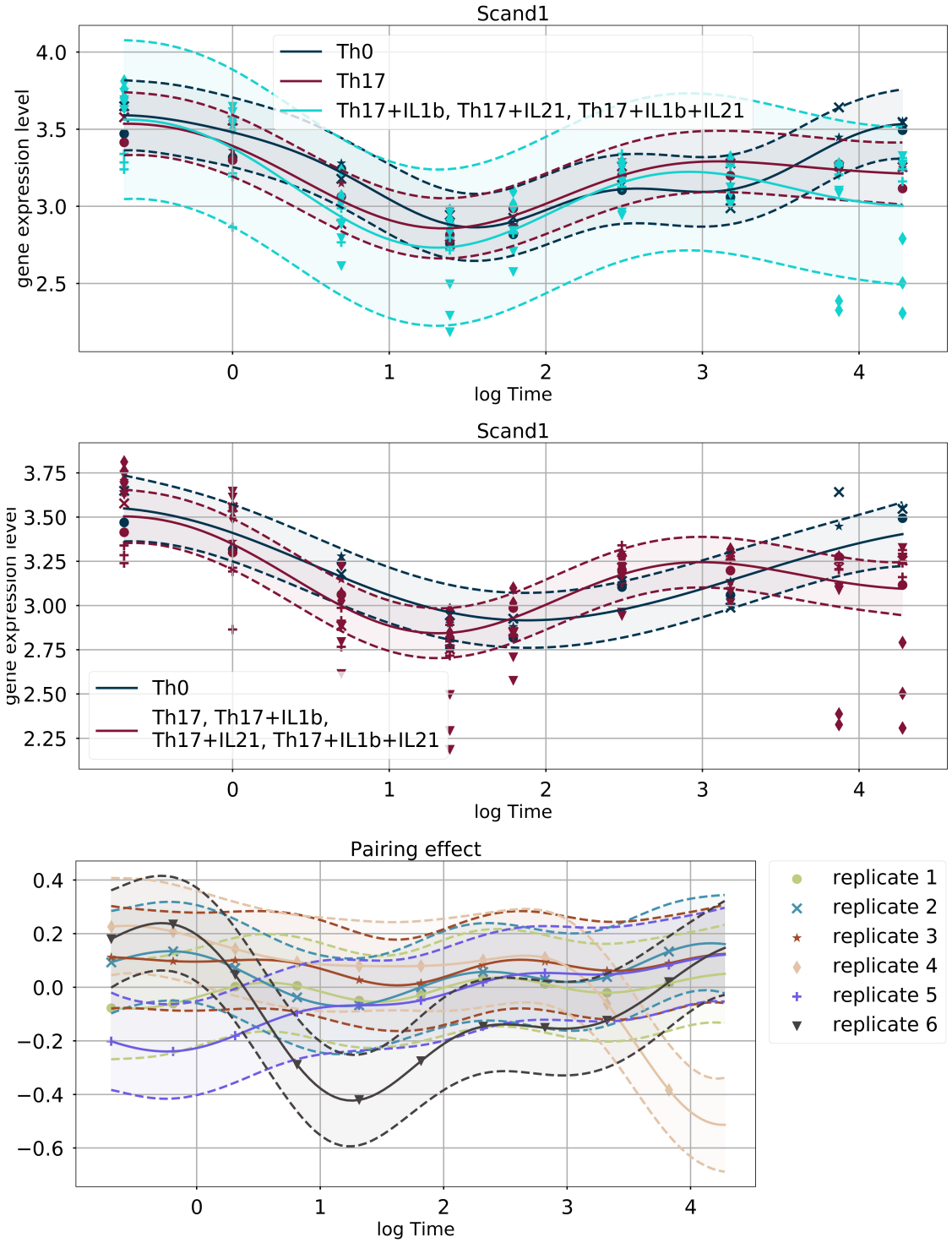

Figure 6: (top) The result of the base model on the gene *Scand1*. (middle) The result of the pairing effect model on the gene *Scand1* and (bottom) the relative pairing effect learnt from the data.

References

[1] Laura L Elo, Henna Järvenpää, Soile Tuomela, Sunil Raghav, Helena Ahlfors, Kirsti Laurila, Bhawna Gupta, Riikka J Lund, Johanna Tahvanainen, R David Hawkins, et al. Genome-wide profiling of interleukin-4 and stat6 transcription factor regulation of human th2 cell programming. *Immunity*, 32(6):852–862, 2010.

[2] Soile Tuomela, Sini Rautio, Helena Ahlfors, Viveka Öling, Verna Salo, Ubaid Ullah, Zhi Chen, Saara Hämälistö, Subhash K Tripathi, Tarmo Äijö, et al. Comparative analysis of human and mouse transcriptomes of th17 cell priming. *Oncotarget*, 7(12):13416, 2016.

[3] Matthew E Ritchie, Belinda Phipson, DI Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, 43(7):e47–e47, 2015.

[4] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Nature Precedings*, pages 1–1, 2010.

[5] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[6] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[7] Claudia Angelini, Daniela De Canditiis, Margherita Mutarelli, and Marianna Pensky. A bayesian approach to estimation and testing in time-course microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 2007.

[8] Claudia Angelini, Luisa Cutillo, Daniela De Canditiis, Margherita Mutarelli, and Marianna Pensky. Bats: a bayesian user-friendly software for analyzing time series microarray experiments. *BMC bioinformatics*, 9(1):415, 2008.

[9] J Straube, KA Lê Cao, and E Huang. lmms: Linear mixed effect model splines for modelling and analysis of time course data. *R package version*, 1(3), 2016.

[10] David S Fischer, Fabian J Theis, and Nir Yosef. Impulse model-based differential expression analysis of time course sequencing data. *Nucleic acids research*, 46(20):e119–e119, 2018.

[11] Ming Yuan. Flexible temporal expression profile modelling using the gaussian process. *Computational statistics & data analysis*, 51(3):1754–1764, 2006.

[12] Oliver Stegle, Katherine J Denby, Emma J Cooke, David L Wild, Zoubin Ghahramani, and Karsten M Borgwardt. A robust bayesian two-sample test for detecting intervals of differential gene expression in microarray time series. *Journal of Computational Biology*, 17(3):355–367, 2010.

[13] Tarmo Äijö, Sanna M Edelman, Tapio Lönnberg, Antti Larjo, Henna Kallionpää, Soile Tuomela, Emilia Engström, Riitta Lahesmaa, and Harri Lähdesmäki. An integrative computational systems biology approach identifies differentially regulated dynamic transcriptome signatures which drive the initiation of human t helper cell differentiation. *BMC genomics*, 13(1):572, 2012.

[14] Markus Heinonen, Olivier Guipaud, Fabien Milliat, Valérie Buard, Béatrice Micheau, Georges Tarlet, Marc Benderitter, Farida Zehraoui, and Florence d'Alché Buc. Detecting time periods of differential gene expression using gaussian processes: an application to endothelial cells exposed to radiotherapy dose fraction. *Bioinformatics*, 31(5):728–735, 2014.

[15] Tarmo Äijö, Vincent Butty, Zhi Chen, Verna Salo, Subhash Tripathi, Christopher B Burge, Riitta Lahesmaa, and Harri Lähdesmäki. Methods for time series analysis of rna-seq data with application to human th17 cell differentiation. *Bioinformatics*, 30(12):i113–i120, 2014.

[16] James Hensman, Neil D Lawrence, and Magnus Rattray. Hierarchical bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC bioinformatics*, 14(1):252, 2013.

[17] Daniel Spies, Peter F Renz, Tobias A Beyer, and Constance Ciaudo. Comparative analysis of differential gene expression tools for rna sequencing time course data. *Briefings in bioinformatics*, 20(1):288–298, 2017.

[18] Lu Cheng, Siddharth Ramchandran, Tommi Vatanen, Niina Lietzen, Riitta Lahesmaa, Aki Vehtari, and Harri Lähdesmäki. An additive gaussian process regression model for interpretable non-parametric analysis of longitudinal data. *Nature Communications*, 10(1798), 2019.

[19] Juho Timonen, Henrik Mannerström, Aki Vehtari, and Harri Lähdesmäki. An interpretable probabilistic machine learning method for heterogeneous longitudinal studies, 2019.

[20] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.