Modeling dynamic gene expression from RNA-seq time series data using hierarchical Gaussian process-linked negative binomial regression

Spencer Woody*
spencer.woody@utexas.edu

May 16, 2017

Abstract

RNA-seq has emerged as the predominant technology for gene expression analysis. Modeling approaches must take into account the fact that there data are in the form of overdispersed counts. Existing models use the negative binomial distribution. However, there are few existing analysis pipelines which can infer a continuous time course of gene expression. In this paper we present a negative binomial regression model with a logit link on an underlying latent nonparametric time course function taken from a hierarchical Gaussian process to model dynamic gene expression. The advantages of this approach are that it can pool information across replicates and handle the case of samples collected at irregular time intervals. We also present a Gibbs sampler for implementing the model. The model is applied to an RNA-seq time series dataset from an experiment tracking the cellular response of *E. coli* to starvation conditions. We conclude by listing potential extensions of the model and other future work to be done. This paper is written as a final project for the spring 2017 semester course SDS 383D course taught by Professor James Scott at UT-Austin.

1 Introduction

Recently there has been a proliferation of next-generation sequencing technologies used in bioinformatics. RNA-seq has become the new dominant tool for gene expression analysis, which may be used to infer gene regulatory networks and measure cellular response to external stimuli, to name just a few examples.

As opposed to microarray data, where gene expression is measured as a continuous variable, RNA-seq data are in the form of counts and are thus discrete. Furthermore, RNA-seq data are often overdispersed (i.e., the variance is larger than the mean), so the Poisson distribution, which has equal variance and mean, is inappropriate to use. Popular analysis pipelines, such as DESeq2 [1] and edgeR [2], use the negative binomial distribution. These techniques often fit a generalized linear model (GLM) with some sort of link function, such as the logarithmic link. This may be well suited for tasks such as differential expression between several conditions (e.g. experiment and control), but they are not adept at modeling the continuous time course of gene expression which is likely nonlinear.

*The University of Texas at Austin, Department of Statistics and Data Science. Data and R script used for this analysis available at `github.com/spencerwoody/SDS383D/tree/master/FinalProject`

In response I propose a negative binomial regression model with a logit link on a latent hierarchical Gaussian process. The motivation for this model comes from [3], where the authors use a hierarchical Gaussian process to model time series microarray data for samples with multiple replicates. Their approach accounts for hierarchy at the replicate level, but may also be extended to cases like data fusion where an experimenter would also like to pool information across related groups. In our case, we account for the discrete nature of RNA-seq data with the linked negative binomial regression. This allows us to use the same Gaussian process framework in a discrete data context, and the same extensions mentioned in [3] like clustering and data fusion should be applicable to this case too.

2 Model

Let y_{nrt} represent the read count at the time point $t > 0$ for replicate $r \in \{1, 2, \ldots, R_n\}$ of gene $n \in \{1, 2, \ldots N\}$. Then y_{nrt} follows a negative-binomial distribution,

$$(y_{nrt}|\alpha_n, \psi_{nr}(t)) \sim \text{NB}\left(\alpha_n, \frac{\exp\left[\psi_{nr}(t)\right]}{1 + \exp\left[\psi_{nr}(t)\right]}\right),$$

where α_n is the gene-specific dispersion parameter. For clarity, the parametrization of the negative binomial distribution used in this paper is given in Appendix A. For the sake of this project, I used a plug-in estimate of the dispersion parameter from [1], which uses an empirical Bayesian shrinkage estimate of dispersions by pooling across genes with similar mean normalized counts. As for the success probability parameter, there is a logit link on $\psi_{nr}(t)$, a hierarchical Gaussian process described by

$$\begin{aligned}(\psi_{nr}(t)|g_n(t)) &\sim \text{GP}(g_n(t), k_\psi(t, t')) \\ g_n(t) &\sim \text{GP}(0, k_g(t, t')),\end{aligned}$$

for some covariance functions $k_\psi(t, t')$ and $k_g(t, t')$, respectively parameterized by hyperparameters θ_ψ and θ_g (which might both be vectors). Notice that the expectation of y_{nrt} is

$$\mathbb{E}(y_{nrt}|\alpha_n, \psi_{nr}(t)) = \alpha_n \cdot \exp\left[\psi_{nr}(t)\right]. \tag{1}$$

We can consider $g_n(t)$ to be the underlying gene-level time course function, and then each $\psi_{nr}(t)$ represents a replicate-level deviation away from this gene-level function.

3 Inference

3.1 Gibbs sampler

In order to implement the model, we use a data augmentation strategy using the Pólya-Gamma distribution, originally introduced in [4] and specifically applied to GLMs for NB regression in [5], so that we have conditionally conjugate distributions to sample from for the Gibbs sampler. Specifically we introduce a latent ω_{nrt} corresponding to each y_{nrt} which has the conditional distribution

$$(\omega_{nrt}|y_{nrt}) \sim \text{PG}(y_{nrt} + \alpha_n, 0).$$

The key property of the Pólya-Gamma distribution which lends itself well for the Gibbs sampler is the expectation

$$\mathbb{E}_{\omega_{nrt}}\left[\exp\left(-\omega_{nrt}\left[\psi_{nr}(t)\right]^2/2\right)\right] = \cosh^{-(y_{nrt}+\alpha_n)}(\psi_{nr}(t)/2).$$

As demonstrated in both [4] and [5], the joint likelihood of y_{nrt} given ω_{nrt} and $\psi_{nr}(t)$ may then be written as

$$\begin{aligned} p(y_{nrt}|\psi_{nr}(t),\omega_{nrt}) &\propto \frac{\left(\exp\left[\psi_{nr}(t)\right]\right)^{y_{nrt}}}{\left(1+\exp\left[\psi_{nr}(t)\right]\right)^{\alpha_n+y_{nrt}}} \\ &= \frac{2^{-(y_{nrt}+\alpha_n)}\cdot\exp\left(\frac{y_{nrt}-\alpha_n}{2}\psi_{nr}(t)\right)}{\cosh^{y_{nrt}+\alpha_n}(\psi_{nr}(t)/2)} \\ &\propto \exp\left(\frac{y_{nrt}-\alpha_n}{2}\psi_{nr}(t)\right)\mathbb{E}_{\omega_{nrt}}\left[\exp\left(-\omega_{nrt}\left[\psi_{nr}(t)\right]^2/2\right)\right]. \end{aligned}$$

Now suppose we have observations at a vector of times $\mathbf{t}_{nr}$ which has length T_{nr}. The data vector of read counts is $\mathbf{y}_{nr} = \{y_{nrt}\}_{t\in\mathbf{t}_{nr}}$ which is associated with draws from the GP, $\boldsymbol{\psi}_{nr} = \{\psi_{nr}(t)\}_{t\in\mathbf{t}_{nr}}$. Then there is the latent variable vector $\boldsymbol{\omega}_{nr} = \{\omega_{nrt}\}_{t\in\mathbf{t}_{nr}}$. Define the diagonal matrix $\boldsymbol{\Omega}_{nr} = \text{diag}\left(\boldsymbol{\omega}_{nr}\right)$. Finally define the vector $\mathbf{g}_n$ be a vector of draws from the GP $g_n(t,t')$ at times $\mathbf{t}_{nr}$ and the matrix $\mathbf{K}_\psi(\mathbf{t}_{nr},\mathbf{t}_{nr'})$ such that it's (i,j) element is $k_\psi(\mathbf{t}_{nr}[i],\mathbf{t}_{nr'}[j])$ and $\mathbf{K}_g(\mathbf{t}_{nr},\mathbf{t}_{nr'})$ is defined similarly. Using the definition of the Gaussian process and the properties of the multivariate normal distribution, we can find the marginal prior of distribution of $\boldsymbol{\psi}_{nr}$,

$$\begin{aligned} p(\boldsymbol{\psi}_{nr}|\mathbf{g}_n,\theta_\psi) &\sim \mathcal{N}\left(\mathbf{f}_n,\mathbf{K}_\psi(\mathbf{t}_{nr},\mathbf{t}_{nr})\right) \\ p(\mathbf{f}_n|\theta_g) &\sim \mathcal{N}\left(\mathbf{0},\mathbf{K}_g(\mathbf{t}_{nr},\mathbf{t}_{nr})\right) \\ \Rightarrow p(\boldsymbol{\psi}_{nr}|\theta_\psi,\theta_g) &\sim \mathcal{N}\left(\mathbf{0},\mathbf{K}_\psi(\mathbf{t}_{nr},\mathbf{t}_{nr})+\mathbf{K}_g(\mathbf{t}_{nr},\mathbf{t}_{nr})\right). \end{aligned}$$

Let the vector $\boldsymbol{\theta} = (\theta_\psi,\theta_g)^T$ contain the hyperparameters of both covariance functions, $k_g(t,t')$ and $k_\psi(t,t')$. Now consider the collapsed vector across all replicates. We can write the marginal prior of the concatenated vector $\boldsymbol{\psi}_n = \{\boldsymbol{\psi}_{nr}\}_{r=1}^{R_n}$ as

$$p(\boldsymbol{\psi}_n|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0},\mathbf{K}_n)$$

where the matrix $\mathbf{K}_n$ is a $R_n \times R_n$ arrangement of matrices, each of which has dimension $T_{nr}\times T_{nr'}$ and is taken as

$$\mathbf{K}_n[r,r'] = \text{cov}(\boldsymbol{\psi}_{nr},\boldsymbol{\psi}_{nr'}) = \begin{cases}\mathbf{K}_g(\mathbf{t}_{nr},\mathbf{t}_{nr})+\mathbf{K}_\psi(\mathbf{t}_{nr},\mathbf{t}_{nr}) & \text{if } r=r' \\ \mathbf{K}_g(\mathbf{t}_{nr},\mathbf{t}_{nr'}) & \text{otherwise}\end{cases}. \tag{2}$$

In plain English, (2) captures the hierarchical nature of the data; samples from the same gene and same replicate will have covariance coming from both the gene-level and replicate-level functions, $g_n(t)$ and $\psi_{nr}(t)$, while samples from the same gene but different replicates will have covariance only coming from the gene-level function. The conditional posterior of $\boldsymbol{\psi}_n$, given the values of $\boldsymbol{\omega}_n$ and the data vector $\mathbf{y}_n$ is

$$\begin{aligned} p(\boldsymbol{\psi}_n|\mathbf{y}_n,\boldsymbol{\omega}_n,\boldsymbol{\theta}), &\propto p(\boldsymbol{\psi}_n|\boldsymbol{\theta})\prod_{r=1}^{R_n}\prod_{t\in\mathbf{t}_{nr}}p(y_{nrt}|\psi_{nr}(t),\omega_{nrt}) \\ &\propto p(\boldsymbol{\psi}_n|\boldsymbol{\theta})\prod_{r=1}^{R_n}\prod_{t\in\mathbf{t}_{nr}}\exp\left[-\frac{\omega_{nrt}}{2}\left(\psi_{nr}(t)-\frac{y_{nrt}-\alpha_n}{2\omega_{nrt}}\right)^2\right], \text{define } z_{nrt}=\frac{y_{nrt}-\alpha_n}{2}, \\ &\propto p(\boldsymbol{\psi}_n|\boldsymbol{\theta})\cdot\exp\left[-\frac{1}{2}\left(\boldsymbol{\psi}_n-\boldsymbol{\Omega}_n^{-1}\mathbf{z}_n\right)^T\boldsymbol{\Omega}_n\left(\boldsymbol{\psi}_n-\boldsymbol{\Omega}_n^{-1}\mathbf{z}_n\right)\right] \\ &\propto \mathcal{N}\left(\boldsymbol{\psi}_n|\boldsymbol{\Sigma}_n\mathbf{z}_n,\boldsymbol{\Sigma}_n\right), \text{ with } \boldsymbol{\Sigma}_n=\left(\mathbf{K}_n^{-1}+\boldsymbol{\Omega}_n\right)^{-1}. \end{aligned}$$

The conditional posterior of each ω_{nrt} is

$$\begin{aligned} p(\omega_{nrt}|y_{nrt}, \psi_{nr}(t)) &\propto \left[\exp\left(-\omega_{nrt}\left[\psi_{nr}(t)\right]^2/2\right)\right] \cdot \mathrm{PG}(\omega_{nrt}|y_{nrt}+\alpha_n, 0) \\ &\propto \mathrm{PG}(\omega_{nrt}|y_{nrt}+\alpha_n, \psi_{nr}(t)). \end{aligned}$$

Therefore the Gibbs sampler involves iteratively sampling from

$$(\boldsymbol{\psi}_n|\mathbf{y}_n, \boldsymbol{\omega}_n, \theta) \sim \mathcal{N}\left(\boldsymbol{\psi}_n|\boldsymbol{\Sigma}_n\mathbf{z}_n, \boldsymbol{\Sigma}_n\right), \text{ with } \boldsymbol{\Sigma}_n = \left(\mathbf{K}_n^{-1} + \boldsymbol{\Omega}_n\right)^{-1} \tag{3}$$

$$(\omega_{nrt}|y_{nrt}, \psi_{nr}(t)) \sim \mathrm{PG}(\omega_{nrt}|y_{nrt}+\alpha_n, \psi_{nr}(t)). \tag{4}$$

3.2 Prediction of gene- and replicate-level time course

At each iteration of the Gibbs sampler, we can also sample from the marginal posterior distribution of the underlying gene- and replicate-level time series functions. That is to say, we sample $\mathbf{g}_n^\star$ which is a series of draws from $g_n(t)$ at "new" times $\mathbf{t}_n^\star$, and we also sample $\boldsymbol{\psi}_{nr}^\star$ which is $\psi_{nr}(t)$ at "new" times $\mathbf{t}_{nr}^\star$ for each r. The respective joint distributions between these vectors and $\boldsymbol{\psi}_n$ are

$$\begin{bmatrix}\boldsymbol{\psi}_n \\ \mathbf{g}_n^\star\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}_n & \mathbf{K}_{n\star}^T \\ \mathbf{K}_{n\star} & \mathbf{K}_{n\star\star}\end{bmatrix}\right)$$

$$\begin{bmatrix}\boldsymbol{\psi}_n \\ \boldsymbol{\psi}_{nr}^\star\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}_n & \mathbf{K}_{nr\star}^T \\ \mathbf{K}_{nr\star} & \mathbf{K}_{nr\star\star}\end{bmatrix}\right)$$

with $\mathbf{K}_{n\star}$ and $\mathbf{K}_{nr\star}$ are defined element-wise such that

$$\begin{aligned} \mathbf{K}_{n\star}[i,j] &= \mathrm{cov}\left(\mathbf{g}_n^\star[i], \boldsymbol{\psi}_n[j]\right) = k_g(\mathbf{t}_n^\star[i], \mathbf{t}_n[j]) \\ \mathbf{K}_{nr\star}[i,j] &= \mathrm{cov}\left(\boldsymbol{\psi}_{nr}^\star[i], \boldsymbol{\psi}_n[j] \in \boldsymbol{\psi}_{nr'}\right) = \begin{cases} k_g(\mathbf{t}_{nr}^\star[i], \mathbf{t}_n[j]) + k_\psi(\mathbf{t}_{nr}^\star[i], \mathbf{t}_n[j]) & \text{if } r = r' \\ k_g(\mathbf{t}_{nr}^\star[i], \mathbf{t}_n[j]) & \text{otherwise} \end{cases}, \end{aligned}$$

and the matrices $\mathbf{K}_{n\star\star}$ and $\mathbf{K}_{nr\star\star}$ are

$$\begin{aligned} \mathbf{K}_{n\star\star} &= \mathbf{K}_g(\mathbf{t}_n^\star, \mathbf{t}_n^\star) \\ \mathbf{K}_{nr\star\star} &= \mathbf{K}_g(\mathbf{t}_{nr}^\star, \mathbf{t}_{nr}^\star) + \mathbf{K}_\psi(\mathbf{t}_{nr}^\star, \mathbf{t}_{nr}^\star). \end{aligned}$$

The conditional distribution of $\mathbf{g}_n^\star$ given $\boldsymbol{\psi}_n$ is

$$(\mathbf{g}_n^\star|\boldsymbol{\psi}_n, \theta) \sim \mathcal{N}\left(\mathbf{K}_{n\star}\mathbf{K}_n^{-1}\boldsymbol{\psi}_n,\ \mathbf{K}_{n\star\star} - \mathbf{K}_{n\star}\mathbf{K}_n^{-1}\mathbf{K}_{n\star}^T\right).$$

Given the fact that the marginal posterior of $\boldsymbol{\psi}_n$ is

$$(\boldsymbol{\psi}_n|\mathbf{y}_n, \boldsymbol{\omega}_n, \theta) \sim \mathcal{N}\left(\boldsymbol{\Sigma}_n\mathbf{z}_n, \boldsymbol{\Sigma}_n\right),$$

and using Lemma B.1 we can write the marginal posterior of $\mathbf{g}_n^\star$ as

$$(\mathbf{g}_n^\star|\mathbf{y}_n, \boldsymbol{\omega}_n, \theta) \sim \mathcal{N}\left(\mathbf{K}_{n\star}\mathbf{K}_n^{-1}\boldsymbol{\Sigma}_n\mathbf{z}_n,\ \mathbf{K}_{n\star}\mathbf{K}_n^{-1}\boldsymbol{\Sigma}_n\mathbf{K}_n^{-1}\mathbf{K}_{n\star}^T + \mathbf{K}_{n\star\star} - \mathbf{K}_{n\star}\mathbf{K}_n^{-1}\mathbf{K}_{n\star}^T\right). \tag{5}$$

Similarly, the marginal posterior of $\boldsymbol{\psi}_{nr}^\star$ is

$$(\boldsymbol{\psi}_{nr}^\star|\mathbf{y}_n, \boldsymbol{\omega}_n, \theta) \sim \mathcal{N}\left(\mathbf{K}_{nr\star}\mathbf{K}_n^{-1}\boldsymbol{\Sigma}_n\mathbf{z}_n,\ \mathbf{K}_{nr\star}\mathbf{K}_n^{-1}\boldsymbol{\Sigma}_n\mathbf{K}_n^{-1}\mathbf{K}_{nr\star}^T + \mathbf{K}_{nr\star\star} - \mathbf{K}_{nr\star}\mathbf{K}_n^{-1}\mathbf{K}_{nr\star}^T\right). \tag{6}$$

Using our Gibbs sampler framework, at each iteration we store draws from (5) and (6). Then we can make a 95% posterior credible band by taking the 2.5% and 97.5% quantiles for the time series at each time in $\mathbf{t}_n^\star$ and $\mathbf{t}_{nr}^\star$.

3.3 Choice of covariance function and hyperparameters

For this analysis I chose the Matérn(5/2) covariance function for both GPs,

$$k_g(t,t') = \tau_g^2 \exp\left\{1+\sqrt{5}\cdot\frac{d}{b_g}+\frac{5}{3}\cdot\frac{d^2}{b_g^2}\right\}\exp\left\{-\sqrt{5}\cdot\frac{d}{b_g}\right\},\ d=\|t-t'\|,$$

$$k_\psi(t,t') = \tau_\psi^2 \exp\left\{1+\sqrt{5}\cdot\frac{d}{b_\psi}+\frac{5}{3}\cdot\frac{d^2}{b_\psi^2}\right\}\exp\left\{-\sqrt{5}\cdot\frac{d}{b_\psi}\right\},\ d=\|t-t'\|,$$

so, for the gene-level covariance function the hyperparameters are $\theta_g = (b_g, \tau_g^2)^T$, and we refer to b_g as the *relative length* parameter and τ_g^2 is the *amplitude* parameter. For the replicate-level covariance function, the hyperparameters are $\theta_\psi = (b_\psi, \tau_\psi^2)^T$. Of course, inferences made from Gaussian processes are highly sensitive to the hyperparameters contained within the covariance function. It is common to use plug-in estimates from type-II marginal likelihood maximization, i.e.

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{y}_n|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \int p(\mathbf{y}_n|\boldsymbol{\psi}_n,\boldsymbol{\theta})p(\boldsymbol{\psi}_n|\boldsymbol{\theta})d\boldsymbol{\psi}_n \tag{7}$$

However, in this case there is no closed form solution to the marginal likelihood function in (7) so this is difficult to do in practice. It might also be preferable to integrate over uncertainty of the hyperparameters, in which case we set the hyperprior $p(\boldsymbol{\theta})$ and then at each iteration of the Gibbs, sample from the density

$$p(\boldsymbol{\theta}|\mathbf{y}_n,\boldsymbol{\omega}_n,\boldsymbol{\psi}_n) \propto p(\boldsymbol{\psi}_n|\mathbf{y}_n,\boldsymbol{\omega}_n,\boldsymbol{\theta})\cdot p(\boldsymbol{\theta}).$$

This presents another issue because we must choose an appropriate hyperprior, which is likely not conjugate to $p(\boldsymbol{\psi}_n|\mathbf{y}_n,\boldsymbol{\omega}_n,\boldsymbol{\theta})$, so then we need to perform a Metropolis-Hastings step at each iteration.

4 Experiment

We implemented our model on an RNA-seq dataset from an experiment in [6], where the authors collected samples from *Escherichia coli* to track cellular response to starvation conditions at nine irregularly spaced time-points: 3, 4, 5, 6, 8, 24, 48, 168, and 336 hours. For this analysis we estimated on a logarithmic time scale to put the times on a more dense space. We coded the Gibbs sampler in R as derived above and used the `BayesLogit` package [4] to sample from the Pólya-Gamma distribution.

As mentioned earlier, the choice of hyperparameters is critical because they are highly influential on the outcome of our results. Our first idea was to implement a Metropolis-Hastings step to sample the hyperparameters at each Gibbs iteration. Before doing this, we tried using an estimate of the hyperparamters found by maximizing the log-likelihood of $p(\boldsymbol{\psi}_n|\mathbf{y}_n,\boldsymbol{\omega}_n,\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ using the output of $\boldsymbol{\psi}_n$ from the Gibbs sampler (which is equivalent to using a MAP estimate with a flat hyperprior), using the `optim` command in R. However, with this approach we found that this optimization method gave wildly varying results, with each estimated hyperparameter ranging across multiple orders of magnitude. This may suggest that the log-likelihood function is very flat, so it is difficult to find a global maximum, which in turn highlights a need to have an appropriate hyperprior. Due to a shortage in available time, for the purposes of this project, the hyperparameters were somewhat arbitrarily set to $b_g = b_\psi = 2$ and $\tau_g = \tau_\psi = 1$

Figure 1 shows the results of the model implemented on the dataset. There were 4000 Gibbs iterations after a burn-in of 1000 iterations. Each row of the figure represents one gene. The ribbon on each plot is the 95% posterior credible band for the function, and the solid black line is the posterior median. The left panel of each row shows the infered gene-level time course function, and the right panel shows each infered replicate-level time course function. On the replicate panels, there is also overlaid data, with each datapoint normalized by the transormation $\log(y_{nrt}/\alpha_n)$, derived from 1, so that the data and the time course function are on the same scale. Some read counts are equal to zero and the logarithmic function is not defined at zero, so these counts are shown with semi-circles along the bottom of the plots. The results show that the model fits the data well, and we are able to estimate a sensible gene-level function in each case. We should not expect the fitted curves to pass through or nearly through all the data because the time course function is related to the *expected* read count for a given time.

5 Conclusion

In this report we were mostly successful in applying to RNA-seq data our negative binomial regression model with a logit-linked nonparametric time course function from a Gaussian process. The Gibbs sampler for making inference is presented, and then implemented in R. However, there remain problems with tuning the hyperparameters of the Gaussian processes, and this issue should be addressed because picking the right hyperparameters is essential to deriving meaningful results. Similar to what is explained in [3], there are several extensions of this model which are worth exploring. For instance, it is possible to generalize to different levels of hierarchy, such as data from different experiments. Clustering genes based on similar expression time dynamics can be used for inferring gene regulatory networks or gauging global cellular response qualitatively through gene ontology. Differential expression analysis, a vital method in transcriptomics, is explored in context of Gaussian processes applied to microarray data in [7], and this should also be extended to the model presented here.

References

[1] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15:550, 2014.

[2] 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, January 2010.

[3] 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.

[4] N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using Polya-Gamma latent variables. *ArXiv e-prints*, May 2012.

[5] M. Zhou, L. Li, D. Dunson, and L. Carin. Lognormal and Gamma Mixed Negative Binomial Regression. *ArXiv e-prints*, June 2012.

[6] John R. Houser, Craig Barnhart, Daniel R. Boutz, Sean M. Carroll, Aurko Dasgupta, Joshua K. Michener, Brittany D. Needham, Ophelia Papoulas, Viswanadham Sridhara, Dariya K.

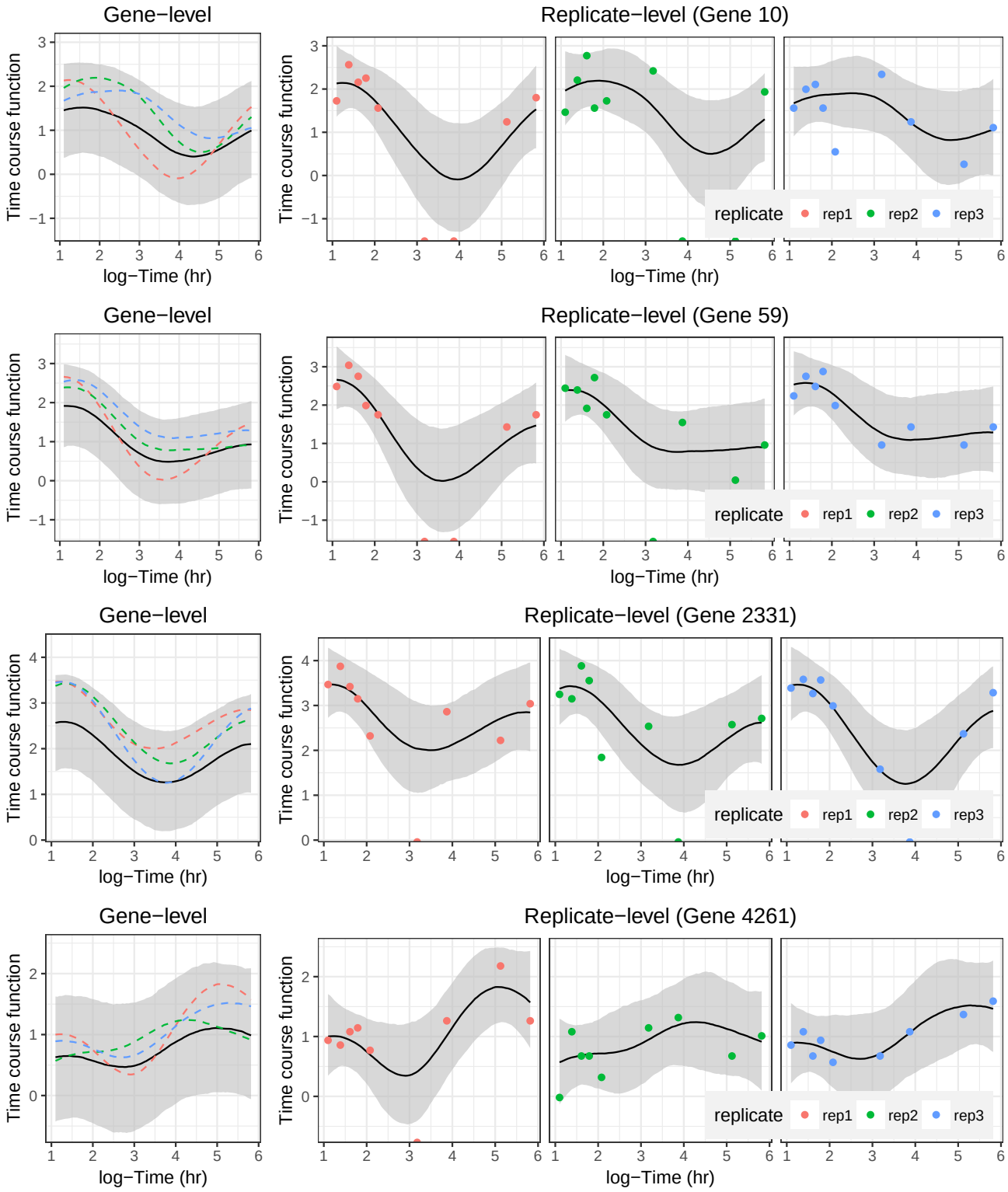

Figure 1: Gene- and replicate-level time series functions for selected genes with overlaid normalized data

Sydykova, Christopher J. Marx, M. Stephen Trent, Jeffrey E. Barrick, Edward M. Marcotte, and Claus O. Wilke. Controlled measurement and comparative analysis of cellular components in e. coli reveals broad regulatory changes in response to glucose starvation. *PLOS Computational Biology*, 11(8):1–27, 08 2015.

[7] Alfredo A. Kalaitzis and Neil D. Lawrence. A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. *BMC Bioinformatics*, 12(1):180, 2011.

A Parameterization of Negative Binomial distribution

Let Y be a negative binomial-distributed random variable which has support over the nonnegative integers with *dispersion* parameter $r \in (0, \infty)$, and *success probability* parameter $p \in (0, 1)$. This may be denoted as

$$Y \sim \mathrm{NB}(r, p),$$

and Y has the probability mass function

$$\begin{aligned}\Pr(Y = k) &= \frac{\Gamma(r+k)}{k!\Gamma(r)} \cdot (1-p)^r p^k \\ &= \binom{k+r-1}{k} \cdot (1-p)^r p^k \ \text{ if } \ r \in \{1, 2, \ldots\}.\end{aligned}$$

The mean and variance of Y are, respectively,

$$\begin{aligned}\mathbb{E}(Y) \equiv \mu &= \frac{p}{1-p} \cdot r \\ \mathrm{Var}(Y) &= \frac{p}{(1-p)^2} \cdot r = \mu + \mu^2/r.\end{aligned}$$

Note that $\mathrm{Var}(Y) > \mathbb{E}(Y)$, which encapsulates the *overdispersion* property of the negative binomial distribution, as opposed to the Poisson distribution which has equal mean and variance. When r is a positive integer, we may interpret Y as the number of successes before the rth failure in a series of independent Bernoulli trials with succeess probability p.

B Lemma

Lemma B.1. *Define the random vectors x and γ such that the conditional distribution of x given γ and the marginal distribution of γ are, respectively,*

$$\begin{aligned}(x|\gamma) &\sim \mathcal{N}_n(A\gamma, \Sigma) \\ \gamma &\sim \mathcal{N}_p(m, V),\end{aligned}$$

where A is a $n \times p$ matrix. Then the joint distribution of $(x, \gamma)^T$ is

$$\begin{bmatrix} x \\ \gamma \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} Am \\ m \end{bmatrix}, \begin{bmatrix} AVA^T + \Sigma & AV \\ VA^T & \Sigma \end{bmatrix} \right). \tag{8}$$

Proof. Equivalently, x may be written as

$$x = A\gamma + \eta,\ \eta \sim \mathcal{N}_n(0, \Sigma)$$

and then $(x, \gamma)^T$ is multivariate normal because it can be written as an affine transformation of independent multivariate normal variables,

$$\begin{bmatrix} x \\ \gamma \end{bmatrix} = \begin{bmatrix} A \\ \mathcal{I}_p \end{bmatrix} \gamma + \begin{bmatrix} \mathcal{I}_n \\ \mathcal{O}_{p \times n} \end{bmatrix} \eta.$$

From this, the mean and covariance matrix in (8) may be derived from properties of the multivatiate normal distribution. □