Bayesian model-based method for clustering gene expression time series with multiple replicates

Elio Nushi[1], François P. Douillard[2], Katja Selby[2], Miia Lindström[2], Antti Honkela[1*]

1 Department of Computer Science, University of Helsinki, Helsinki, Finland
2 Department of Food Hygiene and Environmental Health, Faculty of Veterinary Medicine, University of Helsinki, Helsinki, Finland

* antti.honkela@helsinki.fi

Abstract

In this study, we introduce a Bayesian model-based method for clustering transcriptomics time series data with multiple replicates. This technique is based on sampling Gaussian processes (GPs) within an infinite mixture model from a Dirichlet process (DP). Our method uses multiple GP models to accommodate for multiple differently behaving experimental replicates within each cluster. We call it multiple models Dirichlet process Gaussian process (MMDPGP). We compare our method with state-of-the-art model-based clustering approaches for handling gene expression time series with multiple replicates. We present a case study where all methods are applied for clustering RNA-Seq time series of *Clostridium botulinum* with three different experimental replicates. The results obtained from the gene enrichment analysis showed that the number of significantly enriched sets of genes is larger in the clusters produced by MMDPGP. To demonstrate the accuracy of our method we use it to cluster synthetically generated data sets. The clusters produced by our method on the synthetic data had a significantly higher purity score compared to the state-of-the-art approaches. By modelling each replicate with a separate GP, our method can use the natural variability between experimental replicates to learn more about the underlying biology.

Author summary

In our manuscript we introduce a method called multiple models Dirichlet process Gaussian process (MMDPGP), a novel Bayesian approach for clustering gene expression time series data. Our method stands out by accounting for the variability among multiple experimental replicates within each cluster, a feature that is often overlooked in existing model-based clustering approaches. This allows us to capture the natural variability between replicates as opposed to the crude method of simply averaging the replicates which discards interesting information in the data. By integrating multiple Gaussian process models within an infinite mixture model derived from a Dirichlet process, MMDPGP offers a more nuanced and accurate representation of the biological data. We benchmarked MMDPGP against state-of-the-art methods, by applying them for the purpose of clustering recently collected RNA-Seq time series of the bacterium Clostridium botulinum and performing a gene enrichment analysis on the generated clusters. Additionally, we test the accuracy of our method in comparison with other methods using synthetic data sets. The superior performance of our method in terms of finding significantly enriched gene sets and the clustering accuracy on synthetic data

underscore its robustness and potential for broad applicability in computational biology. Our study addresses a critical gap in the analysis of transcriptomics time series data by explicitly modeling the natural variability across experimental replicates. This advancement not only enhances the accuracy of clustering results but also provides deeper insights into the underlying biological processes. By leveraging Bayesian methods and Gaussian processes, our approach offers a powerful tool that can be adapted and extended for various types of omics data, inspiring further methodological developments in the field.

Competing interests

We declare no competing interests related to this work.

Code availability and implementation

The Python code for implementing our method is publicly available in Zenodo through the following DOI link: `https://doi.org/10.5281/zenodo.11202145`.

Data

The RNA-Seq data used to validate our method in the paper are deposited in GEO at the following link: `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE248529,secureaccesstoken:qfajmuisbvcxran`.

Introduction

Clustering genes based on their transcription dynamics can yield invaluable information about the roles of these genes in different biological processes and their complex interaction network. The transcription dynamics is captured by gene expression time series, which can be collected in multiple replicates to take in consideration the biological variation and measurement errors [1].

Traditional clustering methods such as k-means, hierarchical clustering or self-organizing maps rely on correlation or Euclidean distance, which may not account for dependencies in gene expression levels across time points [2]. Alternatively, model-based clustering methods assume that the data are generated from a mixture of underlying probability distributions, each representing a different cluster [3]. However, previously published model-based clustering methods for gene expression time series have put little to no attention in considering the variability of experimental replicates.

In this work, we present a Bayesian model-based clustering technique for multi-replicate transcriptomics time series, that uses a Dirichlet process (DP) [4] for generating clusters and multiple Gaussian process (GPs) [5] models to represent the different replicates inside each cluster. We call our method multiple models Dirichlet process Gaussian process (MMDPGP) mixture model. This approach incorporates a measure of uncertainty to the cluster assignments. It also provides the added benefit of automatically identifying an optimal number of clusters based on different optimality criteria, instead of arbitrarily setting it to a fixed value like in the case of k-means clustering for instance.

We compare MMDPGP with two similar model-based approaches proposed by Hensman et al. [6] and McDowell et al. [2]. The approach presented by Hensman et al. is called mixture of hierarchical Gaussian processes (MOHGP) and it models the replicates using a single GP model at each cluster and samples the clusters from a DP. The software implementation of MOHGP indicated in the original Hensman et al.

paper [6] was based on a variational inference technique. In order to provide a better comparison we implemented it using a Gibbs sampler written in Python similarly to [2]. On the other hand, McDowell et al. handle multiple experimental replicates by using the average of the replicates at each time point as the data in their GP model. They call their method Dirichlet process Gaussian process (DPGP). Unlike these alternative methods, MMDPGP provides a finer and more detailed modelling of each replicate which could help uncover clusters of genes that always behave similarly with each other, but together may behave differently across different replicates because of natural variability.

We use the methods for the purpose of clustering recently collected RNA-Seq time series. The RNA-Seq data were obtained by collecting RNA samples from growth experiments of three replicate cultures of *Clostridium botulinum* strain ATCC 19397 in different time points.

Related Methods

Clustering gene expression patterns according to their similarity has been the subject of many studies [2, 6–9]. Rasmussen et al. [7] proposed a Dirichlet process mixture (DPM) model as an alternative to the bootstrap approach for modeling uncertainty in gene expression clustering. This method calculates the likelihood of two genes being in the same cluster as a measure of similarity. It does this by considering the primary factors of model uncertainty, such as the number of clusters and their location. They used a Gibbs sampler to implement their model and apply it to cluster high-dimensional non-time series gene expression data.

Medvedovic et al. [8] developed different clustering procedures based on Bayesian mixture models. Their method design specifically targeted microarray gene expression data with experimental replicates. They model the distribution of their data as a Bayesian mixture model and consider both finite and infinite Bayesian mixture models in their modeling approach. However, they do not present any application of their method on time series data.

Cooke et al. [9] introduced an extension to the Bayesian hierarchical clustering (BHC) algorithm [10] for use with time series data, and tested it on microarray time series. BHC uses a one-pass, bottom up greedy approach which assigns each data points to a separate cluster and iteratively merges cluster pairs together similarly to traditional agglomerative clustering. Their algorithm utilizes GP regression to model the structure of the data and can accommodate multiple replicate measurements of genes to enable a better discrimination of gene expression profiles.

Hensman et al. [6] introduced a clustering method which they called mixture of hierarchical Gaussian processes (MOHGP), that is based on GPs and DP. Their method can cluster time series with multiple replicates, but they consider only one way of modeling replicates which consists on clusters that are parameterized by a single GP. They provide an efficient implementation for MOHGP based on a variational inference procedure.

McDowell et al. [2] presented a method for clustering gene expression time series by combining a DP with a GP, and they called it simply DPGP. Their method is specialised in clustering gene expression time series consisting of a single replicate. If multiple replicate measurements are provided, it simply calculates their average and uses that in the rest of the analysis, thus not considering the variability between replicates.

In this work we focus on non-parametric clustering methods for time series which can handle the information provided by different replicates. Here we propose MMDPGP, a method for clustering gene expression time series which leverages the information provided by the variability between different experimental replicates. We

compare our method with two previously published model-based clustering methods: MOHGP (Hensman et al.) and DPGP (McDowell et al.). Similarly to MMDPGP, MOHGP considers the variability between experimental replicates. However, it assumes that the replicate gene measurements within the same cluster are generated by the same cluster-specific GP model, while MMDPGP assumes that the different gene expression replicates within the same cluster are generated by replicate-specific GPs. On the other hand, DPGP (McDowell et al.) ignores the variability between the replicates and uses only their average to fit the GPs in the clustering process. We argue that MMDPGP is superior to the previous methods for clustering gene expression time series with multiple replicates. We demonstrate this through different analyses after applying the methods on both biological data consisting of RNA-Seq time series of *C. botulinum* cultures and synthetically generated data.

Methods

Gaussian Processes

Gaussian processes are non-parametric methods that are used for building probabilistic models of functional data [5]. In many previous studies, GPs have been widely recognized and adopted for modeling transcriptomics data [1, 2, 11–16]. A GP is completely characterized by a mean function $m(t)$ and a covariance (or kernel) function $k(t, t')$, which is often referred to as the kernel of the GP. It defines a probability distribution over functions $f(t)$ which we denote as:

$$f(t) \sim \mathcal{GP}(m(t), k(t, t')). \tag{1}$$

Choosing the kernel is an important step in the process of modeling with GPs as it determines our assumptions about the function that we want to model. The most widely used kernel function for modelling with GPs is the exponentiated quadratic function [5] which is defined as follows:

$$k_{\mathrm{EQ}}(t, t') = \sigma_f^2 \ \exp\left(-\frac{\|t - t'\|^2}{2l^2}\right). \tag{2}$$

The exponentiated quadratic kernel function is characterized by 2 parameters which are the length scale l and the variance σ_f^2. The length scale determines the oscillatory behavior of the function, specifically influencing the extent of fluctuations. Generally, the predictive capability of the model is limited to a range not exceeding l units from the observed data, indicating constraints on the model's extrapolation potential beyond this threshold. The variance determines the average distance of the function away from the mean. We will also consider a third parameter σ_n^2 which is the Gaussian noise and represents the observation noise of the data in the GP model. In the rest of the analysis when we write $k(t, t')$ it is implied the exponentiated quadratic kernel unless specified otherwise. Our method uses GPs as a parametrization for each cluster of genes (i.e., all genes within the same cluster share the same GPs' profiles).

Infinite mixtures of Gaussian processes

Bayesian nonparametric (BNP) techniques differ from traditional methods in addressing the complexity of models. In standard cluster analysis methods like k-means, it is required to evaluate a range of models, each with a different parameter value (e.g., $k = 1, \ldots, N$ clusters), then the optimal model is selected based on various metrics for model comparison. Traditional methods are characterized by a finite number of

parameters. In contrast, BNP methods are grounded in a statistical framework that can accommodate a potentially infinite number of parameters, this characteristic makes them flexible and removes the burden of choosing the number of parameters which could best model the system.

The Dirichlet process (DP) [4] is an example of a BNP technique which we use in our clustering method. DP is a nonparametric probability distribution over probability distributions, and it is used as a flexible prior for unsupervised learning tasks like clustering and density modeling [17]. We denote a probability distribution sampled from a DP as:

$$G \sim DP(\alpha, H), \tag{3}$$

where α is a positive real number called the concentration parameter, and H is a probability distribution called the base measure.

In the context of clustering, a DP is used to generate priors for a Dirichlet process mixture model (DPMM) which is a mixture model that accounts for a theoretically infinite number of mixture components [18]. Several authors have extended the concept of DPMM to include GP models and call them infinite mixture of Gaussian process experts, infinite mixture of global Gaussian processes or Dirichlet process Gaussian process models [2, 6, 19–22].

In this paper we use a Dirichlet process Gaussian process model to cluster time course transcriptomics data using multiple replicates. Below we define the data and the model used for the clustering.

Let $D = \{[(t_j, g_{ij})]_{j=1}^{T},\ i \in \{1, \ldots, N\}\}$, be a set of transcriptomics time series, where g_{ij} represents the gene expression value of gene i at time t_j. We denote by $f_i(t)$ the latent function that models the time series $[(t_j, g_{ij})]_{j=1}^{T}$ representing the gene expression values of gene i at every time point t_j. Then, we define the following generative model for our data:

$$G \sim DP(\alpha, H) \tag{4a}$$

$$\theta_\kappa \sim G \tag{4b}$$

$$f_i(t) \sim \mathcal{GP}(m_\kappa(t), k_\kappa(t, t')), \tag{4c}$$

where G is a sample from $DP(\alpha, H)$ as defined in Eq (3) (small values of α generate less clusters with bigger sizes, while large values generate more clusters with smaller sizes), and $\theta_\kappa = \{m_\kappa(t), l_\kappa, \sigma_{f,\kappa}, \sigma_{n,\kappa}\}$ is a set of GP latent parameters for cluster κ which is drawn from the distribution G. Finally, we set $f_i(t)$ to be a sample drawn from the GP with mean function $m_{\theta_\kappa}(t)$, exponentiated quadratic kernel parameterized by $l_\kappa, \sigma_{f,\kappa}$, and observation noise $\sigma_{n,\kappa}$. It is possible to express the conditional distribution of a randomly sampled cluster parameter θ_κ dependent on all the previously sampled parameters $\theta_{\setminus\kappa}$ by integrating over G and obtain:

$$P(\theta_\kappa | \theta_{\setminus\kappa}) \propto \alpha P(\theta_\kappa | H) + \sum_{\theta_i \in \theta_{\setminus\kappa}} \delta(\theta_i = \theta_\kappa), \tag{5}$$

where δ is the Kronecker delta function, and $\theta_{\setminus\kappa}$ is the set of all sampled parameters except for θ_κ. This suggests a practical solution to the problem of sampling from a DP because drawing a concrete distribution G is impossible in practice.

Model selection for fitting multiple replicates

Gene expression time series can be collected in multiple replicates to take in consideration the biological variation and measurement errors [1]. However, the

information provided by multiple replicates is often disregarded by clustering methods or they are used merely for estimating an average value which is later used in the downstream analysis as it is the case in [2].

In this paper we construct and analyze 3 different approaches which take in consideration all replicate measurements of genes in the process of clustering genes by using a DP and GPs. The first method, MOHGP, was proposed by Hensman et al. [6]. The second method, DPGP, was proposed by McDowell et al. [2]. The third is our proposed approach, that we call multiple models DPGP (MMDPGP). Conceptually they involve changing Eq (4c) in the generative model defined by Eq (4a)-(4c). Below we explain the details for each approach.

To account for R different replicates, we extend the notation used for defining our transcriptomics time series data D in the previous section by introducing a superscript r ($r \in \{1, \ldots, R\}$) which denotes the specific replicate time series (i.e., $D = \{[(t_j, g_{ij}^{(r)})]_{j=1}^{T},\ i \in \{1, \ldots, N\},\ \text{and}\ r \in \{1, \ldots, R\}\}$).

MOHGP (Hensman et al.)

In the case of MOHGP for each cluster κ, Hensman et al. [1] define a regression problem over functions $f_i : t \mapsto f_i(t)$, which are generated by the following model:

$$f_i(t) \sim \mathcal{GP}(m_\kappa(t), k_\kappa(t, t')). \tag{6}$$

We denote by $g_i^{(r)} \in D$ the vector $[g_{ij}^{(r)}]_{j=1}^{T}$ representing the gene expression values of gene i at every time point t_j of replicate r. The model assumes that they are noisy observations of $f_i(t)$. If we denote by $\mu_i = [f_i(t_j)]_{j=1}^{T}$ the vector representing the values of $f_i(t)$ at each time point t_j, then the conditional distribution for $g_i^{(r)}$ can be written as follows:

$$g_i^{(r)} | f_i(t) \sim \mathcal{N}(\mu_i, \sigma_n^2 \mathbb{I}), \tag{7}$$

where σ_n^2 represents the noise variance between all replicates across all genes inside the cluster.

To make inference easier, $f_i(t)$ is marginalized out and the model is expressed by conditioning on $m_\kappa(t)$. In the marginalised model, the covariance between two observations $g_{ij}^{(r)}$ and $g_{ij'}^{(r')}$ will then be

$$\mathrm{Cov}[g_{ij}^{(r)}, g_{ij'}^{(r')}] = k(t_j, t_{j'}) + \sigma_n^2 \delta(t_j = t_{j'}) \delta(r = r'), \tag{8}$$

where δ is the Kronecker delta.

Let K_{HE_κ} be the covariance matrix defined by the kernel function in Eq (8). If we denote by $\mu_\kappa = [m_\kappa(t_j)]_{j=1}^{T}$, the vectors of observed values $g_i^{(r)}$ can then be sampled from the following multivariate normal distribution:

$$\begin{bmatrix} g_i^{(1)} \\ \vdots \\ g_i^{(R)} \end{bmatrix} | m_\kappa(t) \sim \mathcal{N}\left(\begin{bmatrix} \mu_\kappa \\ \vdots \\ \mu_\kappa \end{bmatrix}, K_{\mathrm{HE}_\kappa} \right). \tag{9}$$

The value of μ_κ is determined as an approximation by setting it as $\mu_\kappa \approx \frac{1}{N_\kappa} \sum_{i=1}^{N_\kappa} \mu_i$, where N_κ is the number of genes in the cluster. The generative model MOHGP can be

written as:

$$
\begin{aligned}
G &\sim DP(\alpha, H) \\
\theta_\kappa &\sim G \\
\begin{bmatrix} g_i^{(1)} \\ \vdots \\ g_i^{(R)} \end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix} \mu_\kappa \\ \vdots \\ \mu_\kappa \end{bmatrix}, K_{\mathrm{HE}_\kappa} \right).
\end{aligned} \tag{10}
$$

The computational complexity for fitting the GP of a single cluster in this model is therefore $\mathcal{O}(N_\kappa(RT)^3)$ in time and $\mathcal{O}(N_\kappa(RT)^2)$ in space, where N_κ, T, R represent the number of genes in the cluster, the number of time points for which we have measurements and the number of replicates.

DPGP (McDowell et al.)

Similarly to the previous approach, DPGP assumes that the latent mean function of each observed gene expression time series is sampled from cluster specific GP like in Eq 6. However, they limit their model by considering gene expression time series, which consist only of a single replicate. When multiple replicates are provided, DPGP collapses them into a single time series by computing their mean at each observed time point. Following the derivations of MOHGP from the previous section, but in the case of a single replicate, the generative model for DPGP can be written as:

$$
\begin{aligned}
G &\sim DP(\alpha, H) \\
\theta_\kappa &\sim G \\
g_i &\sim \mathcal{N}(\mu_\kappa, K_{\mathrm{HE}_\kappa}).
\end{aligned} \tag{11}
$$

The computational complexity for fitting the GP of a single cluster for DPGP is $\mathcal{O}(N_\kappa T^3)$ in time and $\mathcal{O}(N_\kappa T^2)$ in space.

MMDPGP

In the case of MMDPGP we take another approach by assuming that the set of functions $f_i : t \mapsto f_i(t)$ are generated from replicate specific GPs within each cluster κ and we denote this association with superscript r (i.e, $f_i^{(r)}(t)$). The generative model for $f_i^{(r)}(t)$) will then be

$$
f_i^{(r)}(t) \sim \mathcal{GP}\left(m_\kappa^{(r)}(t), k_\kappa^{(r)}(t,t')\right), \quad r \in \{1, \ldots, R\}. \tag{12}
$$

If we denote by $\mu_i^{(r)} = [f_i^{(r)}(t_j)]_{j=1}^T$ the vector of evaluations of $f_i^{(r)}(t)$ at time points t_j, the vector $g_i^{(r)}$ representing the gene expression values of gene i at every time point t_j of replicate r can be sampled from the following replicate specific conditional distribution:

$$
g_i^{(r)} | f_i^{(r)} \sim \mathcal{N}\left(\mu_i^{(r)}, (\sigma_n^{(r)})^2 \mathbb{I}\right), \tag{13}
$$

where $(\sigma_n^{(r)})^2$ is the observation noise between genes of the same replicate in the cluster.

In order to perform inference on the model we can marginalize out $f_i^{(r)}(t)$ and express the model conditioned on $m_\kappa^{(r)}(t)$. In the marginalized model the expression values of the same gene from different replicate measurements have a zero covariance, as they come from different GPs, we can reduce the computations by building smaller

covariance matrices with the observations from each replicate separately. If we denote by $\mu_\kappa^{(r)} = [m_\kappa^{(r)}(t)]_{t=1}^T$, and by $K_{\mathrm{MM}_\kappa}^{(r)}$ the covariance matrix in Eq (12) for genes in replicate r, we can sample $g_i^{(r)}$ as:

$$g_i^{(r)} | m_\kappa^{(r)}(t) \sim \mathcal{N}\left(\mu_\kappa^{(r)}, K_{\mathrm{MM}_\kappa}^{(r)} + (\sigma_n^{(r)})^2 \mathbb{I}\right). \tag{14}$$

We can approximate the value of μ_κ as $\mu_\kappa^{(r)} \approx \frac{1}{N_\kappa} \sum_{i=1}^{N_\kappa} \mu_i^{(r)}$, where N_κ is the number of genes in cluster κ. The generative model of MMDPGP will then be:

$$\begin{aligned} G &\sim DP(\alpha, H) \\ \theta_\kappa &\sim G \\ g_i^{(r)} &\sim \mathcal{N}\left(\mu_\kappa^{(r)}, K_{MM_\kappa}^{(r)}\right). \end{aligned} \tag{15}$$

The computational complexity for fitting the GP of a single cluster in the case of MMDPGP is $\mathcal{O}(N_\kappa R T^3)$ in time and $\mathcal{O}(N_\kappa R T^2)$ in space, which is significantly lower than MOHGP but larger than DPGP.

Implementation

In this section, we focus exclusively on describing our implementations for MMDPGP and MOHGP (Hensman et al.). In the case of DPGP (McDowell et al.), we utilized their software, which is extensively detailed in [2].

In the Gaussian process Dirichlet process model (Eq (4a)-(4c)) the DP generates the distributions parameters $\theta_\kappa = \{m_\kappa^{(r)}(t), l_\kappa^{(r)}, \sigma_{f,\kappa}^{(r)}, \sigma_{n,\kappa}^{(r)}\}_{r=1}^R$ for the cluster specific GP(s). For $m_\kappa^{(r)}(t)$ we denote by $\mu_\kappa^{(r)}$ the vector of its evaluations on the observed time points. We set the following conjugate priors for the variance and observation noise as suggested in [2], which fit the characteristics of our data:

$$\begin{aligned} \left(\sigma_{n,\kappa}^{(r)}\right)^2 &\sim \mathrm{InverseGamma}(12, 2) \\ \left(\sigma_{f,\kappa}^{(r)}\right)^2 &\sim \mathrm{logNormal}(0, 1). \end{aligned} \tag{16}$$

We set bounding constrains on the length scale $l_\kappa^{(r)}$ such that $\Delta t_{\min} \leq l_\kappa^{(r)} \leq \Delta t_{\max}$, where $\Delta t_{min} = \min(t_i - t_{i-1})$ and $\Delta t_{max} = \max(t_i - t_j)$ as described in [23], where t_i, t_j are time points from the gene expression time series. Finally, we sample the means of the replicates $\mu_i^{(r)}$ following Eq (6) and Eq (12) for MOHGP and MMDPGP accordingly. We initialise the parameters of each cluster with some default values (e.g., $(l_0, \sigma_{f_0}, \sigma_{n_0}) = (1, 1, 0.2)$) to determine the initial covariance matrices K_{MM_0}, K_{HE_0} and for μ_0 we sample a multivariate normal with zero mean, and covariance matrices K_{HE_0} and K_{MM_0} for each method respectively.

For the purpose of our clustering method we have used an implementation [2] of Neal's Gibbs sampling "Algorithm 8" [24], which yields the posterior probability for assigning a gene expression time series to a particular cluster. If we denote by c_i the cluster of gene i and $c_{\setminus i}$ the set of clusters of each gene except for gene i, the algorithm determines the probability for assigning gene i to any cluster κ conditioned on all other cluster assignments as follows:

$$P(c_i = \kappa | [g_i^{(r)}]_{r=1}^R, c_{\setminus i}, \alpha, \theta_\kappa) \propto P(c_i = \kappa | c_{\setminus i}, \alpha) P([g_i^{(r)}]_{r=1}^R | c_i = \kappa, \theta_\kappa). \tag{17}$$

$P(c_i = \kappa | c_{\setminus i}, \alpha)$ is a prior probability which does not depend on any characteristics of the time series for gene i but is based solely on the information about cluster κ:

$$P(c_i = \kappa | c_{\setminus i}, \alpha) \propto \begin{cases} \frac{\alpha/m}{\alpha+N-1} & \text{if the cluster is empty or gene } i \text{ is the only element,} \\ \frac{\sum_{j=1}^{N} \delta(c_j=\kappa)}{\alpha+N-1} & \text{otherwise,} \end{cases} \tag{18}$$

where m is the number of new clusters at each iteration which we set to 4, α is the clustering concentration parameter of the DP which we set to 1 as it has been found convenient on other gene clustering applications [25], N is the total number of clustered genes. However, while testing for different values of α and m we noticed that their impact on the number of clusters generated by MMDPGP is insignificant due to the good fit of the genes in the existing clusters.

$P([g_i^{(r)}]_{r=1}^R | c_i = \kappa, \theta_\kappa)$ is the likelihood that the time series $[g_i^{(r)}]_{r=1}^R$ for all replicate measurements of gene i are generated by the GP(s) which parameterize cluster κ, and is given by Eq (19) and Eq (20) for MOHGP and MMDPGP respectively. Additionally, μ_κ, $\mu_\kappa^{(r)}$ are calculated as averages, like we indicated in MOHGP and MMDPGP sections.

$$P([g_i^{(r)}]_{r=1}^R | c_i = \kappa, \theta_\kappa) \propto \begin{cases} N([g_i^{(r)}]_{r=1}^R | [\mu_0]_{r=1}^R, K_{\text{HE}_0}) & \text{if the cluster is empty,} \\ N([g_i^{(r)}]_{r=1}^R | [\mu_\kappa]_{r=1}^R, K_{\text{HE}_\kappa}) & \text{otherwise,} \end{cases} \tag{19}$$

$$P([g_i^{(r)}]_{r=1}^R | c_i = \kappa, \theta_\kappa) \propto \begin{cases} \prod_{r=1}^R N(g_i^{(r)} | \mu_0, K_{\text{MM}_0}^{(r)}) & \text{if the cluster is empty,} \\ \prod_{r=1}^R N(g_i^{(r)} | \mu_\kappa^{(r)}, K_{\text{MM}_\kappa}^{(r)}) & \text{otherwise,} \end{cases} \tag{20}$$

In order to select the optimal clusters for the genes we use a posterior similarity matrix which tells the proportion of samples generated by the Gibbs sampler for which every pair of genes (i, j) is found to be on the same cluster as described in [2].

Data description and preprocessing

The data that we used consisted of gene expression time series composed of RNA-Seq experiments sampled during different growth stages of *C. botulinum*.

RNA-Seq data collection

The RNA-Seq data were obtained by collecting RNA samples from growth experiments of three replicate cultures of *C. botulinum* strain ATCC 19397 at 13 different time points. RNA samples were collected for each replicate at consistent time intervals corresponding to the growth experiment, specifically at 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15 and 24 hours post-initiation. Then, 92 ERCC RNA spike-in controls [26, 27] were added to the collected samples and they were analyzed using paired-end sequencing with Ilumina's NextSeq technology.

Data preprocessing

Genes were assigned to COG (i.e., Clusters of Orthologous Groups of proteins) [28] categories using *EggNOG-mapper* [29] and used in further analysis, as detailed below. We then carried out an operon identification analysis using *Rockhopper* [30]. *Operons* are clusters of co-expressed genes with related functions [31] and are often situated adjacently on the DNA strand, leading to their collective activation. In the clustering analyses presented here we have used the information about operons as an informative prior by enforcing genes within the same operon to be on the same cluster. This would make the GP(s) of each cluster better capture the dynamic patterns of genes that are

potentially involved in the same processes in what it is called as guilt by association [32]. The gene expression data which consist of values in TPM (i.e., transcripts per million) units were then normalised using a spike-in-based normalisation. Genes with low expression values were then filtered as low values are not strongly reliable and can introduce a potential source of noise in the analysis. A detailed description of the data preprocessing including data preparation, normalisation and filtering is included in S1 Appendix.

Before fitting the transcriptomics time courses to our model we artificially pulled the last two time points of each time series closer to the rest - that is, the 15h and 24h time points were reassigned to 13.5h and 15h time points respectively. This is because the model assumes that the transcription dynamics of the genes happened with the same intensity through out the time course, which is not a realistic assumption in our case (i.e., in the beginning of the life-cycle [2h-12h] we expect rapid non-monotonic changes, while at the end the changes are very small and monotonic). Alternatively, the problem could be addressed by using a non-stationary kernel similarly to [33], but given that the change of transcription dynamic is expected only on the later time points we implemented this simpler approach.

Results

Clustering of the genes using RNA-Seq time series

We applied the multi-replicate clustering methods MMDPGP, MOHGP [6] and DPGP [2] on the RNA-Seq data to group the genes according to their transcription dynamics. Even though we used the same values for initializing the parameters of the methods we obtained very different results. In the case of MMDPGP we acquired 26 different clusters, MOHGP method returned 116 clusters where most of the clusters have a small number of genes, while DPGP yielded only 13 clusters. The reason for this is because in the case of MOHGP during each iteration of the Gibbs sampler the likelihood of each gene to belong to a new cluster is higher. This is because there is only one GP for each cluster which captures the variation between all replicates, and given a not negligible difference between replicates, it often results in a poor fit for the gene under consideration, thus the gene is assigned to a new cluster. In the case of MMDPGP however, every cluster has a separate GP for each replicate and the biological replicates are compared on how well they fit in the replicate specific GP at each cluster, thus resulting in a better fit at the cluster whose genes express similarly on the same replicate. DPGP on the other hand ignores the variation between replicates and performs clustering using the replicates' means. When performing the mean of the replicates, the replicate's gene specific nuances vanish resulting in a larger number of similar time series, hence the smaller number of clusters. Fig 1 shows the GP models for three clusters generated by the methods. We can see how in the case of MMDPGP the GPs of different replicates differ because even though the replicates are similar they have their own specific characteristics depending on natural variation between the different replicates. When exploited, this natural variation can be a useful source of additional knowledge during the clustering procedure.

Biological significance analysis

To understand the biological relevance of our results, we carried out a gene enrichment analysis using COG annotations assigned to each gene (see "Data pre-processing" subsection). We chose this method because the standard gene ontology (GO) enrichment analysis, was not available for *C. botulinum* that is not annotated in the GO

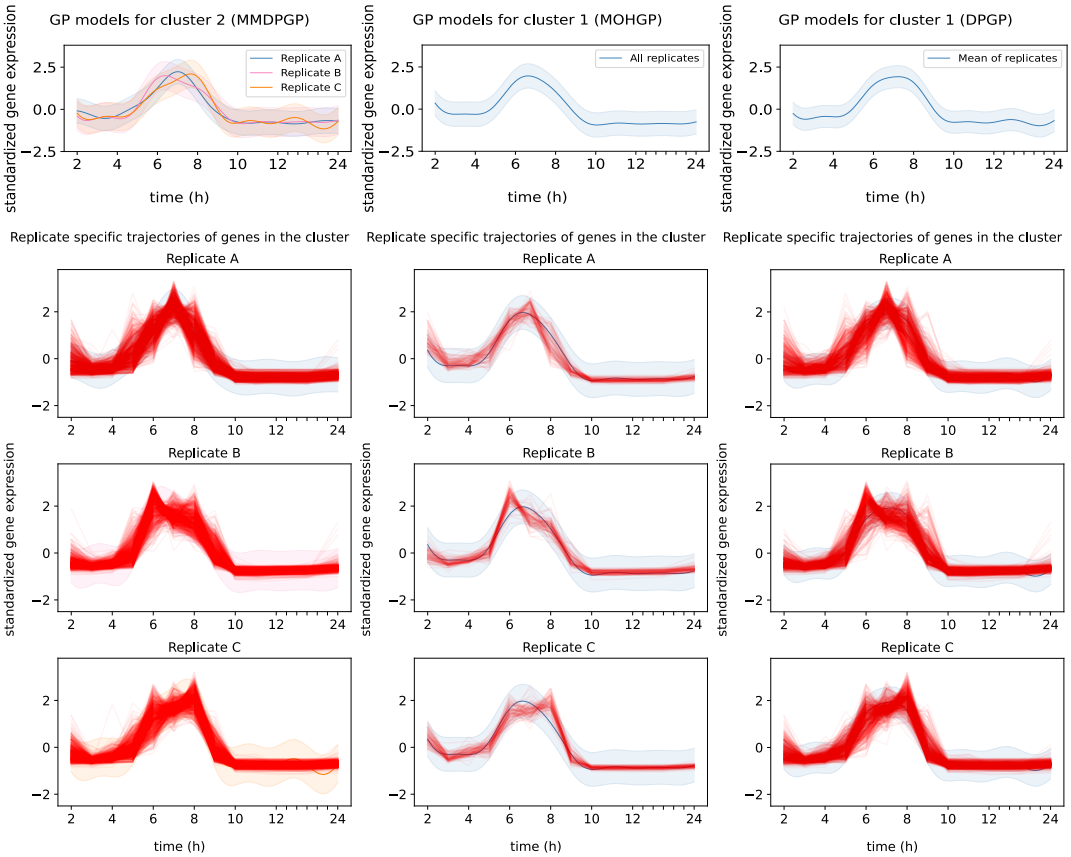

Fig 1. Comparison between 3 clusters generated by MMDPGP (left column plots), MOHGP [6] (middle column plots), and DPGP [2] (right column plots). The three replicates in the MOHGP's cluster and DPGP's cluster are modeled by one GP (top middle column plot and top right column plot). The three replicates in the MMDPGP cluster are modeled by three different GPs (top left plot). The replicate-specific trajectories of the genes in each cluster are plotted separately on top of their respective GP. In the case of MOHGP and DPGP they are plotted on top of the same GP model which characterizes the respective cluster. In the case of of MMDPGP cluster the trajectories are plotted on top of their respective replicate specific GP. The x-axis is drawn to indicate the real time of the RNA-Seq measurements.

database. By using COG annotations, it is still possible to group the genes in a meaningful way and explore their potential functions. Fig 2 shows the significant COG enrichment across clusters obtained from each of the methods. The significant COG enrichment was determined by considering those which have a FDR-corrected p-value smaller than 0.05. The significantly enriched COG classes at each cluster are marked with -$\log_{10}(p)$ while the rest are marked with 0. The level of significance for the significantly enriched COG classes is indicated by the number of stars in the heatmap.

The heatmap for MMDPGP reveals a total of 5 significantly enriched COG classes and significant enrichment was detected in 5 clusters. In contrast, the heatmaps for MOHGP and DPGP methods show 2 and 4 detections of significant enrichments respectively. For higher levels of significance (i.e., $p \leq 0.01$), MMDPGP still reveals 5 significantly enriched COG classes, while the numbers drop to 1 and 2 significant enrichments for the other methods. Additionally, all the significantly enriched COG classes with $p \leq 0.01$, which are detected in MOHGP and DPGP, are also detected in

MMDPGP. These indicate that the genes within each cluster of MMDPGP display patterns of higher biological similarity, compared to those inside the clusters that were generated by the other methods.

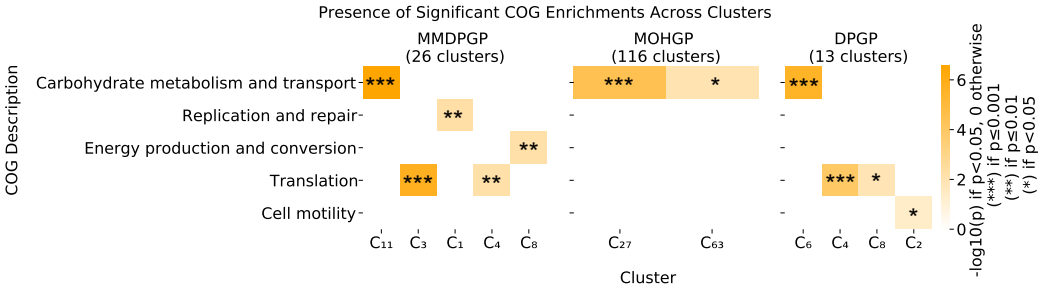

Fig 2. Gene enrichment analysis based on COG for MMDPGP (proposed), MOHGP [6] and DPGP [2]. Presence of significant COG enrichment is shown by $-\log_{10}(p)$ for the COG categories with $p < 0.05$ at each cluster. COG categories which do not pass the p-value threshold are marked with 0. The enrichment significance of the COG classes is indicated by the number of stars in the corresponding class.

Leave-one-out predictive likelihood

To estimate the quality of the fit of the genes in their corresponding cluster, we performed a leave-one-out predictive likelihood analysis as described in [34]. This is a standard measure for performing model selection with GPs. In order to do it, we refit the data of each cluster to its respective GP model by removing one time point from each gene/operon at a time. At each iteration, we calculated the log-likelihood of the model to predict the observed gene expression values at the time point that was removed. We did this for all time points (i.e., 13 time points) and we divided the numbers with the total number of single observations, to express the results in natural logarithm units per observation (nat/obs.). This normalisation allows for a fair comparison, given that in the case of DPGP the number of observations is 3 times smaller because they use only the mean of the 3 replicates as observation data. Fig 3 shows the results of the leave-one-out predictive likelihood analysis. We can see that MOHGP performs slightly better in the analysis by having a higher predictive likelihood for the time points overall compared to MMDPGP. This result can be explained by the fact that in the case of MOHGP most of the clusters have a very small number of genes which results in a better modeling of the data, but at the expense of producing less biologically meaningful cluster as indicated by the gene enrichment analysis.

Performance on synthetic data with different amount of noise

We tested the classification accuracy of our methods by generating time series data from 6 different clusters. Each cluster was represented by a GP with a different mean function, and same kernel parameters (l, σ_f) and observation noise σ_n^2. We sampled 10 different time series from each GP with mean $m(t) \in \{\sin(x/2), -\sin(x/2), \sin(x/4), -\sin(x/4), \cos(x/2), -\cos(x/2)\}$ and introduced 2 additional replicates. For the first 5 clusters the replicates were generate by adding random noise σ. The additional replicates from the last cluster were obtained by horizontally shifting to the right (e.g., suggesting genes which are expressed earlier or later in different replicates) the initial samples by Δ and 2Δ respectively.

We simulated the process multiple times for different values of the GP model variance $\sigma_f^2 \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, replicate noise standard deviation

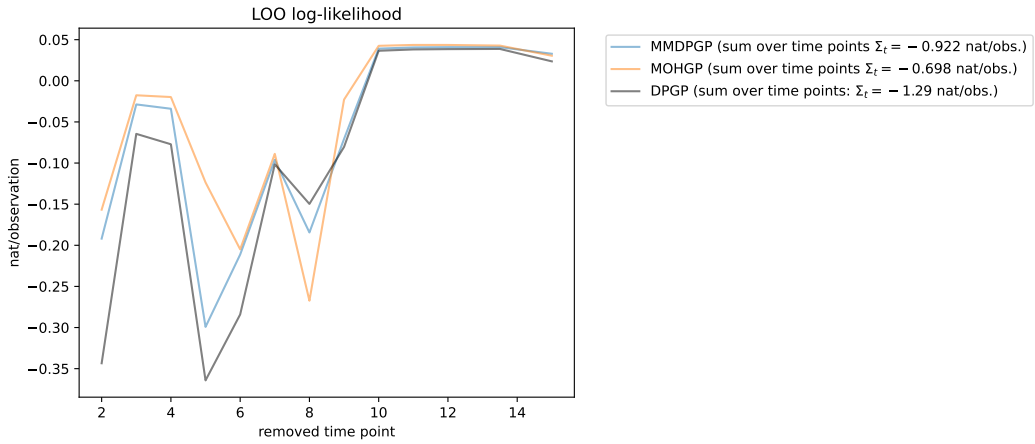

Fig 3. Sum of predictive log-likelihoods at each time point for the different methods (i.e., MMDPGP (proposed), MOHGP [6] and DPGP [2]) on the left out time points for each observation. Each point represents the sum of log-likelihoods of the observed gene expression values at the removed time point over all the posterior cluster models divided by the number of observations. In the legend we can also see the sums over all the removed time points, for each model.

$\sigma \in \{0, 0.2, 0.4, 0.5\}$, and length of phase shift $\Delta \in \{0, 0.3, 0.5\}$. The length scale and model noise were set to default values (i.e., $(l, \sigma_n^2) = (2, 1)$). Each simulation resulted in clusters' data with different attributes that are determined by the triple $(\sigma_f^2, \sigma, \Delta)$. Then, we run the clustering methods with each cohort of generated data. In order to evaluate the results obtained by the methods we used a measure of purity for the clusters as defined in [35]. Purity assesses clustering quality by evaluating how well it avoids mixing different types of objects within the same cluster. This is a desired property in our context if we want to generate hypothesis about the functionality of unknown genes based on genes which we know in the same cluster. Fig 4 shows the purity values of the clusters obtained by running MMDPGP, MOHGP and DPGP methods on different cohorts of synthetic data. As it can be seen MMDPGP outperforms the other methods at all cohorts of synthetic data in terms of cluster purity measure. This means that MMDPGP is less likely to put time series which are originally generated by different clusters into the same one.

Discussion

Dirichlet processes (DP) and Gaussian processes (GP) can be combined and used to cluster functional data without making strong assumptions such as the number of clusters to group the data. In this paper we proposed a model-based method for the purpose of clustering gene expression time series with multiple replicates. Our method uses a different GP to represent each replicate inside every cluster, for this reason we call it multiple models Dirichlet process Gaussian process (MMDPGP). We compared our method with two similar approaches, MOHGP, proposed by Henseman et al. [6], and DPGP, proposed by McDowell et al. [2]. We described the three methods and their generative model assumptions in detail. Python implementations were also provided for MMDPGP and MOHGP. One clear advantage of MMDPGP over MOHGP is its lower computational complexity which makes it useful for handling larger amount of data (e.g., data consisting of dense time series of thousands of genes, with many replicates). On the other hand even though DPGP has an even lower computational complexity

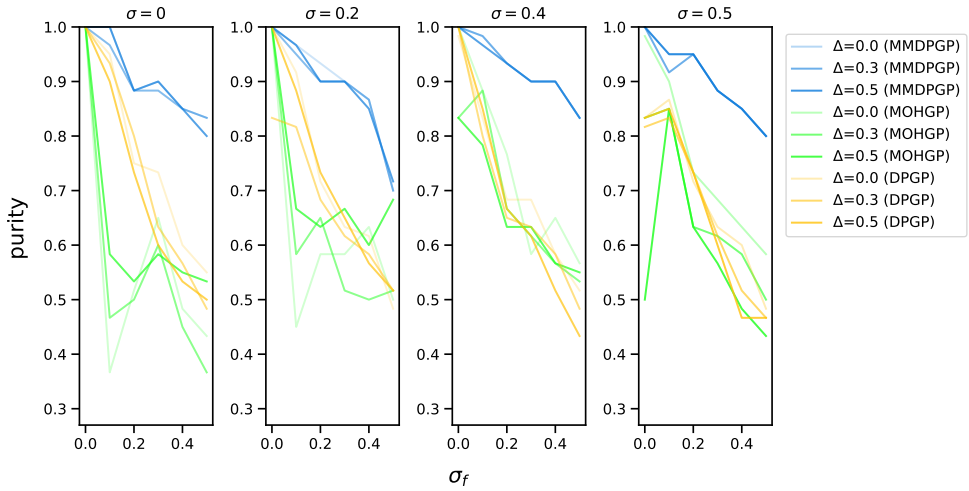

Fig 4. Cluster purity of the results obtained by applying MMDPGP (blue), MOHGP [6] (green) and DPGP [2] (amber) on synthetic data generated with different parameters $(\sigma_f, \sigma, \Delta)$.

compared to MMDPGP, it ignores the variation between replicates, which can be a useful source of information for clustering the genes in a biologically meaningful way.

We applied the clustering methods on a novel RNA-Seq time series data set consisting of three experimental replicates. We performed a COG enrichment analysis on the clusters produced by each method and showed that in the case of MMDPGP a higher number of significant enrichments (i.e., $p < 0.05$) are detected. Additionally for higher levels of significance (i.e., $p < 0.01$) not only the number of significant enriched classes is higher in the case of MMDPGP but all the significant enrichments detected by the other methods were also detected by MMDPGP.

The leave-one-out predictive likelihood analysis showed that MOHGP performed better than MMDPGP, but that is because the size of the clusters returned by MOHGP are very small which can cause an overfit to the GP of the clusters, thus yielding higher predictive likelihood overall.

Finally, to demonstrate the accuracy of MMDPGP we generated synthetic data from six different clusters by introducing different levels of noise and tested the performance of each method by measuring the cluster purity of the results. MMDPGP outperformed the other methods by scoring a significantly higher cluster purity as shown in Fig 4. We believe that MMDPGP can be useful for clustering gene expression time series data with multiple replicates because it effectively captures the information provided by experimental replicates and uses it to provide a meaningful discrimination between gene expression profiles. By modeling each replicate with a separate GP it provides more details for each individual replicate which can be useful for clustering in the cases when replicates differ, for instance because of different experimental conditions.

Supporting information

S1 Appendix. Data preparation and normalisation.

Contributions and Acknowledgments

E.N. and A.H. developed the method; E.N. implemented the model and performed computational analyses; F.D., K.S., and M.L. provided the RNA-Seq data and biological expertise; E.N wrote the manuscript; all authors reviewed the manuscript. We declare no competing interests related to this work. The present study was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC-CoG whyBOTher, grant 683099), the HiLife Fellows Program and by the University of Helsinki. We thank the Biomedicum Functional Genomics Unit at the Helsinki Institute of Life Science and Biocenter Finland at the University of Helsinki for performing the sequencing of the samples analyzed in this work. We are grateful to Hanna Korpunen for her excellent technical work in the laboratory.

References

1. Hensman J, Lawrence ND, Rattray M. Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. BMC Bioinformatics. 2013;14(1):1–12.

2. McDowell IC, Manandhar D, Vockley CM, Schmid AK, Reddy TE, Engelhardt BE. Clustering gene expression time series data using an infinite Gaussian process mixture model. PLoS Computational Biology. 2018;14(1):e1005896.

3. Yeung KY, Fraley C, Murua A, Raftery AE, Ruzzo WL. Model-based clustering and data transformations for gene expression data. Bioinformatics. 2001;17(10):977–987. doi:10.1093/bioinformatics/17.10.977.

4. Ferguson TS. A Bayesian analysis of some nonparametric problems. The Annals of Statistics. 1973;1(2):209–230.

5. Rasmussen C, Williams C. Gaussian Processes for Machine Learning. Cambridge, MA, USA: MIT Press; 2006.

6. Hensman J, Rattray M, Lawrence ND. Fast nonparametric clustering of structured time-series. IEEE Trans Pattern Anal Mach Intell. 2014;37(2):383–393.

7. Rasmussen C, De la Cruz BJ, Ghahramani Z, Wild DL. Modeling and visualizing uncertainty in gene expression clusters using Dirichlet process mixtures. IEEE/ACM Trans Comput Biol Bioinform. 2008;6(4):615–628.

8. Medvedovic M, Yeung KY, Bumgarner RE. Bayesian mixture model based clustering of replicated microarray data. Bioinformatics. 2004;20(8):1222–1232.

9. Cooke EJ, Savage RS, Kirk PD, Darkins R, Wild DL. Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements. BMC Bioinformatics. 2011;12:1–12.

10. Heller KA, Ghahramani Z. Bayesian hierarchical clustering. In: Proceedings of the 22nd international conference on Machine learning; 2005. p. 297–304.

11. Heinonen M, Guipaud O, Milliat F, Buard V, Micheau B, Tarlet G, et al. Detecting time periods of differential gene expression using Gaussian processes: an application to endothelial cells exposed to radiotherapy dose fraction. Bioinformatics. 2014;31(5):728–735. doi:10.1093/bioinformatics/btu699.

12. Reid JE, Wernisch L. Pseudotime estimation: deconfounding single cell time series. Bioinformatics. 2016;32(19):2973–2980. doi:10.1093/bioinformatics/btw372.

13. Lawrence N, Sanguinetti G, Rattray M. Modelling transcriptional regulation using Gaussian Processes. In: Advances in Neural Information Processing Systems. vol. 19; 2006.Available from: `https://proceedings.neurips.cc/paper_files/paper/2006/file/f42c7f9c8aeab0fc412031e192e2119d-Paper.pdf`.

14. Honkela A, Girardot C, Gustafson E, Liu YH, Furlong E, Lawrence N, et al. Model-based method for transcription factor target identification with limited data. Proc Natl Acad Sci USA. 2010;107:7793–7798. doi:10.1073/pnas.0914285107.

15. Jo K, Sung I, Lee D, Jang H, Kim S. Inferring transcriptomic cell states and transitions only from time series transcriptome data. Scientific Reports. 2021;11(1):12566.

16. Gao P, Honkela A, Rattray M, Lawrence ND. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. Bioinformatics. 2008;24(16):i70–i75. doi:10.1093/bioinformatics/btn278.

17. Murphy KP. Probabilistic Machine Learning: Advanced Topics. MIT Press; 2023. Available from: `http://probml.github.io/book2`.

18. Li Y, Schofield E, Gönen M. A tutorial on Dirichlet process mixture modeling. Journal of Mathematical Psychology. 2019;91:128–144.

19. Rasmussen C, Ghahramani Z. Infinite mixtures of Gaussian process experts. Advances in Neural Information Processing Systems. 2001;14.

20. Pradier MF, Perez-Cruz F. Infinite mixture of global Gaussian processes. In: Bayesian Non-parametric: the Next Generation Workshop in Neural Information Processing Systems; 2015.

21. Meeds E, Osindero S. An alternative infinite mixture of Gaussian process experts. Advances in Neural Information Processing Systems. 2005;18.

22. Sun S. Infinite mixtures of multivariate Gaussian processes. In: 2013 International Conference on Machine Learning and Cybernetics. vol. 3. IEEE; 2013. p. 1011–1016.

23. Topa H, Honkela A. Gaussian process modelling of multiple short time series. arXiv preprint arXiv:12102503. 2012;.

24. Neal RM. Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics. 2000; p. 249–265.

25. Medvedovic M, Sivaganesan S. Bayesian infinite mixture model based clustering of gene expression profiles. Bioinformatics. 2002;18(9):1194–1206.

26. The External RNA Controls Consortium. The external RNA controls consortium: a progress report. Nature Methods. 2005;2(10):731–734.

27. Jiang L, Schlesinger F, Davis CA, Zhang Y, Li R, Salit M, et al. Synthetic spike-in standards for RNA-seq experiments. Genome Research. 2011;21(9):1543–1551.

28. Tatusov RL, Koonin EV, Lipman DJ. A genomic perspective on protein families. Science. 1997;278(5338):631–637.

29. Cantalapiedra CP, Hernández-Plaza A, Letunic I, Bork P, Huerta-Cepas J. EggNOG-mapper v2: functional annotation, orthology assignments, and domain prediction at the metagenomic scale. Molecular Biology and Evolution. 2021;38(12):5825–5829.

30. Tjaden B. A computational system for identifying operons based on RNA-seq data. Methods. 2020;176:62–70. doi:https://doi.org/10.1016/j.ymeth.2019.03.026.

31. Osbourn AE, Field B. Operons. Cellular and Molecular Life Sciences. 2009;66(23):3755–3775.

32. Vey G. Metagenomic guilt by association: an operonic perspective. PLOS One. 2013;8(8):e71484.

33. Heinonen M, Guipaud O, Milliat F, Buard V, Micheau B, Tarlet G, et al. Detecting time periods of differential gene expression using Gaussian processes: an application to endothelial cells exposed to radiotherapy dose fraction. Bioinformatics. 2015;31(5):728–735.

34. Vehtari A, Mononen T, Tolvanen V, Sivula T, Winther O. Bayesian leave-one-out cross-validation approximations for Gaussian latent variable models. Journal of Machine Learning Research. 2016;17(103):1–38.

35. Schütze H, Manning CD, Raghavan P. Introduction to Information Retrieval. Cambridge University Press; 2008.