

BCClong: An R Package for Bayesian Consensus Clustering for Multiple Longitudinal Features

Zhiwen Tan, Chang Shen, Zihang Lu

Abstract It is very common nowadays for a study to collect multiple longitudinal features, and appropriately integrating these features simultaneously for defining individual subgroups (i.e., clusters) becomes increasingly crucial to understanding population heterogeneity and predicting future outcomes. The aim of this paper is to describe a new package, BCClong, which implements a Bayesian consensus clustering (BCC) model for multiple longitudinal features. Compared to existing packages, several key features make the BCClong package appealing: (a) it allows simultaneous clustering of mixed-type (e.g., continuous, discrete, and categorical) longitudinal features, (b) it allows each longitudinal feature to be collected from different sources, with measurements taken at distinct sets of time points (known as irregularly sampled longitudinal data), and (c) it relaxes the assumption that all features have the same clustering structure by estimating the feature-specific (local) clusterings and consensus (global) clustering. Using two real data examples, we provide a tutorial with step-by-step instructions on how to use the package.

1 Introduction

Cluster analysis has been widely used to identify subgroups of the study population, and it is also known as an unsupervised learning method in the machine learning literature. When longitudinal data is available, identifying distinct developmental patterns is a common goal in many research studies. Traditional statistical models often focus on clustering a single longitudinal feature, which may ignore the complexity of the clustering structures and the dynamic relationship when multiple longitudinal features are available. It is very common nowadays for a study to collect multiple features, and appropriately integrating multiple longitudinal features simultaneously for defining individual subgroups (i.e., clusters) becomes increasingly crucial to understanding population heterogeneity and predicting future outcomes. In particular, compared to clustering a single longitudinal feature, simultaneously clustering multiple longitudinal features (also known as joint clustering) facilitates the discovery of co-occurring trajectory patterns for multiple features of interest. Several R packages have been developed to cluster longitudinal data with multiple features. For model-based clustering, the **gbmt** package (Magrini, 2022a) and the **flexmix** package (Leisch, 2004) are developed to implement the group-based multi-trajectory analysis (GBMT) (Nagin et al., 2018; Magrini, 2022b). Both packages implement the EM algorithm, allowing unbalanced data and missing values, and can only be applied to continuous features. Notably, a SAS procedure PROC TRAJ is also developed for this model and has been widely used (Nagin et al., 2018). The GBMT assumes that the features and repeated measures are independent conditional on the cluster membership. Alternatively, the **lcmm** and **mixAK** packages are developed under the framework of mixed-effect models. The dependence between features and repeated measurements is captured by joint modeling of the random effects. Specifically, the **lcmm** package is designed for performing analyses such as latent class mixed models for a single longitudinal feature, joint latent class mixed models for multiple longitudinal features, as well as joint latent class mixed models for longitudinal features and time-to-event outcomes. While this package allows modeling multiple continuous or categorical longitudinal data of the same type, it does not support clustering mixed data types (e.g., continuous and categorical) simultaneously. On the other hand, the **mixAK** package can handle same-type (e.g., multiple continuous) or mixed-type (e.g., continuous, discrete, and categorical) features. For nonparametric clustering, the **kml3d** package (Genolini et al., 2013) is developed to implement the K-means clustering for multiple longitudinal features, which is an extension of the **kml** package (Genolini and Falissard, 2009) for a single longitudinal feature. However, this package can only be applied to continuous features and does not allow for missing data. See Lu et al. (2023) for a review and comparison of these packages. The aim of this paper is to describe a new package, the **BCClong** package, which implements the BCC model for multiple longitudinal features (Lu and Lou, 2022b; Tan et al., 2022). Compared to existing packages, several key features make the **BCClong** package appealing: (a) it allows simultaneous clustering of mixed-type (e.g., continuous, discrete, and categorical) longitudinal features, (b) it allows each longitudinal feature to be collected from different sources, with measurements taken at distinct sets of time points (known as irregularly sampled longitudinal data), and (c) it relaxes the assumption that all features have the same clustering structure by estimating the feature-specific (local) clusterings and overall (global) clustering. This model is very flexible, which facilitates the interpretation of clustering results and

enhances its practical utilities.

2 Models and software

The methodology of the BCC model for clustering multiple longitudinal features implemented in the **BCClong** package is described in detail in Lu and Lou (2022b) (for multiple continuous features) and in Tan et al. (2022) (for multiple mixed-type features). In this section, we provide a brief overview of this model and the **BCClong** package. Without loss of generality, we use the term "features" to represent variables of interest used for the cluster analysis; however, these variables can also be a patient's traits, biomarkers, risk factors, exposures, and outcomes of interest in different contexts and studies.

2.1 Bayesian Consensus Clustering for Multiple Longitudinal Features

The BCC model is originally proposed as a flexible approach to model the dependence and heterogeneity of the data collected from multiple sources (Lock and Dunson, 2013). Instead of arriving at a single overall clustering structure, the BCC model allows each feature to follow feature-specific (local) clustering and these local clusterings are aggregated to find a consensus (global) clustering. Let $L_{i,r} = 1, \dots, K$ denote the feature-specific (local) cluster label for individual i and feature r for $r = 1, \dots, R$, and $C_i = 1, \dots, K$ denote the consensus (global) cluster label for individual i , where K denotes the number of clusters and R denotes the number of longitudinal features. Let $\mathbf{y}_{i,r} = (y_{i1,r}, \dots, y_{in_{i,r},r})^\top$ denote the measurement for individual i and feature r , where $y_{ij,r}$ denotes the observation j of individual i for feature r , and $n_{i,r}$ is the number of measurements, for $i = 1, \dots, N, j = 1, \dots, n_{i,r}, r = 1, \dots, R$. The BCC model assumes that $\mathbf{L}_r = (L_{1,r}, \dots, L_{N,r})$ is dependent on $\mathbf{C} = (C_1, \dots, C_N)$ through $P(L_{i,r} = k | C_i) = \vartheta(k, C_i, \alpha_r)$, where α_r adjusts the dependence function $\vartheta(\cdot)$. The conditional model can be specified as

$$P(L_{i,r} = k | \mathbf{y}_{i,r}, C_i, \gamma_{k,r}, \beta_{ik,r}) \propto \vartheta(k, C_i, \alpha_r) f_{k,r}(\mathbf{y}_{i,r} | \gamma_{k,r}, \beta_{ik,r}) \quad (1)$$

where $\gamma_{k,r}$ and $\beta_{ik,r}$ denote the fixed effects and random effects for individual i belonging to cluster k , where $k = 1, \dots, K$. To specify $\vartheta(k, C_i, \alpha_r)$, the package uses the following function (Lock and Dunson, 2013).

$$\vartheta(k, C_i, \alpha_r) = \begin{cases} \alpha_r & \text{if } C_i = L_{i,r} \\ (1 - \alpha_r) / (K - 1) & \text{otherwise} \end{cases} \quad (2)$$

Therefore, α_r (for $r = 1, \dots, R$) can also be viewed as an adherence parameter representing the degree of agreement between $L_{i,r}$ and C_i . Moreover, $f_{k,r}(\mathbf{y}_{i,r} | \gamma_{k,r}, \beta_{ik,r})$ is a distribution from the exponential family with the dispersion parameter $\phi_{k,r}$ and the fully specified mean function is given by

$$h_{k,r}^{-1}(E(\mathbf{y}_{i,r} | \beta_{ik,r}, \gamma_{k,r})) = \boldsymbol{\eta}_{ik,r} = \mathbf{x}_{i,r}^\top \boldsymbol{\gamma}_{k,r} + \mathbf{Z}_{i,r}^\top \boldsymbol{\beta}_{ik,r} \quad (3)$$

where $h_{k,r}^{-1}$ is a canonical link function for the mean of the feature r in cluster k . Also, $\boldsymbol{\eta}_{ik,r} = (\eta_{i1k,r}, \dots, \eta_{in_{i,r}k,r})^\top$ is the linear predictor for feature r , $\mathbf{x}_{i,r}$ is a $p_r \times n_{i,r}$ vector of predictors, $\boldsymbol{\gamma}_{k,r}$ is the corresponding vector of coefficients, $\mathbf{Z}_{i,r}$ is a $q_r \times n_{i,r}$ vector of predictors, and $\boldsymbol{\beta}_{ik,r} | \cdot \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{k,r})$. In addition, the distribution of $\mathbf{y}_{i,r}$ can be written as

$$P(\mathbf{y}_{i,r} | \beta_{ik,r}, \gamma_{k,r}, \phi_{k,r}) = \prod_{j=1}^{n_{i,r}} \exp \left\{ \frac{y_{ij,r} \eta_{ijk,r} - g_{k,r}(\eta_{ijk,r})}{\phi_{k,r}} + w_{k,r}(y_{ij,r}, \phi_{k,r}) \right\} \quad (4)$$

where $g(\eta)$ and $w(y, \phi)$ are appropriate distribution-specific functions. For example, for features with Gaussian distributions, $g(\eta) = \eta^2/2$ and $w(y, \phi) = \frac{\log(2\pi\phi)}{2} - \frac{y^2}{2\phi}$. Furthermore, for three features with Gaussian, Poisson, and Binomial distributions, respectively, the model can be specified as:

$$\begin{aligned} (1): E(\mathbf{y}_{i,1} | \gamma_{k,1}, \beta_{ik,1}) &= \boldsymbol{\eta}_{ik,1} = \mathbf{x}_{i,1}^\top \boldsymbol{\gamma}_{k,1} + \mathbf{Z}_{i,1}^\top \boldsymbol{\beta}_{ik,1} \\ (2): \log(E(\mathbf{y}_{i,2} | \gamma_{k,2}, \beta_{ik,2})) &= \boldsymbol{\eta}_{ik,2} = \mathbf{x}_{i,2}^\top \boldsymbol{\gamma}_{k,2} + \mathbf{Z}_{i,2}^\top \boldsymbol{\beta}_{ik,2} \\ (3): \text{logit}(E(\mathbf{y}_{i,3} | \gamma_{k,3}, \beta_{ik,3})) &= \boldsymbol{\eta}_{ik,3} = \mathbf{x}_{i,3}^\top \boldsymbol{\gamma}_{k,3} + \mathbf{Z}_{i,3}^\top \boldsymbol{\beta}_{ik,3} \end{aligned}$$

where $i = 1, \dots, N, j = 1, \dots, n_{i,r}$ and $r = 1, 2, 3$. The model also involves dispersion parameters $\phi_{k,r} = \sigma_{k,r}^2$ for Gaussian distribution. The corresponding dispersion parameters for logistic regression and Poisson regression are both 1, for $r = 1, \dots, R$ and $k = 1, \dots, K$. Given this proposed model, the

complete-data likelihood can be written as

$$\mathbb{L}^{Bayes}(\Theta | \mathbf{y}_i, \beta_i; \mathbf{x}_i) = \sum_{k_1, \dots, k_R} \left(\prod_k \prod_{r=1}^R \theta(L_{i,r} = k_r, C_i = k, \alpha_r) \pi_k \right) \prod_{r=1}^R f(\mathbf{y}_{i,r} | \gamma_{ik_r, r}, \beta_{ik_r, r}; \mathbf{x}_{i,r}) \quad (5)$$

where $\mathbf{y}_i = (\mathbf{y}_{i,1}, \dots, \mathbf{y}_{i,R})$, $\mathbf{x}_i = (\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,R})$, $\beta_i = (\beta_{i,1}, \dots, \beta_{i,R})$, $\pi_k = P(C_i = k)$ and $\Theta = (\alpha, \pi, \gamma, \Sigma, \phi)$.

2.2 Bayesian Inference

For computational convenience, the **BCClong** package uses independent conjugate priors for the model parameters. Specifically, for the adherence parameters, the prior distribution is $\alpha_r \sim \text{TBeta}(\delta_{1,r}, \delta_{2,r}, 1/K)$ for $r = 1, \dots, R$, where TBeta denotes a truncated Beta distribution ranged from $[1/K, 1]$. For the cluster probabilities, the prior distribution is $\pi \sim \text{Dirichlet}(\phi_0)$. For the fixed effect coefficients, the prior distribution is $\gamma_{k,r} \sim \text{MVN}(\mathbf{0}, \mathbf{V}_{0k,r})$, where $\mathbf{V}_{0k,r}$ is an $m_{k,r} \times m_{k,r}$ variance-covariance matrix, and $m_{k,r}$ is the dimension of $\gamma_{k,r}$. The dispersion parameter $\phi_{k,r}$ is not constant only when the feature is a Gaussian distribution. In such a case, $\phi_{k,r} = \sigma_{k,r}^2$. The prior distribution is $\sigma_{k,r}^2 \sim \text{IG}(a_{0k,r}, b_{0k,r})$, where $a_{0k,r}$ and $b_{0k,r}$ are the parameters of an inverse gamma distribution. By definition, $\phi_{k,r} = 1$ for features that follow a Poisson or Binomial distribution, for $k = 1, \dots, K$ and $r = 1, \dots, R$. For the variance-covariance matrix of the random effect, the prior distribution is $\Sigma_{k,r}^{-1} \sim \text{Wishart}(\lambda_{0k,r}, (\lambda_{0k,r} \mathbf{\Lambda}_{0k,r})^{-1})$, where the prior for the Wishart distribution is parametrized such that the mean is $\mathbf{\Lambda}_{0k,r}^{-1}$. In the special case when $\Sigma_{k,r}$ is a diagonal matrix, that is, $\Sigma_{k,r} = \text{diag}(\xi_{k1,r}^2, \dots, \xi_{kq_r,r}^2)$, the prior is an inverse gamma distribution, i.e., $\xi_{km,r}^2 \sim \text{IG}(c_{0k,r}, d_{0k,r})$, for $m = 1, \dots, q_r$. For posterior computation, a Gibbs sampling approach to update the model parameters is described as follows. After initializing the parameters, the algorithm repeats the following steps until convergence. At the s step of the iteration,

- Update local cluster membership $L_{i,r}^{(s)}$ given $\{\mathbf{y}_{i,r}, \Theta_r^{(s-1)}, \alpha_r^{(s-1)}, C_i^{(s-1)}, \beta_r^{(s-1)}\}$, for $i = 1, \dots, N$ and $r = 1, \dots, R$.
- Update $\alpha_r^{(s)}$ given $\{C^{(s-1)}, L^{(s)}\}$, for $r = 1, \dots, R$.
- Update $C_i^{(s)}$ given $\{L^{(s)}, \alpha^{(s)}, \pi^{(s-1)}\}$, for $i = 1, \dots, N$.
- Update $\pi^{(s)}$ given $C^{(s)}$.
- Update cluster-specific parameters $\Theta^{(s)} = (\gamma^{(s)}, \Sigma^{(s)}, \sigma^{2(s)})$ and $\beta^{(s)}$. In particular, $\gamma^{(s)}$ and $\beta^{(s)}$ are updated via the Metropolis-Hastings algorithm.

Each MCMC iteration will produce a realization of cluster membership for C, L_1, \dots, L_R . The package uses the mode over all the MCMC samples (after burn-in and thinning) as the point estimates for both feature-specific and global clustering. Label switching is a common phenomenon in mixture models. This problem arises due to both the likelihood and posterior distributions being invariant to permutations of the parameters. The **BCClong** package applies a post-processing algorithm, via the **label.switching** package (Papastamoulis, 2016), to reorder the labels based on Kullback-Leibler divergence (Stephens, 2000). The computation of the BCC model can be carried out using the main function in the **BCClong** package:

```
BCC.multi(mydat, dist, id, time, formula, num.cluster,
         hyper.par, initials, initial.cluster.membership,
         input.initial.local.cluster.membership,
         input.initial.global.cluster.membership,
         burn.in, thin, per, max.iter, ...)
```

The meaning of the key arguments of the **BCC.multi()** function is described as follows.

- **mydat**: list of R longitudinal features (i.e., with a length of R), where R is the number of features. The data should be prepared in a long format (each row is one time point per individual).
- **dist**: a character vector (with a length of R) that determines the distribution for each feature. Possible values are "gaussian" for a continuous feature, "poisson" for a discrete feature (e.g., count data) using a log link, and "binomial" for a dichotomous feature (0/1) using a logit link. A single value (i.e., a length of 1) is recycled if necessary.
- **id**: a list (with a length of R) of vectors of the study ID of individuals for each feature. A single value (i.e., a length of 1) is recycled if necessary.
- **time**: a list (with a length of R) of vectors of time (or age) at which the feature measurements are recorded.

- `formula`: a list (with a length of R) of formulas for each feature. Each formula is a two-sided linear formula object describing both the fixed-effects and random effects part of the model, with the response (i.e., longitudinal feature) on the left of a `~` operator and the terms, separated by `+` operations, to the right. Random-effects terms are distinguished by vertical bars (`|`) separating expressions for design matrices from grouping factors. See the `formula` argument from the **lme4** package.
- `num.cluster`: number of clusters K . A minimum value of K is 2.
- `hyper.par`: hyper-parameters of the prior distributions for the model parameters. The default hyper-parameters values will result in weakly informative prior distributions.
- `initials`: a list of initial values for model parameters.
- `initial.cluster.membership`: a character, which can be "random", "mixAK" or "input". For "random" (default), the local cluster membership is randomly generated from 1 to K for each feature. For "mixAK", the local cluster membership is obtained by fitting the Bayesian mixture model for longitudinal data to each feature through the **mixAK** package for the given K . For both methods, the global cluster membership is set to be equal to the local cluster membership of the first feature. For "input", the user can input initial values for local and global cluster memberships, using the argument `input.initial.local.cluster.membership` and `input.initial.global.cluster.membership`.
- `input.initial.local.cluster.membership`: if `initial.cluster.membership = "input"`, then this argument must not be empty. It is a list with a length of R and each element of the list corresponds to the local clustering of each feature. See the Illustration section for an example.
- `input.initial.global.cluster.membership`: if `initial.cluster.membership = "input"`, by default the `input.initial.global.cluster.membership` is the local cluster membership of the first feature, and therefore this argument need not be supplied, but the user can also use this argument to supply a different initial cluster membership for the global clustering. See the Illustration section for an example.
- `burn.in`: the number of samples discarded. This value must be smaller than `max.iter`.
- `thin`: the number of thinning. For example, if `thin = 10`, then the MCMC chain will keep one sample every 10 iterations.
- `per`: specify how often the MCMC chain will print the iteration number.
- `max.iter`: the number of MCMC iterations.

In the **BCClong** package, the default hyper-parameter values for the prior distributions are set to reflect no prior information regarding the values of these parameters, that is, uninformative priors are used. The argument `hyper.par` allows setting these hyper-parameter values. The default values are:

```
hyper.par = list(a.star = 1, b.star = 1, delta = 1,
                aa0 = 0.001, bb0 = 0.001, cc0 = 0.001, dd0 = 0.001,
                vv0 = 1000)
```

This indicates that for $\delta_{1,r} = \delta_{2,r} = 1$ for $r = 1, \dots, R$ (corresponding to `a.star = 1, b.star = 1`). The elements of $\boldsymbol{\varphi}$ are set to 1 (corresponding to `delta = 1`), and $a_{0k,r} = b_{0k,r} = 0.001$ for $k = 1, \dots, K$ and $r = 1, \dots, R$ (corresponding to `aa0 = 0.001, bb0 = 0.001`). Also, $V_{0k,r}$ is set to be a diagonal matrix with elements of 1000 (corresponding to `vv0 = 1000`). Finally, $\Sigma_{k,r}$ is set to be a diagonal matrix with elements following an inverse gamma distribution (corresponding to `cc0 = 0.001, dd0 = 0.001`). Customized or informative priors can be used by modifying the values in `hyper.par`. By default, the `initials` is set to `NULL` and the **BCClong** will automatically generate initial values for both the local and global cluster memberships and the model parameters. Convergence of MCMC is diagnosed using visual inspection of trace plots as well as using the Geweke statistics (Geweke, 1991). The **BCClong** package provides a function called `traceplot()` to produce trace plots for model parameters of interest. The `traceplot()` function has the following arguments:

```
traceplot(fit, cluster.indx, feature.indx, parameter)
```

The meaning of these arguments is described as follows.

- `fit`: an objective output from `BCC.multi()` function.
- `cluster.indx`: a numeric value. For cluster-specific parameters, specifying `cluster.indx` will generate the trace plot for the corresponding cluster.
- `feature.indx`: a numeric value. For cluster-specific parameters, specifying `feature.indx` will generate the trace plot for the corresponding cluster.
- `parameter`: a character value. Specify which parameter for which the trace plot will be generated. The value can be "PPI" for $\boldsymbol{\pi}$, $\boldsymbol{\alpha}$ for $\boldsymbol{\alpha}$, "GA" for $\boldsymbol{\gamma}$, "SIGMA.SQ.U" for $\boldsymbol{\Sigma}$ and "SIGMA.SQ.E" for $\boldsymbol{\sigma}$.

In addition, the output of summary statistics from the `BCC.multi()` function also includes the Geweke statistics calculated using the `coda` package (Plummer et al., 2006). Briefly, the Geweke statistics determine whether a Markov chain converges or not based on a test for equality of the means of the first and last parts of a Markov chain (by default the first 10% and the last 50%). If the samples are drawn from the stationary distribution of the chain, the two means are equal and Geweke's statistic has an asymptotically standard normal distribution. Practically, if the estimated Geweke statistics fall between -1.96 and 1.96, we consider the model parameter has converged.

2.3 Missing Data, Measurement Time and Frequency

The proposed BCC model for longitudinal data and the `BCClong` package allow the modeling of missing data under the assumption that the data are missing at random (Lu and Lou, 2022b; Tan et al., 2022). No extra steps are required from the users to handle the missing data. The package will keep individuals with at least one observation for any input features and use them for the analysis, whereas the package will remove individuals with no data on all features. The package also allows the number of observations to be different between individuals and features, and these observations can also be measured at time points that are different between individuals and features.

2.4 Determining the Number of Clusters

Determining the number of clusters in model-based clustering is challenging and there are no widely accepted approaches. A review of different approaches within the Bayesian framework can be found in Celeux et al. (2006); Nasserinejad et al. (2017); Merkle et al. (2019). The `BCClong` package offers three criteria to determine the number of clusters, namely the mean adjusted adherence (Lock and Dunson, 2013), the Deviance Information Criterion (DIC) (Spiegelhalter et al., 2002; Celeux et al., 2006), and the widely Applicable Information Criterion (WAIC) (Watanabe, 2010; Gelman et al., 2014). We briefly describe these three approaches in this subsection. For consensus clustering, Lock and Dunson (2013) proposed an empirical separability criterion, which selects the value of K that gives maximum adherence to an overall clustering. For each $K \geq 2$, the estimated adherence parameters $\alpha_r \in [\frac{1}{K}, 1]$ are mapped to the unit interval by the linear transformation $\alpha_r^* = \frac{K\alpha_r - 1}{K - 1}$, thus $\alpha_r^* \in [0, 1]$. One then selects the value of K that results in the highest mean adjusted adherence, which is defined as $\bar{\alpha}^* = \frac{1}{R} \sum_{m=1}^R \alpha_r^*$. This approach selects a model with which the feature-specific clusterings contribute the most information (on average) to the global clustering. A model with a higher $\bar{\alpha}^*$ is preferred. DIC and its variants have also been widely used for clustering multivariate longitudinal data. See references (Lu and Lou, 2019; Neelon et al., 2011; Leiby et al., 2009; Elliott et al., 2005; Frühwirth-Schnatter and Pyne, 2010) for example. In the `BCClong` package, the DIC is computed based on the complete likelihood (also known as DIC_4 in Celeux et al. (2006)), which is defined as $DIC = -4E_{\Theta, C, L}[\log f(\mathbf{y}, C, L | \Theta) | \mathbf{y}] + 2E_{C, L}[\log f(\mathbf{y}, C, L | E[\Theta | \mathbf{y}, C, L]) | \mathbf{y}]$, where $\mathbf{y} = (y_1, \dots, y_N)$ and Θ is defined in equation (5). The effective number of parameters and the posterior mean of deviance are defined as $p_D = -E_{\Theta, C, L}[2 \log f(\mathbf{y}, C, L | \Theta) | \mathbf{y}] + E_{C, L}[2 \log f(\mathbf{y}, C, L | E[\Theta | \mathbf{y}, C, L]) | \mathbf{y}]$ and $D(\Theta) = -E_{\Theta, C, L}[2 \log f(\mathbf{y}, C, L | \Theta) | \mathbf{y}]$, respectively. A model with a lower DIC is preferred. WAIC is an extension of the Akaike Information Criterion (AIC) that is more fully Bayesian than the DIC. Similar to DIC, WAIC estimates the effective number of parameters to adjust for overfitting. The `BCClong` package uses the `LaplacesDemon` package (Hall, 2021) to calculate the WAIC, which is defined as $WAIC = -2(lppd - p_{WAIC})$, where $lppd$ is the log pointwise predictive density, which is equivalent to $lppd = \log \prod_{i=1}^N p_{\text{post}}(\mathbf{y}, C, L)$. Also, $p_{WAIC} = \sum_{i=1}^N \text{var}_{\text{post}}(\log p(\mathbf{y}, C, L | \Theta))$ is an approximation to the number of unconstrained and uninformed parameters, where a parameter counts as 1 when estimated without constraint or any prior information, 0 if fully constrained or all information comes from the prior distribution, or an intermediate number if both the data and prior are informative. The p_{WAIC} defined here corresponds to p_{WAIC2} in Gelman et al. (2014), which is recommended because its results are closer in practice to the results of leave-one-out cross-validation. A model with a lower WAIC is preferred. In the `BCClong` package, the `model.selection.criteria()` function is used to compute the DIC and WAIC:

```
model.selection.criteria(fit, fast_version)
```

- `fit`: an objective output from `BCC.multi()` function.
- `fast_version`: if `fast_version = TRUE` (default), then compute the DIC and WAIC using the first 100 MCMC samples (after burn-in and thinning). If `fast_version = FALSE`, then compute the DIC and WAIC using all MCMC samples (after burn-in and thinning).

2.5 Goodness of Fit

To assess the model goodness of fit, one can compute the posterior predictive check (Gelman et al., 1996). This approach is performed by comparing the observed data with data replicated from the posterior predictive distribution. If the model fits the data well, the replicated data, denoted as \mathbf{y}^{rep} , should have a similar distribution as the observed data \mathbf{y} . Let $T = T(\mathbf{y}, \Phi)$ denote a discrepancy measure, where $\Phi = (\Theta, \beta, L, C)$ denotes all model parameters (including random effects and cluster labels) and T is sample quantiles or residual-based measures. Following Gelman et al. (1996), a natural discrepancy measure is a χ^2 measure. For the BCC model, it is defined as

$$T^{obs}(\mathbf{y}, \Phi) = \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^{n_{i,r}} \sum_{r=1}^R \frac{z_{ik,r} \| \mathbf{y}_{i,r} - h_{k,r}^{-1}(E(\mathbf{y}_{i,r} | \beta_{ik,r}, \gamma_{k,r})) \|^2}{\sigma_{k,r}^2} \quad (6)$$

where $z_{ik,r} = 1$ if $L_{i,r} = k$, and 0 otherwise. This measure is computed at each MCMC step by treating cluster indicator $z_{ik,r}$ and random effects $\beta_{ik,r}$ as observed data. The Bayesian predictive p-value is the probability that the discrepancy measure based on a predictive sample, $T^{rep}(\mathbf{y}^{rep}, \Phi)$ is more extreme than the observed measure $T^{obs}(\mathbf{y}, \Phi)$. This quantity is estimated by computing the proportion of draws in which $T^{rep} > T^{obs}$. A p-value close to 0.5 indicates the model provides a good fit to the data, whereas a p-value close to 0 or 1 indicates a poor fit. In the **BCClong** package, **BayesT0** is used for computing the posterior predictive check.

BayesT(fit)

- fit: an objective output from `BCC.multi()` function.

2.6 Visualization of Trajectory Patterns by Clusters

To plot the longitudinal trajectory of features by local and global clusterings derived from the BCC model, the `trajplot()` function can be used. This function uses the **ggplot2** package internally.

trajplot(fit, feature.ind, which.cluster)

- fit: an objective output from the `BCC.multi()` function.
- feature.ind: a numeric value indicating which feature to plot. The number indicates the order of the feature specified in the `mydat` argument of the `BCC.multi()` function.
- which.cluster: a character value: "global" or "local", indicating whether to plot the trajectory by global cluster or local cluster indices.

3 Illustrations

In this section, we demonstrate the utility of the **BCClong** package (version 1.0.3) using two real data examples. The first example was quality of life data following epilepsy drug treatment, which included three continuous features. This dataset can be found in the **joineRML** package (Hickey et al., 2018). The second example was the Mayo Clinic primary biliary cholangitis (PBC) data, which included three features of mixed types (i.e., continuous, discrete, and categorical). This dataset can be found in **mixAK** package (Komárek and Komárková, 2014).

3.1 Example 1: Quality of Life Data Following Epilepsy Drug Treatment

The data called `epileptic.qol` came from the Standard and New Antiepileptic Drugs (SAND) study (Marson et al., 2007). This is a randomized control trial of standard and new antiepileptic drugs, comparing effects on long-term clinical outcomes. Quality of life (QoL) data were collected by mail at baseline, 3 months, and at 1 and 2 years using validated measures. The first measurement for each individual was the baseline measurement; however, there was variability in the time taken to return the questionnaires. Similarly, the second, third, and fourth follow-up times, which were scheduled for 3 months, 1-year, and 2 years, respectively, also had variability in completion times. The data can be loaded by

```
R> library("joineRML")
R> head(epileptic.qol[,c(1,5,6,7,8)])

  id time anxiety depress aep
1  1  147      11      14  43
```

2	1	259	12	12	51
3	1	519	20	21	63
4	1	906	17	20	53
5	2	134	19	13	45
6	2	258	21	16	50

This is a longitudinal dataset with one row per visit. At each visit, measurements of three continuous features ($R = 3$) are recorded, namely the anxiety score (anxiety), depression score (depress), and Liverpool Adverse Events Profile (aep) defined according to the NEWQOL (Newly Diagnosed Epilepsy Quality of Life) assessment. One could use a spaghetti plot to visualize the trajectories of these features (Figure 1). The goal of the analysis was to identify distinct longitudinal QoL patterns based on these three features simultaneously. Based on the observed trajectories, we used a linear form with a random intercept for each feature with a Gaussian distribution. The model is written as:

$$\begin{aligned} \text{anxiety: } E(\mathbf{y}_{i,1} | \boldsymbol{\gamma}_{k,1}, \boldsymbol{\beta}_{ik,1}) &= \boldsymbol{\eta}_{ik,1} = \mathbf{x}_{i,1}^\top \boldsymbol{\gamma}_{k,1} + \mathbf{Z}_{i,1}^\top \boldsymbol{\beta}_{ik,1} \\ \text{depress: } E(\mathbf{y}_{i,2} | \boldsymbol{\gamma}_{k,2}, \boldsymbol{\beta}_{ik,2}) &= \boldsymbol{\eta}_{ik,2} = \mathbf{x}_{i,2}^\top \boldsymbol{\gamma}_{k,2} + \mathbf{Z}_{i,2}^\top \boldsymbol{\beta}_{ik,2} \\ \text{aep: } E(\mathbf{y}_{i,3} | \boldsymbol{\gamma}_{k,3}, \boldsymbol{\beta}_{ik,3}) &= \boldsymbol{\eta}_{ik,3} = \mathbf{x}_{i,3}^\top \boldsymbol{\gamma}_{k,3} + \mathbf{Z}_{i,3}^\top \boldsymbol{\beta}_{ik,3} \end{aligned}$$

where $\mathbf{x}_{i,r} = (1, \mathbf{t}_{i,r})^\top$ for $r = 1, 2, 3$ and $\mathbf{t}_{i,r} = (t_{i1,r}, \dots, t_{in_{i,r},r})^\top$ is a vector of time values at which the feature r is recorded for individual i . The corresponding coefficients are $\boldsymbol{\gamma}_{k,r} = (\gamma_{k,r1}, \gamma_{k,r2})^\top$. Also, $\mathbf{Z}_{i,r} = \mathbf{1}_{n_{i,r}}$ for $r = 1, 2, 3$. For model-based clustering based on a finite mixture model, it is necessary

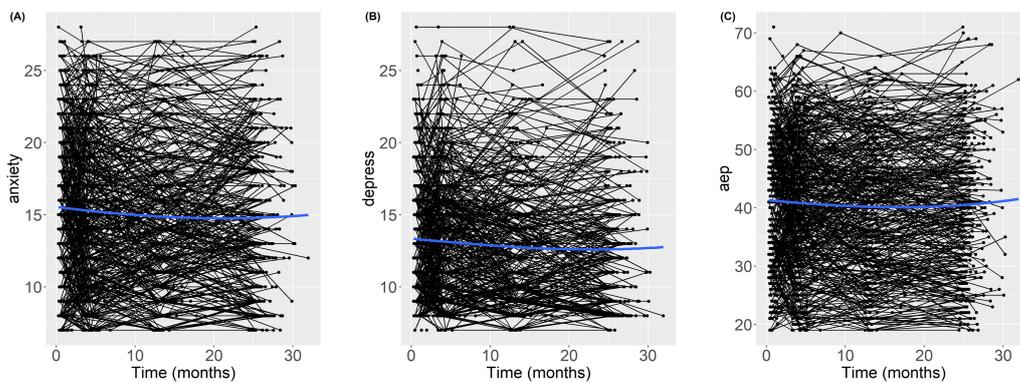


Figure 1: Longitudinal trajectories of three features for the epileptic.qol dataset. A locally weighted scatterplot smoothing curve is overlaid on each panel to provide an estimate of the overall trend. (A) longitudinal trajectories of Anxiety score (anxiety), (B) longitudinal trajectories of depression score (depress), (C) longitudinal trajectories of Liverpool Adverse Events Profile (aep).

to normalize the data within each feature in order to model the growth pattern, not the level. This is because the level could dominate the sample variability so that the clusters will be mainly determined by the level and will not necessarily be homogeneous in trend (Heggeseth and Jewell, 2018). Therefore, in the current analysis, the three features were normalized using the `scale` function before entering the model, using the following commands,

```
R> epileptic.qol$anxiety_scale <- scale(epileptic.qol$anxiety)
R> epileptic.qol$depress_scale <- scale(epileptic.qol$depress)
R> epileptic.qol$aep_scale <- scale(epileptic.qol$aep)
```

Determining the Number of Clusters

The following codes fit the BCC model with the number of clusters ranging from 2 to 5, i.e., $K = 2, \dots, 5$. For each value of K , we saved the mean adjusted adherence (`alpha.adjust`). For each model, we generated 2000 samples and discarded the first 1000 samples. Note that the following part of the codes may be computationally intensive; if users just want to get familiar with the codes and outputs, a smaller number of iterations and burn-in can be used to reduce the computational time, for example, by setting `burn.in = 10`, `per = 1`, and `max.iter = 20`.

```
R> dat <- epileptic.qol
R> alpha.adjust <- NULL
R> for (k in 2:5){
+   fit.BCC <- BCC.multi (
```

```

+   mydat = list(dat$anxiety_scale, dat$depress_scale, dat$aep_scale),
+   dist = c("gaussian"),
+   id = list(dat$id),
+   time = list(dat$time),
+   formula = list(y ~ time + (1|id)),
+   num.cluster = k,
+   burn.in = 1000,
+   thin = 1,
+   per = 100,
+   max.iter = 2000)
+   alpha.adjust <- c(alpha.adjust, fit.BCC$alpha.adjust)
+ }

```

An alternative specification for the formula argument is

```
formula = list(y ~ time + (1|id), y ~ time + (1|id), y ~ time + (1|id))
```

This specification can be used if the users would like to use different functional forms for different features. Each element within the list corresponds to a model for a feature listed in the mydat argument. To compare the model under different numbers of clusters, one can plot the mean adjusted adherence index using the following commands:

```

R> num.cluster <- 2:5
R> plot(num.cluster, alpha.adjust, type = "o",
+       cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5, lwd = 2,
+       xlab = "Number of Clusters",
+       ylab = "mean adjusted adherence", main="mean adjusted adherence")

```

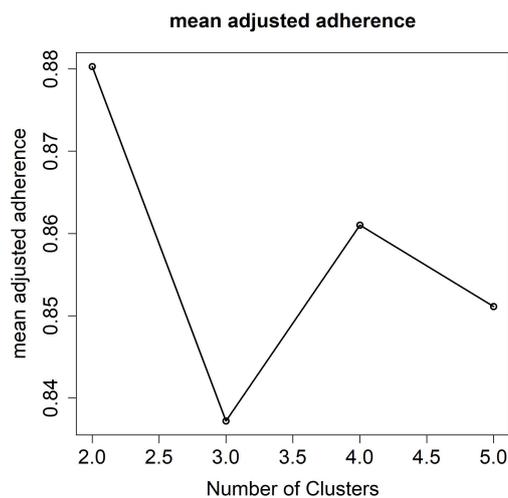


Figure 2: Mean adjusted adherence for $K = 2$ to 5 for the epileptic.qol dataset.

From Figure 2, it was clear that the model with the number of clusters $K = 2$ had the largest mean adjusted adherence (0.88), suggesting that a two-cluster model provided the best fit for the data. To determine the number of clusters based on DIC and WAIC, one can also use the `model.selection.criteria()` function to compute these indices. Next, we used the following commands to fit the final model with $K = 2$.

```

R> fit.BCC2 <- BCC.multi (
+   mydat = list(dat$anxiety_scale, dat$depress_scale, dat$aep_scale),
+   dist = c("gaussian"),
+   id = list(dat$id),
+   time = list(dat$time),
+   formula = list(y ~ time + (1|id)),
+   num.cluster = 2,
+   burn.in = 1000,
+   thin = 1,
+   per = 100,
+   max.iter = 2000)

```

One can apply `print()` and `summary()` functions to the object returned from `BCC.multi()` to obtain summary information of the result, for example,

```
> print(fit.BCC2)
> summary(fit.BCC2)
```

Both functions will return result summaries such as the number of individuals, number of features included in the analysis, and tabulation for global and local clusterings.

Posterior Summary Statistics and Trace Plots

One can print the summary statistics for all the model parameters using the following commands (results not shown):

```
print(fit.BCC2$summary.stat)
```

As an illustration, we print the estimated cluster probabilities $\pi = (\pi_1, \pi_2)$ as follows:

```
R> print(fit.BCC2$summary.stat$PPI)

           [,1]      [,2]
mean      0.43608373 0.56391627
sd        0.02367214 0.02367214
2.5%      0.39326189 0.51584563
97.5%     0.48415437 0.60673811
geweke.stat -0.78671117 0.78671117
```

The summary statistics provided the posterior mean, standard deviation (sd), 2.5%tile, 97.5%tile, and the Geweke statistics. For example, the posterior mean for π_1 and π_2 were 0.44 and 0.56, respectively. This suggested that the estimated cluster proportions for Clusters 1 and 2 are 44% and 56%, respectively. The Geweke statistics for π_1 and π_2 fell between -2 and 2, suggesting that the two parameters converge to a stationary distribution. One can also use the following commands to print the adherence parameters for the three features, i.e., $\alpha = (\alpha_1, \alpha_2, \alpha_3)$:

```
R> print(fit.BCC2$summary.stat$ALPHA)

           [,1]      [,2]      [,3]
mean      0.97572877 0.91727462 0.92427049
sd        0.01230074 0.01764104 0.01771331
2.5%      0.94682192 0.88115409 0.88730582
97.5%     0.99620462 0.95091755 0.95581869
geweke.stat 0.10674171 0.05585855 -1.13303537
```

The output indicated that the posterior means for the adherence parameters were 0.98, 0.92, and 0.92, respectively. This suggested that all three features highly adhered to the global clustering. The Geweke statistics suggested that all the parameters converge to a stationary distribution. One can print similar information for the fixed effect coefficients γ_k using commands `print(fit.BCC2$summary.stat$GA)`, for the variance-covariance matrix of the random effects Σ_k using commands `print(fit.BCC2$summary.stat$SIGMA.SQ.U)`, for the residual variance of continuous features σ_k using commands `print(fit.BCC2$summary.stat$SIGMA.SQ.E)`, for $k = 1, \dots, K$. To visualize the MCMC chain for model parameters, one can use the `traceplot()` function to generate the trace plots. For example, to generate a trace plot for cluster probabilities $\pi = (\pi_1, \pi_2)$ and the adherence parameters $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, the following commands were used:

```
R> traceplot(fit = fit.BCC2, parameter = "PPI", ylab = "pi", xlab = "MCMC samples")
R> traceplot(fit = fit.BCC2, parameter = "ALPHA", ylab = "alpha", xlab = "MCMC samples")
```

The outputs are displayed in Figure 3 and Figure 4, respectively. These trace plots suggested that π_1 , π_2 , α_1 , α_2 , and α_3 are mixing well and converging. In addition, one can generate trace plots for cluster-specific parameters, for example, the fixed effect regression coefficients, $\gamma_{k,r} = (\gamma_{k,r1}, \gamma_{k,r2})$, with the first element as the intercept and the second element as the slope. The following commands generate trace plots (results not shown) for Cluster 1 of the first feature (anxiety), second (depress), and third (aep) features, that is, $\gamma_{1,1} = (\gamma_{1,11}, \gamma_{1,12})$, $\gamma_{1,2} = (\gamma_{1,21}, \gamma_{1,22})$, and $\gamma_{1,3} = (\gamma_{1,31}, \gamma_{1,32})$.

```
R> traceplot(fit = fit.BCC2, cluster.indx = 1, feature.indx = 1,
+           parameter = "GA", ylab = "GA", xlab = "MCMC samples")
R> traceplot(fit = fit.BCC2, cluster.indx = 1, feature.indx = 2,
+           parameter = "GA", ylab = "GA", xlab = "MCMC samples")
R> traceplot(fit = fit.BCC2, cluster.indx = 1, feature.indx = 3,
+           parameter = "GA", ylab = "GA", xlab = "MCMC samples")
```

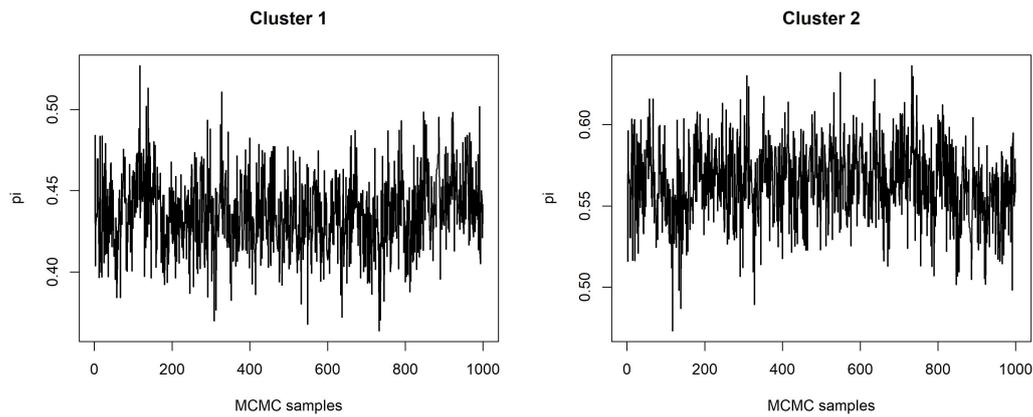


Figure 3: Trace plots for cluster probabilities π_1 and π_2 .

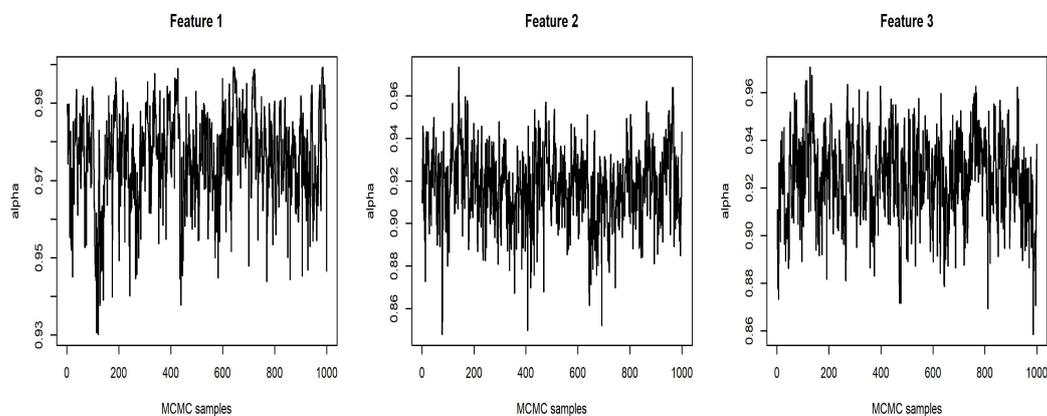


Figure 4: Trace plots for adherence parameters $\alpha_1, \alpha_2, \alpha_3$.

Also, the following commands generated trace plots (results not shown) for Cluster 2 of the first feature (anxiety), second (depress), and third (aep) features, that is, $\gamma_{2,1} = (\gamma_{2,11}, \gamma_{2,12})$, $\gamma_{2,2} = (\gamma_{2,21}, \gamma_{2,22})$, and $\gamma_{2,3} = (\gamma_{2,31}, \gamma_{2,32})$.

```
R> traceplot(fit = fit.BCC2, cluster.indx = 2, feature.indx = 1,
+           parameter = "GA", ylab = "GA", xlab="MCMC samples")
R> traceplot(fit = fit.BCC2, cluster.indx = 2, feature.indx = 2,
+           parameter = "GA", ylab = "GA", xlab = "MCMC samples")
R> traceplot(fit = fit.BCC2, cluster.indx = 2, feature.indx = 3,
+           parameter = "GA", ylab = "GA", xlab = "MCMC samples")
```

Similarly, one can use `parameter = "SIGMA.SQ.U"` to generate the elements of the variance-covariance matrix (Σ_k) for the random effects, and `parameter = "SIGMA.SQ.E"` for the residual variance of continuous features σ_k , for $k = 1, \dots, K$.

Longitudinal Profiles by Local and Global Clusters

The package uses the mode over all the MCMC samples (after burn-in and thinning) as the point estimates for both feature-specific and global clusterings. The following commands computed the number of individuals belonging to each cluster for both feature-specific and global clusterings.

```
R> table(fit.BCC2$cluster.local[[1]])
 1  2
231 309
R> table(fit.BCC2$cluster.local[[2]])
 1  2
223 317
```

```
R> table(fit.BCC2$cluster.local[[3]])
 1  2
267 273
R> table(fit.BCC2$cluster.global)
 1  2
236 304
```

That is, for clustering based on the first feature (anxiety) L_1 , 231 individuals were assigned to Cluster 1 and 309 were assigned to Cluster 2. For clustering based on the second feature (depress) L_2 , 223 individuals were assigned to Cluster 1 and 317 were assigned to Cluster 2. For clustering based on the third feature (aep) L_3 , 267 individuals were assigned to Cluster 1 and 273 were assigned to Cluster 2. For global clustering, C , 236 individuals were assigned to Cluster 1 and 304 were assigned to Cluster 2. To plot the longitudinal trajectory of features by local and global clusterings, the `trajplot()` function can be used. Example codes for plotting the trajectory of the features in the `epileptic.qol` data are provided as follows:

```
R> gp1 <- trajplot(fit = fit.BCC2, feature.ind = 1,
+   which.cluster = "local.cluster",
+   title = bquote(paste("Local Clustering (",
+   hat(alpha)[1] == .(round(fit.BCC2$alpha[1], 2)), ")")),
+   xlab = "time (months)", ylab = "anxiety", color = c("#00BA38", "#619CFF"))
R> gp2 <- trajplot(fit = fit.BCC2, feature.ind = 2,
+   which.cluster = "local.cluster",
+   title = bquote(paste("Local Clustering (",
+   hat(alpha)[2] == .(round(fit.BCC2$alpha[2], 2)), ")")),
+   xlab = "time (months)", ylab = "depress", color = c("#00BA38", "#619CFF"))
R> gp3 <- trajplot(fit = fit.BCC2, feature.ind = 3,
+   which.cluster = "local.cluster",
+   title = bquote(paste("Local Clustering (",
+   hat(alpha)[3] == .(round(fit.BCC2$alpha[3], 2)), ")")),
+   xlab = "time (months)", ylab = "aep", color = c("#00BA38", "#619CFF"))
R> gp4 <- trajplot(fit = fit.BCC2, feature.ind = 1,
+   which.cluster = "global.cluster",
+   title = "Global Clustering", xlab = "time (months)", ylab = "anxiety",
+   color = c("#00BA38", "#619CFF"))
R> gp5 <- trajplot(fit = fit.BCC2, feature.ind = 2,
+   which.cluster = "global.cluster",
+   title = "Global Clustering", xlab = "time (months)", ylab = "depress",
+   color = c("#00BA38", "#619CFF"))
R> gp6 <- trajplot(fit = fit.BCC2, feature.ind = 3,
+   which.cluster = "global.cluster",
+   title = "Global Clustering", xlab = "time (months)", ylab = "aep",
+   color = c("#00BA38", "#619CFF"))
```

The `cowplot` package (Wilke, 2024) is used here to combine all six plots together into one figure.

```
R> library("cowplot")
R> dev.new(width = 180, height = 120)
R> plot_grid(gp1, gp2, gp3, gp4, gp5, gp6,
+   labels = c("(A)", "(B)", "(C)", "(D)", "(E)", "(F)"),
+   ncol = 3, align = "v" )
```

The results are displayed in Figure 5. The top panel (A, B, and C) shows the three features plotted by local clustering indices, whereas the bottom panel (D, E, F) shows the three features plotted by global clustering indices. Individuals in Cluster 1 had lower anxiety scores, depression scores, and aep scores, which represented a better health condition compared to those in Cluster 2. The local clustering based on the anxiety score highly adhered to the global clustering ($\hat{\alpha}_1 = 0.98$), followed by aep ($\hat{\alpha}_2 = 0.92$) and depress scores ($\hat{\alpha}_3 = 0.92$), respectively. This suggested that all three features highly adhered to the global clustering, and the anxiety score contributed the most information to determine the global clustering.

Goodness of Fit

To assess the model goodness of fit, one can use `BayesT0` for computing the posterior predictive check. The commands of the two-cluster model for `epileptic.qol` data are as follows:

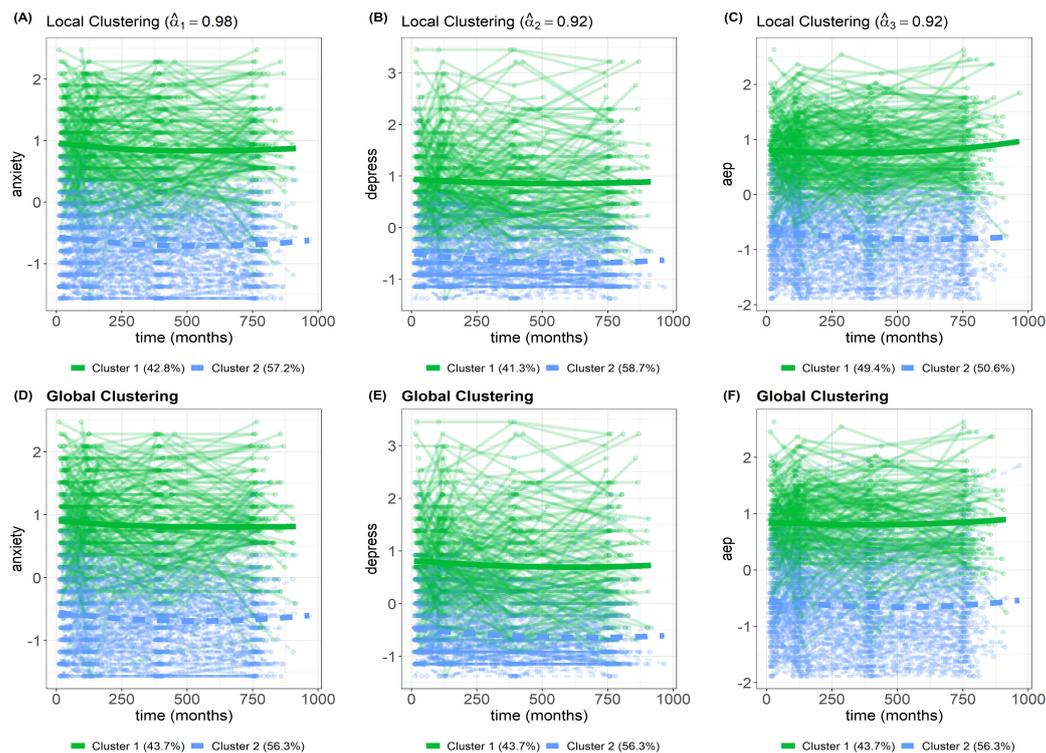


Figure 5: Longitudinal trajectories for features by local and global clusterings `epileptic.qol` data. Locally weighted scatterplot smoothing curves are overlaid on each panel to provide an estimate of the overall trend. (A) Longitudinal trajectories of anxiety plotted by local clustering L_1 . (B) Longitudinal trajectories of depress plotted by local clustering L_2 . (C) Longitudinal trajectories of aep plotted by local clustering L_3 . (D) Longitudinal trajectories of anxiety plotted by global clustering C . (E) Longitudinal trajectories of depress plotted by global clustering C . (F) Longitudinal trajectories of aep plotted by global clustering C .

```
R> res <- BayesT(fit = fit.BCC2)
R> plot(log(res$T.obs), log(res$T.rep), xlim = c(8.45,8.7), ylim = c(8.45,8.7), cex = 1.5,
+       xlab = "Observed T statistics (in log scale)",
+       ylab = "Predicted T statistics (in log scale)")
R> abline(0, 1, lwd = 2,col = 2)
```

The result is displayed in Figure 6. This figure suggested that there was no systematic pattern (e.g., consistent overestimation or underestimation) between the observed T (in log scale) and predicted T (in log scale) statistics, indicating that the model provided a good fit to the data. To further evaluate the goodness of fit using an objective measure, the Bayesian p-value can be calculated, using the following commands,

```
R> p.value <- sum(res$T.rep > res$T.obs)/length(res$T.rep)
R> p.value

[1] 0.559
```

The Bayesian p-value of 0.559 corresponds to the proportion of samples above the diagonal, further suggesting that there is no clear evidence of model misspecification or that the model provides a poor fit to the data. In addition, the output also provided the posterior cluster probability, which is a measure of cluster uncertainty. The following commands generated a boxplot for the posterior cluster probabilities by clusters.

```
R> fit.BCC2$cluster.global <- factor(fit.BCC2$cluster.global,
+   labels = c("Cluster 1", "Cluster 2"))
R> boxplot(fit.BCC2$postprob ~ fit.BCC2$cluster.global, ylim = c(0.5, 1),
+   xlab = "", ylab = "Posterior Cluster Probability")
```

The results were displayed in Figure 7. The posterior cluster probabilities for a majority of individuals in both Clusters 1 and 2 were close to 1, except for a few individuals. This suggested that the cluster membership for individuals is robust.

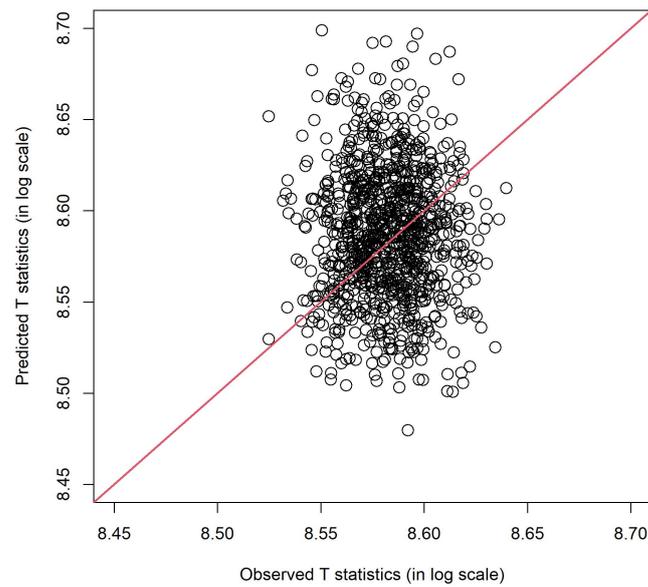


Figure 6: Posterior predictive check: observed T (in log scale) and predicted T (in log scale) statistics.

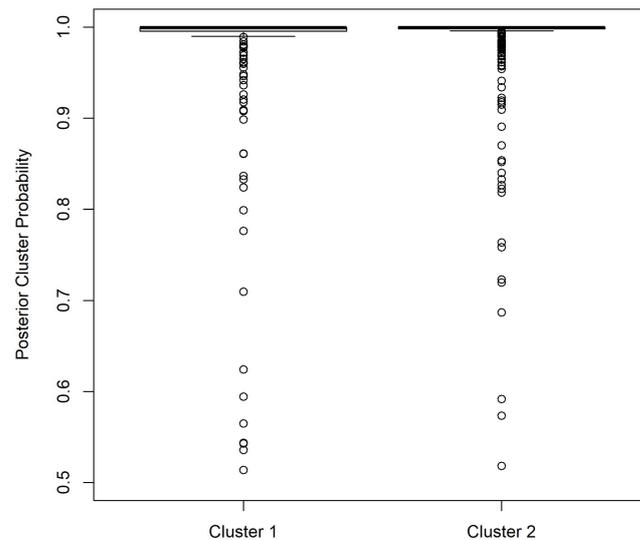


Figure 7: Boxplot for the posterior cluster probabilities by clusters.

Using a Different Set of Initial Values for Local and Global Cluster Memberships

It is a good practice to fit the model with different starting values to ensure the model converges and the clustering results are stable. By default, the initial values of the local cluster membership are randomly generated from 1 to K for each feature, and the global cluster membership is set to be equal to the local cluster membership of the first feature. One can also supply the local cluster membership by fitting the Bayesian mixture model for longitudinal data to each feature through the `mixAK` package for the given K . To do so, the argument `initial.cluster.membership = "mixAK"` can be added to the `BCC.multi` function. One can also provide user input initial values for the cluster membership, by using `initial.cluster.membership = "input"`, and then supply the initial values using the `input.initial.local.cluster.membership` and `input.initial.global.cluster.membership` argu-

ments. For example, if the user would like to refit the model using clustering results generated from the previous model (`fit.BCC2`), the following code can be used:

```
> fit.BCC2a <- BCC.multi (
+   mydat = list(dat$anxiety_scale, dat$depress_scale, dat$aep_scale),
+   dist = c("gaussian"),
+   id = list(dat$id),
+   time = list(dat$time),
+   formula = list(y ~ time + (1|id)),
+   initial.cluster.membership = "input",
+   input.initial.local.cluster.membership = list(fit.BCC2$cluster.local[[1]],
+   fit.BCC2$cluster.local[[2]], fit.BCC2$cluster.local[[3]]),
+   input.initial.global.cluster.membership = fit.BCC2$cluster.global,
+   num.cluster = 2,
+   burn.in = 1000,
+   thin = 1,
+   per = 100,
+   max.iter = 2000)
```

Comparing Clustering Results to Other Model Specifications

To evaluate the robustness of the resulting clusters, one can compare the clustering results to more complicated models. We fit a BCC model with a linear form using a random intercept and slope, and a BCC model with a quadratic form using a random intercept and slope. The commands are as follows:

```
R> fit.BCC2b <- BCC.multi (
+   mydat = list(dat$anxiety_scale, dat$depress_scale, dat$aep_scale),
+   dist = c("gaussian"),
+   id = list(dat$id),
+   time = list(dat$time),
+   formula = list(y ~ time + (1 + time|id)),
+   num.cluster = 2,
+   print.info = "FALSE",
+   burn.in = 1000,
+   thin = 1,
+   per = 100,
+   max.iter = 2000)
R> fit.BCC2c <- BCC.multi (
+   mydat = list(dat$anxiety_scale, dat$depress_scale, dat$aep_scale),
+   dist = c("gaussian"),
+   id = list(dat$id),
+   time = list(dat$time),
+   formula = list(y ~ time + time2 + (1 + time|id)),
+   num.cluster = 2,
+   print.info = "FALSE",
+   burn.in = 1000,
+   thin = 1,
+   per = 100,
+   max.iter = 2000)
> fit.BCC2b$cluster.global <- factor(fit.BCC2b$cluster.global,
+   labels = c("Cluster 1", "Cluster 2"))
> table(fit.BCC2$cluster.global, fit.BCC2b$cluster.global)
```

	Cluster 1	Cluster 2
Cluster 1	223	13
Cluster 2	1	303

The agreement between the first model (linear model with a random intercept) and the second model (linear model with a random intercept and random slope) was $(223 + 303)/540 = 97\%$.

```
R> fit.BCC2c$cluster.global <- factor(fit.BCC2c$cluster.global,
+   labels = c("Cluster 1", "Cluster 2"))
R> table(fit.BCC2$cluster.global, fit.BCC2c$cluster.global)
```

	Cluster 1	Cluster 2
Cluster 1	227	9

Cluster 2 2 302

The agreement between the first model (linear model with a random intercept) and the second model (quadratic model with a random intercept and random slope) was $(226 + 301)/540 = 98\%$. This suggested that fitting a more complicated model does not substantially change the individual cluster membership, and further attests that the estimated cluster membership was robust to different model specifications.

3.2 Example 2: Mayo Clinic Primary Biliary Cholangitis Data

The second dataset was from a randomized placebo-controlled trial of the drug D-penicillamine for Primary Biliary Cholangitis conducted between 1974 and 1984 at the Mayo Clinic Primary Biliary Cholangitis (PBC). The dataset used here contains multiple laboratory results collected longitudinally on 312 randomized PBC patients. Example features (variables) include serum bilirubin (mg/dl), serum albumin (mg/dl), platelet count, serum glutamic-oxaloacetic transaminase and the presence of blood vessel malformations in the skin. The dataset was analyzed previously by [Komárek and Komárková \(2013\)](#) using the `mixAK` package. The analysis goal was to identify subgroups of patients with similar profiles based on multiple features which may serve as critical information regarding patients' prognostics. To illustrate the clustering process of mixed-type longitudinal features, we followed [Komárek and Komárková \(2013\)](#) analysis and chose three features of interest, namely the `lbili` (log of serum bilirubin), platelet count, and `spiders` (presence of blood vessel malformations in the skin). Also, following [Komárek and Komárková \(2013\)](#), we included $N = 260$ individuals known to be alive at 910 days of follow-up, and only the longitudinal measurements up to this point were considered (known as the `PBC910` data). The median (min, max) number of observations for these markers was 4 (1, 5). Given a small number of observations available for each individual, we considered a model with only a random intercept, i.e., using the following specification:

```
formula = list(y ~ time + (1|id))
```

For a complete analysis, one can follow the steps outlined in Example 1 to determine the number of clusters and to evaluate the model performance using several different approaches (e.g., posterior predictive check). Here, as an illustration, and for the purpose of comparison, we only fitted a model with two clusters (i.e., $K = 2$) as in [Komárek and Komárková \(2013\)](#). The data is available through the `mixAK` package. The following commands prepared the dataset for analysis.

```
R> library("mixAK")
R> data(PBC910)
```

Using the argument `dist = c("gaussian", "poisson", "binomial")` to specify the distributions for the three features, the following commands performed a BCC model with $K = 2$.

```
R > fit.BCC2 <- BCC.multi(
+   mydat = list(PBC910$lbili, PBC910$platelet, PBC910$spiders),
+   dist = c("gaussian", "poisson", "binomial"),
+   id = list(PBC910$id),
+   time = list(PBC910$month),
+   formula = list(y ~ time + (1|id)),
+   num.cluster = 2,
+   burn.in = 10000,
+   thin = 10,
+   per = 1000,
+   max.iter = 20000)
```

The following commands generated the longitudinal profile plots for the three features (`lbili`, platelet, `spiders`).

```
R> gp1 <- trajplot(fit = fit.BCC2, feature.ind = 1,
+   which.cluster = "local.cluster",
+   title = bquote(paste("Local Clustering (",
+   hat(alpha)[1] == .(round(fit.BCC2$alpha[1], 2)), ")")),
+   xlab = "months", ylab = "lbili", color = c("#00BA38", "#619CFF"))
R> gp2 <- trajplot(fit = fit.BCC2, feature.ind = 2,
+   which.cluster = "local.cluster",
+   title = bquote(paste("Local Clustering (",
+   hat(alpha)[2] == .(round(fit.BCC2$alpha[2], 2)), ")")),
+   xlab = "months", ylab = "platelet", color = c("#00BA38", "#619CFF"))
R> gp3 <- trajplot(fit = fit.BCC2, feature.ind = 3,
```

```

+       which.cluster = "local.cluster",
+       title = bquote(paste("Local Clustering (",
+       hat(alpha)[3] == .(round(fit.BCC2$alpha[3],2)),")"),
+       xlab = "months", ylab = "spiders", color = c("#00BA38", "#619CFF"))
R> gp4 <- trajplot(fit = fit.BCC2, feature.ind = 1,
+       which.cluster = "global.cluster",
+       title = "Global Clustering",
+       xlab = "months", ylab = "lbili", color = c("#00BA38", "#619CFF"))
R> gp5 <- trajplot(fit = fit.BCC2, feature.ind = 2,
+       which.cluster = "global.cluster",
+       title = "Global Clustering",
+       xlab = "months", ylab = "platelet", color = c("#00BA38", "#619CFF"))
R> gp6 <- trajplot(fit = fit.BCC2, feature.ind = 3,
+       which.cluster = "global.cluster",
+       title = "Global Clustering",
+       xlab = "months", ylab = "spiders", color = c("#00BA38", "#619CFF"))

```

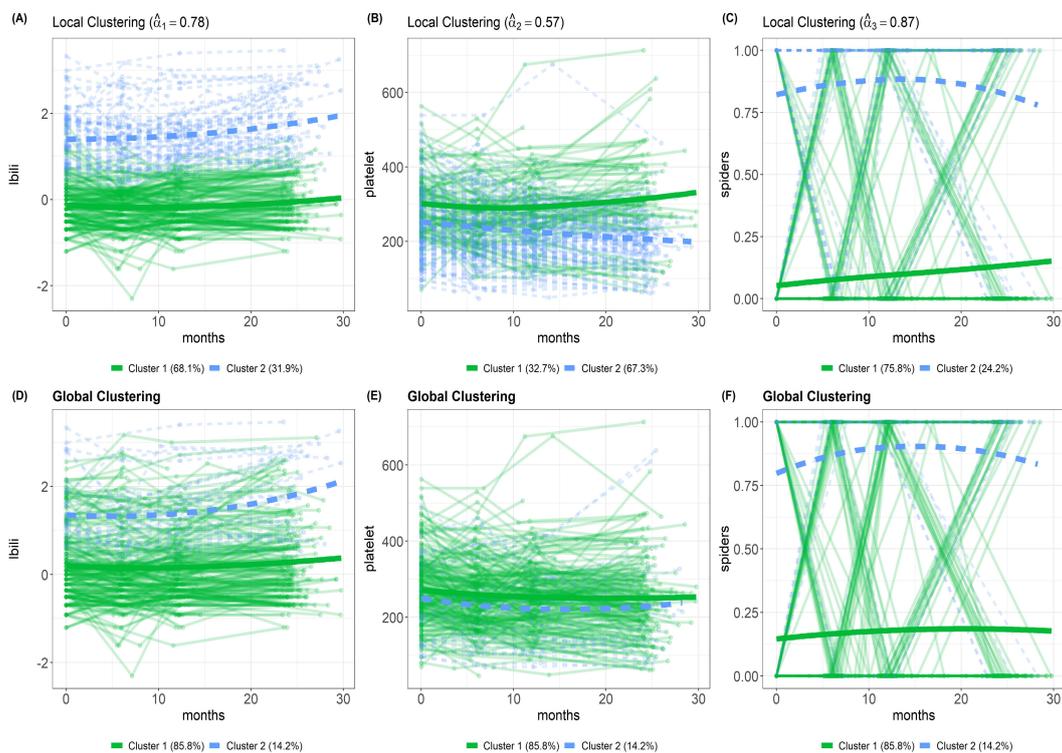


Figure 8: Longitudinal trajectories for features by local and global clusterings for **PBC910** data. Locally weighted scatterplot smoothing curves are overlaid on each panel to provide an estimate of the overall trend. (A) Longitudinal trajectories of *lbili* plotted by local clustering L_1 . (B) Longitudinal trajectories of *platelet* plotted by local clustering L_2 . (C) Longitudinal trajectories of *spiders* plotted by local clustering L_3 . (D) Longitudinal trajectories of *lbili* plotted by global clustering C . (E) Longitudinal trajectories of *platelet* plotted by global clustering C . (F) Longitudinal trajectories of *spiders* plotted by global clustering C .

To compare the results with an existing model, we fit a Bayesian mixture model using the **mixAK** package using the following commands (see (Komárek and Komárková, 2014) for details of using this package):

```

R> mod <- GLMM_MCMC(y = PBC910[, c("lbili", "platelet", "spiders")],
+       dist = c("gaussian", "poisson(log)", "binomial(logit)",
+       id = PBC910[, "id"],
+       x = list(lbili = "empty", platelet = "empty", spiders = PBC910[, "month"]),
+       z = list(lbili = PBC910[, "month"], platelet = PBC910[, "month"], spiders = "empty"),
+       random.intercept = rep(TRUE, 3), prior.b = list(Kmax = 2),
+       nMCMC = c(burn = 100, keep = 1000, thin = 10, info = 100),
+       parallel = FALSE)
R> mod <- NMixRelabel(mod, type = "stephens", keep.comp.prob = TRUE)

```

```
R> cluster.mixAK <- apply(mod[[1]]$poster.comp.prob, 1, which.max)
R> table(mixAK = cluster.mixAK, BCClong = fit.BCC2$cluster.global)
```

```
      BCClong
mixAK  1   2
      1 158  3
      2  65 34
```

The agreement between the two models for the **PBC910** data was $(158 + 34)/260 = 74\%$.

4 Summary and discussion

In this paper, we described a BCC model for clustering longitudinal data with multiple features. Using two real-life and open-access data, we provided step-by-step guidance on using the **BCClong** package and interpreting the results. The results also demonstrated that the **BCClong** package is a useful tool for clustering longitudinal data and it enhanced our understanding of the heterogeneity underlying each feature and across features. Of note, the reason we require the user to create three separate lists of data vectors (instead of a data frame) when using `BCC.multi()`, in specifying the arguments `mydat`, `id`, and `time` is to allow more flexibility in modeling longitudinal features with distinct structures. This is particularly useful when these features are collected from multiple data sources (with different study designs), and therefore the time scale and data collection frequency could differ between features. Additional details regarding the functions and their arguments can be found in the package vignettes. The BCC model implemented using the **BCClong** package yields both feature-specific (local) clusterings and consensus (global) clustering. In practice, global clustering is often of greater interest, as it encompasses information from all features. It can be used to associate with exposures and predict long-term outcomes. Conceptually, feature-specific clustering can be viewed as clustering based on a single feature alone, whereas global clustering can be viewed as a weighted average clustering across all features, with weights being the adherence parameters. In order to capture the complexity of the underlying population heterogeneity, it is of great importance to consider multiple features of interest simultaneously, as we often do for cross-sectional data. Several directions can be considered in future studies. For example, incorporating variable selection (e.g., spike-and-slab and shrinkage) priors (Lu and Lou, 2022a, 2021) for fixed and random effects will enhance the flexibility of the package and allow it to be applied to high-dimensional settings. In addition, determining the number of clusters can be achieved by using a Dirichlet process mixture model (Escobar, 1994; Lu and Chandra, 2024; Lu et al., 2024). The package does not support predicting the cluster membership of a new individual or future trajectories for a given individual. These functions will be considered in future versions of the package. Finally, the current version of the **BCClong** package relies on a standard MCMC algorithm to estimate the model, which is generally slow and requires particular care for the user to check convergence. While the computational time is reasonable for our current applications, to scale up the model to a large dataset, other computational algorithms such as distributed stochastic gradient MCMC (Ahn et al., 2014) and variational inference (Blei et al., 2017) can be considered in future studies.

Computational details

Most of the functions in the **BCClong** package are written in C++. The output shown in this article was obtained using R version 4.2.1 (June, 2022), **BCClong** 1.0.3, and the following contributed packages which are the dependencies or imports of the **BCClong** package: `cluster` (2.1.4) (Maechler et al., 2022), `coda` (0.19-4) (Plummer et al., 2006), `ggplot2` (3.4.0) (Wickham, 2016), `label.switching` (1.8) (Papastamoulis, 2016), `LaplacesDemon` (16.1.6) (Hall, 2021), `lme4` (1.1-31) (Bates et al., 2015), `MASS` (7.3-58.1) (Ripley et al., 2013), `mclust` (6.0.0) (Scrucca et al., 2023), `MCMCpack` (1.6-3) (Martin et al., 2011), `mixAK` (5.5) (Komárek and Komárková, 2014), `mvtnorm` (1.1-3) (Genz et al., 2021), `nnet` (7.3-18) (Venables and Ripley, 2002), `Rcpp` (1.0.9) (Eddelbuettel and François, 2011), `Rmpfr` (0.8-9) (Maechler, 2022), `truncdist` (1.0-2) (Novomestky et al., 2016). R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/>.

Acknowledgments

ZT is supported by a Dean's Doctoral Award from Queen's University. ZL is supported by a Discovery Grant funded by the Natural Sciences and Engineering Research Council of Canada.

References

- S. Ahn, B. Shahbaba, and M. Welling. Distributed stochastic gradient MCMC. In *International conference on machine learning*, pages 1044–1052. PMLR, 2014. URL <https://proceedings.mlr.press/v32/ahn14.html>. [p57]
- D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. doi: 10.18637/jss.v067.i01. R package version 1.1-31. [p57]
- D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. URL <https://doi.org/10.1080/01621459.2017.1285773>. [p57]
- G. Celeux, F. Forbes, C. P. Robert, D. M. Titterton, et al. Deviance information criteria for missing data models. *Bayesian Analysis*, 1(4):651–673, 2006. doi: 10.1214/06-BA122. [p45]
- D. Eddelbuettel and R. François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011. doi: 10.18637/jss.v040.i08. R package version 1.0.9. [p57]
- M. R. Elliott, J. J. Gallo, T. R. Ten Have, H. R. Bogner, and I. R. Katz. Using a bayesian latent growth curve model to identify trajectories of positive affect and negative events following myocardial infarction. *Biostatistics*, 6(1):119–143, 2005. doi: 10.1093/biostatistics/kxh022. [p45]
- M. D. Escobar. Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277, 1994. URL <https://doi.org/10.2307/2291223>. [p57]
- S. Frühwirth-Schnatter and S. Pyne. Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions. *Biostatistics*, 11(2):317–336, 2010. URL <https://doi.org/10.1093/biostatistics/kxp062>. [p45]
- A. Gelman, X.-L. Meng, and H. Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, pages 733–760, 1996. URL <https://www.jstor.org/stable/24306036>. [p46]
- A. Gelman, J. Hwang, and A. Vehtari. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016, 2014. URL <https://doi.org/10.1007/s11222-013-9416-2>. [p45]
- C. Genolini and B. Falissard. Kml: k-means for longitudinal data. *Computational Statistics*, 25(2): 317–328, 2009. doi: 10.1007/s00180-009-0178-4. [p41]
- C. Genolini, J.-B. Pingault, T. Driss, S. Côté, R. E. Tremblay, F. Vitaro, C. Arnaud, and B. Falissard. Kml3d: a non-parametric algorithm for clustering joint trajectories. *Computer Methods and Programs in Biomedicine*, 109(1):104–111, 2013. URL <https://doi.org/10.1016/j.cmpb.2012.08.016>. [p41]
- A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2021. URL <https://CRAN.R-project.org/package=mvtnorm>. R package version 1.1-3. [p57]
- J. Geweke. *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*, volume 196. Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN, USA, 1991. [p44]
- B. Hall. *LaplacesDemon: Complete Environment for Bayesian Inference*, 2021. URL <https://CRAN.R-project.org/package=LaplacesDemon>. R package version 16.1.6. [p45, 57]
- B. C. Heggseth and N. P. Jewell. How Gaussian mixture models might miss detecting factors that impact growth patterns. *The Annals of Applied Statistics*, 12(1):222–245, 2018. URL <https://doi.org/10.1214/17-aos1066>. [p47]
- G. L. Hickey, P. Philipson, A. Jorgensen, and R. Kolamunnage-Dona. joineRML: a joint model and software package for time-to-event and multivariate longitudinal outcomes. *BMC Medical Research Methodology*, 18(1):1–14, 2018. URL <https://doi.org/10.1186/s12874-018-0502-1>. [p46]
- A. Komárek and L. Komárková. Clustering for multivariate continuous and discrete longitudinal data. *The Annals of Applied Statistics*, 7(1):177–200, 2013. doi: 10.1214/12-AOAS580. [p55]
- A. Komárek and L. Komárková. Capabilities of R package mixak for clustering based on multivariate continuous and discrete longitudinal data. *Journal of Statistical Software*, 59(12):1–38, 2014. URL <https://doi.org/10.18637/jss.v059.i12>. R package version 5.5. [p46, 56, 57]

- B. E. Leiby, M. D. Sammel, T. R. Ten Have, and K. G. Lynch. Identification of multivariate responders and non-responders by using Bayesian growth curve latent class models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 58(4):505–524, 2009. URL <https://doi.org/10.1111/j.1467-9876.2009.00663.x>. [p45]
- F. Leisch. FlexMix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, 11(8):1–18, 2004. doi: 10.18637/jss.v011.i08. [p41]
- E. F. Lock and D. B. Dunson. Bayesian consensus clustering. *Bioinformatics*, 29(20):2610–2616, 2013. doi: 10.1093/bioinformatics/btt425. [p42, 45]
- Z. Lu and N. K. Chandra. A sparse factor model for clustering high-dimensional longitudinal data. *Statistics in Medicine*, 43(19):3633–3648, 2024. URL doi:10.1002/sim.10151. [p57]
- Z. Lu and W. Lou. Shape invariant mixture model for clustering non-linear longitudinal growth trajectories. *Statistical Methods in Medical Research*, 28(12):3769–3784, 2019. URL <https://doi.org/10.1177/0962280218815301>. [p45]
- Z. Lu and W. Lou. Bayesian approaches to variable selection in mixture models with application to disease clustering. *Journal of Applied Statistics*, pages 1–21, 2021. URL <https://doi.org/10.1080/02664763.2021.1994529>. [p57]
- Z. Lu and W. Lou. Bayesian approaches to variable selection: a comparative study from practical perspectives. *International Journal of Biostatistics*, 18(1):89–108, May 2022a. URL <https://doi.org/10.1515/ijb-2020-0130>. [p57]
- Z. Lu and W. Lou. Bayesian consensus clustering for multivariate longitudinal data. *Statistics in Medicine*, 41(1):108–127, 2022b. doi: 10.1002/sim.9225. [p41, 42, 45]
- Z. Lu, M. Ahmadiankalati, and Z. Tan. Joint clustering multiple longitudinal features: A comparison of methods and software packages with practical guidance. *Statistics in Medicine*, 42(29):5513–5540, 2023. URL <https://doi.org/10.1002/sim.9917>. [p41]
- Z. Lu, P. Subbarao, and W. Lou. A Bayesian latent class model for integrating multi-source longitudinal data: application to the CHILd cohort study. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 73(2):398–419, 2024. URL <https://doi.org/10.1093/jrssc/qlad100>. [p57]
- M. Maechler. *Rmpfr: R MPFR - Multiple Precision Floating-Point Reliable*, 2022. URL <https://CRAN.R-project.org/package=Rmpfr>. R package version 0.8-9. [p57]
- M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2022. URL <https://CRAN.R-project.org/package=cluster>. R package version 2.1.4. [p57]
- A. Magrini. *gbmt: Group-Based Multivariate Trajectory Modeling*, 2022a. URL <https://CRAN.R-project.org/package=gbmt>. R package version 0.1.3. [p41]
- A. Magrini. Assessment of agricultural sustainability in european union countries: A group-based multivariate trajectory approach. *Asta Advances in Statistical Analysis*, 106(4):673–703, 2022b. URL <https://doi.org/10.1007/s10182-022-00437-9>. [p41]
- A. Marson, R. Appleton, G. Baker, D. Chadwick, J. Doughty, B. Eaton, C. Gamble, A. Jacoby, P. Shackley, D. Smith, et al. A randomised controlled trial examining the longer-term outcomes of standard versus new antiepileptic drugs. the sanad trial. *Health Technology Assessment-southampton-*, 11(37), 2007. URL <https://doi.org/10.3310/hta11370>. [p46]
- A. D. Martin, K. M. Quinn, and J. H. Park. MCMCpack: Markov chain Monte Carlo in R. *Journal of Statistical Software*, 42(9):22, 2011. doi: 10.18637/jss.v042.i09. R package version 1.6-3. [p57]
- E. C. Merkle, D. Furr, and S. Rabe-Hesketh. Bayesian comparison of latent variable models: Conditional versus marginal likelihoods. *Psychometrika*, 84(3):802–829, 2019. URL <https://doi.org/10.1007/s11336-019-09679-0>. [p45]
- D. S. Nagin, B. L. Jones, V. L. Passos, and R. E. Tremblay. Group-based multi-trajectory modeling. *Statistical Methods in Medical Research*, 27(7):2015–2023, 2018. URL <https://doi.org/10.1177/0962280216673085>. [p41]
- K. Nasserinejad, J. van Rosmalen, W. de Kort, and E. Lesaffre. Comparison of criteria for choosing the number of classes in Bayesian finite mixture models. *Plos One*, 12(1):e0168838, 2017. URL <https://doi.org/10.1371/journal.pone.0168838>. [p45]

- B. Neelon, G. K. Swamy, L. F. Burgette, and M. L. Miranda. A Bayesian growth mixture model to examine maternal hypertension and birth outcomes. *Statistics in Medicine*, 30(22):2721–2735, 2011. URL <https://doi.org/10.1002/sim.4291>. [p45]
- F. Novomestky, S. Nadarajah, and M. F. Novomestky. *truncdist: Truncated Random Variables*, 2016. URL <https://CRAN.R-project.org/package=truncdist>. R package version 1.0-2. [p57]
- P. Papastamoulis. label.switching: An R package for dealing with the label switching problem in MCMC outputs. *Journal of Statistical Software, Code Snippets*, 69(1):1–24, 2016. doi: 10.18637/jss.v069.c01. R package version 1.8. [p43, 57]
- M. Plummer, N. Best, K. Cowles, and K. Vines. Coda: Convergence diagnosis and output analysis for mcmc. *R News*, 6(1):7–11, 2006. URL <https://journal.r-project.org/archive/>. R package version 0.19-4. [p45, 57]
- B. Ripley, B. Venables, D. M. Bates, K. Hornik, A. Gebhardt, D. Firth, and Ripley. *MASS: Support Functions and Datasets for Venables and Ripley's MASS*, 2013. URL <https://CRAN.R-project.org/package=crosstalk>. R package version 7.3-58.1. [p57]
- L. Scrucca, C. Fraley, T. B. Murphy, and A. E. Raftery. *Model-Based Clustering, Classification, and Density Estimation Using mclust in R*. Chapman and Hall/CRC, 2023. ISBN 978-1032234953. doi: 10.1201/9781003277965. URL <https://mclust-org.github.io/book/>. R package version 6.0.0. [p57]
- D. J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4): 583–639, 2002. URL <https://doi.org/10.1111/1467-9868.00353>. [p45]
- M. Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000. URL <https://doi.org/10.1111/1467-9868.00265>. [p43]
- Z. Tan, C. Shen, P. Subbarao, W. Lou, and Z. Lu. A joint modeling approach for clustering mixed-type multivariate longitudinal data: Application to the CHILD cohort study. *Arxiv Preprint Arxiv:2210.08385*, 2022. URL <https://doi.org/10.48550/arXiv.2210.08385>. [p41, 42, 45]
- W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL <https://www.stats.ox.ac.uk/pub/MASS4/>. ISBN 0-387-95457-0, R package version 7.3-18. [p57]
- S. Watanabe. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11(Dec):3571–3594, 2010. URL <http://jmlr.org/papers/v11/watanabe10a.html>. [p45]
- H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL <https://ggplot2.tidyverse.org>. [p57]
- C. O. Wilke. *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*, 2024. URL <https://CRAN.R-project.org/package=cowplot>. R package version 1.1.3. [p51]

Zhiwen Tan
Department of Public Health Sciences
Queen's University

Chang Shen
Department of Electrical and Computer Engineering
Queen's University

Zihang Lu
Department of Public Health Sciences
and Department of Mathematics and Statistics
Queen's University
zihang.lu@queensu.ca