MicroBVS-vignette

Matthew D. Koslovsky and Marina Vannucci

Introduction

In this vignette, we provide worked examples on simulated data to demonstrate how to apply the Bayesian variable selection methods for compositional data referenced in [1], [2], and [3].

Dirichlet-Multinomial Bayesian Variable Selection

Having installed and loaded the `MicroBVS` package into the R environment (See README for instructions), generate a simulated data set for a Dirichlet-Multinomial model using the `simulate_DM()` function.

```r
library(MicroBVS)
```

```r
data_DM <- simulate_DM( rho = 0.3 )
#> Loading required package: dirmult
#> Loading required package: MASS
#> Loading required package: matrixcalc
```

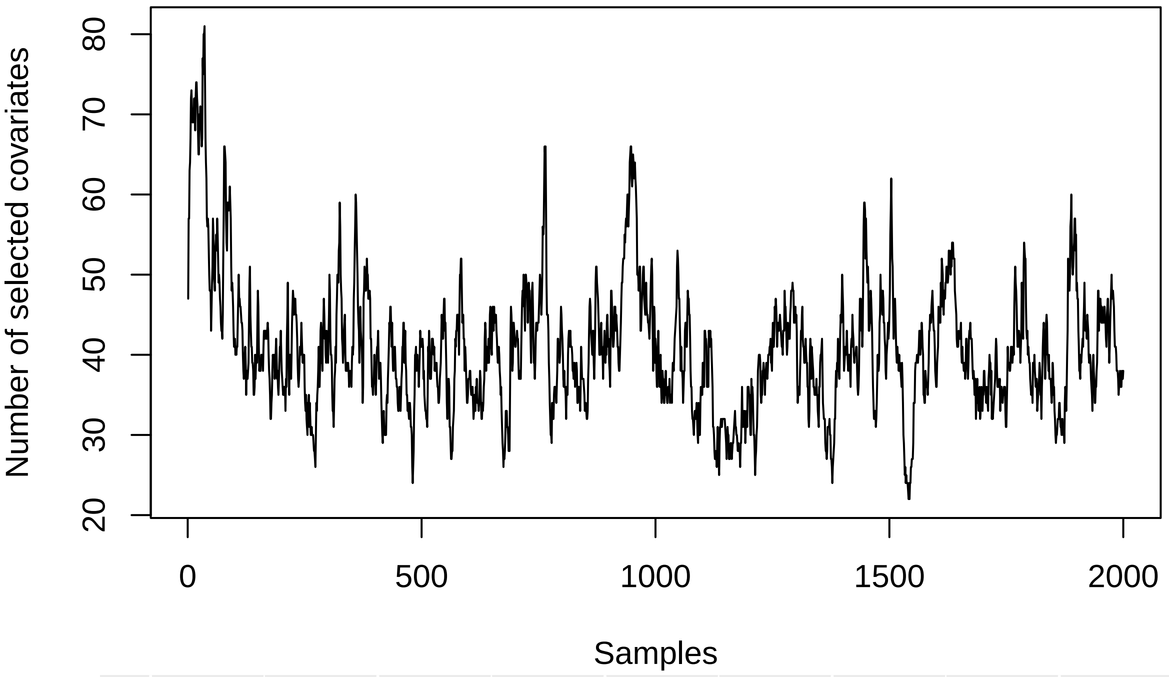
By default 100 subjects (`n_obs = 100`) with 75 taxa (`n_taxa = 75`) and 30 covariates (`n_vars = 30`) are simulated. The function requires specification of `rho` or `Sigma`, where `rho` sets the covariance matrix between simulated covariates as $\sigma_{ij} = \rho^{|i-j|}$ and `Sigma` is simply a given covariance matrix. Of the 30×75 potential covariate-taxon associations, 4 of the covariates are associated with 4 of the simulated taxa. Additional arguments are available to control to minimum (`beta_min`) and maximum (`beta_max`) true regression coefficients, signal-to-noise ratio (`signoise`), minimum (`n_reads_min`) and maximum (`n_reads_max`) number of taxa reads, and the dispersion factor (`theta0`).

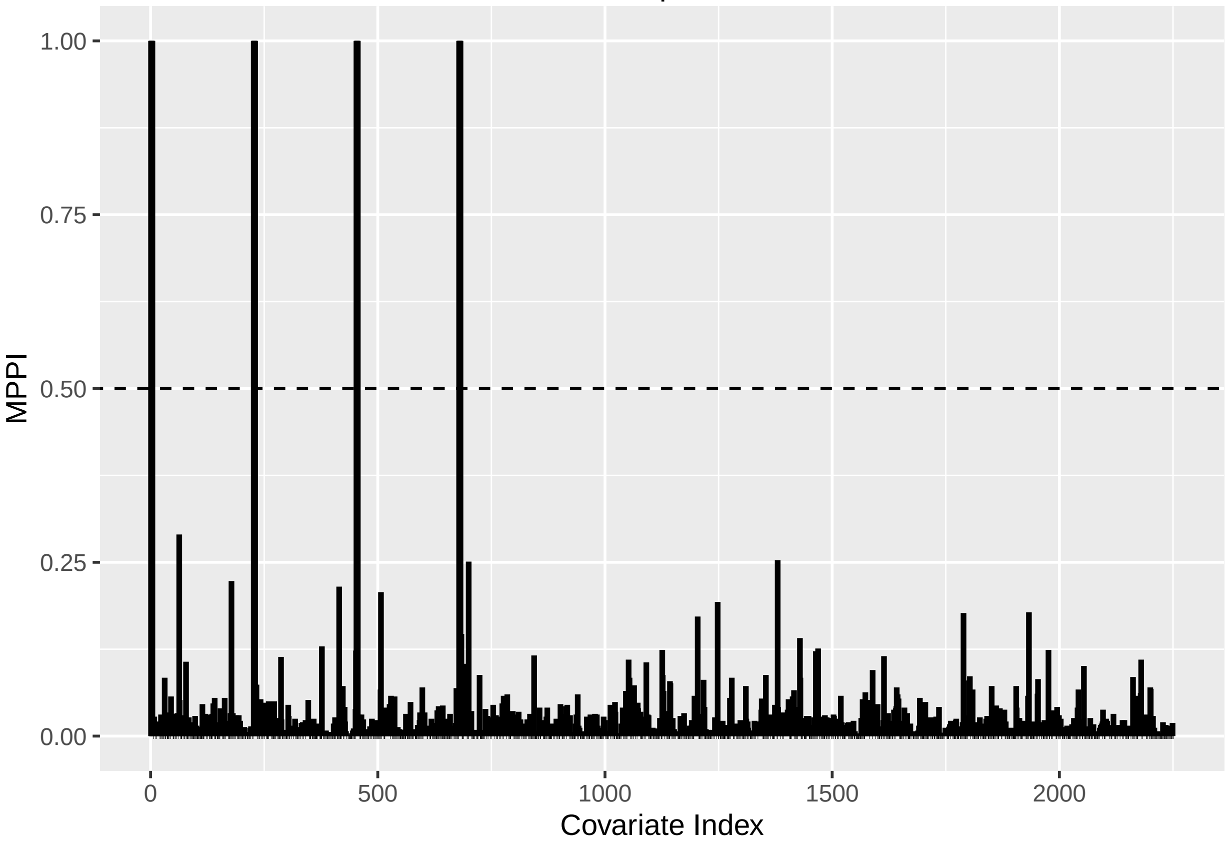
To run the model with a Beta-Binomial prior, simply input the data to the `DMbvs_R()` function and set `prior = "BB"`.

```r
model1 <-  DMbvs_R( z = data_DM$Y, x = data_DM$X[,-1], prior = "BB",  seed = 1)
```

By default, the algorithm is run for $20,000$ iterations, thinning to every 10^{th} iteration, and the hyperparameters for the Beta-Binomial prior are $a = 1$ and $b = 9$. The `model1` object contains a list of the MCMC samples for the taxon-specific intercept terms, α, the inclusion indicators, ζ, and corresponding regression coefficients, φ. To extract selected terms from the model, run

```r
selected1 <-  selected_DM( dm_obj = model1, burnin = 1000, plotting = T)
```





Inclusion is determined with the marginal posterior probability of inclusion (MPPI) for each branch-by-covariate inclusion indicator. By default, the MPPI threshold for significant terms is set to 0.5. `selected1` contains a two element list, where the first element is a number of selected covariate-taxon by 2 dimensional matrix where the first (second) column represents the corresponding row (column) in the φ matrix selected. The second element is a matrix of corresponding MPPIs for each covariate-taxon pair. By setting `plotting = T`, a plot of the sum of covariate-taxon pairs active in each MCMC iteration as well as plot of the MPPIs with the provided significance level is generated.

Next, we demonstrate how to run the DM model when there is an underlying graphical relation between covariates. To simulate these data, we first need to define the graphical structure. Here, we generate a graphical structure similar to our simualtion study.

```
      Sigma <- matrix(0,30,30)
      Sigma[1:5,1:5] <- 0.7
      Sigma[6:10,6:10] <- 0.5
      Sigma[11:15,11:15] <- 0.3
      diag(Sigma) <- 1
      G <- (Sigma != 0 )*1
      data_DM_G <- simulate_DM( Sigma = Sigma )
```

Assuming the true graphical structure is known, we can run the DM model with a MRF prior with known graphical structure.

```
model2 <-  DMbvs_R( z = data_DM_G$Y, x = data_DM_G$X[,-1], prior = "MRF_fixed", G = G, seed = 1)
```

By default, the hyperparameters for the MRF prior are set to $a_G = \log(0.1/0.9)$ and $b_G = 0.5$. Similarly, the `selected_DM()` function can be used to extract the results. If the graphical structure is unknown, simply run

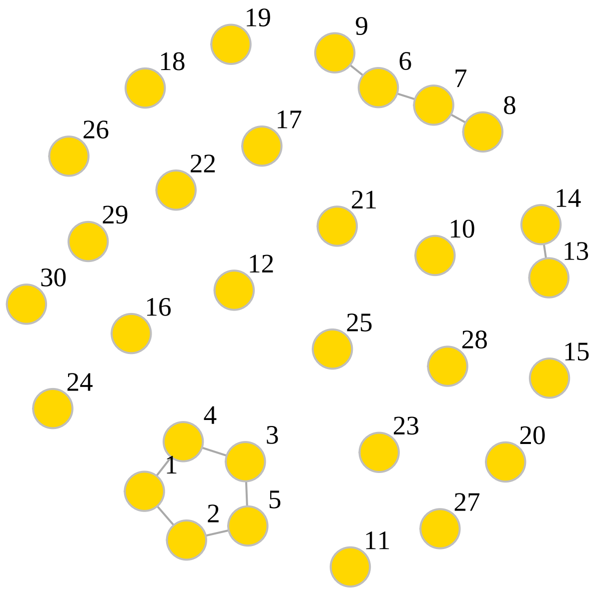
```
model3 <-  DMbvs_R( z = data_DM_G$Y, x = data_DM_G$X[,-1], prior = "MRF_unknown", seed = 1)
```

The output of `model3` is very similar to `model1` and `model2`, with the exception that it additionally includes MCMC samples for the learned graphical structure. To additionally extract the learned graph, run

```
selected3 <-  selected_DM( dm_obj = model3, burnin = 1000, plotting = F, G = T)
```

The third element in the list `estimated_G`, represents the posterior edge inclusion for the graphical structure. By thresholding, an adjacency matrix can be obtained to plot the corresponding graphical structure using the `igraph` package.

```
library( igraph )
graph <- ( selected3$estimated_G > 0.5 )*1
adj <- graph_from_adjacency_matrix( graph , mode = "undirected" )
plot( adj, vertex.color = "gold", vertex.size = 15,  vertex.frame.color = "gray", vertex.label.color =
```



In this example, the model was able to identify the stronger relations ($\sigma_{ij} = 0.7$) between covariates, but selected fewer edges with $\sigma_{ij} = 0.3$ and $\sigma_{ij} = 0.5$. In addition to the data, the true regression coefficients are saved in the `data_DM_G` object. Comparing these to the output, we see the model performed quite well in this toy example.

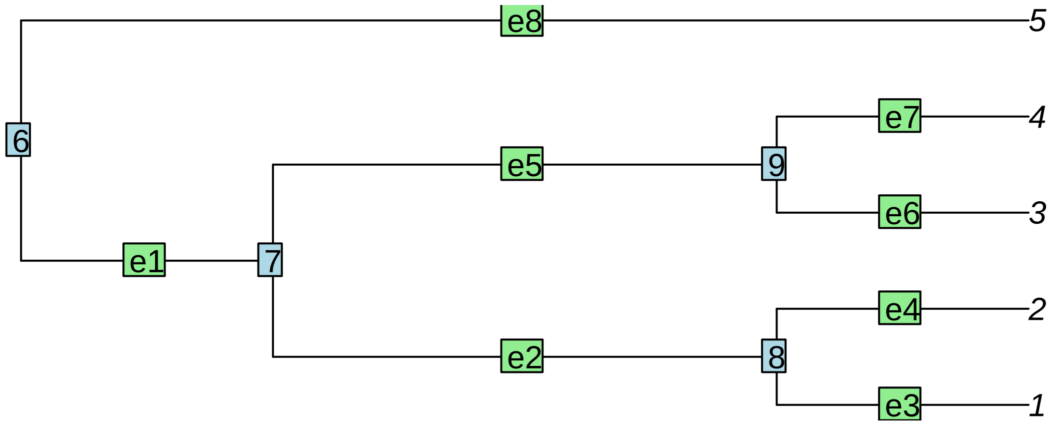
```
which( data_DM_G$betas[,-1] != 0, arr.ind = T )
#>       row col
#>  [1,]   1   1
#>  [2,]   2   1
#>  [3,]   3   1
#>  [4,]   4   1
#>  [5,]   2   4
#>  [6,]   3   4
#>  [7,]   4   4
#>  [8,]   5   4
#>  [9,]   3   7
#> [10,]   4   7
#> [11,]   5   7
#> [12,]   6   7
#> [13,]   4  10
#> [14,]   5  10
#> [15,]   6  10
#> [16,]   7  10
selected3$selected_zeta
#>       row col
#>  [1,]   1   1
#>  [2,]   2   1
#>  [3,]   3   1
#>  [4,]   4   1
#>  [5,]   2   4
#>  [6,]   3   4
#>  [7,]   4   4
#>  [8,]   5   4
#>  [9,]   3   7
#> [10,]   4   7
#> [11,]    5    7
#> [12,]    6    7
#> [13,]    4   10
#> [14,]    5   10
#> [15,]    6   10
#> [16,]    7   10
```

Dirichlet-tree Multinomial Bayesian Variable Selection

Next, we demonstrate how to run the model on compositional data that follow a tree-like structure. To simulate data, run

```
data_DTM <- simulate_DTM( rho = 0.3 )
```

Similar to `simulate_DM`, `simulate_DTM` requires `rho` or `Sigma` to be specified. By default, 100 subjects (`subject_sim = 100`) with 50 covariates (`covariates_sim = 50`), and a random tree with 5 leaf nodes (`num_leaf = 5`) are simulated. The function generates six data objects. Namely, the compositional counts (`Y`), covariates (`X`), true regression coefficients (`phi_sim`), true inclusion indicators (`zeta_sim`), the random tree (`tree`), which is a tree object generated by the `ape` package, and the simulated covariance matrix for the covariates (`Sigma`). The default settings generate a tree structure with *L=5* leaves with *(L-1)x2 = 8* total branches. In the following plot, the branches or edges are labeled e1-e8 and the nodes are labeled 1-9. Node 6 is referred to as the root node, nodes 7-9 are internal nodes, and nodes 1-5 are leaf nodes.



Using the simulated data, we can run the model to identify which of the branch-by-covariate combinations are significant in the model.

```
model4 <- DTMbvs_R( tree = data_DTM$tree, Y = data_DTM$Y, X = data_DTM$X, aa = 0.1, bb = 0.9)
```

```
str(model4)
#> List of 3
#>  $ alpha: num [1:8, 1:5000] -1.163 -0.158 -0.516 -0.628 1.114 ...
#>  $ zeta : num [1:8, 1:50, 1:5000] 0 0 0 0 0 0 0 0 0 0 ...
#>  $ psi  : num [1:8, 1:50, 1:5000] 0 0 0 0 0 0 0 0 0 0 ...
```

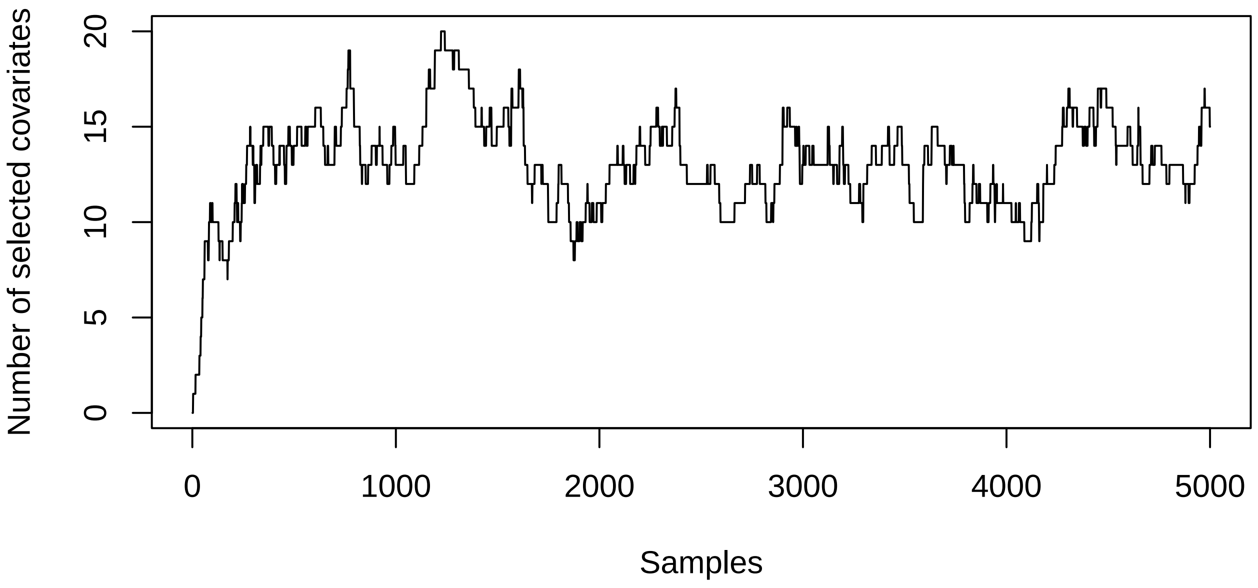
The results object, `model4`, is a *3*-element list of the $s = 1,\ldots,S$ MCMC samples for the $\sum_{v\in V}|C_v|$ branch-specific intercept terms α_{vc}, $\sum_{v\in V}|C_v| \times P$ branch-by-covariate inclusion indicators ζ_{vcp}, and their respective regression coefficients φ_{vcp}. Note that by default, there are S = *5,000* MCMC samples, since the total *50,000* iterations of the MCMC chain were thinned by every *10^th^* iteration. Here, we place a weakly-informative prior for each branch-by-covariate inclusion indicator. Specifically, we set the model to use a beta-binomial prior for inclusion probabilities with hyperparameters $a = 0.1$ and $b = 0.9$.

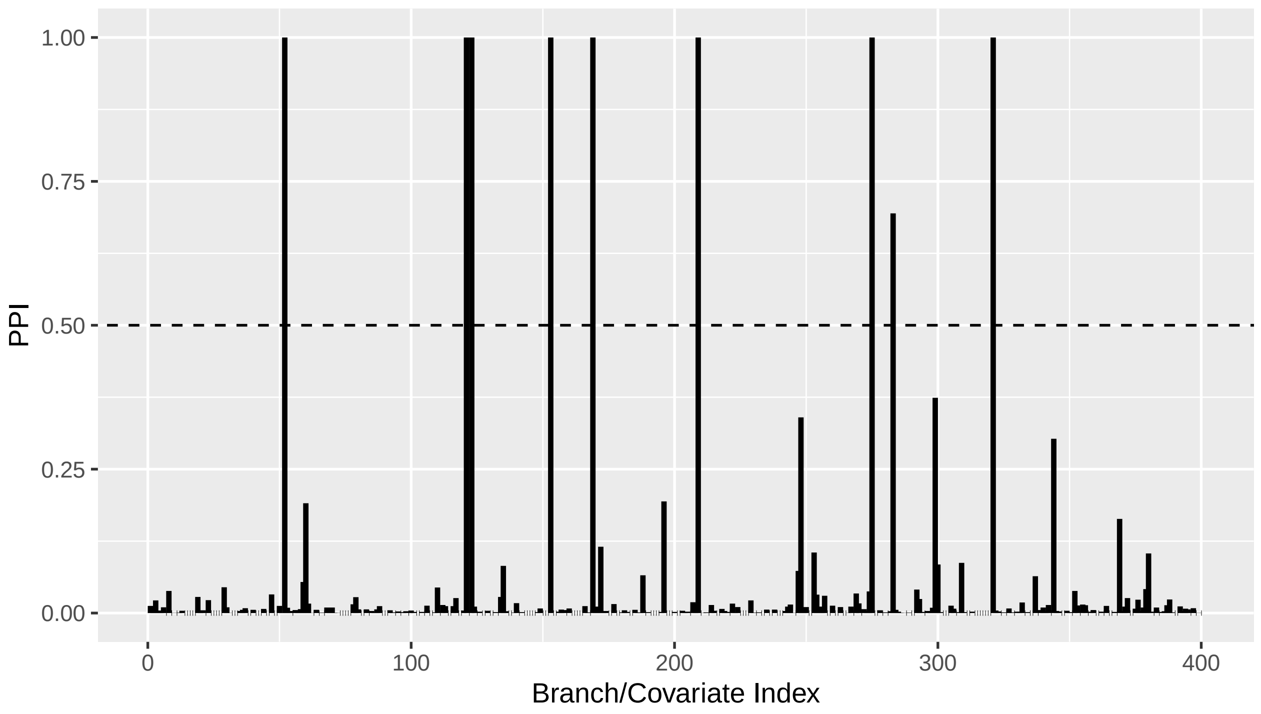
The ordering of the branches corresponds to the edges between nodes stored in the tree object, as defined by the `ape` package. For example, branch 1 corresponds to the edge between node 6 and 7.

```
 data_DTM$tree$edge
#>      [,1] [,2]
#> [1,]    6    7
#> [2,]    7    8
#> [3,]    8    1
#> [4,]    8    2
#> [5,]    7    9
#> [6,]    9    3
#> [7,]    9    4
#> [8,]    6    5
```

Using a threshold of *MPPI* $\geq$ *0.50* with a burn-in of *2,500* iterations, we identify 9 associations. By setting `plotting = TRUE`, we obtain a plot of the number of covariates selected at each MCMC iteration and a plot of the MPPI for each branch-by-covariate index.

```
edge_lab <- c( "e1", "e2", "e3", "e4", "e5", "e6", "e7", "e8")
selected4 <- selected_DTM( dtm_obj = model4, threshold = c( 0.5 ), burnin = 2500, plotting = TRUE, edge
```





Branch	Covariate
e4	7
e1	16
e3	16
e1	20
e1	22
e1	27
e3	35
e3	36
e1	41

The true branch-by-covariates in this model are

Branch	Covariate
e4	7
e4	8
e1	16
e3	16
e1	20
e1	22
e4	25
e1	27
e3	27
e4	28
e3	35
e3	36
e3	38
e1	41
e4	48

Comparatively, we see that for this example, the model was able to identify 9/15 of the true associations with no false positives.

Joint Dirichlet-Multinomial and Linear Regression Model Bayesian Variable Selection

This section demonstrates how to run the method proposed in *A Bayesian model of microbiome data for simultaneous identification of covariate associations and prediction of phenotypic outcomes* by Koslovsky, MD, Hoffman, KL, Daniels, CR, and Vannucci, M. This joint model can be used to simultaneously identify clinical covariates associated with microbial composition data and predict a (continuous) phenotypic response using information contained in the compositional data.

To simulate the covariates, compositional data, and continous outcome, run

```r
data_DMLM <- simulate_DMLM( rho = 0.3 )
```

By default 50 subjects (`subject_sim = 50`) with 50 taxa (`B_sim = 50`) and 50 covariates (`covariates_sim = 50`) are simulated. Of these, 10 covariates (`active_cov = 10`) are associated with the taxa counts, and the first 10 balances are associated with the continous outcome. The covariates are simulated from a normal distribution with covariance structure similar as above. Balances are constructed using sequential binary separation.

To run the model, simply imput the data to the `dm_lm_bvs_R()` function.

```r
model5 <- dm_lm_bvs_R( y = data_DMLM$Y, z = data_DMLM$Z, x = data_DMLM$X, seed = 1 )
```

By default, the algorithm is run for 10,000 iterations, thinning to every 10^{th} iteration. The prior probabilities of inclusion for covariates and balances are both 10%. For large sample sizes, adjust the `rate` (default 1) parameter which controls the proportion of individuals updated with sampling the auxiliary parameter `c`. See the R documentation for various options involving the default parameterization. The `model5` object contains a list (in order) of the MCMC samples for α, ζ, φ, ψ, ξ, Ω, and G following the notation used in the main manuscript. Using the `selected_DMLM()` function, the user can extract the selected covariates and balances in the model. By default, the threshold for active terms is 0.5. For example,

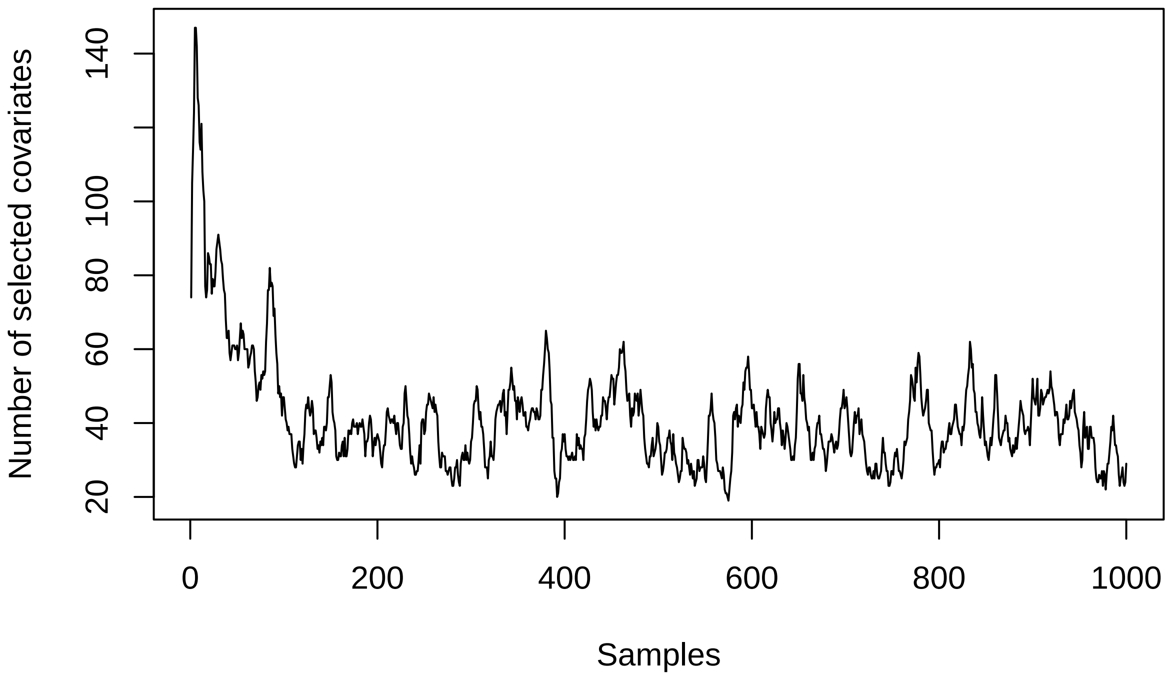
```r
selected5 <- selected_DMLM( model5, burnin = 500 )
```

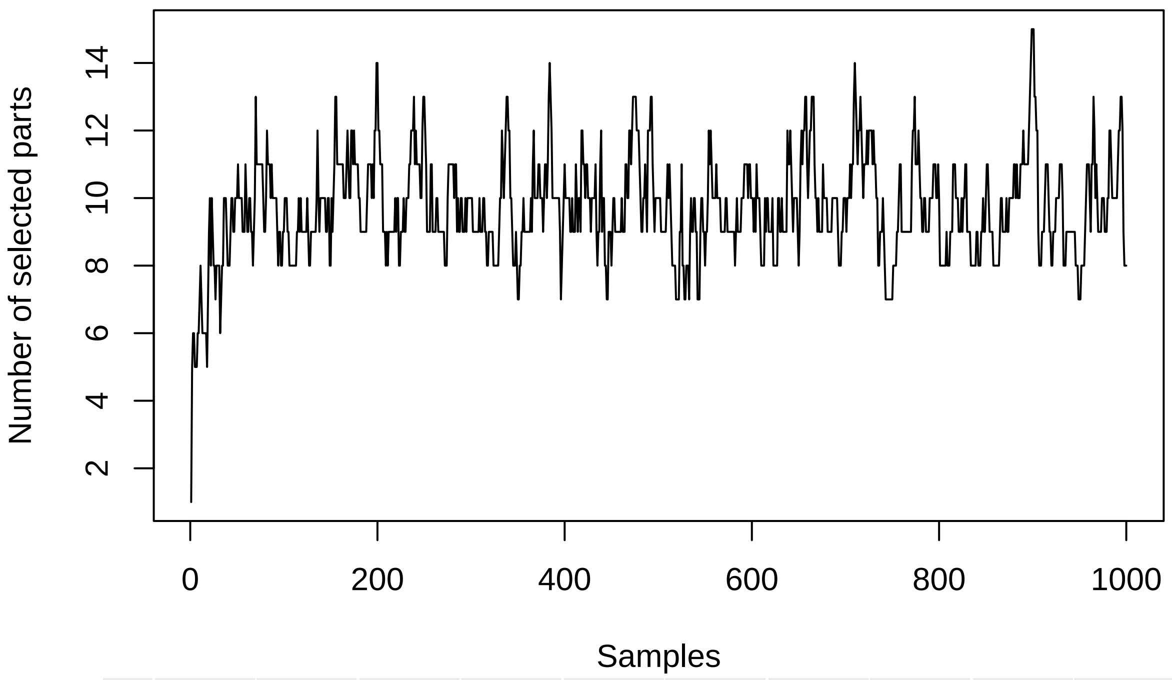
The `selected5` object contains the selected covariates and balances, as well as the marginal posterior probabilities of inclusion (MPPI) for all of the terms. We can see that the model does a good job recovering the active terms in this example:

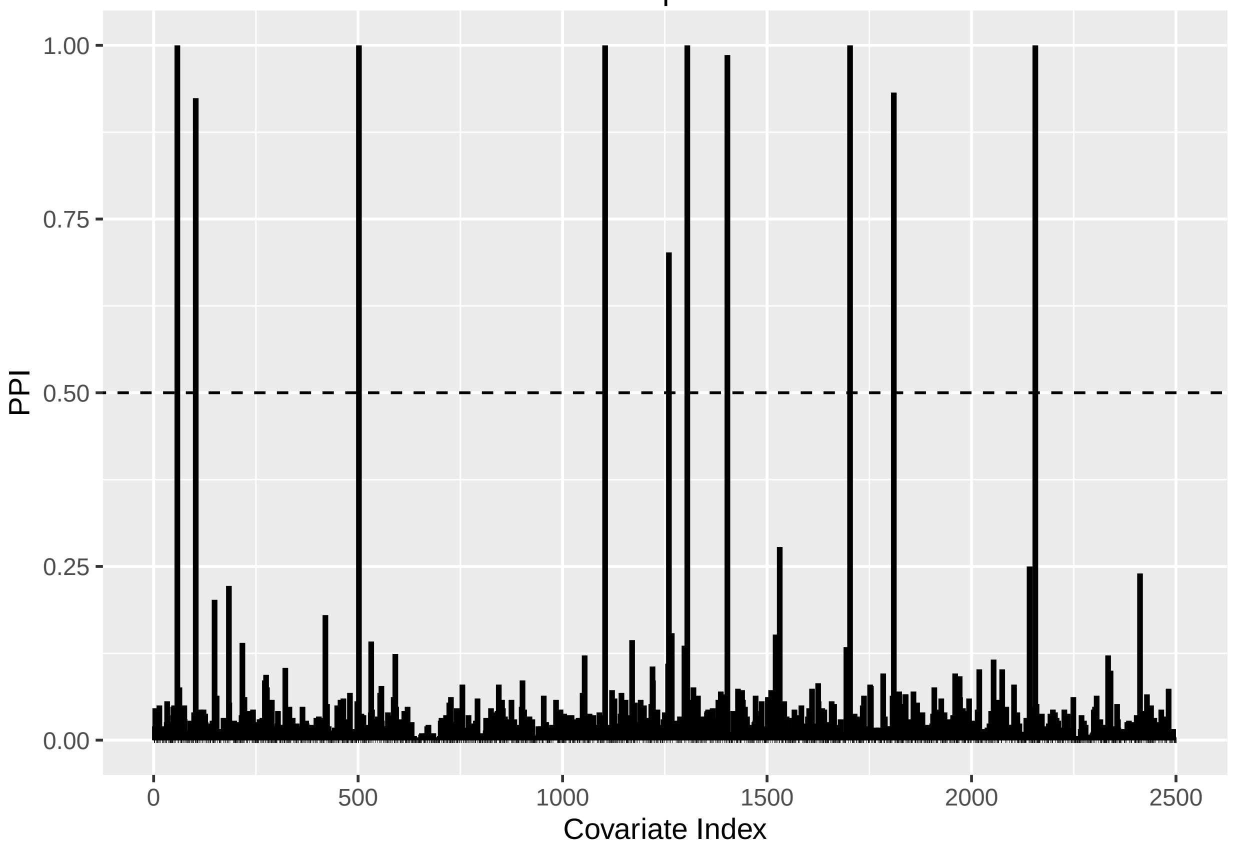
```r
# Selected covariate-taxon relations
selected5$selected_zeta
#>      row col
#> [1,]   8   2
#> [2,]   3   3
#> [3,]   2  11
#> [4,]   4  23
#> [5,]  10  26
#> [6,]   5  27
#> [7,]   3  29
#> [8,]   3  35
#>  [9,]  10  37
#> [10,]   6  44
# True covariate-taxon relations
data_DMLM$true_cov
#>       [,1] [,2]
#>  [1,]   10   37
#>  [2,]    6   44
#>  [3,]    3   35
#>  [4,]   10   26
#>  [5,]    8    2
#>  [6,]    3   29
#>  [7,]    3    3
#>  [8,]    2   11
#>  [9,]    4   23
#> [10,]    5   27
# Selected balances (First 10 active in true model)
selected5$selected_xi
#> [1] 1 2 3 4 5 7 8 9
```

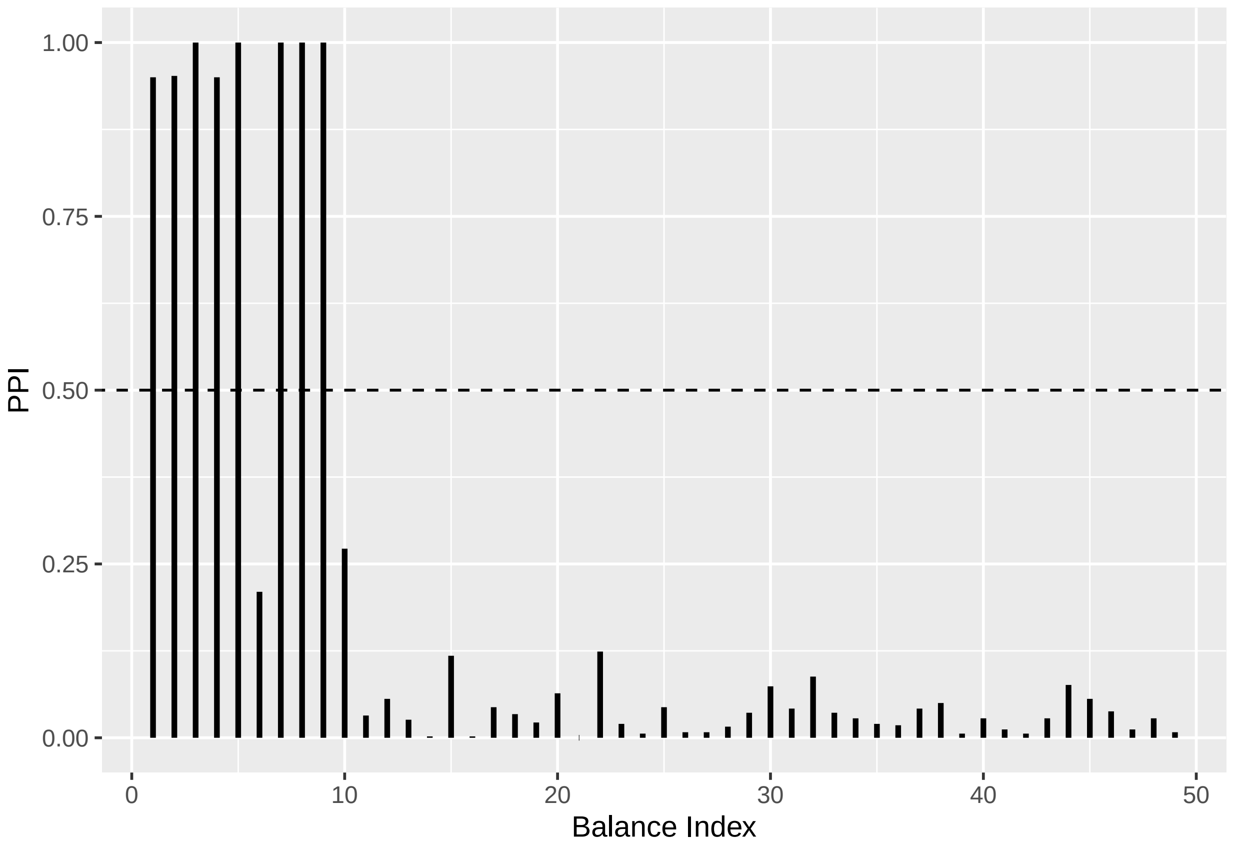
For inference, the `selected_DMLM()` function can be used to generate plots of the MPPIs for both levels of the model as well as plots for the number of terms active in the model at each MCMC iteration. The latter can be used to help assess convergence of the model. To obtain these plots, simply run

```
selected5 <- selected_DMLM( model5, burnin = 500, plotting = TRUE )
```




Number of selected parts
2
4
6
8
10
12
14
0
200
400
600
800
1000
Samples


PPI
0.00
0.25
0.50
0.75
1.00
0
500
1000
1500
2000
2500
Covariate Index



We also provide an alternative sampler that additionally samples the variance and regression coefficients in the linear portion of the model. This approach is typically faster to implement than the orignal version of the sampler described above. To implement this version of the model, run

```
model6 <- dm_lm_bvs_full_R( y = data_DMLM$Y, z = data_DMLM$Z, x = data_DMLM$X, seed = 1 )
```

In this example, we find similar selection performance as `model5`, with the exception of a missed covariate associated with the relative abundances.

```
selected6 <- selected_DMLM( model6, burnin = 500 )
```

```
# Selected covariate-taxon relations
selected6$selected_zeta
#>       row col
#>  [1,]   8   2
#>  [2,]   3   3
#>  [3,]   2  11
#>  [4,]   4  23
#>  [5,]   5  27
#>  [6,]   3  29
#>  [7,]   3  35
#>  [8,]  10  37
#>  [9,]   6  44
# True covariate-taxon relations
data_DMLM$true_cov
#>       [,1] [,2]
#>  [1,]   10   37
#>  [2,]    6   44
#>  [3,]    3   35
#>  [4,]   10   26
#>  [5,]    8    2
#>  [6,]    3   29
#>  [7,]    3    3
#>  [8,]    2   11
#>  [9,]    4   23
#> [10,]    5   27
# Selected balances (First 10 active in true model)
selected6$selected_xi
#> [1] 1 2 3 4 5 7 8 9
```

References

1. Wadsworth D, Argiento R, Guindani M, Galloway-Pena J, Shelburne SA, Vannucci M. An integrative Bayesian Dirichlet-multinomial regression model for the analysis of taxonomic abundances in microbiome data. BMC Bioinformatics. 2017;18:94.
2. Koslovsky MD, Vannucci M. MicroBVS: Dirichlet-Tree Multinomial Regression Models with Bayesian Variable Selection - an R Package. BMC Bioinformatics (Accepted). 2020.
3. Koslovsky MD, Hoffman KL, Daniel CR, Vannucci M. A Bayesian model of microbiome data for simultaneous identification of covariate associations and prediction of phenotypic outcomes. Annals of Applied Statistics (In Press). 2020.