BVSMed-vignette

Jingyan Fu, Matthew Koslovsky, Andreas Neophytou, and Marina Vannucci

Introduction

In this vignette, we provide worked examples on simulated data to demonstrate how to apply the proposed Bayesian joint model to identify overall and relative mediation effects in compositional data in [1]. Having installed and loaded the MicroBVS package into the R environment (See README for instructions), generate a simulated data set for the joint model using the `Simulate_MedDM()` function.

```
library( MicroBVS )

# Set the number of taxa J in the data set
  n_taxa <- 20

# Without covariate
 data_DM <- Simulate_MedDM( n_obs = 100, n_taxa = n_taxa, seed = 123)

# With one binary covariate
 data_DM2 <-  Simulate_MedDM( n_obs = 100, n_taxa = n_taxa, bin_cov = 1, seed = 123)
```

Here, the taxa proportions are assumed to follow a Dirichlet distribution, the multivariate counts are simulated from a multinomial distribution, and the continuous outcome is generated using the log of the active mediators' taxa proportions. In general, our model is flexible to other compositional data structures, as demonstrated in the main manuscript.

By default, 200 subjects (`n_obs = 200`) with 50 taxa (`n_taxa = 50`) and a binary treatment without additional covariates are simulated. To generate additional covariates, the user can specify the number of binary and continuous covariates in both levels of the model using the `bin_cov` and `con_cov` arguments, respectively. The range of the covariates' corresponding regression coefficients in both levels of the model is set to $[-2, 2]$ by default (i.e., `covar_min = -2` and `covar_max = 2`). In the Dirichlet portion of the model, the intercept terms for each taxa, α_j, are uniformly sampled from Unif(-1,1) (i.e., `alpha_intercept_min = -1` and `alpha_intercept_max = 1`) by default. The treatment effects for each taxa, ϕ_j, are uniformly sampled from Unif(1.5,3) (i.e., `phi_min = 1.5` and `phi_max = 3`) for the vector of active mediators (`active = c(1,2,3)`) by default. The corresponding log taxa proportion for the active mediators' regression coefficients, β_j, must sum to zero and are defaulted to `beta_coeff = c(1.3, -0.7, -0.6)`. The intercept and direct treatment effect in the linear portion of the model `c0 = 0` and `trt_coeff = 1`, respectively.

The output of the `Simulate_MedDM()` function contains a list with the continuous outcome, compositional counts and proportions, active mediator indicies, treatment assignments, auxillary covariates (if generated), true parameters in the Dirichlet-multinomial and linear models, as well as the seed used to generate the data.

MCMC Algorithm

To implement the Metropolis-Hastings within Gibbs MCMC algorithm for the proposed mediation model, call the `MCMC_Med()` function, which requires the treatment assignment (`trt`), continuous outcome (`Y`), taxa

counts (`Z`), as well as the column index in `Z` which corresponds to the taxon to investigate as a potential mediator (`taxa`). In this example, we investigate the relative mediation/indirect effect of the first taxon in `Z` (i.e., `taxa = 1`). For large sample sizes, adjust the `rate` (default 1) parameter which controls the proportion of individuals updated with sampling the auxiliary parameter `k`.

```
model1 <- MCMC_Med( trt = data_DM$trt, Y = data_DM$Y, Z = data_DM$Z, taxa = 1, seed =1234 )
model2 <- MCMC_Med( trt = data_DM2$trt, Y = data_DM2$Y, Z = data_DM2$Z,
                    covariate = data_DM2$covariate, taxa = 1, seed = 1234 )
```

By default, the algorithm is run for 5000 iterations, thinning to every 10^{th} iteration, with the first 500 as burn-in, and the hyperparameters for the Beta-Binomial prior are non-informative with $a = 1$ and $b = 1$. Other hyperparameters for the inverse-Gamma distribution are also set as non-informative ($h_\alpha, h_\beta, a_0, b_0, a_m, b_m = 1$), and the variance for the slabs are set to 10. The `model1` object contains a list of the MCMC samples for the taxon-specific intercept terms, α_j, taxa proportions, ψ_j, regression coefficients in Dirichlet model, ϕ_{jp}, their corresponding inclusion indicators, ζ_{jp}, regression coefficients in the linear model, β, their corresponding inclusion indicators, ξ, as well as the assumed hyperparameters.

Inference

The presence of a relative mediation effect is determined with the marginal posterior probability of inclusion (MPPI) for each pair of taxa-specific inclusion indicators in the function `Selection_Med1()` and `Selection_Med3()`, which is used for Strategy 1 and the first part of Strategy 3 in the main manuscript. By default, the MPPI threshold for significant terms is set to 0.5 (`threshold = c(0.5,0.5)`), and we calculate the 95% credible intervals (`quantile = c(0.025, 0.975)`) for the relative mediation effect corresponding to the j^{th} taxa specified in `MCMC_Med()` and the overall mediation effect. The function `Selection_Med2()` is used to perform Strategy 2 and the second part of Strategy 3 in the main manuscript. This function reports the relative mediation effect for all the taxa in the mediator matrix and the overall effect. We first demonstrate how to perform selection for an individual taxon and then provide code to search through all potential relative medatiors using Strategy 1.

```
result <- Selection_Med1( model = model1 )
```

To test the relative mediation effect for each taxon using Strategy 1, simply iterate through each taxon using a for-loop as follows:

```
# Loop over each taxon to identify relative mediation effects. Note, this will
# run the MCMC_Med function J times.
  selected_marginal_mediators <- matrix( ncol = 4, nrow = n_taxa)
  for( i in 1:n_taxa ){
      model_temp <- MCMC_Med( trt = data_DM$trt, Z = data_DM$Z,
                              covariate = data_DM$covariate, Y = data_DM$Y,
                              seed = 123, taxa = i )
      result <- Selection_Med1( model = model_temp )
      selected_marginal_mediators[ i, 1 ] <- result$selected
      selected_marginal_mediators[ i, 2 ] <- result$marginal_mean
      selected_marginal_mediators[ i, 3:4 ] <-  result$marginal_quantile
  }
  colnames( selected_marginal_mediators ) <- c( "Selected", "Posterior Mean", "LB", "UB" )
```

Here, we find that the model was able to correctly identify the 3rd and 4th taxon as relative mediators but not the 5th.

Selected	Posterior Mean	LB	UB
0	0.0022480	0.0000000	0.0335294
0	-0.0008711	-0.0176778	0.0000000
1	-3.2370336	-4.7060926	-2.0438573
1	0.7594560	0.2926164	1.3023693
0	0.0326314	0.0000000	0.4212707
0	-0.0006224	0.0000000	0.0000000
0	-0.0000372	0.0000000	0.0000000
0	-0.0002296	0.0000000	0.0000000
0	0.0001315	-0.0020087	0.0038242
0	0.0000087	0.0000000	0.0000000
0	-0.0010271	-0.0076007	0.0000000
0	-0.0004503	-0.0009609	0.0000000
0	0.0000000	0.0000000	0.0000000
0	-0.0002345	0.0000000	0.0000000
0	0.0016022	0.0000000	0.0320278
0	-0.0016935	-0.0244277	0.0000000
0	0.0004209	0.0000000	0.0000000
0	-0.0001079	0.0000000	0.0000000
0	0.0002188	0.0000000	0.0001595
0	-0.0182482	-0.0920918	0.0000000

The overall mediation effect in this example is estimated as:

```
#> -3.87725 ( -5.86269 , -1.784489 )
```

Users can perform variable selection for covariates in both the Dirichlet-multinomial and the outcome regression model by specifying `covar_select = TRUE` and notating which covariates they want to perform selection on. Here, we find that the model was able to recover both covariates in the DM and LM portion of the model, respectively.

```
covariate_info <- Selection_Med1( model = model2, covar_select = TRUE )
covariate_info$covar_result
#> [1] TRUE TRUE
```

To test the relative mediation effect for all taxa and calculate the overall mediation effect with Strategy 2, use the following:

```
  selected_marginal_mediators2 <- matrix( ncol = 4, nrow = n_taxa )
  result2 <- Selection_Med2( model = model1 )
  selected_marginal_mediators2[,1] = result2$selected
  selected_marginal_mediators2[,2] = result2$marginal_mean
  selected_marginal_mediators2[,3] = result2$marginal_lowerquantile
  selected_marginal_mediators2[,4] = result2$marginal_upperquantile
  colnames( selected_marginal_mediators2 ) <- c( "Selected", "Posterior Mean", "LB", "UB" )
```

Here, we find that the 2nd strategy was able to correctly identify the 3rd taxon as a mediator, but failed to identify the 4th and 5th.

Selected	Posterior Mean	LB	UB
1	0.0045824	0.0000000	0.0622085
0	-0.3313175	-0.6059581	0.3040302
1	-3.8140988	-5.9030287	-2.2849528
0	-0.5723203	-1.4207458	0.0378788
0	0.0192920	-0.0608955	0.0609109
0	-0.0023874	-0.0154579	0.0503359
0	-0.0061889	-0.0154057	0.0028431
1	-0.0101269	-0.0518621	-0.0003133
0	-0.0061983	-0.0228992	0.0057480
0	-0.0061534	-0.0169055	0.0035601
0	-0.0070295	-0.0267485	0.0043969
0	-0.0061713	-0.0224791	0.0114561
0	-0.0075708	-0.0233529	0.0028183
0	-0.0112930	-0.0483237	0.0037316
0	0.0021709	-0.0156885	0.0746390
0	-0.0077482	-0.0294124	0.0045602
0	-0.0090899	-0.0429547	0.0068892
0	-0.0070893	-0.0306444	0.0066575
0	0.0071682	-0.0258179	0.1063160
0	-0.0181324	-0.1228788	0.0122865

Next, we demonstrate how to implement Strategy 3. Note that if the model identifies more than one relative mediator, the input mediator matrix is truncated to the active terms, and Strategy 2 is re-run to obtain the relative indirect effects.

```
  result3 <- Selection_Med3( model = model1 )
  result3$message
#> [1] "more than 1 mediators founded, process the input data"
```

```
### If no mediator found, then we explore the direct effect
  if( result3$message == "No mediator found" ){
  cat("No mediator identified, and the direct effect is", result3$direct_mean, "(",
  result3$direct_quantile,")")
  }

### If 1 mediator is identified, then we automatically have its relative mediation effect
  taxa_order <- 1:n_taxa
  if(result3$message == "1 mediator found"){
  selected_marginal_mediators3 <- c(taxa_order[result3$selected],result3$marginal_mean, result3$marginal
  cat("Only mediator ",selected_marginal_mediators3, "identified", "and the direct effect is", result3$d
  result3$direct_quantile,")")
  }

### If more than 1 mediator is identified, we truncate the mediator matrix and re-run Strategy 2 to est
  if( result3$message == "more than 1 mediators founded, process the input data" ){
  selected_taxa = result3$selected
  taxa_numbers = taxa_order[result3$selected]
  cat("The potential mediators to process are:",taxa_numbers)
  model_temp <- MCMC_Med( trt = data_DM$trt, Z = data_DM$Z[,selected_taxa], covariate = data_DM$Covariat
  selected_marginal_mediators3 <- matrix(ncol = 5, nrow = sum(selected_taxa))
  result3 <- Selection_Med2( model = model_temp)
```

```
selected_marginal_mediators3[,1] = c(taxa_numbers[3],taxa_numbers[-3])
selected_marginal_mediators3[,2] = result3$selected
selected_marginal_mediators3[,3] = result3$marginal_mean
selected_marginal_mediators3[,4] = result3$marginal_lowerquantile
selected_marginal_mediators3[,5] = result3$marginal_upperquantile
colnames( selected_marginal_mediators3 ) <- c("Taxa", "Selected", "Posterior Mean", "LB", "UB" )
}
```

```
kable( selected_marginal_mediators3, align = 'c')
```

Taxa	Selected	Posterior Mean	LB	UB
5	1	1.0306656	0.4547525	1.7663757
3	1	-3.5904960	-6.2230094	-1.9184941
4	0	0.3066807	-0.1362619	0.8927129

We can see that the 3rd strategy correctly identifies the 3rd, 4th, and 5th taxon.

1. Fu J, Koslovsky MD, Neophytou AM, Vannucci M. A Bayesian Joint Model for Mediation Effect Selection in Compositional Microbiome Data. 2023.