A User Mannual for SOHPIE-DNA

Seungjun Ahn

12/2/2022

We introduce the SOHPIE-DNA, a pseudo-value regression approach that determines whether a microbial taxa is significantly differentially connected (DC) between groups with the presence of additional clinical covariate. Of important note, the main difference between our recently devised method, namely a Pseudo-value Regression Approach for Network Analysis (PRANA) (Ahn *et al.*, 2022) with SOHPIE-DNA is that PRANA can be applied on gene expression data only.

Requirements

Please download the following R code from my GitHub repository (https://github.com/sjahnn/SOHPIE-DNA) to use our method.

- `Thetahat_MiNW.R` is to calculate the degree centrality of a taxa from the estimated association matrix of an observed OTU table.
- `SOHPIE-DNA_main.R` is the main code for the analysis.

In addition, please install these R packages prior to use SOHPIE-DNA.

```
# library(robustbase) # To fit robust regression.
# library(SpiecEasi) # To obtain SparCC association network.
# library(parallel) # To utilize mclapply() -- parallel computing
# library(dplyr)  # To use bind() later.
# library(fdrtool) # To compute the q-values.
```

Example

Please download `cleaned_amgut.RDS` from my GitHub respository. This contains clinical and OTU data for 268 subjects and 138 taxa. In this user manual, we will use a subset of the data from the American Gut Project (McDonald, D. *et al.*, 2018), which is available in the SpiecEasi R package (Kurtz, Z. D. *et al.*, 2015) as a toy example. The data in this toy example consists of 30 out of 138 taxa from 50 out of 268 subjects.

Please provide the directory information where your two R codes and RDS data file downloaded from the GitHub repository.

```
# dir_main = "your file directory where you saved two R codes."
# dir_data = "your file directory where you saved RDS data file from the repository."
```

Load the dataset and R code that calculates the degree centrality of a node.

```
# This will load the amgut dataset.
combined.amgut.data = readRDS(file.path(dir_data, "cleaned_amgut.rds"))
combined.amgut.data = combined.amgut.data[1:50, ]
# Load the function to calculate the degree centality measure (thetahat in the paper.
source(file.path(dir_main, "Thetahat_MiNW.R"))
```

`est_method` below is to specify the method to estimate the association matrix given the observed OTU table. In our paper, the Sparse Correlations for compositional Data (SparCC) is used.

```
# Estimate the association matrix/network via SparCC.
est_method = sparcc

## Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combined.amgut.data[ , 8:37] # OTU table part of the combined data.
## Note: Please comment out above and uncomment the line below
##       if you wish to replicate what was done in our paper with 138 taxa.
#OTUtab = combined.amgut.data[ , 8:ncol(combined.amgut.data)]


## Clinical/demographic covariates (phenotypic data):
phenodat = combined.amgut.data[, 1:7] # first column is ID, so not using it.
```

Up to this point, we are ready to use SOHPIE-DNA. See the next step.

Differential Network Analysis with SOHPIE-DNA.

The main variable in this analysis is the binary indicator for migraine headache (yes or no). In Step 1 below, we obtain the indices of subjects who are 'migraineurs' vs. 'non-migraineurs.' These indices are used to dichotomize the OTU table into 'migraineurs (Group B; i.e. case)' and 'non-migraineurs (Group A; i.e. control).' This is important as the we estimate the group-specific $p \times p$ association matrices.

```
##############################################################################
#  STEP 1. Estimate an association matrix through the SparCC.
##############################################################################
# Indices for non-migraineurs; namely Group A
newindex_A = which(combined.amgut.data$bin_migraine == 0)
# Indices for migraineurs; namely Group B
newindex_B = which(combined.amgut.data$bin_migraine == 1)

# OTU table for Group A using the indices above.
OTUtabA = OTUtab[newindex_A, ]
# OTU table for Group B using the indices above.
OTUtabB = OTUtab[newindex_B, ]

n_A <- length(newindex_A) # Sample size for Group A.
n_B <- length(newindex_B) # Sample size for Group B.

# Estimate an association matrix for each group.
sparcc.matA = est_method(data = OTUtabA)$Cor
sparcc.matB = est_method(data = OTUtabB)$Cor
```

In Step 2, we compute the degree centrality of each taxon by taking the marginal sums of the group-specific association matrices that are acquired from Step 1 above. The degree centrality corresponds to the $\hat{\theta}_k$ for each taxa k. Please make sure you download `Thetahat_MiNW.R` from my repository to use `thetahats` function. Otherwise, you will encounter an error.

```
##############################################################################################
#  STEP 2. Calculate \hat{\theta}_{k} by taking the marginal sum of the
#          association matrix to obtain the degree centrality measure for each taxon.
##############################################################################################
thetahat_grpA = thetahats(sparcc.matA)
thetahat_grpB = thetahats(sparcc.matA)
```

In Step 3, the association matrices will be re-estiamated using the same OTU table but without the ith subject, where $i = 1, \ldots, n$. Please be aware that this may take some time depending on the network size

and/or sample size.

```
#############################################################################################
#  STEP 3. Re-estimate association matrix using the same OTU table without i-th subject.
#          Then, calculate \hat{\theta}_{k(i)} from the re-estimated association matrix.
#############################################################################################
# Re-estimation part
sparcc.mat_drop_grpA <- mclapply(newindex_A, function(j) sparcc(OTUtabA[-j, ])$Cor)
sparcc.mat_drop_grpB <- mclapply(newindex_B, function(j) sparcc(OTUtabB[-j, ])$Cor)
```

`mclapply` is a parallelized version of `lapply`. The number of cores can be adjusted by specifying `mc.cores` option in the `mclapply` function.

Below is to calculate $\hat{\theta}_{k(i)}$, the marginal sums for the re-estimated association matrices obtained earlier.

```
# thetahat_{-i} for each taxa
thetahat_drop_grpA <- sapply(sparcc.mat_drop_grpA, thetahats)
thetahat_drop_grpB <- sapply(sparcc.mat_drop_grpB, thetahats)
```

We have the main ingredients $\hat{\theta}_k$ and $\hat{\theta}_{k(i)}$ from Step 2 and 3, and now we need to calculate jackknife pseudo-values, denoted as $\tilde{\theta}_{ik}$ in Step 4.

The input for `thetatilde` function requires $\hat{\theta}_k$, $\hat{\theta}_k$, and the sample size for each groups.

```
#############################################################################################
#  STEP 4. Calculate jackknife pseudovalues (\tilde{\theta}_{ik} using \hat{\theta}_{k}
#          and \hat{\theta}_{k(i)}.
#############################################################################################
thetatildefun <- function(thetahatinput, thetahatdropinput, sizegroup) {
        thetatildeout = matrix(NA, ncol=length(thetahatinput), nrow=sizegroup)
        thetatildeout = sapply(1:nrow(thetahatdropinput), function(k) {
                sizegroup * thetahatinput[k] - (sizegroup - 1) * thetahatdropinput[k, ]
        })
        return(thetatildeout)
}

thetatilde_grpA = thetatildefun(thetahat_grpA, thetahat_drop_grpA, n_A)
thetatilde_grpB = thetatildefun(thetahat_grpB, thetahat_drop_grpB, n_B)
thetatilde = rbind(thetatilde_grpA, thetatilde_grpB)
colnames(thetatilde) = colnames(OTUtab) # Map the column names (taxa names)
```

In Step 5, a robust regression is fitted to regress the pseudo-values on a set of clinical covariates which includes age, sex, dental floss frequency, exercise frequency, alcohol consumption, and pet ownership. In this example, the main grouping variable is the binary migraine indicator.

```
#############################################################################################
#  STEP 5. Fit a robust regression model
#############################################################################################
pseudo.reg.res <- lapply(1:ncol(thetatilde), function(i) {
        m <- thetatilde[, i]
        df <- data.frame(phenodat,
                         m = m)
        fit <- ltsReg(m ~ bin_migraine + age + sex + bin_floss + bin_exercise +
                              cat_alcohol + bin_dog, data = df, mcd=FALSE)
        return(fit)
})
```

Obtain the p-values (and beta coefficients in case) from the fitted models.

```
### Obtain p-values (and coefficient estimates if interested) for each taxa:
beta_hat = vector(mode = "list", ncol(thetatilde))
p_values = vector(mode = "list", ncol(thetatilde))
k = NULL
for(k in 1:ncol(thetatilde)) {
        # The beta coefficients for each model:
        beta_hat[[k]] <- summary(pseudo.reg.res[[k]])$coef[-1, "Estimate"]
        # p-values for each model:
        p_values[[k]] <- summary(pseudo.reg.res[[k]])$coef[-1, "Pr(>|t|)"]

}

# Convert list into data.frame.
beta_hat = as.data.frame(bind_rows(beta_hat))
# Map the taxa names to the data.frame for betahats.
rownames(beta_hat) <- colnames(OTUtab)

# Convert list into data.frame.
p_values = as.data.frame(bind_rows(p_values))
# Map the taxa names to the data.frame for p-values.
rownames(p_values) <- colnames(OTUtab)
```

Here, we would like to make a statement whether whether a taxa is differentially connected (DC) between migraineurs and non-migraineurs. thus, we subset the vector of p-values for migraine indicator variable is extracted from the `p_values` data.frame.

```
# p-values for binary migraine indicator accounting for other clinical covariates.
binmigraine_pval = p_values[, 1]
```

The q-value is used to appropriately control the false discovery rate incurred among a set of DC taxa from the multiple hypothesis testing.

```
# Compute the q-values :
q_values = fdrtool(binmigraine_pval, statistic = "pvalue", plot=FALSE, verbose = FALSE)$qval
```

```
## Warning in fdrtool(binmigraine_pval, statistic = "pvalue", plot = FALSE, : There
## may be too few input test statistics for reliable FDR calculations!
```

```
# Map the taxa names to the data.frame for q-values
names(q_values) <- colnames(OTUtab)
```

Lastly, return the taxa names of the significantly differentially connected (DC) taxa from SOHPIE-DNA.

```
sigDCtaxa = q_values[which(q_values < 0.05)]
names(sigDCtaxa)
```

```
## [1] "187524" "512309"
```

References

[1] Ahn, S., Grimes, T., & Datta, S. (2022). A pseudo-value regression approach for differential network analysis of co-expression data. Revised and Resubmitted to *BMC Bioinformatics*.

[2] McDonald, D. *et al.* American gut: an open platform for citizen science microbiome research. *mSystems*. **3**, e00031-18 (2018).

[3] Kurtz, Z. D. *et al.* Sparse and compositionally robust inference of microbial ecological networks. *PLoS Computational Biology*. **11**, e1004226 (2015).