IFAA

IFAA is a robust approach to make inference on the association of covariates with the absolute abundance (AA) of microbiome in an ecosystem. It can be also directly applied to relative abundance (RA) data to make inference on AA because the ratio of two RA is equal ratio of their AA. This algorithm can estimate and test the associations of interest while adjusting for potential confounders. High-dimensional covariates are handled with regularization. The estimates of this method have easy interpretation like a typical regression analysis. This algorithm can find optimal reference taxa/OTU/ASV and control FDR by permutation. The IFAA package also offers the 'MZILN' function for estimating and testing associations of abundance ratios with covariates.

To model the association, the following equation is used:

$$\log(\mathcal{Y}_i^k)|\mathcal{Y}_i^k > 0 = \beta^{0k} + X_i^T\beta^k + W_i^T\gamma^k + Z_i^T b_i + \epsilon_i^k, \quad k = 1, ..., K+1,$$

where

- $\mathcal{Y}_i^k$ is the AA of taxa k in subject i in the entire ecosystem.
- X_i is the covariate matrix.
- W_i is the confounder matrix.
- Z_i is the design matrix for random effects.
- β^k is the regression coefficients that will be estimated and tested with the `IFAA()` function.

The challenge in microbiome analysis is that we can not oberve $\mathcal{Y}_i^k$. What is observed is its small proportion: $Y_i^k = C_i\mathcal{Y}_i^k$ where C_i is an unknown number between 0 and 1 that denote the observed proportion. The IFAA method successfuly addressed this challenge.

Package installation

To install, type the following command in R console:

```
install.packages("IFAA", repos = "http://cran.us.r-project.org")
```

The package could be also installed from GitHub using the following code:

```
require(devtools)
devtools::install_github("gitlzg/IFAA")
```

Input for IFAA() function

Most of the time, users just need to feed the first five inputs to the function: `MicrobData`, `CovData`, `linkIDname`, `testCov` and `ctrlCov`. All other inputs can just take their default values. Below are all the inputs of the functions

- `MicrobData`: Microbiome data matrix containing microbiome abundance with each row per sample and each column per taxon/OTU/ASV. It should contain an `"id"` variable to correspond to the `"id"` variable in the covariates data: `CovData`. This argument can also take file directory path. For example, `MicrobData="C://...//microbiomeData.tsv"`.

- `CovData`: Covariates data matrix containing covariates and confounders with each row per sample and each column per variable. It should also contain an `"id"` variable to correspond to the `"id"` variable in the microbiome data: `MicrobData`. This argument can also take file directory path. For example, `CovData="C://...//covariatesData.tsv"`.
- `linkIDname`: Variable name of the `"id"` variable in both `MicrobData` and `CovData`. The two data sets will be merged by this `"id"` variable.
- `testCov`: Covariates that are of primary interest for testing and estimating the associations. It corresponds to X_i in the equation. Default is `NULL` which means all covariates are `testCov`.
- `ctrlCov`: Potential confounders that will be adjusted in the model. It corresponds to W_i in the equation. Default is `NULL` which means all covariates except those in `testCov` are adjusted as confounders.
- `testMany`: This takes logical value `TRUE` or `FALSE`. If `TRUE`, the `testCov` will contain all the variables in `CovData` provided `testCov` is set to be `NULL`. The default value is `TRUE` which does not do anything if `testCov` is not `NULL`.
- `ctrlMany`: This takes logical value `TRUE` or `FALSE`. If `TRUE`, all variables except `testCov` are considered as control covariates provided `ctrlCov` is set to be `NULL`. The default value is `FALSE`.
- `nRef`: The number of randomly picked reference taxa used in phase 1. Default number is `40`.
- `nRefMaxForEsti`: The maximum number of final reference taxa used in phase 2. The default is `2`.
- `refTaxa`: A vector of taxa names. These are reference taxa specified by the user to be used in phase 1 if the user believe these taxa are indepenent of the covariates. If the number of reference taxa is less than 'nRef', the algorithm will randomly pick extra reference taxa to make up 'nRef'. The default is `NULL` since the algorithm will pick reference taxa randomly.
- `adjust_method` The adjusting method used for p value adjustment. Default is "BY" for dependent FDR adjustment. It can take any adjustment method for p.adjust function in R.
- `fdrRate`: The false discovery rate for identifying taxa/OTU/ASV associated with `testCov`. Default is `0.15`.
- `paraJobs`: If `sequentialRun` is `FALSE`, this specifies the number of parallel jobs that will be registered to run the algorithm. If specified as `NULL`, it will automatically detect the cores to decide the number of parallel jobs. Default is `NULL`.
- `bootB`: Number of bootstrap samples for obtaining confidence interval of estimates in phase 2 for the high dimensional regression. The default is `500`.
- `standardize`: This takes a logical value `TRUE` or `FALSE`. If `TRUE`, all design matrix X in phase 1 and phase 2 will be standardized in the analyses. Default is `FALSE`.
- `sequentialRun`: This takes a logical value `TRUE` or `FALSE`. Default is `FALSE`. This argument could be useful for debug.
- `refReadsThresh`: The threshold of proportion of non-zero sequencing reads for choosing the reference taxon in phase 2. The default is `0.2` which means at least 20% non-zero sequencing reads.
- `taxkeepThresh`: The threshold of number of non-zero sequencing reads for each taxon to be included into the analysis. The default is `1` which means taxon with at least `1` sequencing reads will be included into the analysis.
- `SDThresh`: The threshold of standard deviations of sequencing reads for been chosen as the reference taxon in phase 2. The default is `0.05` which means the standard deviation of sequencing reads should be at least `0.05` in order to be chosen as reference taxon.
- `SDquantilThresh`: The threshold of the quantile of standard deviation of sequencing reads, above which could be selected as reference taxon. The default is `0`.

- `balanceCut`: The threshold of the proportion of non-zero sequencing reads in each group of a binary variable for choosing the final reference taxa in phase 2. The default number is `0.2` which means at least 20% non-zero sequencing reads in each group are needed to be eligible for being chosen as a final reference taxon.
- `seed`: Random seed for reproducibility. Default is `1`. It can be set to be NULL to remove seeding.

Output for IFAA() function

The estimation results are saved in the following lists:

- `sig_results`: A list containing estimating results that are statistically significant.
- `full_results`: A list containing all estimating results. NA denotes unestimable.

The covariates data used in the analyses including `testCov` and `ctrlCov` is saved in the following object:

- `covariatesData`: A dataset containing covariates and confounders used in the analyses

Examples

The example datasets `dataM` and `dataC` are included in the package. They could be accessed by:

```
library(IFAA)

data(dataM)
dim(dataM)
#> [1] 40 61
dataM[1:5, 1:8]
#>   id rawCount1 rawCount2 rawCount3 rawCount4 rawCount5 rawCount6 rawCount7
#> 1  1         4        49         2         0       360       222         4
#> 2  2         0        20        14         0        86       211         5
#> 3  3         3         0         3         7         0        57         0
#> 4  4         9        18         5        31        42        58         8
#> 5  5         0         2         1        19        15        67         6

data(dataC)
dim(dataC)
#> [1] 40  4
dataC[1:3, ]
#>   id        v1        v2        v3
#> 1  1  58.06969 -49.90376 -15.30643
#> 2  2  25.96522 -68.58894 -23.10992
#> 3  3 193.71625 124.40186 119.56747
```

Both the microbiome data `dataM` and the covariates data `dataC` contain 40 samples (i.e., 40 rows).

- `dataM` contains 60 taxa with absolute abundances and these are gut microbiome.
- `dataC` contains 3 covariates.

Next we analyze the data to test the association between microbiome and the variable `"v1"` while adjusting for the variables (potential confounders) `"v2"` and `"v3"`.

```
results <- IFAA(MicrobData = dataM,
                CovData = dataC,
                linkIDname = "id",
                testCov = c("v1"),
                ctrlCov = c("v2","v3"),
```

```
                fdrRate = 0.15)
#> Data dimensions (after removing missing data if any):
#> 40 samples
#> 60 taxa/OTU/ASV
#> 1 testCov variables in the analysis
#> These are the testCov variables:
#> v1
#> 2 ctrlCov variables in the analysis
#> These are the ctrlCov variables:
#> v2, v3
#> 0 binary covariates in the analysis
#> 25.71 percent of microbiome sequencing reads are zero
#> Start Phase 1 analysis
#> 6 parallel jobs are registered for analyzing 40 reference taxa in Phase 1
#> 33 percent of phase 1 analysis has been done
#> 6 parallel jobs are registered for analyzing 20 reference taxa in Phase 1
#> 67 percent of phase 1 analysis has been done
#> Phase 1 analysis used 0.72 minutes
#> Start Phase 2 parameter estimation
#> Start estimation for the 1th final reference taxon:
#> Estimation done for the 1th final reference taxon:  and it took 0.012 minutes
#> Start estimation for the 2th final reference taxon:
#> Estimation done for the 2th final reference taxon:  and it took 0.009 minutes
#> Phase 2 parameter estimation done and took 0.021 minutes.
#> The entire analysis took 0.74 minutes
```

In this example, we are only interested in testing the associations with `"v1"` which is why `testCov=c("v1")`. The variables `"v2" and "v3"` are adjusted as potential confounders in the analyses. The final analysis results are saved in the list `sig_results`:

```
results$sig_results
#> $v1
#>              estimate      SE est     CI low      CI up  adj p-value
#> rawCount18 0.02549632 0.004759192 0.01616830 0.03482434 1.138360e-05
#> rawCount36 0.02646417 0.005062444 0.01654178 0.03638656 1.542983e-05
#> rawCount41 0.03042955 0.004687643 0.02124177 0.03961733 2.291142e-08
```

The results found three taxa `"rawCount18"`, `"rawCount36"`, `"rawCount41"` associated with `"v1"` while adjusting for `"v2" and "v3"`. The regression coefficients and their 95% confidence intervals are provided. These coefficients correspond to β^k in the model equation.

The interpretation is that

- Every unit increase in `"v1"` is associated with approximately 2.5% increase in the absolute abundance of `"rawCount18"`, approximately 2.6% increase in the absolute abundance of `"rawCount36"`, and approximately 3.0% increase in the absolute abundance of `"rawCount41"` in the entire gut ecosystem.

All the analyzed covariates including `testCov` and `ctrlCov` can be extracted using the object `covariatesData`. The covariates data of the first 10 subjects can extracted as follows:

```
results$covariatesData[1:10,]
#>    id         v1         v2          v3
#> 1   1  58.069691  -49.90376  -15.306430
#> 2   2  25.965216  -68.58894  -23.109922
#> 3   3 193.716251  124.40186  119.567468
#> 4   4  72.156467  -98.48536    2.877972
```

```
#> 5   5  98.062712   23.55358  -79.893161
#> 6   6  83.094848 -116.95821 -107.641285
#> 7   7   8.217154 -205.64480 -139.958481
#> 8   8  36.169820   58.95708   26.890379
#> 9   9 152.786131  162.60935  138.731954
#> 10 10  41.621790   65.15427   59.974310
```

MZILN() function

The IFAA package can also implement the Multivariate Zero-Inflated Logistic Normal (MZILN) regression model for estimating and testing the association of abundance ratios with covariates. The `MZILN()` function estimates and tests the associations of user-specified abundance ratios with covariates. When the denominator taxon of the ratio is independent of the covariates, 'MZILN()' should generate similar results as 'IFAA()'. The regression model of 'MZILN()' can be expressed as follows:

$$\log\left(\frac{\mathcal{Y}_i^k}{\mathcal{Y}_i^{K+1}}\right)|\mathcal{Y}_i^k > 0, \mathcal{Y}_i^{K+1} > 0 = \alpha^{0k} + \mathcal{X}_i^T\alpha^k + \epsilon_i^k, \quad k = 1, ..., K,$$

where

- $\mathcal{Y}_i^k$ is the AA of taxa k in subject i in the entire ecosystem.
- $\mathcal{Y}_i^{K+1}$ is the reference taxon (specified by user).
- $\mathcal{X}_i$ is the covariate matrix for all covariates including confounders.
- α^k is the regression coefficients that will be estimated and tested.

Input for MZILN() function

Most of the time, users just feed the first six inputs to the function: `MicrobData`, `CovData`, `linkIDname`, `targetTaxa`, `refTaxa` and `allCov`. All other inputs can just take their default values. All the inputs for 'MZILN()' are:

- `MicrobData`: Microbiome data matrix containing microbiome abundance with each row per sample and each column per taxon/OTU/ASV. It should contain an `"id"` variable to correspond to the `"id"` variable in the covariates data: `CovData`. This argument can also take file directory path. For example, `MicrobData="C://...//microbiomeData.tsv"`.
- `CovData`: Covariates data matrix containing covariates and confounders with each row per sample and each column per variable. It should also contain an `"id"` variable to correspond to the `"id"` variable in the microbiome data: `MicrobData`. This argument can also take file directory path. For example, CovData="C://...//covariatesData.tsv".
- `linkIDname` Variable name of the `"id"` variable in both `MicrobData` and `CovData`. The two data sets will be merged by this `"id"` variable.
- `targetTaxa` The numerator taxa names specified by users for the targeted ratios. Default is NULL in which case all taxa are numerator taxa (except the taxa in the argument 'refTaxa').
- `refTaxa` Denominator taxa names specified by the user for the targeted ratios. This could be a vector of names.
- `allCov` All covariates of interest (including confounders) for estimating and testing their associations with the targeted ratios. Default is 'NULL' meaning that all covariates in covData are of interest.
- `adjust_method` The adjusting method for p value adjustment. Default is "BY" for dependent FDR adjustment. It can take any adjustment method for p.adjust function in R.
- `fdrRate` The false discovery rate for identifying ratios associated with `allCov`. Default is `0.15`.

- `paraJobs` If `sequentialRun` is `FALSE`, this specifies the number of parallel jobs that will be registered to run the algorithm. If specified as `NULL`, it will automatically detect the cores to decide the number of parallel jobs. Default is `NULL`.
- `bootB` Number of bootstrap samples for obtaining confidence interval of estimates for the high dimensional regression. The default is `500`.
- `taxkeepThresh` The threshold of number of non-zero sequencing reads for each taxon to be included into the analysis. The default is `1` which means taxon with at least `1` sequencing reads will be included into the analysis.
- `standardize` This takes a logical value `TRUE` or `FALSE`. If `TRUE`, all design matrix X in the analysis will be standardized. Default is `FALSE`.
- `sequentialRun` This takes a logical value `TRUE` or `FALSE`. Default is `TRUE`. It can be set to be "FALSE" to increase speed if there are multiple taxa in the argument 'refTaxa'.
- `seed` Random seed for reproducibility. Default is `1`. It can be set to be NULL to remove seeding.

Output for MZILN() function

The estimation results are saved in the following lists:

- `targettaxa_result_list`: A list containing estimating results for targeted ratios. Only available when targetTaxa is non-empty.
- `sig_results`: A list containing estimating results for all significant ratios

All covariates data used in the analysis is saved in the following object:

- `covariatesData`: A dataset containing all covariates used in the analyses

Examples

We use the same example data The example dataset as that for illustrating the IFAA function. `dataM` and `dataC` are included in the package. They could be accessed by:

```
data(dataM)
dim(dataM)
#> [1] 40 61
dataM[1:5, 1:8]
#>   id rawCount1 rawCount2 rawCount3 rawCount4 rawCount5 rawCount6 rawCount7
#> 1  1         4        49         2         0       360       222         4
#> 2  2         0        20        14         0        86       211         5
#> 3  3         3         0         3         7         0        57         0
#> 4  4         9        18         5        31        42        58         8
#> 5  5         0         2         1        19        15        67         6

data(dataC)
dim(dataC)
#> [1] 40  4
dataC[1:3, ]
#>   id        v1        v2        v3
#> 1  1  58.06969 -49.90376 -15.30643
#> 2  2  25.96522 -68.58894 -23.10992
#> 3  3 193.71625 124.40186 119.56747
```

Both the microbiome data `dataM` and the covariates data `dataC` contain 40 samples (i.e., 40 rows).

- `dataM` contains 60 taxa with absolute abundances and these are gut microbiome.

- `dataC` contains 3 covariates.

Next we analyze the data to test the associations between the ratio "rawCount18/rawCount11" and all the three variables `"v1"`, `"v2"` and `"v3"` in a multivariate model where all `"v1"`, `"v2"` and `"v3"` are independent variables simultaneously.

```
results <- MZILN(MicrobData = dataM,
                 CovData = dataC,
                  linkIDname = "id",
                  targetTaxa = "rawCount18",
                  refTaxa=c("rawCount11"),
                  allCov=c("v1","v2","v3"),
                  fdrRate=0.15)
#> Data dimensions (after removing missing data if any):
#> 40 samples
#> 60 taxa/OTU/ASV
#> 3 testCov variables in the analysis
#> These are the testCov variables:
#> v1, v2, v3
#> 0 ctrlCov variables in the analysis
#> 0 binary covariates in the analysis
#> 25.71 percent of microbiome sequencing reads are zero
#> Estimation done for the 1th reference taxon: rawCount11 and it took 0.01 minutes
#> The entire analysis took 0.01 minutes
```

The final analysis results are saved in the list `targettaxa_result_list`:

```
results$targettaxa_result_list
#> $rawCount11
#> $rawCount11$v1
#>               estimate      SE est     CI low      CI up adj p-value
#> rawCount18 0.02310369 0.005570789 0.01218495 0.03402244 0.003085391
#>
#> $rawCount11$v2
#>                estimate      SE est       CI low       CI up adj p-value
#> rawCount18 0.002604117 0.003173695 -0.003616325 0.008824558           1
#>
#> $rawCount11$v3
#>                 estimate      SE est      CI low        CI up adj p-value
#> rawCount18 -0.006250528 0.002814316 -0.01176659 -0.000734468   0.8055964
```

The regression coefficients and their 95% confidence intervals are provided. These coefficients correspond to α^k in the model equation, and can be interpreted as the associations between the covariates and log-ratio of `"rawCount18"` over ' "rawCount11" '.

The interpretation for the results is that

- Every unit increase in `"v1"` is associated with approximately 2.3% increase in the abundance ratio of `"rawCount18"` over `"rawCount11"` (while controlling for `"v2"` and `"v3"`); Every unit increase in `"v2"` is associated with approximately 0.26% increase in the abundance ratio of `"rawCount18"` over `"rawCount11"` (while controlling for `"v1"` and `"v3"`), but not statistically significant; Every unit increase in `"v3"` is associated with approximately -0.63% decrease in the abundance ratio of `"rawCount18"` over `"rawCount11"` (while controlling for `"v1"` and `"v2"`), but not statistically significant.

We can also extract all the ratios (with `"rawCount11"` being the denominator taxon) that are significantly associated with any of the covariates as follows:

```
results$sig_results
#> $rawCount11
#> $rawCount11$v1
#>              estimate      SE est     CI low      CI up  adj p-value
#> rawCount18 0.02310369 0.005570789 0.01218495 0.03402244 0.0030853909
#> rawCount36 0.02965713 0.005894748 0.01810342 0.04121084 0.0001341658
#> rawCount41 0.02562728 0.005462932 0.01491993 0.03633463 0.00037377775
```

The interpretation for the results is that

- Every unit increase in `"v1"` is associated with approximately 2.3% increase in the abundance ratio of `"rawCount18"` over `"rawCount11"` (while controlling for `"v2"` and `"v3"`), and it is statistically significant; Every unit increase in `"v1"` is also associated with approximately 3.0% increase in the abundance ratio of `"rawCount36"` over `"rawCount11"` (while controlling for `"v2"` and `"v3"`), and it is statistically significant; Every unit increase in `"v1"` is also associated with approximately 2.6% decrease in the abundance ratio of `"rawCount41"` over `"rawCount11"` (while controlling for `"v2"` and `"v3"`), and it is statistically significant.

All covariates used in the analysis can be extracted using the object `covariatesData`. The covariates data of the first 10 subjects are extracted as follows:

```
results$covariatesData[1:10,]
#>    id         v1         v2          v3
#> 1   1  58.069691  -49.90376  -15.306430
#> 2   2  25.965216  -68.58894  -23.109922
#> 3   3 193.716251  124.40186  119.567468
#> 4   4  72.156467  -98.48536    2.877972
#> 5   5  98.062712   23.55358  -79.893161
#> 6   6  83.094848 -116.95821 -107.641285
#> 7   7   8.217154 -205.64480 -139.958481
#> 8   8  36.169820   58.95708   26.890379
#> 9   9 152.786131  162.60935  138.731954
#> 10 10  41.621790   65.15427   59.974310
```