Package 'SparseMCMM'

April 18, 2019

Title SparseMCMM: Estimating and testing the microbial causal mediation effect with the high-dimensional and compositional microbiome data

Version 0.1.0

Date 2019-04-17

Author Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

Maintainer
Chan Wang <Chan.Wang@nyulangone.org> and Huilin Li <Huilin.Li@nyulangone.org>

Description Sparse Microbial Causal Mediation Model (SparseMCMM) is designed for the high dimensional and compositional microbiome data in a typical three-factor (treatment,microbiome and outcome) causal study design. This model involves linear log-contrast regression and Dirichlet regression to model the causal mediation relationships of treatment, microbiome, covariates and outcomes.Under four sufficient identifiable assumptions, SparseMCMM gives the causal direct effect of treatment and causal mediation effect of the microbiota at both community level and individual, as well as, two tests OME and CME testing the overall mediation effect. Regularization techniques are used to identify signature causal microbes (midiators). This package has four main functions: alpha.estimates (estimating parameters in the linear log-contrast model), beta.estimates (estimating parameters in the Dirichlet regression model), CausalE (calculating causal DE, ME and individual ME estimates) and SparseMCMM (summaring results, calculating statistical significances of OME and CME with permutation procedure, and calculating 95% confidence interval (CI) estimates of component-wise MEs for the causal mediators with bootstrapping procedure). Finally, SparseMCMM can give a clear and sensible causal path among treatment, microbiome composition and outcome. Both simulation studies and real data applications showed the superb performance of SparseMCMM.

Depends Compositional, stats, nloptr, R (>= 3.5.3)

License GPL (>= 2)

Encoding UTF-8

LazyData true

URL https://github.com/chanw0/SparseMCMM

BugReports http://github.com/chanw0/SparseMCMM/issues

R topics documented:

`SparseMCMM-package` *SparseMCMM: Estimating and testing the microbial causal mediation effect with the high-dimensional and compositional microbiome data*

Description

Sparse Microbial Causal Mediation Model (SparseMCMM) is designed for the high dimensional and compositional microbiome data in a treatment-microbiome-outcome causal study. SparseMCMM utilizes the linear log-contrast regression and Dirichlet regression to quantify the causal direct effect of the treatment and the causal mediation effect of the microbiome on the outcome under the counterfactual framework while addressing the compositional structure of microbiome data. Further it implements regularization techniques to handle the high-dimensional microbial mediators and identify the signature causal microbes.

SparseMCMM consists of three steps:

Step 1: Report the estimates of DE, ME and TE respectively under the sufficient causal assumptions mentioned in the Supplementary Materials, Section S1.

Step 2: Report the overall mediation test results: OME test can determine whether the overall mediation effect of microbiome is significant, and CME test can determine whether at least one individual microbe has a significant mediation effect on the outcome.

Step 3: Report the point and 95% confidence interval estimates of ME_j for each signature causal microbe identified by the regularization technique only if CME test is significant at step 2.

Consequentially, SparseMCMM provides a clear and sensible causal path analysis among treatment, compositional microbiome and outcome. Both simulation studies and real data applications showed the superb performance of SparseMCMM.

Details

Package:	SparseMCMM
Type:	Package
Version:	1.0
Date:	2019-04-17
License:	GPL (>= 2)

Author(s)

Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

Maintainer: Chan Wang <Chan.Wang@nyulangone.org>, Huilin Li <Huilin.Li@nyulangone.org>

References

Wang C, Hu J, Blaser M J, Li H (2019). Estimating and testing the microbial causal mediation effect with the high-dimensional and compositional microbiome data.

`alpha.estimates` *Parameter estimation for the linear log-contrast model*

Description

This function gives the numerical estimates of the parameters in model (1) by minimizing the penalized sum of squared residuals measuring the discrepancy between the observed and predicted outcome with sequential quadratic programming (SQP) method. With the heredity constraint, two penalties are incorporated to preform variable selection in high dimensional linear regression with interaction (optimization problem (7)). The tunning parameter is determined by Bayesian information criterion (BIC).

Usage

```
alpha.estimates(Treatment,otu.com,outcome,covariates=NULL,
                    penalty.lambda1=seq(0,1,0.1),penalty.lambda2=seq(0,2,0.2),
                        low.bound=NULL,up.bound=NULL,max.iter=3000)
```

Arguments

`Treatment`	A numeric vector of the binary treatment (takes the value 1 if it is assigned to the treatment group and takes the value 0 if assigned to the control group) with length = sample size (n).
`otu.com`	A n*p numeric matrix containing compositional microbiome data. Each row represents a subject, and each column represents a taxon (given the rank, for example, the genus rank) or an OTU. The row sum equals 1.
`outcome`	A numeric vector of the continuous outcome with length = sample size (n).
`covariates`	An optional matrix containing covariates which need to be adjusted in the model.Each row represents a subject, and each column represents a covariate. Default=NULL.
`penalty.lambda1`	A numeric vector constaing the candidated tunning parameters for the first penalty function.Default=seq(0,1,0.1).
`penalty.lambda2`	A numeric vector constaing the candidated tunning parameters for the second penalty function.Default=seq(0,2,0.2).
`low.bound`	A numeric vector with lower bounds of the controls, Default=NULL, there is no lower bound.
`up.bound`	A numeric vector with upper bounds of the controls, Default=NULL, there is no upper bound.
`max.iter`	An integer value, the maximum number of iterations, Default=3000.

Value

A numeric vector, the parameters'estimates. There is no optimal numerical estimates if return 0.

Author(s)

Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

References

Wang C, Hu J, Blaser M J, Li H (2019). Estimating and testing the microbial causal mediation effect with the high-dimensional and compositional microbiome data.

Examples

```
# require(SparseMCMM)
#
# ########### generation data
# ### Sample size and number of mediators
# sample.num=100
# otu.num=10
#
# ###Treatment
# Treatment=rep(c(0,1),each=sample.num/2)
# ######Two covariates
# covariates=cbind(sample(c(1,0),sample.num,replace = TRUE),rnorm(sample.num))
#
# ### parameters
# beta0=c(0.6, -0.3, 0.8, -1.4, -1.2, -1.4, -1.3, -1.0, -0.2, 0.6)
# betaT=rep(0,otu.num)
# betaT[c(1,3)]=c(0.4,0.2)
# betaX=matrix(0,otu.num,2)
#
# alpha0=0
# alphaT=1
# alphaZ=alphaC=rep(0,otu.num)
# alphaZ[c(1,3)]=c(0.7,-0.7)
# alphaC[c(1,3)]=c(0.15,-0.15)
# alphaX=c(0,0)
#
#
# ############Microbiome data
# library(dirmult)
# X=cbind(rep(1,sample.num),covariates,Treatment) #n*(1+q+p)
# b=cbind(beta0,betaX,betaT) #num.otu*(1+q+p)
# gamma.simu=exp(X %*% t(b)) # n * num.otu
# otu.com=t(apply(gamma.simu,1,rdirichlet,n=1))
#
# ##################Outcome  data
# X=cbind(rep(1,sample.num),Treatment,covariates,log(otu.com),log(otu.com)*Treatment)
# b=c(alpha0,alphaT,alphaX,alphaZ,alphaC)
# outcome=c(b%*%t(X)+rnorm(sample.num,mean = 0, sd =1))
#
# ########
# alpha.estimates(Treatment,otu.com,outcome,covariates)
```

`beta.estimates` *Parameter estimation for the Dirichlet regression model*

Description

This function gives the numerical estimates of the parameters in model (2) by minimizing the penalized log-likilihood of Dirichlet regression model with Newton-Raphson algorithm. L_1 penalty is incorporated to select the microbes (mediators) whose relative abundances are altered by the treatment. The tunning parameter is determined by Bayesian information criterion (BIC).

Usage

```
beta.estimates(Treatment,otu.com,covariates=NULL,penalty.lambda=seq(0,1,0.1),
                        low.bound=NULL,up.bound=NULL,max.iter=3000)
```

Arguments

`Treatment`	A numeric vector of the binary treatment (takes the value 1 if it is assigned to the treatment group and takes the value 0 if assigned to the control group) with length = sample size (n).
`otu.com`	A n*p numeric matrix containing compositional microbiome data. Each row represents a subject, and each column represents a taxon (given the rank, for example, the genus rank) or an OTU. The row sum equals 1.
`covariates`	An optional matrix containing covariates which need to be adjusted in the model.Each row represents a subject, and each column represents a covariate. Default=NULL.
`penalty.lambda`	A numeric vector constaing the candidated tunning parameters.Default=seq(0,1,0.1).
`low.bound`	A numeric vector with lower bounds of the controls, Default=NULL, there is no lower bound.
`up.bound`	A numeric vector with upper bounds of the controls, Default=NULL, there is no upper bound.
`max.iter`	An integer value, the maximum number of iterations, Default=3000.

Value

A numeric vector, the parameters'estimates. There is no optimal numerical estimates if return 0.

Author(s)

Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

References

Wang C, Hu J, Blaser M J, Li H (2019). Estimating and testing the microbial causal mediation effect with the high-dimensional and compositional microbiome data.

Examples

```
# require(SparseMCMM)
#
# ########### generation data
# ### Sample size and number of mediators
# sample.num=100
# otu.num=10
#
# ###Treatment
# Treatment=rep(c(0,1),each=sample.num/2)
# ######Two covariates
# covariates=cbind(sample(c(1,0),sample.num,replace = TRUE),rnorm(sample.num))
#
# ### parameters
# beta0=c(0.6, -0.3, 0.8, -1.4, -1.2, -1.4, -1.3, -1.0, -0.2, 0.6)
# betaT=rep(0,otu.num)
# betaT[c(1,3)]=c(0.4,0.2)
# betaX=matrix(0,otu.num,2)
#
# alpha0=0
# alphaT=1
# alphaZ=alphaC=rep(0,otu.num)
# alphaZ[c(1,3)]=c(0.7,-0.7)
# alphaC[c(1,3)]=c(0.15,-0.15)
# alphaX=c(0,0)
#
#
# ############Microbiome data
# library(dirmult)
# X=cbind(rep(1,sample.num),covariates,Treatment) #n*(1+q+p)
# b=cbind(beta0,betaX,betaT) #num.otu*(1+q+p)
# gamma.simu=exp(X %*% t(b)) # n * num.otu
# otu.com=t(apply(gamma.simu,1,rdirichlet,n=1))
#
# ##################Outcome  data
# X=cbind(rep(1,sample.num),Treatment,covariates,log(otu.com),log(otu.com)*Treatment)
# b=c(alpha0,alphaT,alphaX,alphaZ,alphaC)
# outcome=c(b%*%t(X)+rnorm(sample.num,mean = 0, sd =1))
#
# ########
# beta.estimates(Treatment,otu.com,covariates)
```

`CausalEffect` *Calculating causal DE, ME and component-wise ME estimates based on models (1)-(2)*

Description

This function calculates the causal DE of treatment, ME and component-wise ME of microbiome on the outcome under the counterfactual framework with the estimated coefficients in models (1)-(2).

Usage

```
CausalE(otu.com,alpha.estimation,beta.estimation,covariate.fix)
```

Arguments

otu.com	A n*p numeric matrix containing compositional microbiome data. Each row represents a subject, and each column represents a taxon (given the rank, for example, the genus rank) or an OTU. The row sum equals 1.
alpha.estimation	The numeric vector containing estimated coefficients from alpha.estimates function
beta.estimation	The numeric vector containing estimated coefficients from beta.estimates function
covariate.fix	An optional vector containing the given values for covariates. Default=NULL.

Value

A list which contains two elements:

Esitmated Causal Effects It has three values: direct effect (DE), mediation effect (ME) and total effect (TE)

Estimated component-wise Mediation Effects A numeric vector containing component-wise MEs for all mediators

Author(s)

Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

References

Wang C, Hu J, Blaser M J, Li H (2019). Estimating and testing the microbial causal mediation effect with the high-dimensional and compositional microbiome data.

Examples

```
# require(SparseMCMM)
#
# ########### generation data
# ### Sample size and number of mediators
# sample.num=100
# otu.num=10
#
# ###Treatment
# Treatment=rep(c(0,1),each=sample.num/2)
# ######Two covariates
# covariates=cbind(sample(c(1,0),sample.num,replace = TRUE),rnorm(sample.num))
#
# ### parameters
# beta0=c(0.6, -0.3, 0.8, -1.4, -1.2, -1.4, -1.3, -1.0, -0.2, 0.6)
# betaT=rep(0,otu.num)
# betaT[c(1,3)]=c(0.4,0.2)
# betaX=matrix(0,otu.num,2)
```

```
#
# alpha0=0
# alphaT=1
# alphaZ=alphaC=rep(0,otu.num)
# alphaZ[c(1,3)]=c(0.7,-0.7)
# alphaC[c(1,3)]=c(0.15,-0.15)
# alphaX=c(0,0)
#
#
# ############Microbiome data
# require(dirmult)
# X=cbind(rep(1,sample.num),covariates,Treatment) #n*(1+q+p)
# b=cbind(beta0,betaX,betaT) #num.otu*(1+q+p)
# gamma.simu=exp(X %*% t(b)) # n * num.otu
# otu.com=t(apply(gamma.simu,1,rdirichlet,n=1))
#
# ##################Outcome  data
# X=cbind(rep(1,sample.num),Treatment,covariates,log(otu.com),log(otu.com)*Treatment)
# b=c(alpha0,alphaT,alphaX,alphaZ,alphaC)
# outcome=c(b%*%t(X)+rnorm(sample.num,mean = 0, sd =1))
#
# ########
# beta.estimation=beta.estimates(Treatment,otu.com,covariates)
#
# alpha.estimation=alpha.estimates(Treatment,otu.com,outcome,covariates)
#
# CausalE(otu.com,alpha.estimation,beta.estimation,covariate.fix=c(0,0))
```

`SparseMCMM` *A main function in SparseMCMM framework*

Description

This function gives estimated DE, estimated ME, estimated TE, and estimated component-wise MEs, statistical significances of OME and CME with permutation procedure, as well as, 95% CIs of component-wise MEs for the causal mediators with bootstrapping procedure, based on models (1)-(2).

Usage

```
SparseMCMM(Treatment,otu.com,outcome,covariates=NULL,covariate.fix=NULL,
       dirichlet.penalty=seq(0,1,0.1),lm.penalty1=seq(0,1,0.1),lm.penalty2=seq(0,2,0.2),
          low.bound1=NULL,up.bound1=NULL,low.bound2=NULL,up.bound2=NULL,
           max.iter=3000,num.per=NULL,bootstrap=NULL)
```

Arguments

`Treatment`	A numeric vector of the binary treatment (takes the value 1 if it is assigned to the treatment group and takes the value 0 if assigned to the control group) with length = sample size (n).
`otu.com`	A n*p numeric matrix containing compositional microbiome data. Each row represents a subject, and each column represents a taxon (given the rank, for example, the genus rank) or an OTU. The row sum equals 1.

`outcome` A numeric vector of the continuous outcome with length = sample size (n).

`covariates` An optional matrix containing covariates which need to be adjusted in the model.Each row represents a subject, and each column represents a covariate. Default=NULL.

`covariate.fix` An optional vector containing the given values for covariates. Default=NULL.

`dirichlet.penalty` A numeric vector constaing the candidated tunning parameters used in beta.estimates function .Default=seq(0,1,0.1).

`lm.penalty1` A numeric vector constaing the candidated tunning parameters for the first penalty function in the alpha.estimates.Default=seq(0,1,0.1).

`lm.penalty2` A numeric vector constaing the candidated tunning parameters for the second penalty function in the alpha.estimates.Default=seq(0,2,0.2).

`low.bound1` A numeric vector with lower bounds of the controls used in alpha.estiates function, Default=NULL, there is no lower bound.

`up.bound1` A numeric vector with upper bounds of the controls used in alpha.estiates function, Default=NULL, there is no upper bound.

`low.bound2` A numeric vector with lower bounds of the controls used in beta.estiates function, Default=NULL, there is no lower bound.

`up.bound2` A numeric vector with upper bounds of the controls used in beta.estiates function, Default=NULL, there is no upper bound.

`max.iter` An integer value, the maximum number of iterations, Default=3000.

`num.per` An integer value, the number of permuations. statistical significances of tests TME and CME are calculated based on these permutations. Default=NULL, No calculation for hypothesis test.

`bootstrap` An integer value, the number of bootstrappings. 95% CIs of component-wise MEs are calculated based on these bootstrappings. Default=NULL, no calculation for 95% CIs.

Value

A list which contains four elements:

Esitmated Causal Effects It has three values: direct effect (DE), mediation effect (ME) and total effect (TE)

Estimated component-wise Mediation Effects A numeric vector containing component-wise MEs for all mediators

Test P-values for tests OME and CME if num.per is not NULL, otherwise, no this item.

Compontent-wise ME 95% CI A numeric matrix if bootstrap is not NULL, otherwise, no this item.

Author(s)

Chan Wang, Jiyuan Hu, Martin J. Blaser, Huilin Li.

References

Wang C, Hu J, Blaser M J, Li H (2019). Estimating and testing the microbial causal mediation effect with the high-dimensional and compositional microbiome data.

Examples

```
# ############ generation data
# ### Sample size and number of mediators
# sample.num=100
# otu.num=10
#
# ###Treatment
# Treatment=rep(c(0,1),each=sample.num/2)
# ######Two covariates
# covariates=cbind(sample(c(1,0),sample.num,replace = TRUE),rnorm(sample.num))
#
# ### parameters
# beta0=c(0.6, -0.3, 0.8, -1.4, -1.2, -1.4, -1.3, -1.0, -0.2, 0.6)
# betaT=rep(0,otu.num)
# betaT[c(1,3)]=c(0.4,0.2)
# betaX=matrix(0,otu.num,2)
#
# alpha0=0
# alphaT=1
# alphaZ=alphaC=rep(0,otu.num)
# alphaZ[c(1,3)]=c(0.7,-0.7)
# alphaC[c(1,3)]=c(0.15,-0.15)
# alphaX=c(0,0)
#
#
# #############Microbiome data
# library(dirmult)
# X=cbind(rep(1,sample.num),covariates,Treatment) #n*(1+q+p)
# b=cbind(beta0,betaX,betaT) #num.otu*(1+q+p)
# gamma.simu=exp(X %*% t(b)) # n * num.otu
# otu.com=t(apply(gamma.simu,1,rdirichlet,n=1))
#
# ###################Outcome  data
# X=cbind(rep(1,sample.num),Treatment,covariates,log(otu.com),log(otu.com)*Treatment)
# b=c(alpha0,alphaT,alphaX,alphaZ,alphaC)
# outcome=c(b%*%t(X)+rnorm(sample.num,mean = 0, sd =1))
#
#
#
# ##### SparseMCMM function
# SparseMCMM(Treatment,otu.com,outcome,covariates,covariate.fix=c(0,0),
#           num.per=10,bootstrap=10)
```

Index