**Welcome to cmLIS - Covariate-Modulated Large-Scale Multiple Testing under Dependence**

Jiangzhou Wang, Pengfei Wang\*, Tingting Cui and Wensheng Zhu

December 21, 2020

**Description**

cmLIS is a procedure to perform testing tens of thousands of hypotheses simultaneously that not only takes into account local correlations among tests but also accommodates the covariate information by leveraging a covariate-modulated hidden Markov model (HMM).

**Citations**

1. Sun W., Cai T. Large-scale multiple testing under dependence. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2009;71(2):393–424.

2. Wang J., Wang P., Cui T., Zhu W. Covariate-Modulated Large-Scale Multiple Testing under Dependence.

**Downloads**

* Some core codes of cmLIS procedure are available on GitHub (???????). This repository contains the following files:

rdata\_HMM.R

cmLIS.func.R

cmLIS.DP.func.R

mt.hmm.R.txt

**Illustrations of core R functions**

1. rdata\_HMM

Description:

Generating the observed z-values and the states of hypotheses that are based on covariate-modulated HMM.

Usage:

rdata\_HMM(NUM, pii, A, sigma\_0, prob, alpha, sigma\_1, X)

Arguements:

NUM: the number of multiple hypotheses

pii=(pii[1], pii[2]): the initial state distribution

A=(A[0,0], A[0,1]\\ A[1,0], A[1,1]): the transition matrix

sigma\_0: standard deviation of null density

prob: proportion of mixed components under the alternative hypothesis

alpha=(alpha\_0, alpha\_1,..., alpha\_p)^T: parameter vector with

mu(x)=x\_1\*alpha\_1+...+x\_p\*alpha\_p

sigma\_1: standard deviation of non-null density

X=(x11, ..., x1p \\ ... \\ xn1, ..., xnp): design matrix

Values:

o=z: continuous observed data

s=theta: binary unobserved states

1. cmLIS.func

Description:

For the observed z-values, Fitting the covariate-modulated HMM with given L by using the Bayesian sampling algorithm.

Usage:

cmLIS.func(z\_score, X\_covariates, L, niter)

Arguements:

z\_score=(z\_score[1], ..., z\_score[m]): the z-scores

X\_covariates=(x[1,1], ..., x[1,p] \\ ... \\ x[n,1], ..., x[n,p]): the design matrix

L: the number of mixture components

niter: number of iterations of MCMC algorithm

Values:

cmLIS: covariate-modulated local index of signicance (cmLIS)

BIC.value: BIC value

1. cmLIS.DP.func

Description:

For the observed z-values, fitting the covariate-modulated non-parametric Bayesian hierarchical model with unknown L by using the non-parametric Bayesian algorithm.

Usage:

cmLIS.func (z\_score, X\_covariates, niter, Rep, Gibbs\_Iter)

Arguements:

z\_score=(z\_score[1], ..., z\_score[m]): the z-scores

X\_covariates=(x[1,1], ..., x[1,p] \\ ... \\ x[n,1], ..., x[n,p]): the design matrix

niter, Rep, Gibbs\_Iter: number of iterations of the external and internal loops of the MCMC algorithm

Values:

cmLIS.DP: covariate-modulated local index of signicance (cmLIS)

1. mt.hmm

Description:

Conducting cmLIS procedure when a pre-specified nominal level is given.

Usage:

mt.hmm(cmLIS, q)

Arguements:

cmLIS: cmLIS multiple testing statistics

q: the pre-specified nominal level

Values:

nr: the number of rejected hypotheses

th: the threshold

re: the rejected hypotheses

ac: the accepted hypotheses

de: the decision rule

**Examples:**

## the number of observed z-values

NUM<-5000

## the initial state distribution

pii<-c(0, 1)

## initialize the transition matrx

A<-matrix(c(0.9, 0.1, 0.15, 0.85), 2, 2, byrow=TRUE)

## initialize parameter set of the null and non-null distributions

sigma\_0<-1

sigma\_1<-c(1, 1)

alpha<-matrix(c(c(0.5, -2, 1), c(1, 2, 3)), 2, 3, byrow=TRUE)

L<-nrow(alpha)

p<-ncol(alpha)

X<-matrix(0, nrow=NUM, ncol=p)

X[, 1]<-rep(1, NUM)

X[, 2]<-rnorm(NUM, 0, 1)

X[, 3]<-rnorm(NUM, 0, 1)

prob<-c(0.8, 0.2)

## Generating the observed z-valuse and the states of hypotheses that are based on

## covariate-modulated hidden Markov models.

rdata<-rdata\_HMM(NUM, pii, A, sigma\_0, prob, alpha, sigma\_1, X)

z<-rdata$o

theta<-rdata$s

## Calculating the cmLIS multiple testing statistics by using Bayesian MCMC algorithm

res<-cmLIS.func(z, X, L=2 ,niter=300)

## Calculating the cmLIS.DP multiple testing statistics by using the non-parametric

## Bayesian MCMC algorithm

res.DP <-cmLIS.DP.func(z, X, niter=300, Rep=6, Gibbs\_Iter=4)

## Conducting cmLIS and cmLIS.DP procedures given the pre-specified level is 0.1.

level <- 0.1

res\_cmLIS <- mt.hmm(res$cmLIS, level)$de # reject or accept

res\_cmLIS.DP <- mt.hmm(res.DP$cmLIS.DP, level)$de # reject or accept

## compute FDR, FNR and ATP for cmLIS

N10<-length(which(res\_cmLIS-theta==1))

N01<-length(which(theta-res\_cmLIS==1))

R<-length(which(res\_cmLIS==1))+0.0001

S<- NUM -R

FDR\_cmLIS<-N10/R

FNR\_cmLIS<-N01/S

ATP\_cmLIS<-R-N10

## compute FDR, FNR and ATP for cmLIS.DP

N10<-length(which(res\_cmLIS.DP-theta==1))

N01<-length(which(theta-res\_cmLIS.DP==1))

R<-length(which(res\_cmLIS.DP==1))+0.0001

S<- NUM -R

FDR\_cmLIS.DP<-N10/R

FNR\_cmLIS.DP<-N01/S

ATP\_cmLIS.DP<-R-N10

## In real data analysis, the number of mixture components for the non-null distributions is usually unknown, thus we need to either firstly use the Bayesian information criterion (BIC) to select L or to use the non-parametric Bayesian method.