Package ‘sparseMbClust’

November 11, 2022

Type Package

Title Bayesian Mixture Model for Microbiome Data

Version 1.0

Date 2022-11-11

Author Yushu Shi

Maintainer Yushu Shi <shiyushu2006@gmail.com>

Depends Rcpp (>= 0.12.4),

Description
Use Dirichlet process mixtures/mixture of finite mixtures of Dirichlet Multinomial distributions and Metropolis sampling to achieve two-way clustering for microbiome data, i.e., OTUs contributing to the clustering will be selected during the clustering processes.

License GPL (>= 2)

LinkingTo Rcpp

Imports gplots, mcclust

NeedsCompilation yes

R topics documented:

`children` *A mircobiome dataset with samples from African children and Italian children*

Description

A mircobiome dataset with samples from African children and Italian children. Rows are the OTUs identified, while columns are the observations in the study.

References

De Filippo, Carlotta and Cavalieri, Duccio and Di Paola, Monica and Ramazzotti, Matteo and Poullet, Jean Baptiste and Massart, Sebastien and Collini, Silvia and Pieraccini, Giuseppe and Lionetti, Paolo (2010) Impact of diet in shaping gut microbiota revealed by a comparative study in children from Europe and rural Africa *Proceedings of the National Academy of Sciences*, vol 107, num 33, 14691-14696

`PerformClustering` *Bayesian mixture model for Microbiome data*

Description

Use Dirichlet process mixture or Finite Mixture of Dirichlet Multinomial distributions and Metropolis algorithm to cluster observations and select informative OTUs in Microbiome data.

Usage

```
PerformClustering(otutable, ClusteringMethod, c=NULL, gamma=NULL,
alpha1=1, alpha2=1, nu=1,
w=0.5,beta1=1, beta2=1, a=1, b=1, pargamma=1, lambda=1,
totaliter=20000,burnin=10000,thin=50)
```

Arguments

`otutable` OTU table. Rows correspond to OTUs, while columns correspond to observations. Row names and column names must be given.

`ClusteringMethod` Users need to specify the method “DP" or “MFM".

`c` An integer vector of the observation assignment. If not specified, all the observations will be assigned to one cluster initially.

`gamma` An integer vector of initial discriminating OTUs. If not specified, all the OTUs will be chosen as discriminating initially.

`alpha1` The Dirichlet distribution parameter for all the variables in non-discriminating group.

`alpha2` The Dirichlet distribution parameter for all the variables in discriminating group.

`nu` Initial guess of the concentration parameter.

`w` Initial guess of the proportion of sequences from informative OTUs.

`beta1` Parameter for the Beta prior of the proportion of sequences assigned to discriminating group. The default is 1.

`beta2` Parameter for the Beta prior of the proportion of sequences assigned to discriminating group. The default is 1.

`a` Parameter for the gamma prior of the concentration parameter of DP. The default is 1.

`b` Parameter for the gamma prior of the concentration parameter of DP. The default is 1.

`pargamma` Parameter for the Dirichlet prior in MFM model. The default is 1.

`lambda` Parameter for the Poisson prior in MFM model. The default is 1.

`totaliter` Total number of iterations. The default is 20000.

`burnin` Number of burnin iterations. The default is 10000.

`thin` Number of thinnins. The default is 50.

Details

We introduce a binary vector γ, each entry γ_j indicates whether the corresponding feature j is informative for clustering, 1 for informative, 0 for not. We denote the number of informative features d_γ, and the length of vector γ as d.

Parameter of each observation θ_i has the part shared by all the observations $p_1, p_2, \dots, p_{d-d_\gamma}$ and the part specific to the cluster it belongs to, W_{c_i}, $q_{c_i,1}, q_{c_i,2}, \dots, q_{c_i,d_\gamma}$. Given the parameters belonging to its cluster, each observation is from a multinomial with parameters $(W_{c_i}p_1, W_{c_i}p_2 \dots W_{c_i}p_{d-d_\gamma}, (1-W_{c_i})q_{c_i,1}, (1-W_{c_i})q_{c_i,2}, \dots (1-W_{c_i})q_{c_i,d_\gamma})$, where W_{c_i} controls the proportion of sequences belonging to noise OTUs in cluster c_i. We use conjugate priors by setting $W_{c_i} \sim \mathrm{Beta}(\beta_1, \beta_2)$, $p_1, p_2, \dots, p_{d-d_\gamma} \sim \mathrm{Dirichlet}(\alpha, \alpha, \dots, \alpha)$, $q_{c_i,1}, q_{c_i,2}, \dots, q_{c_i,d_\gamma} \sim \mathrm{Dirichlet}(\alpha, \alpha, \dots, \alpha)$.

For Dirichlet process mixture,

$$\begin{aligned} \mathbf{X_i}|\theta_i &\sim \mathrm{F}(\theta_i), \\ \theta_i|G &\sim G, \\ G &\sim \mathrm{DP}(G_0, \nu), \\ \nu &\sim \mathrm{Ga}(\nu_1, \nu_2). \end{aligned}$$

Here, θ_i is the set of corresponding parameters for observation i. F is the mixing kernel introduced above. G is the distribution realized by the Dirichlet processes, whose expectation is the base distribution G_0.

For mixture of finite mixtures model,

$$\begin{aligned} M &\sim p_m \\ (\pi_1, \dots, \pi_K) &\sim \mathrm{Dirichlet}_m(\eta, \dots, \eta) \\ c_1, \dots, c_n &\sim \pi \\ \theta_1, \dots, \theta_k &\sim G_0 \\ \mathbf{X_i} &\sim F(\theta_{c_i}). \end{aligned}$$

Here K is the underlying number of components from the population. We give $K-1$ a $\mathrm{Poisson}(\lambda)$ distribution. The vector $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ is the probability of belonging to a component for a random sample. $c_1, \dots, c_N$ are the component indicators for the samples. G_0 is the base distribution.

Value

`crec` a cluster assignment record matrix, the columns correspond to observations, while the rows correspond to saved iterations.

`gammarec` a feature selection record matrix, the columns correspond to OTUs, while the rows correspond to saved iterations.

Examples

```
  ## Not run:
library("sparseMbClust")
library("gplots")
library("mcclust")
data(children)
children<-as.matrix(children)
result<-PerformClustering(children,"DP",totaliter=2000,burnin=1000,thin=5)
matcount<-matrix(rep(0,29^2),nrow=29)
```

```
for(iter in 1:200){
  for(i in 1:29){
    for(j in 1:29){
      if(result$crec[iter,i]==result$crec[iter,j]){
        matcount[i,j]<-matcount[i,j]+1
      }
    }
  }
}

matcount<-matcount/200
colnames(matcount)<-colnames(children)
rownames(matcount)<-colnames(children)
cols=seq(0,1,length=100)
heatmap.2(matcount,Rowv = NULL,Colv = NULL,
          trace="none",
          breaks=cols,dendrogram="none")

truth<-gsub("\..*","",colnames(children))
csubsmall<-result$crec
for(iter in 1:nrow(result$crec)){
  csubsmall[iter,]<-as.numeric(factor(result$crec[iter,]))
}

psm2 <- comp.psm(csubsmall)
mbind2 <- minbinder(psm2)
#Rand index
arandi(mbind2$cl, truth)

## End(Not run)
```

Index