Package 'MDM'

November 13, 2013

Type Package

Title Analysis of Chromatin Interactions Using MC_DIST Model, Two-Step Model and One-Step Model.

Version 1.0

Date 2012-07-19

Author Liang Niu

Maintainer Liang Niu <niu.13@osu.edu>

Description A package for the MC_DIST model,Two-Step Model and One-Step Model.

Depends R (>= 2.10.0)

SystemRequirements GNU Scientific Library (GSL)

License GPL-2

LazyLoad yes

R topics documented:

MDM-package *Analysis of Chromatin Interaction Using MC_DIST Model, Two-Step Model and One-Step Model*

Description

MDM provides functions for the analysis of chromatin interactions using MC_DIST model (one sample problem), Two-Step model and One-Step model.

Details

MDM provides a function MCDIST to detect ture chromatin interactions using a dataset obtained from a chromatin looping experiment, such as a Hi-C, ChIA-PET, or modified Hi-C experiment. It also provides trace.MCDIST to draw the MCMC plots for the output of the MCDIST function.

MDM provides a function MDTS to peoform the second step of the Two-Step model for detecting chromatin interactions with different intensities in two samples. This second step is to classify the chromatin interactions (claimed as true interactions from the first step) into three categories: 0 (interactions with higher intensity in sample 1 than that in sample 2), 1 (interactions with the same intensity in sample 1 and sample 2) and 2 (interactions with lower intensity in sample 1 than that in sample 2) It also provides trace.MDTS to draw the MCMC trace plots for the output of the MDTS function.

MDM provides a function MDOS to peoform the One-Step model for detecting chromatin interactions with different intensities in two samples. This is to classify the chromatin interactions into six categories: 0 (true interactions in both sample, with higher intensity in sample 1 than that in sample 2), 1 (true interactions with the same intensity in sample 1 and sample 2), 2 (true interactions in both sample, with lower intensity in sample 1 than that in sample 2), 3 (true interactions in sample 1, but false in sample 2), 4 (true interactions in sample 2, but false in sample 1) and 5 (false interactions in both sample 1 and sample 2). It also provides trace.MDOS to draw the MCMC trace plots for the output of the MDOS function.

Author(s)

Liang Niu <niu.13@osu.edu>

MCDIST *Analysis of Chromatin Interaction Using MC_DIST Model*

Description

Given count, marginal count and distance information, plus the tuning parameters for the MCMC algorithm, the MCDIST function returns the posterior probability of being a true interaction for each observation, acceptance rates of the parameters and the stored samples for the parameters.

Usage

```
MCDIST(count,rmcd,
  init1=list("z"=rep(1,length(count)),
    "lambda1"=rep(model$cut.off+5.0,length(count)),
    "lambda0"=rep(model$cut.off+1.0,length(count)),
    "pi"=rep(0.5,length(count))),
  init2=list("d"=0.1,"sigma2"=1.0,"r0"=1.0,"b"=1.0),
  proposal=list("d"=0.1,"r0"=0.1,"b"=0.1,"lambda1"=0.1,"lambda0"=0.1),
  model=list("cut.off"=1,"UUB"=1000,"VUB"=1000^2,"DUB"=10000),
  running=list("iter"=1000,"thin"=10,"burnin"=200,"seed"=2000),
  monitor=list(
  "obs1"=match(min(abs(count-quantile(count)[2])),
               abs(count-quantile(count)[2])),
  "obs2"=match(min(abs(count-quantile(count)[3])),
               abs(count-quantile(count)[3])),
  "obs3"=match(min(abs(count-quantile(count)[4])),
               abs(count-quantile(count)[4]))))
```

Arguments

`count`	a vector specifying the observation (count) to be analyzed.
`rmcd`	a vector specifying the ratio of marginal count to distance for each observation, with the same length as count.
`init1`	an optional list of four numeric vectors with names "z", "lambda1","lambda0" and "pi". Each vector is of length n and gives the initial values of the corresponding parameter.
`init2`	an optional list of four numeric values with names "d", "sigma2" (sigma square), "r0" and "b". Each value is the initial value of the corresponding parameter.
`proposal`	an optional list of five numeric values with names "d", "r0", "b", "lambda1" and "lambda0". Each value is the proposal standard deviation for the corresponding parameter.
`model`	an optional list of four numeric values with names "cut.off" (cut off value for data), "UUB" (upperbound of uniform distributions for r0 and b), "VUB" (square of the upperbound of uniform distributions for sigma) and "DUB" (upperbound of uniform distribution for d), which are the model setting parameters.
`running`	an optional list of four numeric values with names "iter" (number of iterations), "thin" (thinning number), "burnin" (burnin number) and "seed" (random seed), which are the parameters to set up MCMC.
`monitor`	an optional list of three integers with names "obs1", "obs2" and "obs3", which are the indices of the three monitored observations. The default are the three observations with counts closest to the 25%, 50% and 75% percentile of the data, respectively. In case of multiple matches, the first one in the data set is used.

Value

A list including elements

`result`	a dataframe containg the analysis result. It has two columns with names "posterior.probability" and "interaction.type".

`acceptance` a list of acceptance rates with names "r0", "b", "d", "lambda1.min", "lambda1.max", "lambda0.min" and "lambda0.max". The last four values give the two ranges of acceptance rates for lambda1 and lambda0.

`samples` a list of 13 numeric vectors of stored samples (after thining) with names "d", "sigma2" (sigma square), "r0", "b", "pi_obs1", "lambda1_obs1", "lambda0_obs1", "pi_obs2", "lambda1_obs2", "lambda0_obs2", "pi_obs3", "lambda1_obs3", "lambda0_obs3", the last nine vectors give the samples of pi, lambda1 and lambda0 for the three monitored observations.

`computing.time` a vaule of computing time, in unit of seconds.

Author(s)

Liang Niu

Examples

```
data("simdata")
n=length(simdata$count)
output=MCDIST(count=simdata$count,rmcd=simdata$mc/simdata$distance,
              init1=list("z"=rep(1,n),"lambda1"=rep(30.0,n),
                         "lambda0"=rep(1.0,n),"pi"=rep(0.5,n)),
              init2=list("d"=0.5,"sigma2"=10.0,"r0"=5.0,"b"=1.0),
              proposal=list("d"=0.036,"r0"=0.015,"b"=0.054,
                            "lambda1"=0.6,"lambda0"=1.1))
output$result$interaction.type
```

MDOS *One-Step Model*

Description

The one-step model for detecting chromatin interactions with different intensities in two samples.

Usage

```
MDOS(count.1,count.2,dirichlet.const,
        init1=list("z"=rep(1,length(count.1)),
                   "lambda0"=rep(model$cut.off+10.0,length(count.1)),
                   "lambda1"=rep(model$cut.off+5.0,length(count.1)),
                   "lambda2"=rep(model$cut.off+5.0,length(count.1)),
                   "lambda3"=rep(model$cut.off+10.0,length(count.1)),
                   "lambda4"=rep(model$cut.off+1.0,length(count.1)),
                   "lambda5"=rep(model$cut.off+1.0,length(count.1)),
                   "mu0"=rep(model$cut.off+5.0,length(count.1)),
                   "mu2"=rep(model$cut.off+10.0,length(count.1)),
                   "mu3"=rep(model$cut.off+1.0,length(count.1)),
                   "mu4"=rep(model$cut.off+10.0,length(count.1)),
                   "mu5"=rep(model$cut.off+1.0,length(count.1)),
                   "w0"=rep(1/6,length(count.1)),
                   "w1"=rep(1/6,length(count.1)),
                   "w2"=rep(1/6,length(count.1)),
```

```
                "w3"=rep(1/6,length(count.1)),
                "w4"=rep(1/6,length(count.1)),
                "w5"=rep(1/6,length(count.1))),
          init2=list("d"=1.0,"sigma2"=1.0,"r0"=1.0,"b"=1.0),
          proposal=list("d"=1.0,"r0"=0.1,"b"=0.1,
                        "lambda0"=0.1,"lambda1"=0.1,
                        "lambda2"=0.1,"lambda3"=0.1,
                        "lambda4"=0.1,"lambda5"=0.1,
                        "mu0"=0.1,"mu2"=0.1,"mu3"=0.1,
                        "mu4"=0.1,"mu5"=0.1),
          model=list("cut.off"=1,"UUB"=1000,"VUB"=1000^2,"DUB"=100000),
          running=list("iter"=1000,"thin"=10,"burnin"=200,"seed"=2000),
          monitor=list("obs1"=match(min(abs(count.1-quantile(count.1)[2])),
                                    abs(count.1-quantile(count.1)[2])),
                       "obs2"=match(min(abs(count.1-quantile(count.1)[3])),
                                    abs(count.1-quantile(count.1)[3])),
                       "obs3"=match(min(abs(count.1-quantile(count.1)[4])),
                                    abs(count.1-quantile(count.1)[4]))))
```

Arguments

`count.1`	a vector specifying the observations (counts)of DNA fragment pairs claimed as true pairs (pairs formed by true chromatin interactions)in sample one by MC_DIST model (the model in the first step of the two-step model).
`count.2`	a vector specifying the observations (counts)of DNA fragment pairs claimed as true pairs (pairs formed by true chromatin interactions)in sample two by MC_DIST model (the model in the first step of the two-step model).
`dirichlet.const`	a list of six numeric vectors with names "c0", "c1", "c2", "c3", "c4" and "c5". Each vector is of length n (length of count.1 and count.2) and specifies the corresponding constant factors in the dirichlet distributions for the weights of the six components of the model.
`init1`	an optional list of eighteen numeric vectors with names "z", "lambda0", "lambda1", "lambda2", "lambda3", "lambda4", "lambda5","mu0", "mu2", "mu3", "mu4", "mu5","w0", "w1", "w2","w3", "w4" and "w5". Each vector is of length n and gives the initial values of the corresponding parameter.
`init2`	an optional list of four numeric values with names "d", "sigma2", "r0" and "b". Each value is the initial value of the corresponding parameter.
`proposal`	an optional list of fourteen numeric values with names "d", "r0", "b", "lambda0", "lambda1", "lambda2", "lambda3", "lambda4", "lambda5", "mu0", "mu2", "mu3", "mu4" and "mu5". Each value is the proposal standard deviation for the corresponding parameter.
`model`	an optional list of four numeric values with names "cut.off" (cut off value for data), "UUB" (upperbound of uniform distributions for r0 and b), "VUB" (upperbound of uniform distribution for sigma2) and "DUB" (upperbound of uniform distribution for d), which are the model setting parameters.
`running`	an optional list of four numeric values with names "iter" (number of iterations), "thin" (thinning number), "burnin" (burnin number) and "seed" (random seed), which are the parameters to set up MCMC.
`monitor`	an optional list of three integers with names "obs1", "obs2" and "obs3", which are the indices of the three monitored observations. The default are the three

observations with counts closest to the 25%, 50% and 75% percentile of count.1, respectively. In case of multiple matches, the first one in count.1 is used.

Value

A list including elements

`result` a dataframe containg the analysis result. It has seven columns with names "posterior.probability.0", "posterior.probability.1", "posterior.probability.2", "posterior.probability.3", "posterior.probability.4", "posterior.probability.5" and "interaction.category".

`acceptance` a list of acceptance rates with names "r0", "b", "d", "lambda0.min", "lambda0.max", "lambda1.min", "lambda1.max", "lambda2.min", "lambda2.max","lambda3.min", "lambda3.max", "lambda4.min", "lambda4.max", "lambda5.min", "lambda5.max","mu0.min", "mu0.max","mu2.min", "mu2.max","mu3.min", "mu3.max","mu4.min", "mu4.max","mu5.min", "mu5.max". The last 22 values give the ranges of acceptance rates for lambda0, lambda1, lambda2, lambda3, lambda4, lambda5, mu0, mu2, mu3, mu4 and mu5.

`samples` a list of 55 numeric vectors of stored samples (after thining) with names "d", "sigma2", "r0", "b", "w0_obs1", "w1_obs1", "w2_obs1", "w3_obs1", "w4_obs1", "w5_obs1", "lambda0_obs1", "lambda1_obs1", "lambda2_obs1", "lambda3_obs1", "lambda4_obs1", "lambda5_obs1", "mu0_obs1", "mu2_obs1", "mu3_obs1", "mu4_obs1", "mu5_obs1", "w0_obs2", "w1_obs2", "w2_obs2", "w3_obs2", "w4_obs2", "w5_obs2", "lambda0_obs2", "lambda1_obs2", "lambda2_obs2", "lambda3_obs2", "lambda4_obs2", "lambda5_obs2", "mu0_obs2", "mu2_obs2", "mu3_obs2", "mu4_obs2", "mu5_obs2", "w0_obs3", "w1_obs3", "w2_obs3", "w3_obs3", "w4_obs3", "w5_obs3", "lambda0_obs3", "lambda1_obs3", "lambda2_obs3", "lambda3_obs3", "lambda4_obs3", "lambda5_obs3", "mu0_obs3", "mu2_obs3", "mu3_obs3", "mu4_obs3", "mu5_obs3", the last 51 vectors give the samples of pi, lambda's and mu's for the three monitored observations.

`computing.time` a vaule of computing time, in unit of seconds.

Author(s)

Liang Niu

Examples

```
data("simdata.OS")
n=length(simdata.OS$count.1)
output=MDOS(count.1=simdata.OS$count.1,count.2=simdata.OS$count.2,
            dirichlet.const=list("c0"=simdata.OS$dirichlet.c0,
                                 "c1"=simdata.OS$dirichlet.c1,
                                 "c2"=simdata.OS$dirichlet.c2,
                                 "c3"=simdata.OS$dirichlet.c3,
                                 "c4"=simdata.OS$dirichlet.c4,
                                 "c5"=simdata.OS$dirichlet.c5),
            init1=list("z"=rep(1,n),
                       "lambda0"=rep(20.0,n),"lambda1"=rep(15.0,n),
                       "lambda2"=rep(10.0,n),"lambda3"=rep(15.0,n),
                       "lambda4"=rep(2.0,n),"lambda5"=rep(3.0,n),
                       "mu0"=rep(10.0,n),"mu2"=rep(20.0,n),
                       "mu3"=rep(4.0,n),"mu4"=rep(10.0,n),
```

```
                        "mu5"=rep(2.0,n),
                        "w0"=rep(0.2,n),"w1"=rep(0.1,n),
                        "w2"=rep(0.1,n),"w3"=rep(0.2,n),
                        "w4"=rep(0.1,n),"w5"=rep(0.3,n)),
            init2=list("d"=5000,"sigma2"=35^2,"r0"=6.0,"b"=2.0),
            proposal=list("d"=100.0,"r0"=0.008,"b"=0.025,
                          "lambda0"=1.0,"lambda1"=1.0,"lambda2"=1.0,
                          "lambda3"=1.0,"lambda4"=1.0,"lambda5"=1.0,
                          "mu0"=0.8,"mu2"=0.8,"mu3"=0.8,
                          "mu4"=0.8,"mu5"=0.8))
output$result$interaction.category
```

MDTS *Second Step of Two-Step Model*

Description

The second step of the two-step model for detecting chromatin interactions with different intensities in two samples.

Usage

```
MDTS(count.1,count.2,dirichlet.const,
     init1=list("z"=rep(1,length(count.1)),
                "lambda0"=rep(model$cut.off+10.0,length(count.1)),
                "lambda1"=rep(model$cut.off+5.0,length(count.1)),
                "lambda2"=rep(model$cut.off+5.0,length(count.1)),
                "mu0"=rep(model$cut.off+5.0,length(count.1)),
                "mu2"=rep(model$cut.off+10.0,length(count.1)),
                "w0"=rep(1/3,length(count.1)),
                "w1"=rep(1/3,length(count.1)),
                "w2"=rep(1/3,length(count.1))),
     init2=list("d"=0.1,"r1"=1.0,"a"=1.0),
     proposal=list("d"=0.1,"r1"=0.1,"a"=0.1,"lambda0"=0.1,
                   "lambda1"=0.1,"lambda2"=0.1,
                   "mu0"=0.1,"mu2"=0.1),
     model=list("cut.off"=1,"UUB"=1000,"DUB"=100000),
     running=list("iter"=1000,"thin"=10,"burnin"=200,"seed"=2000),
     monitor=list("obs1"=match(min(abs(count.1-quantile(count.1)[2])),
                               abs(count.1-quantile(count.1)[2])),
                  "obs2"=match(min(abs(count.1-quantile(count.1)[3])),
                               abs(count.1-quantile(count.1)[3])),
                  "obs3"=match(min(abs(count.1-quantile(count.1)[4])),
                               abs(count.1-quantile(count.1)[4]))))
```

Arguments

`count.1`	a vector specifying the observations (counts)of DNA fragment pairs claimed as true pairs (pairs formed by true chromatin interactions)in sample one by MC_DIST model (the model in the first step of the two-step model).

`count.2` a vector specifying the observations (counts)of DNA fragment pairs claimed as true pairs (pairs formed by true chromatin interactions)in sample two by MC_DIST model (the model in the first step of the two-step model).

`dirichlet.const` a list of three numeric vectors with names "c0", "c1" and "c2". Each vector is of length n (length of count.1 and count.2) and specifies the corresponding constant factors in the dirichlet distributions for the weights of the three components of the model.

`init1` an optional list of nine numeric vectors with names "z", "lambda0", "lambda1", "lambda2", "mu0", "mu2", "w0", "w1" and "w2". Each vector is of length n and gives the initial values of the corresponding parameter.

`init2` an optional list of three numeric values with names "d", "r1" and "a". Each value is the initial value of the corresponding parameter.

`proposal` an optional list of eight numeric values with names "d", "r1", "a", "lambda0", "lambda1", "lambda2", "mu0" and "mu2". Each value is the proposal standard deviation for the corresponding parameter.

`model` an optional list of three numeric values with names "cut.off" (cut off value for data), "UUB" (upperbound of uniform distributions for r1 and a) and "DUB" (upperbound of uniform distribution for d), which are the model setting parameters.

`running` an optional list of four numeric values with names "iter" (number of iterations), "thin" (thinning number), "burnin" (burnin number) and "seed" (random seed), which are the parameters to set up MCMC.

`monitor` an optional list of three integers with names "obs1", "obs2" and "obs3", which are the indices of the three monitored observations. The default are the three observations with counts closest to the 25%, 50% and 75% percentile of count.1, respectively. In case of multiple matches, the first one in count.1 is used.

Value

A list including elements

`result` a dataframe containg the analysis result. It has four columns with names "posterior.probability.0", "posterior.probability.1", "posterior.probability.2" and "interaction.category".

`acceptance` a list of acceptance rates with names "r1", "a", "d", "lambda0.min", "lambda0.max", "lambda1.min", "lambda1.max", "lambda2.min", "lambda2.max","mu0.min", "mu0.max","mu2.min" "mu2.max",. The last ten values give the ranges of acceptance rates for lambda0, lambda1, lambda2, mu0 and mu2.

`samples` a list of 27 numeric vectors of stored samples (after thining) with names "d", "r1", "a", "w0_obs1", "w1_obs1", "w2_obs1", "lambda0_obs1", "lambda1_obs1", "lambda2_obs1", "mu0_obs1", "mu2_obs1", "w0_obs2", "w1_obs2", "w2_obs2", "lambda0_obs2", "lambda1_obs2", "lambda2_obs2", "mu0_obs2", "mu2_obs2", "w0_obs3", "w1_obs3", "w2_obs3", "lambda0_obs3", "lambda1_obs3", "lambda2_obs3", "mu0_obs3" and "mu2_obs3", the last 24 vectors give the samples of pi, lambda's and mu's for the three monitored observations.

`computing.time` a vaule of computing time, in unit of seconds.

Author(s)

Liang Niu

Examples

```
data("simdata.TS")
n=length(simdata.TS$count.1)
output=MDTS(count.1=simdata.TS$count.1,count.2=simdata.TS$count.2,
            dirichlet.const=list("c0"=simdata.TS$dirichlet.c0,
                                 "c1"=simdata.TS$dirichlet.c1,
                                 "c2"=simdata.TS$dirichlet.c2),
            init1=list("z"=rep(1,n),"lambda0"=rep(10.0,n),
                       "lambda1"=rep(5.0,n),"lambda2"=rep(5.0,n),
                       "mu0"=rep(5.0,n),"mu2"=rep(10.0,n),
                       "w0"=rep(0.3,n),"w1"=rep(0.4,n),"w2"=rep(0.3,n)),
            init2=list("d"=0.5,"r1"=10.0,"a"=1.0),
            proposal=list("d"=0.031,"r1"=0.01,"a"=0.036,"lambda0"=0.7,
                          "lambda1"=0.7,"lambda2"=0.7,"mu0"=0.7,"mu2"=0.7))
output$result$interaction.category
```

---

simdata *Simulated Dataset for MC_DIST Model*

Description

A simulated dataset with marginal count and distance information.

Usage

```
simdata
```

Format

A dataframe with 8 columns:"frag1.chr","frag1.ID","frag2.chr","frag2.ID","count", "mc", "distance" and "true.interaction". The first 4 columns specify the two fargments of each pair; the "count" is the observed frequency; the "mc" is the marginal count information; "distance" is the distance information; and "true.interaction" indicates the interaction type (1=true, 0=false).

---

simdata.OS *Simulated Dataset for the One-Step Model*

Description

A simulated dataset with information for the prior (dirichlet) distributions for the weights of the six components of the model for the One-Step Model.

Usage

```
simdata.OS
```

Format

A dataframe with 13 columns:"frag1.chr","frag1.ID","frag2.chr","frag2.ID","count.1","count.2", "dirichlet.c0", "dirichlet.c1","dirichlet.c2", "dirichlet.c3", "dirichlet.c4","dirichlet.c5" and "interaction.category". The first 4 columns specify the two fargments of each pair; the "count.1" is the observed frequency in sample 1; the "count.2" is the observed frequency in sample 2; the "dirichlet.c0","dirichlet.c1", "dirichlet.c2", "dirichlet.c3", "dirichlet.c4" and "dirichlet.c5"are the six constant factors in the dirichlet distributions for the weights of the three components of the model; and "interaction.category" indicates the true interaction category.

---

simdata.TS *Simulated Dataset for the Second Step of the Two-Step Model*

Description

A simulated dataset with information for the prior (dirichlet) distributions for the weights of the three components of the model.

Usage

```
simdata.TS
```

Format

A dataframe with 10 columns:"frag1.chr","frag1.ID","frag2.chr","frag2.ID","count.1","count.2", "dirichlet.c0", "dirichlet.c1","dirichlet.c2" and "interaction.category". The first 4 columns specify the two fargments of each pair; the "count.1" is the observed frequency in sample 1; the "count.2" is the observed frequency in sample 2; the "dirichlet.c0","dirichlet.c1" and "dirichlet.c2" are the three constant factors in the dirichlet distributions for the weights of the three components of the model; and "interaction.category" indicates the true interaction category.

---

trace.MCDIST *Trace Plots of MCMC Output for MCDIST Model*

Description

Plot 13 trace plots for the MCMC samples obtained by MCDIST function. The plots are the four plots for d, σ^2, r_0 and b; the three plots for π_i, λ_{1i} and λ_{0i} for the observation 1; the three plots for π_i, λ_{1i} and λ_{0i} for the observation 2; and the three plots for π_i, λ_{1i} and λ_{0i} for the observation 3.

Usage

```
trace.MCDIST(output)
```

Arguments

output	an object obtained by MCDIST function.

Author(s)

Liang Niu

Examples

```
data("simdata")
n=length(simdata$count)
output=MCDIST(count=simdata$count,rmcd=simdata$mc/simdata$distance,
            init1=list("z"=rep(1,n),"lambda1"=rep(30.0,n),
                       "lambda0"=rep(1.0,n),"pi"=rep(0.5,n)),
            init2=list("d"=0.5,"sigma2"=10.0,"r0"=5.0,"b"=1.0),
            proposal=list("d"=0.036,"r0"=0.015,"b"=0.054,
                          "lambda1"=0.6,"lambda0"=1.1))
trace.MCDIST(output)
```

---

trace.MDOS *Trace Plots of MCMC Output for the One-Step Model*

Description

Plot 55 trace plots for the MCMC samples obtained by MDOS function. The plots are the four plots for d, σ^2, r_1 and a; the 18 plots for w_{0i}, w_{1i}, w_{2i}, w_{3i}, w_{4i}, w_{5i} of the three observations; the 33 plots for λ_{0i}, λ_{1i}, λ_{2i}, λ_{3i}, λ_{4i}, λ_{5i}, μ_{0i}, μ_{2i}, μ_{3i}, μ_{4i} and μ_{5i} for the three observations.

Usage

```
trace.MDOS(output)
```

Arguments

output	an object obtained by MDOS function.

Author(s)

Liang Niu

Examples

```
data("simdata.OS")
n=length(simdata.OS$count.1)
output=MDOS(count.1=simdata.OS$count.1,count.2=simdata.OS$count.2,
            dirichlet.const=list("c0"=simdata.OS$dirichlet.c0,
                                 "c1"=simdata.OS$dirichlet.c1,
                                 "c2"=simdata.OS$dirichlet.c2,
                                 "c3"=simdata.OS$dirichlet.c3,
                                 "c4"=simdata.OS$dirichlet.c4,
                                 "c5"=simdata.OS$dirichlet.c5),
            init1=list("z"=rep(1,n),
                       "lambda0"=rep(20.0,n),"lambda1"=rep(15.0,n),
                       "lambda2"=rep(10.0,n), "lambda3"=rep(15.0,n),
                       "lambda4"=rep(2.0,n),"lambda5"=rep(3.0,n),
                       "mu0"=rep(10.0,n),"mu2"=rep(20.0,n),
                       "mu3"=rep(4.0,n),"mu4"=rep(10.0,n),"mu5"=rep(2.0,n),
                       "w0"=rep(0.2,n),"w1"=rep(0.1,n),"w2"=rep(0.1,n),
                       "w3"=rep(0.2,n),"w4"=rep(0.1,n),"w5"=rep(0.3,n)),
            init2=list("d"=5000,"sigma2"=35^2,"r0"=6.0,"b"=2.0),
            proposal=list("d"=100.0,"r0"=0.008,"b"=0.025,"lambda0"=1.0,
```

```
                          "lambda1"=1.0,"lambda2"=1.0,"lambda3"=1.0,
                          "lambda4"=1.0,"lambda5"=1.0,"mu0"=0.8,
                          "mu2"=0.8,"mu3"=0.8,"mu4"=0.8,"mu5"=0.8))
trace.MDOS(output)
```

`trace.MDTS` *Trace Plots of MCMC Output for the Second Step of the Two-Step Model*

Description

Plot 27 trace plots for the MCMC samples obtained by MDTS function. The plots are the three plots for d, r_1 and a; the nine plots for w_{0i}, w_{1i}, w_{2i} of the three observations; the 15 plots for λ_{0i}, λ_{1i} and λ_{2i}, μ_{0i} and μ_{2i} for the three observations.

Usage

```
trace.MDTS(output)
```

Arguments

`output`	an object obtained by MDTS function.

Author(s)

Liang Niu

Examples

```
data("simdata.TS")
n=length(simdata.TS$count.1)
output=MDTS(count.1=simdata.TS$count.1,count.2=simdata.TS$count.2,
            dirichlet.const=list("c0"=simdata.TS$dirichlet.c0,
                                 "c1"=simdata.TS$dirichlet.c1,
                                 "c2"=simdata.TS$dirichlet.c2),
            init1=list("z"=rep(1,n),"lambda0"=rep(10.0,n),
                       "lambda1"=rep(5.0,n),"lambda2"=rep(5.0,n),
                       "mu0"=rep(5.0,n),"mu2"=rep(10.0,n),
                       "w0"=rep(0.3,n),"w1"=rep(0.4,n),"w2"=rep(0.3,n)),
            init2=list("d"=0.5,"r1"=10.0,"a"=1.0),
            proposal=list("d"=0.031,"r1"=0.01,"a"=0.036,
                          "lambda0"=0.7,"lambda1"=0.7,
                          "lambda2"=0.7,"mu0"=0.7,"mu2"=0.7))
trace.MDTS(output)
```

Index