

# Package ‘PLMIX’

December 21, 2016

**Type** Package

**Title** Bayesian Analysis of Finite Mixtures of Plackett-Luce Models for Partial Rankings/Orderings

**Version** 1.0

**Date** 2016-12-21

**Description** Fit finite mixtures of Plackett-Luce models for partial top rankings/orderings within the Bayesian framework. It provides MAP point estimates via EM algorithm and posterior MCMC simulations via Gibbs Sampling. It also fits MLE as a special case of the noninformative Bayesian analysis with vague priors.

**License** GPL (>= 2)

**Imports** Rcpp (>= 0.12.7), abind, foreach, combinat, FSA, rccd

**LinkingTo** Rcpp

**RoxygenNote** 5.0.1

**Suggests** gtools, label.switching, MCMCpack, doParallel

**LazyData** true

**NeedsCompilation** yes

**Author** Cristina Mollica [aut, cre],  
Luca Tardella [aut]

**Maintainer** Cristina Mollica <cristina.mollica@uniroma1.it>

**Repository** CRAN

**Date/Publication** 2016-12-21 17:02:02

## R topics documented:

|                            |   |
|----------------------------|---|
| bicPLMIX . . . . .         | 2 |
| binary_group_ind . . . . . | 3 |
| d_apa . . . . .            | 4 |
| d_carconf . . . . .        | 5 |
| d_dublinwest . . . . .     | 6 |
| d_german . . . . .         | 7 |
| d_nascar . . . . .         | 8 |

|                               |    |
|-------------------------------|----|
| freq_to_unit . . . . .        | 9  |
| gibbsPLMIX . . . . .          | 9  |
| Loglikelihood . . . . .       | 11 |
| make_complete . . . . .       | 12 |
| make_partial . . . . .        | 13 |
| mapPLMIX . . . . .            | 15 |
| mapPLMIX_multistart . . . . . | 17 |
| paired_comparisons . . . . .  | 19 |
| PLMIX . . . . .               | 20 |
| ppcheckPLMIX . . . . .        | 21 |
| ppcheckPLMIX_cond . . . . .   | 24 |
| rank_ord_switch . . . . .     | 26 |
| rank_summaries . . . . .      | 27 |
| rPLMIX . . . . .              | 28 |
| selectPLMIX . . . . .         | 29 |
| unit_to_freq . . . . .        | 31 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>33</b> |
|--------------|-----------|

---

|          |  |
|----------|--|
| bicPLMIX | <i>BIC for a mixture of Plackett-Luce models</i> |
|----------|--|

---

## Description

Compute BIC value for a mixture of Plackett-Luce models fitted to partial orderings.

## Usage

```
bicPLMIX(max_log_lik, ref_known, weights, pi_inv, ref_vary)
```

## Arguments

|             |   |
|-------------|---|
| max_log_lik | Maximized log-likelihood value.   |
| ref_known   | Logical: whether the component-specific reference orders are known (not to be estimated). |
| weights     | Numeric vector of $G$ mixture weights.  |
| pi_inv      | Numeric $N \times K$ data matrix of partial orderings.                                    |
| ref_vary    | Logical: whether the reference orders vary across mixture components.                     |

## Details

The max\_log\_lik argument corresponding to the MLE solution can be obtained from the MAP estimation with flat priors by using the [mapPLMIX](#) function with the default prior setting.

**Value**

A list of two named objects:

|             |                           |
|-------------|---------------------------|
| max_log_lik | The max_log_lik argument. |
| bic         | BIC value.                |

**Author(s)**

Cristina Mollica and Luca Tardella

**References**

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

Mollica, C., Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

**See Also**

[mapPLMIX](#)

**Examples**

```
library(PLMIX)
data(d_carconf)
K <- ncol(d_carconf)
G <- 3
n.starting=2
outputMAP_multistart <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=G,
                                           n_start=n.starting, n_iter=400*G)

outputMAP_multistart$mod$bic
# Equivalently,
bicPLMIX(max_log_lik=max(outputMAP_multistart$mod$log_lik), ref_known=TRUE,
          weights=outputMAP_multistart$mod$W_map, pi_inv=d_carconf, ref_vary=FALSE)$bic
```

---

|                  |                                       |
|------------------|---------------------------------------|
| binary_group_ind | <i>Binary group membership matrix</i> |
|------------------|---------------------------------------|

---

**Description**

Construct the binary group membership matrix from the multinomial classification vector.

**Usage**

```
binary_group_ind(class, G)
```

**Arguments**

class          Numeric vector of class memberships.  
G                Number of possible different classes.

**Value**

Numeric  $\text{length}(\text{class}) \times G$  matrix of binary group memberships.

**Author(s)**

Cristina Mollica and Luca Tardella

**Examples**

```
library(PLMIX)
binary_group_ind(c(3,1,5),6)
```

---

d\_apa

*American Psychological Association Data (partial orderings)*

---

**Description**

The popular American Psychological Association dataset (d\_apa) contains the results of the voting ballots of the 1980 presidential election. A total of  $N = 15449$  voters ranked a maximum of  $K = 5$  candidates, conventionally classified as research psychologists (candidate 1 and 3), clinical psychologists (candidate 4 and 5) and community psychologists (candidate 2). The winner of the election was candidate 3. The dataset is composed of partial top orderings of varying lengths. Missing positions are denoted with zero entries.

**Usage**

```
data(d_apa)
```

**Format**

Data matrix of partial orderings with  $N = 15449$  rows and  $K = 5$  columns, named as:

rank1 Candidate occupying position 1 (most-liked).  
rank2 Candidate occupying position 2.  
rank3 Candidate occupying position 3.  
rank4 Candidate occupying position 4.  
rank6 Candidate occupying position 5 (least-liked).

## References

- Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* published online, DOI: 10.1007/s11336-016-9530-0.
- Diaconis, P. W. (1988). Group representations in probability and statistics. *Lecture Notes-Monograph Series*, pages 94–96.
- Diaconis, P. W. (1987). Spectral analysis for ranked data. Technical Report 282, Dept of Statistics, Stanford University.

## Examples

```
data(d_apa)
head(d_apa)
```

---

|           |  |
|-----------|--|
| d_carconf | <i>Car Configurator Data (partial orderings)</i> |
|-----------|--|

---

## Description

The Car Configurator dataset (d\_carconf) came up from a marketing study aimed at investigating customer preferences toward different car features. A sample of  $N = 435$  customers were asked to construct their car by using an online configurator system and to choose among  $K = 6$  car modules in order of preference. The car features are labeled as: 1 = price, 2 = exterior design, 3 = brand, 4 = technical equipment, 5 = producing country and 6 = interior design. The survey did not require a complete ranking elicitation, therefore the dataset is composed of partial top orderings of varying lengths. Missing positions are denoted with zero entries.

## Usage

```
data(d_carconf)
```

## Format

Data matrix of partial orderings with  $N = 435$  rows and  $K = 6$  columns, named as:

- rank1 Item occupying position 1 (most-liked).
- rank2 Item occupying position 2.
- rank3 Item occupying position 3.
- rank4 Item occupying position 4.
- rank5 Item occupying position 5.
- rank6 Item occupying position 6 (least-liked).

## References

- Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* published online, DOI: 10.1007/s11336-016-9530-0.
- Hatzinger, R., Dittrich, R. (2012). Prefmod: An R package for modeling preferences based on paired comparisons, rankings, or ratings. *Journal of Statistical Software*, **48**(10), pages 1–31.
- Dabic, M., Hatzinger, R. (2009). Zielgruppenadaequade Ablaeufe in Konfigurationssystemen - eine empirische Studie im Automobilmarkt - Partial Rankings. In Hatzinger, R., Dittrich, R. and Salzberger, T., editors, *Praeferenzanalyse mit R: Anwendungen aus Marketing, Behavioural Finance und Human Resource Management*. Wien: Facultas.

## Examples

```
data(d_carconf)
head(d_carconf)
```

---

|              |   |
|--------------|---|
| d_dublinwest | <i>Dublin West Data (partial orderings)</i> |
|--------------|---|

---

## Description

The Dublin West dataset (`d_dublinwest`) contains some results of the voting ballots of the 2002 Irish general election from the Dublin West constituency. The Irish voting system allows voters to rank the candidates in order of preferences, rather than only specify the favorite one. In the Dublin West constituency,  $N = 29988$  voters ranked a maximum of  $K = 9$  candidates, labeled as: 1 = Bonnie R., 2 = Burton J., 3 = Doherty-Ryan D., 4 = Higgins J., 5 = Lenihan B., 6 = McDonald M., 7 = Morrissey T., 8 = Smyth J. and 9 = Terry S.. The dataset is composed of partial top orderings of varying lengths. Missing positions are denoted with zero entries.

## Usage

```
data(d_dublinwest)
```

## Format

Data matrix of partial orderings with  $N = 29988$  rows and  $K = 9$  columns. Each row lists the candidates from the most-liked (`rank1`) to the least-liked (`rank9`) in a given voting ballot.

## Source

The 2002 Dublin West data have been downloaded from <http://www.preflib.org/> Preflib: A Library for Preferences. In that repository, preferences with ties are also included. The original source was publicly available from the Dublin County Returning Officer at the following URL: <https://dublincountyreturningofficer.com/>.

## References

- Gormley, I. C., Murphy, T. B. (2008). Exploring Voting Blocs Within the Irish Electorate: A Mixture Modeling Approach. *Journal of the American Statistical Association*, **103**(483), pages 1014–1027.
- Gormley, I. C., Murphy, T. B. (2009). A grade of membership model for rank data. *Bayesian Analysis*, **4**(2), pages 65–295.
- Mattei, N., Walsh, T. (2013) PrefLib: A Library of Preference Data. *Proceedings of Third International Conference on Algorithmic Decision Theory (ADT 2013)*. Springer, Lecture Notes in Artificial Intelligence, November 13-15, 2013.

## Examples

```
data(d_dublinwest)
head(d_dublinwest)
```

---

d\_german

*German sample Data (complete rankings)*

---

## Description

The German sample dataset (d\_german) is part of a comparative cross-sectional study on political actions and mass participation involving five Western countries. The dataset regards a sample of  $N = 2262$  German respondents who were asked to rank  $K = 4$  political goals in order of desirability, namely: 1 = maintaining order in the nation, 2 = giving people more say in the decisions of government, 3 = fighting rising prices and 4 = protecting freedom of speech. The dataset is composed of complete orderings.

## Usage

```
data(d_german)
```

## Format

Data matrix of complete orderings with  $N = 2262$  rows and  $K = 4$  columns, named as:

- rank1 Political goal occupying position 1 (most desirable).
- rank2 Political goal occupying position 2.
- rank3 Political goal occupying position 3.
- rank4 Political goal occupying position 4 (least desirable).

## References

- Barnes, S. H. et al. (1979). Political action. Mass participation in five Western democracies. London: Sage.
- Croon, M. A. (1989). *Latent class models for the analysis of rankings*. *Advances in psychology*, **60**, pages 99–121.

### Examples

```
data(d_german)
head(d_german)
```

---

|          |                                 |
|----------|---------------------------------|
| d_nascar | NASCAR Data (partial orderings) |
|----------|---------------------------------|

---

### Description

The NASCAR dataset (d\_nascar) collects the results of the 2002 season of stock car racing held in the United States. The 2002 championship consisted of  $N = 36$  races, with 43 car drivers competing in each race. A total of  $K = 87$  drivers participated in the 2002 season, taking part to a different number of races: some of them competed in all the races, some others in only one. The results of the entire 2002 season were collected in the form of top-43 orderings, where the position of the not-competing drivers in each race is assumed lower than 43th, but undetermined. Missing positions are denoted with zero entries.

### Usage

```
data(d_nascar)
```

### Format

Data matrix of partial orderings with  $N = 36$  rows and  $K = 87$  columns. Each row lists the car drivers from the top position (rank1) to the bottom one (rank87) in a given race. Columns from 44th to 87th are filled with zeros, because only 43 drivers competed in each race.

### Source

The NASCAR dataset in the MATLAB format used by Hunter, D. R. (2004) can be downloaded from <http://sites.stat.psu.edu/~dhunter/code/btmatlab/>. At the same link, a .xls file with drivers' names is also available.

### References

- Caron, F., Doucet, A. (2012). Efficient Bayesian inference for Generalized Bradley-Terry models. *J. Comput. Graph. Statist.*, **21**(1), pages 174–196.
- Guiver, J., Snelson, E. (2009). Bayesian inference for Plackett-Luce ranking models. In Bottou, L. and Littman, M., editors, *Proceedings of the 26th International Conference on Machine Learning - ICML 2009*, pages 377–384. Omnipress.
- Hunter, D. R. (2004). MM algorithms for Generalized Bradley-Terry models. *Ann. Statist.*, **32**(1), pages 384–406.

### Examples

```
data(d_nascar)
head(d_nascar)
```



---

|              |  |
|--------------|--|
| freq_to_unit | <i>Individual rankings/orderings from the frequency distribution</i> |
|--------------|--|

---

**Description**

Construct the dataset of individual rankings/orderings from the frequency distribution of the observed sequences.

**Usage**

```
freq_to_unit(freq_distr)
```

**Arguments**

freq\_distr      Numeric matrix of the observed sequences with the corresponding frequencies indicated in the last  $(K + 1)$ -th column.

**Value**

Numeric  $N \times K$  data matrix of observed individual sequences.

**Author(s)**

Cristina Mollica and Luca Tardella

**Examples**

```
library(PLMIX)
library(combinat)
K <- 4
permutation_matrix <- t(matrix(unlist(permn(x=K)),nrow=K,ncol=factorial(K)))
aggregate_data <- cbind(permutation_matrix,sample(factorial(K)))
aggregate_data

freq_to_unit(freq_distr=aggregate_data)
```

---

|            |  |
|------------|--|
| gibbsPLMIX | <i>Gibbs sampling for a Bayesian mixture of Plackett-Luce models</i> |
|------------|--|

---

**Description**

Perform Gibbs sampling simulation for a Bayesian mixture of Plackett-Luce models fitted to partial orderings.

**Usage**

```
gibbsPLMIX(pi_inv, K, G, init = list(z = NULL, p = NULL), n_iter = 1000,
  n_burn = 500, hyper = list(shape0 = matrix(1, nrow = G, ncol = K), rate0 =
  rep(0.001, G), alpha0 = rep(1, G)))
```

**Arguments**

|        |   |
|--------|---|
| pi_inv | Numeric $N \times K$ data matrix of partial orderings.  |
| K      | Number of possible items.   |
| G      | Number of mixture components.   |
| init   | List of named objects with initialization values: p is a numeric $G \times K$ matrix of component-specific support parameters; z is a numeric $N \times G$ matrix of binary component memberships. If not supplied (NULL), initialization values are randomly generated. Default is NULL.       |
| n_iter | Number of total MCMC iterations.  |
| n_burn | Number of initial burn-in samples removed from the MCMC sample.   |
| hyper  | List of named objects with hyperparameter values for prior specification: shape0 is a numeric $G \times K$ matrix of shape hyperparameters; rate0 is a numeric vector of $G$ rate hyperparameters; alpha0 is a numeric vector of $G$ Dirichlet hyperparameters. Default is vague prior setting. |

**Details**

The size  $L$  of the final posterior MCMC sample is equal to `n_iter-n_burn`.

**Value**

A list of named objects:

|          |   |
|----------|---|
| W        | Numeric $L \times G$ matrix with posterior MCMC samples of the mixture weights.                             |
| P        | Numeric $L \times (G * K)$ matrix with posterior MCMC samples of the component-specific support parameters. |
| log_lik  | Numeric vector of posterior log-likelihood values.  |
| deviance | Numeric vector of posterior deviance values ( $-2 * \log\_lik$ ).   |

**Author(s)**

Cristina Mollica and Luca Tardella

**References**

- Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.
- Mollica, C., Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

**Examples**

```

library(PLMIX)
data(d_carconf)
K <- ncol(d_carconf)
G <- 3
mcmc_iterations=30
burnin=10
outputGIBBS <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=G,
                          n_iter=mcmc_iterations, n_burn=burnin)

str(outputGIBBS)
outputGIBBS$P
outputGIBBS$W

```

---

|               |   |
|---------------|---|
| Loglikelihood | <i>Likelihood and Log-likelihood evaluation for a mixture of Plackett-Luce models</i> |
|---------------|---|

---

**Description**

Compute either the log-likelihood or the likelihood of the Plackett-Luce mixture model parameters for a partial ordering dataset.

**Usage**

```

likPLMIX(p, ref_order, weights, pi_inv)

loglikPLMIX(p, ref_order, weights, pi_inv)

```

**Arguments**

|           |   |
|-----------|---|
| p         | Numeric $G \times K$ matrix of component-specific support parameters. |
| ref_order | Numeric $G \times K$ matrix of component-specific reference orders.   |
| weights   | Numeric vector of $G$ mixture weights.                                |
| pi_inv    | Numeric $N \times K$ data matrix of partial orderings.                |

**Value**

Compute either the log-likelihood or the likelihood of the Plackett-Luce mixture model parameters for a partial ordering dataset.

**Author(s)**

Cristina Mollica and Luca Tardella

## References

- Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.
- Mollica, C., Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

## Examples

```
library(PLMIX)
data(d_apa)
K <- ncol(d_apa)
G <- 3
support_par <- matrix(1:(G*K), nrow=G, ncol=K)
weights_par <- c(0.50, 0.25, 0.25)
loglikPLMIX(p=support_par, ref_order=matrix(1:K, nrow=G, ncol=K, byrow=TRUE),
            weights=weights_par, pi_inv=d_apa)
```

---

make\_complete

*Completion of partial rankings/orderings*

---

## Description

Return complete rankings/orderings from partial sequences relying on a random generation of the missing positions.

## Usage

```
make_complete(data, format = c("ordering", "ranking"), nranked = NULL,
             probitems = rep(1, K))
```

## Arguments

|           |  |
|-----------|--|
| data      | Numeric $N \times K$ data matrix of partial sequences.   |
| format    | Character string indicating the format of the data argument.   |
| nranked   | Optional numeric vector of length $N$ with the number of items ranked by each sample unit.   |
| probitems | Numeric vector with the $K$ item-specific probabilities to be employed for the random generation of the missing positions (normalization is not necessary). See Details for further explanation. Default is equal probabilities. |

## Details

The completion of the partial top rankings/orderings is performed according to the Plackett-Luce scheme, that is, with a sampling without replacement of the not-ranked items by using the positive values in the probitems argument as support parameters. The returned completedata matrix has the same format of the input data.

**Value**

A list of two named objects:

|              |   |
|--------------|---|
| completedata | Numeric $N \times K$ data matrix of complete sequences.   |
| nranked      | Numeric vector of length $N$ with the number of items ranked by each sample unit of the input data. |

**Author(s)**

Cristina Mollica and Luca Tardella

**References**

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

**Examples**

```
library(PLMIX)
data(d_dublinwest)
head(d_dublinwest)
K <- ncol(d_dublinwest)
top_item_freq <- tabulate(d_dublinwest[,1], nbins=K)
set.seed(57524)
d_dublinwest_completed <- make_complete(data=d_dublinwest, format="ordering",
                                       probitems=top_item_freq)
head(d_dublinwest_completed$completedata)
```

---

make\_partial

*Censoring of complete rankings/orderings*

---

**Description**

Return partial top rankings/orderings from complete sequences obtained either with user-specified censoring patterns or with a random truncation.

**Usage**

```
make_partial(data, format = c("ordering", "ranking"), nranked = NULL,
             probcensoring = rep(1, K - 1))
```

**Arguments**

|        |  |
|--------|--|
| data   | Numeric $N \times K$ data matrix of complete sequences.      |
| format | Character string indicating the format of the data argument. |

|               |   |
|---------------|---|
| nranked       | Numeric vector of length $N$ with the desired number of items ranked by each sample unit after censoring. If not supplied (NULL), the censoring patterns are randomly generated according to the probabilities in the probcensoring argument.                         |
| probcensoring | Numeric vector of length $(K - 1)$ with the probability of each censoring pattern to be employed for the random truncation of the complete sequences. It works only if nranked argument is NULL. See Details for further explanation. Default is equal probabilities. |

### Details

The censoring of the complete sequences in the data argument can be performed in: (i) a deterministic way, by specifying the number of top positions to be retained for each sample unit in the nranked argument; (ii) a random way, by sequentially specifying the probabilities of the top-1, top-2,...,top- $(K - 1)$  censoring patterns in the probcensoring argument. Recall that a top- $(K - 1)$  sequence corresponds to a complete ordering/ranking. The returned partialdata matrix has the same format of the input data with missing positions denoted with zero entries.

### Value

A list of two named objects:

|             |   |
|-------------|---|
| partialdata | Numeric $N \times K$ data matrix of partial (censored) sequences.                                 |
| nranked     | Numeric vector of length $N$ with the number of items ranked by each sample unit after censoring. |

### Author(s)

Cristina Mollica and Luca Tardella

### References

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

### Examples

```
library(PLMIX)
data(d_german)
N <- nrow(d_german)
head(d_german)
set.seed(57524)
d_german_censored <- make_partial(data=d_german, format="ordering",
                                probcensoring=c(0.3, 0.3, 0.4))
head(d_german_censored$partialdata)
round(table(d_german_censored$nranked)/N, 2)
```

mapPLMIX

*MAP estimation for a Bayesian mixture of Plackett-Luce models***Description**

Perform MAP estimation via EM algorithm for a Bayesian mixture of Plackett-Luce models fitted to partial orderings.

**Usage**

```
mapPLMIX(pi_inv, K, G, init = list(p = NULL, omega = NULL), n_iter = 1000,
  hyper = list(shape0 = matrix(1, nrow = G, ncol = K), rate0 = rep(0, G),
  alpha0 = rep(1, G)), eps = 10-12, centered_start = FALSE,
  plot_objective = TRUE)
```

**Arguments**

|                |   |
|----------------|---|
| pi_inv         | Numeric $N \times K$ data matrix of partial orderings.  |
| K              | Number of possible items.   |
| G              | Number of mixture components.   |
| init           | List of named objects with initialization values: p is a numeric $G \times K$ matrix of component-specific support parameters; omega is a numeric vector of $G$ mixture weights. Default is NULL. In this case initialization values are randomly generated with a uniform distribution. With the optional argument centered_start one can draw from a different distribution enforcing the expectation of the support parameters to coincide with the relative frequency that each item has been ranked top. |
| n_iter         | Maximum number of EM iterations.  |
| hyper          | List of named objects with hyperparameter values for prior specification: shape0 is a numeric $G \times K$ matrix of shape hyperparameters; rate0 is a numeric vector of $G$ rate hyperparameters; alpha0 is a numeric vector of $G$ Dirichlet hyperparameters. Default is noninformative (flat) prior setting.   |
| eps            | Tolerance value for the convergence criterion.  |
| centered_start | Logical: whether a random start whose support parameters and weights are constrained to be centered around the observed relative frequency that each item has been ranked top. Default is FALSE. Ignored when init is not NULL.   |
| plot_objective | Logical: whether the objective function should be plotted. Default is FALSE.  |

**Details**

Under noninformative (flat) prior setting, the EM algorithm for MAP estimation corresponds to the EMM algorithm described by Gormley and Murphy (2006) to perform frequentist inference. Thus, in this case the MAP solution coincides with the MLE. In this case, also the log\_lik and objective output vectors coincide.

**Value**

A list of named objects:

|                |  |
|----------------|--|
| P_map          | Numeric $G \times K$ matrix with the MAP estimates of the component-specific support parameters. |
| Rho_map        | Numeric $G \times K$ matrix with the MAP estimates of the component-specific reference orders.   |
| W_map          | Numeric vector of the $G$ MAP estimates of the mixture weights.                                  |
| z_hat          | Numeric $N \times G$ matrix of estimated posterior component membership probabilities.           |
| classification | Numeric vector of $N$ component memberships based on MAP allocation.                             |
| log_lik        | Numeric vector of log-likelihood values at each iteration.                                       |
| objective      | Numeric vector of objective function values at each iteration.                                   |
| max_objective  | Maximized objective function value.  |
| bic            | BIC value.   |
| conv           | Binary convergence indicator: 1 = convergence has been achieved, 0 = otherwise.                  |

**Author(s)**

Cristina Mollica and Luca Tardella

**References**

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

Mollica, C., Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

**Examples**

```
library(PLMIX)
data(d_carconf)
K <- ncol(d_carconf)
G <- 3
outputMAP <- mapPLMIX(pi_inv=d_carconf, K=K, G=G, n_iter=400*G)
str(outputMAP)
outputMAP$P_map
outputMAP$W_map
```



---

mapPLMIX\_multistart     *MAP estimation for a Bayesian mixture of Plackett-Luce models with multiple starting values*

---

### Description

Perform MAP estimation via EM algorithm with multiple starting values for a Bayesian mixture of Plackett-Luce models fitted to partial orderings.

### Usage

```
mapPLMIX_multistart(pi_inv, K, G, n_start = 1, init = list(list(p = NULL,
  omega = NULL))[rep(1, n_start)], n_iter = 200, hyper = list(shape0 =
  matrix(1, nrow = G, ncol = K), rate0 = rep(0, G), alpha0 = rep(1, G)),
  eps = 10^(-6), plot_objective = FALSE, init_index = 1:n_start,
  parallel = FALSE, centered_start = FALSE)
```

### Arguments

|                |   |
|----------------|---|
| pi_inv         | Numeric $N \times K$ data matrix of partial orderings.  |
| K              | Number of possible items.   |
| G              | Number of mixture components.   |
| n_start        | Number of starting values.  |
| init           | List of $n\_start$ lists of named objects with initialization values: $p$ is a numeric $G \times K$ matrix of component-specific support parameters; $\omega$ is a numeric vector of $G$ mixture weights. Default is NULL. In this case initialization values are randomly generated with a uniform distribution. With the optional argument <code>centered_start</code> one can draw from a different distribution enforcing the expectation of the support parameters to coincide with the relative frequency that each item has been ranked top. |
| n_iter         | Maximum number of EM iterations.  |
| hyper          | List of named objects with hyperparameter values for prior specification: $shape0$ is a numeric $G \times K$ matrix of shape hyperparameters; $rate0$ is a numeric vector of $G$ rate hyperparameters; $alpha0$ is a numeric vector of $G$ Dirichlet hyperparameters. Default is noninformative (flat) prior setting.   |
| eps            | Tolerance value for the convergence criterion.  |
| plot_objective | Logical: whether the objective function should be plotted. Default is FALSE.  |
| init_index     | Numeric vector of the elements of the <code>init</code> argument to be actually launched. Useful to select the most promising starting values identified after a preliminary run. Default is all the starting points in the <code>init</code> argument.   |
| parallel       | Logical: whether parallelization should be used. Default is FALSE.  |
| centered_start | Logical: whether a random start whose support parameters and weights are constrained to be centered around the observed relative frequency that each item has been ranked top. Default is FALSE. Ignored when <code>init</code> is not NULL.  |

**Details**

Under noninformative (flat) prior setting, the EM algorithm for MAP estimation corresponds to the EMM algorithm described by Gormley and Murphy (2006) to perform frequentist inference. Thus, in this case the MAP solution coincides with the MLE. In this case, also the `log_lik` and objective output vectors coincide. The best model in terms of maximized posterior distribution is returned.

**Value**

A list of named objects:

|                            |   |
|----------------------------|---|
| <code>mod</code>           | List of named objects describing the best model in terms of maximized posterior distribution. See output values of the single-run <code>mapPLMIX</code> function for a detailed explanation of the list elements. |
| <code>max_objective</code> | Numeric vector of the maximized objective function values for each initialization.  |
| <code>convergence</code>   | Binary vector of length <code>length(init_index)</code> with convergence indicators for each initialization: 1 = convergence has been achieved, 0 = otherwise.  |

**Author(s)**

Cristina Mollica and Luca Tardella

**References**

- Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.
- Mollica, C., Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

**See Also**

[mapPLMIX](#)

**Examples**

```
library(PLMIX)
data(d_carconf)
K <- ncol(d_carconf)
G <- 3
n.starting=2
outputMAP_multistart <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=G,
                                           n_start=n.starting, n_iter=400*G)

str(outputMAP_multistart)
outputMAP_multistart$mod$P_map
outputMAP_multistart$mod$W_map
```

---

|                    |  |
|--------------------|--|
| paired_comparisons | <i>Paired comparison matrix for a partial ordering/ranking dataset</i> |
|--------------------|--|

---

**Description**

Construct the paired comparison matrix for a partial ordering/ranking dataset.

**Usage**

```
paired_comparisons(data, format = c("ordering", "ranking"), nranked = NULL)
```

**Arguments**

|         |  |
|---------|--|
| data    | Numeric $N \times K$ data matrix of partial sequences.                                     |
| format  | Character string indicating the format of the data argument.                               |
| nranked | Optional numeric vector of length $N$ with the number of items ranked by each sample unit. |

**Value**

Numeric  $K \times K$  paired comparison matrix: the  $(i, i')$ -th entry indicates the number of sample units that preferred item  $i$  to item  $i'$ .

**Author(s)**

Cristina Mollica and Luca Tardella

**References**

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

**Examples**

```
library(PLMIX)
data(d_dublinwest)
paired_comparisons(data=d_dublinwest, format="ordering")
```

**Description**

The PLMIX package for R provides functions to fit and analyze finite mixtures of Plackett-Luce models for partial top rankings/orderings within the Bayesian framework. It provides MAP point estimates via EM algorithm and posterior MCMC simulations via Gibbs Sampling. It also fits MLE as a special case of the noninformative Bayesian analysis with vague priors.

**Details**

The Plackett-Luce model is one of the most popular and frequently applied parametric distributions to analyze partial top rankings/orderings of a finite set of items. The present package allows to account for unobserved sample heterogeneity of partially ranked data with a model-based analysis relying on Bayesian finite mixtures of Plackett-Luce models. The package provides a suite of functions that covers the fundamental phases of a model-based analysis:

**Ranking data manipulation** `rank_ord_switch` switches data format from rankings to orderings and vice-versa; `make_complete` performs random completion of partial orderings/rankings data; `make_partial` performs censoring of complete data; `binary_group_ind` constructs binary group membership matrix from mixture component labels; `freq_to_unit` transforms from frequency distribution representation to individual orderings/rankings; `unit_to_freq` does the vice-versa.

**Ranking data simulation** `rPLMIX` draws a random sample of complete rankings/orderings according to a finite mixture of plackett-Luce model specification.

**Ranking data description** `rank_summaries` provides basic summary statistics of ranking/ordering data, `paired_comparisons` computes the paired comparison frequencies.

**Model estimation** `mapPLMIX` performs MAP estimation via EM algorithm; `mapPLMIX_multistart` does the same with multiple starting values; `gibbsPLMIX` provides Bayesian analysis with MCMC posterior simulation via Gibbs sampling; `likPLMIX` and `loglikPLMIX` provide likelihood and loglikelihood evaluation for a mixture of Plackett-Luce model.

**Model selection** `selectPLMIX` computes alternative Bayesian model selection criteria; `bicPLMIX` computes the BIC value.

**Model assessment** `ppcheckPLMIX` and `ppcheckPLMIX_cond` provide posterior predictive diagnostics.

Data have to be supplied as an object of class `matrix` with missing positions denoted as zero entries. For a more efficient implementation of the methods, partial sequences with a single missing entry should be preliminarily filled in, as they correspond to complete rankings/orderings. In the present setting, ties are not allowed. Some quantities frequently recalled in the manual are the following:

$N$  Sample size.

$K$  Number of possible items.

$G$  Number of mixture components.

$L$  Size of the final posterior MCMC sample (after burn-in phase).

**Author(s)**

Cristina Mollica and Luca Tardella

Maintainer: Cristina Mollica <[cristina.mollica@uniroma1.it](mailto:cristina.mollica@uniroma1.it)>

**References**

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online): <http://dx.doi.org/10.1007/s11336-016-9530-0>

---

 ppcheckPLMIX

---

*Posterior predictive check for mixtures of Plackett-Luce models*


---

**Description**

Perform posterior predictive check to assess the goodness-of-fit of Bayesian mixtures of Plackett-Luce models with a different number of components. See Details for further explanation.

**Usage**

```
ppcheckPLMIX(pi_inv, seq_G, MCMCsampleP, MCMCsampleW, MAPestP = vector(mode =
  "list", length = length(seq_G)), MAPestW = vector(mode = "list", length =
  length(seq_G)), label_switching_adj = FALSE, top1 = TRUE, paired = TRUE,
  adj_post_sample = FALSE, parallel = FALSE)
```

**Arguments**

|                     |   |
|---------------------|---|
| pi_inv              | Numeric $N \times K$ data matrix of partial orderings.  |
| seq_G               | Numeric vector with the number of components of the considered Plackett-Luce mixtures.  |
| MCMCsampleP         | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric $L \times (G * K)$ matrix with the posterior MCMC samples of the component-specific support parameters.  |
| MCMCsampleW         | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric $L \times G$ matrix with the posterior MCMC samples of the mixture weights.  |
| MAPestP             | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric $G \times K$ matrix with the MAP estimates of the component-specific support parameters. If <code>label_switching_adj</code> argument is TRUE, this argument is necessary to be used as pivot in the Pivotal Relabeling Algorithm. |
| MAPestW             | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric vector of $G$ MAP estimates of the mixture weights. If <code>label_switching_adj</code> argument is TRUE, this argument is necessary to be used as pivot in the Pivotal Relabeling Algorithm.                                      |
| label_switching_adj | Logical: whether MCMC samples have to be processed to remove the label switching phenomenon. See Details for further explanation. Default is FALSE.   |

|                 |  |
|-----------------|--|
| top1            | Logical: whether the posterior predictive $p$ -value based on top item frequencies has to be computed. Default is TRUE.          |
| paired          | Logical: whether the posterior predictive $p$ -value based on paired comparison frequencies has to be computed. Default is TRUE. |
| adj_post_sample | Logical: whether MCMC samples adjusted for label switching have to be returned in the output. Default is FALSE.                  |
| parallel        | Logical: whether parallelization should be used. Default is FALSE.   |

### Details

The ppcheckPLMIX function returns two posterior predictive  $p$ -values based on chi squared discrepancy variables involving, respectively, (i) top item frequencies and (ii) paired comparison frequencies. In the presence of partial sequences in the input `pi_inv`, the same missingness patterns of the observed dataset (i.e., the number of items ranked by each sample unit) are reproduced on the replicated datasets from the posterior predictive distribution.

The ppcheckPLMIX function also performs the label switching adjustment of the posterior MCMC samples via Pivotal Relabeling Algorithm, by employing the `pra` function of the `label.switching` package.

### Value

A list of named objects:

|                  |   |
|------------------|---|
| post_pred_pvalue | Numeric $\text{length}(\text{seq}_G) \times 2$ matrix of posterior predictive $p$ -values based on top item and paired comparison frequencies. If <code>top1</code> and/or <code>paired</code> arguments are FALSE, corresponding matrix entries are NA.                                  |
| final_sampleP    | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric $G \times K \times L$ array with the MCMC samples of the component-specific support parameters adjusted for label switching. If <code>adj_post_sample</code> argument is FALSE, list elements are NULL. |
| final_sampleW    | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric $L \times G$ matrix with the MCMC samples of the mixture weights adjusted for label switching. If <code>adj_post_sample</code> argument is FALSE, list elements are NULL.                               |

### Author(s)

Cristina Mollica and Luca Tardella

### References

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

**Examples**

```

library(PLMIX)
data(d_carconf)
K <- ncol(d_carconf)
mcmc_iterations=30
burnin=10
outputGIBBS_1 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=1, n_iter=mcmc_iterations,
                           n_burn=burnin)
outputGIBBS_2 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=2, n_iter=mcmc_iterations,
                           n_burn=burnin)
outputCHECK <- ppcheckPLMIX(pi_inv=d_carconf, seq_G=1:2,
                            MCMCsampleP=list(outputGIBBS_1$P, outputGIBBS_2$P),
                            MCMCsampleW=list(outputGIBBS_1$W, outputGIBBS_2$W))
outputCHECK$post_pred_pvalue

# Adjusting the posterior MCMC samples for label switching
library(PLMIX)
data(d_carconf)
K <- ncol(d_carconf)
n.starting=2
GG <- 1
outputMAP_1 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=GG,
                                  n_start=n.starting, n_iter=400*GG)

GG <- 2
outputMAP_2 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=GG,
                                  n_start=n.starting, n_iter=400*GG)

mcmc_iterations=30
burnin=10
outputGIBBS_1 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=1, n_iter=mcmc_iterations,
                           n_burn=burnin, init=list(p=outputMAP_1$mod$P_map,
                                                     z=binary_group_ind(outputMAP_1$mod$classification,G=1)))
outputGIBBS_2 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=2, n_iter=mcmc_iterations,
                           n_burn=burnin, init=list(p=outputMAP_1$mod$P_map,
                                                     z=binary_group_ind(outputMAP_2$mod$classification,G=2)))

library(doParallel)
library(parallel)
# getDoParWorkers()
# registerDoParallel(detectCores())
registerDoParallel(2)
getDoParWorkers()
outputLABELSWITCHING <- ppcheckPLMIX(pi_inv=d_carconf, seq_G=1:2,
                                     MCMCsampleP=list(outputGIBBS_1$P, outputGIBBS_2$P),
                                     MCMCsampleW=list(outputGIBBS_1$W, outputGIBBS_2$W),
                                     MAPestP=list(outputMAP_1$mod$P_map, outputMAP_2$mod$P_map),
                                     MAPestW=list(outputMAP_1$mod$W_map, outputMAP_2$mod$W_map),
                                     label_switching_adj = TRUE, top1 = FALSE, paired = FALSE,
                                     adj_post_sample = TRUE, parallel = TRUE)

str(outputLABELSWITCHING)

```

---

|                   |  |
|-------------------|--|
| ppcheckPLMIX_cond | <i>Conditional posterior predictive check for mixtures of Plackett-Luce models</i> |
|-------------------|--|

---

### Description

Perform conditional posterior predictive check to assess the goodness-of-fit of Bayesian mixtures of Plackett-Luce models with a different number of components. See Details for further explanation.

### Usage

```
ppcheckPLMIX_cond(pi_inv, seq_G, MCMCsampleP, MCMCsampleW,
  MAPestP = vector(mode = "list", length = length(seq_G)),
  MAPestW = vector(mode = "list", length = length(seq_G)),
  label_switching_adj = FALSE, top1 = TRUE, paired = TRUE,
  adj_post_sample = FALSE, parallel = FALSE)
```

### Arguments

|                     |   |
|---------------------|---|
| pi_inv              | Numeric $N \times K$ data matrix of partial orderings.  |
| seq_G               | Numeric vector with the number of components of the considered Plackett-Luce mixtures.  |
| MCMCsampleP         | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric $L \times (G * K)$ matrix with the posterior MCMC samples of the component-specific support parameters.  |
| MCMCsampleW         | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric $L \times G$ matrix with the posterior MCMC samples of the mixture weights.  |
| MAPestP             | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric $G \times K$ matrix with the MAP estimates of the component-specific support parameters. If <code>label_switching_adj</code> argument is TRUE, this argument is necessary to be used as pivot in the Pivotal Relabeling Algorithm. |
| MAPestW             | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric vector of $G$ MAP estimates of the mixture weights. If <code>label_switching_adj</code> argument is TRUE, this argument is necessary to be used as pivot in the Pivotal Relabeling Algorithm.                                      |
| label_switching_adj | Logical: whether MCMC samples have to be processed to remove the label switching phenomenon. See Details for further explanation. Default is FALSE.   |
| top1                | Logical: whether the posterior predictive $p$ -value based on top item frequencies has to be computed. Default is TRUE.   |
| paired              | Logical: whether the posterior predictive $p$ -value based on paired comparison frequencies has to be computed. Default is TRUE.  |
| adj_post_sample     | Logical: whether MCMC samples adjusted for label switching have to be returned in the output. Default is FALSE.   |
| parallel            | Logical: whether parallelization should be used. Default is FALSE.  |



## Details

The `ppcheckPLMIX_cond` function returns two posterior predictive  $p$ -values based on chi squared discrepancy variables involving, respectively, (i) top item frequencies and (ii) paired comparison frequencies. In the presence of partial sequences in the input `pi_inv`, the same missingness patterns of the observed dataset (i.e., the number of items ranked by each sample unit) are reproduced on the replicated datasets from the posterior predictive distribution.

The `ppcheckPLMIX_cond` function also performs the label switching adjustment of the posterior MCMC samples via Pivotal Relabeling Algorithm, by employing the `pra` function of the `label.switching` package.

## Value

A list of named objects:

|                                    |   |
|------------------------------------|---|
| <code>post_pred_pvalue_cond</code> | Numeric $\text{length}(\text{seq}_G) \times 2$ matrix of posterior predictive $p$ -values based on top item and paired comparison frequencies. If <code>top1</code> and/or <code>paired</code> argument is FALSE, corresponding matrix entries are NA.                                    |
| <code>final_sampleP</code>         | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric $G \times K \times L$ array with the MCMC samples of the component-specific support parameters adjusted for label switching. If <code>adj_post_sample</code> argument is FALSE, list elements are NULL. |
| <code>final_sampleW</code>         | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric $L \times G$ matrix with the MCMC samples of the mixture weights adjusted for label switching. If <code>adj_post_sample</code> argument is FALSE, list elements are NULL.                               |

## Author(s)

Cristina Mollica and Luca Tardella

## References

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

## Examples

```
library(PLMIX)
data(d_carconf)
K <- ncol(d_carconf)
mcmc_iterations=30
burnin=10
outputGIBBS_1 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=1, n_iter=mcmc_iterations,
                           n_burn=burnin)
outputGIBBS_2 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=2, n_iter=mcmc_iterations,
                           n_burn=burnin)
outputCHECKCOND <- ppcheckPLMIX_cond(pi_inv=d_carconf, seq_G=1:2,
                                     MCMCsampleP=list(outputGIBBS_1$P, outputGIBBS_2$P),
                                     MCMCsampleW=list(outputGIBBS_1$W, outputGIBBS_2$W))
outputCHECKCOND$post_pred_pvalue
```

---

|                 |   |
|-----------------|---|
| rank_ord_switch | <i>Switch from orderings to rankings and vice versa</i> |
|-----------------|---|

---

**Description**

Convert the format of the input dataset from orderings to rankings and vice versa.

**Usage**

```
rank_ord_switch(data, format = c("ordering", "ranking"), nranked = NULL)
```

**Arguments**

|         |  |
|---------|--|
| data    | Numeric $N \times K$ data matrix of partial sequences.                                     |
| format  | Character string indicating the format of the data argument.                               |
| nranked | Optional numeric vector of length $N$ with the number of items ranked by each sample unit. |

**Value**

Numeric  $N \times K$  data matrix of partial sequences with inverse format.

**Author(s)**

Cristina Mollica and Luca Tardella

**References**

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

Mollica, C., Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

**Examples**

```
library(PLMIX)
data(d_dublinwest)
head(d_dublinwest)
rank_ord_switch(data=head(d_dublinwest), format="ordering")
```

---

|                |   |
|----------------|---|
| rank_summaries | <i>Descriptive summaries for a partial ordering/ranking dataset</i> |
|----------------|---|

---

### Description

Compute rank summaries and censoring patterns for a partial ordering/ranking dataset.

### Usage

```
rank_summaries(data, format = c("ordering", "ranking"), mean_rank = TRUE,
  marginals = TRUE, pairedcomparisons = TRUE)
```

### Arguments

|                   |   |
|-------------------|---|
| data              | Numeric $N \times K$ data matrix of partial sequences.                |
| format            | Character string indicating the format of the data argument.          |
| mean_rank         | Logical: whether the mean rank vector has to be computed.             |
| marginals         | Logical: whether the marginal rank distributions have to be computed. |
| pairedcomparisons | Logical: whether the paired comparison matrix has to be computed.     |

### Value

A list of named objects:

|                   |  |
|-------------------|--|
| nranked           | Numeric vector of length $N$ with the number of items ranked by each sample unit.  |
| nranked_distr     | Frequency distribution of the nranked vector.  |
| missing_positions | Numeric vector of length $K$ with the number of missing positions for each item.   |
| mean_rank         | Numeric vector of length $K$ with the mean rank of each item.  |
| marginals         | Numeric $K \times K$ matrix of the marginal rank distributions: the $(i, j)$ -th entry indicates the number of units that ranked item $i$ in the $j$ -th position. |
| pairedcomparisons | Numeric $K \times K$ paired comparison matrix: the $(i, i')$ -th entry indicates the number of sample units that preferred item $i$ to item $i'$ .                 |

### Author(s)

Cristina Mollica and Luca Tardella

### References

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

**Examples**

```
library(PLMIX)
data(d_carconf)
rank_summaries(data=d_carconf, format="ordering")
```

rPLMIX

*Random sample from a mixture of Plackett-Luce models***Description**

Draw a random sample from a  $G$ -component mixture of Plackett-Luce models.

**Usage**

```
rPLMIX(n = 1, K, G, p = t(matrix(1/K, nrow = K, ncol = G)),
  ref_order = t(matrix(1:K, nrow = K, ncol = G)), weights = rep(1/G, G),
  rankingoutput = FALSE)
```

**Arguments**

|               |   |
|---------------|---|
| n             | Number of observations to be sampled. Default is 1.   |
| K             | Number of possible items.   |
| G             | Number of mixture components.   |
| p             | Numeric $G \times K$ matrix of component-specific support parameters. Default is equal support parameters (uniform mixture components). |
| ref_order     | Numeric $G \times K$ matrix of component-specific reference orders. Default is forward orders (Plackett-Luce mixture components).       |
| weights       | Numeric vector of $G$ mixture weights. Default is equal weights.  |
| rankingoutput | Logical: whether the final simulated dataset should be expressed in the ranking format. Default is FALSE.                               |

**Details**

Positive values are required for  $p$  and  $weights$  arguments (normalization is not necessary). A permutation of the first  $K$  integers has to be specified for each row of the  $ref\_order$  argument. By changing the default setting of the  $ref\_order$  argument, a sample from a  $G$ -component mixture of Extended Plackett-Luce models is returned.

**Value**

If  $G = 1$ , a numeric  $N \times K$  matrix of simulated complete sequences. If  $G > 1$ , a list of two named objects:

|          |  |
|----------|--|
| comp     | Numeric vector of $n$ component memberships.                 |
| sim_data | Numeric $N \times K$ matrix of simulated complete sequences. |

**Author(s)**

Cristina Mollica and Luca Tardella

**References**

Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.

Mollica, C., Tardella, L. (2014). Epitope profiling via mixture modeling for ranked data. *Statistics in Medicine*, **33**(21), pages 3738–3758, ISSN: 0277-6715, DOI: 10.1002/sim.6224.

**Examples**

```
library(PLMIX)
K <- 6
G <- 3
support_par <- matrix(1:(G*K), nrow=G, ncol=K)
weights_par <- c(0.50, 0.25, 0.25)
set.seed(47201)
simulated_data <- rPLMIX(n=5, K=K, G=G, p=support_par, weights=weights_par)
simulated_data$comp
simulated_data$sim_data
```

---

selectPLMIX

*Bayesian selection criteria for mixtures of Plackett-Luce models*


---

**Description**

Compute Bayesian comparison criteria for mixtures of Plackett-Luce models with a different number of components.

**Usage**

```
selectPLMIX(pi_inv, seq_G, MCMCsampleP = vector(mode = "list", length =
  length(seq_G)), MCMCsampleW = vector(mode = "list", length = length(seq_G)),
  MAPestP, MAPestW, log_lik = vector(mode = "list", length = length(seq_G)),
  deviance, post_summary = c("mean", "median"), parallel = FALSE)
```

**Arguments**

|             |  |
|-------------|--|
| pi_inv      | Numeric $N \times K$ data matrix of partial orderings.   |
| seq_G       | Numeric vector with the number of components of the considered Plackett-Luce mixtures.   |
| MCMCsampleP | List of length $\text{length}(\text{seq\_G})$ , whose generic element is a numeric $L \times (G * K)$ matrix with the posterior MCMC samples of the component-specific support parameters. |

|              |   |
|--------------|---|
| MCMCsampleW  | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric $L \times G$ matrix with the posterior MCMC samples of the mixture weights.   |
| MAPestP      | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric $G \times K$ matrix with the MAP estimates of the component-specific support parameters.  |
| MAPestW      | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric vector of $G$ MAP estimates of the mixture weights.   |
| log_lik      | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric vector of posterior log-likelihood values.  |
| deviance     | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric vector of posterior deviance values.  |
| post_summary | Character string indicating the summary statistic for computing the point estimates of the Plackett-Luce mixture parameters from the posterior MCMC sample. This argument is ignored when MAP estimates are supplied in the MAPestP and MAPestW arguments. Default is "mean". |
| parallel     | Logical: whether parallelization should be used. Default is FALSE.  |

### Value

A list of named objects:

|                                |  |
|--------------------------------|--|
| point_estP                     | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric $G \times K$ matrix with the point estimates of the component-specific support parameters.                                 |
| point_estW                     | List of length $\text{length}(\text{seq}_G)$ , whose generic element is a numeric vector with the $G$ point estimates of the mixture weights.  |
| fitting_measures               | Numeric $\text{length}(\text{seq}_G) \times 2$ matrix with the fitting measures given by the posterior expected deviance $D_{\text{bar}}$ and the deviance $D_{\text{hat}}$ evaluated at the point estimate. |
| effective_number_of_parameters | Numeric $\text{length}(\text{seq}_G) \times 2$ matrix of penalty terms $pD$ and $pV$ .   |
| selection_criteria             | Numeric $\text{length}(\text{seq}_G) \times 6$ matrix of Bayesian model selection criteria: DIC1, DIC2, BPIC1, BPIC2, BICM1 and BICM2. See Details for further explanation.                                  |

### Author(s)

Cristina Mollica and Luca Tardella

### References

- Mollica, C., Tardella, L. (2016). Bayesian Plackett-Luce mixture models for partially ranked data. *Psychometrika* (published online), DOI: 10.1007/s11336-016-9530-0.
- Ando, T. (2007). Bayesian predictive information criterion for the evaluation of hierarchical Bayesian and empirical Bayes models. *Biometrika*, **94**(2), pages 443–458.
- Raftery, A. E, Satagopan, J. M., Newton M. A., Krivitsky, P. N. (2007). BAYESIAN STATISTICS 8. *Proceedings of the eighth Valencia International Meeting 2006*, pages 371–416. Oxford University Press.

Gelman, A, Carlin, J. B., Stern, H. S., Rubin, D. B. (2004). Bayesian data analysis. Chapman & Hall/CRC, Second Edition, ISBN: 1-58488-388-X. New York.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **64**(4), pages 583–639.

## Examples

```
library(PLMIX)
data(d_carconf)
K <- ncol(d_carconf)
n.starting=2

GG <- 1
outputMAP_1 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=GG,
                                n_start=n.starting, n_iter=400*GG)

GG <- 2
outputMAP_2 <- mapPLMIX_multistart(pi_inv=d_carconf, K=K, G=GG,
                                n_start=n.starting, n_iter=400*GG)

mcmc_iterations=30
burnin=10

outputGIBBS_1 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=1, n_iter=mcmc_iterations,
                           n_burn=burnin, init=list(p=outputMAP_1$mod$P_map,
                                                     z=binary_group_ind(outputMAP_1$mod$classification,G=1)))
outputGIBBS_2 <- gibbsPLMIX(pi_inv=d_carconf, K=K, G=2, n_iter=mcmc_iterations,
                           n_burn=burnin, init=list(p=outputMAP_1$mod$P_map,
                                                     z=binary_group_ind(outputMAP_2$mod$classification,G=2)))

outputSELECT <- selectPLMIX(pi_inv=d_carconf, seq_G=1:2,
                            MAPestP=list(outputMAP_1$mod$P_map, outputMAP_2$mod$P_map),
                            MAPestW=list(outputMAP_1$mod$W_map, outputMAP_2$mod$W_map),
                            deviance=list(outputGIBBS_1$deviance, outputGIBBS_2$deviance))
outputSELECT$selection_criteria
```

---

unit\_to\_freq

*Frequency distribution from the individual rankings/orderings.*

---

## Description

Construct the frequency distribution of the observed sequences from the dataset of individual rankings/orderings.

## Usage

```
unit_to_freq(data)
```

**Arguments**

data            Numeric  $N \times K$  data matrix of observed individual sequences.

**Value**

Numeric matrix of the observed sequences with the corresponding frequencies indicated in the last  $(K + 1)$ -th column.

**Author(s)**

Cristina Mollica and Luca Tardella

**Examples**

```
library(PLMIX)
data(d_german)
unit_to_freq(data=d_german)
```



# Index

\*Topic **Ranking data, Plackett-Luce model, Mixture models, Bayesian inference, MAP estimation, MCMC approximation**

PLMIX, 20

\*Topic **datasets**

d\_apa, 4

d\_carconf, 5

d\_dublinwest, 6

d\_german, 7

d\_nascar, 8

\*Topic

PLMIX, 20

bicPLMIX, 2, 20

binary\_group\_ind, 3, 20

d\_apa, 4

d\_carconf, 5

d\_dublinwest, 6

d\_german, 7

d\_nascar, 8

freq\_to\_unit, 9, 20

gibbsPLMIX, 9, 20

label\_switching, 22, 25

likPLMIX, 20

likPLMIX (Loglikelihood), 11

Loglikelihood, 11

loglikPLMIX, 20

loglikPLMIX (Loglikelihood), 11

make\_complete, 12, 20

make\_partial, 13, 20

mapPLMIX, 2, 3, 15, 18, 20

mapPLMIX\_multistart, 17, 20

paired\_comparisons, 19, 20

PLMIX, 20

ppcheckPLMIX, 20, 21

ppcheckPLMIX\_cond, 20, 24

pra, 22, 25

rank\_ord\_switch, 20, 26

rank\_summaries, 20, 27

rPLMIX, 20, 28

selectPLMIX, 20, 29

unit\_to\_freq, 20, 31