

# Package: LPKsample (via r-universe)

September 17, 2024

**Type** Package

**Title** LP Nonparametric High Dimensional K-Sample Comparison

**Version** 2.1

**Date** 2020-05-31

**Author** Subhadeep Mukhopadhyay, Kaijun Wang

**Maintainer** Kaijun Wang <kaijunwang.19@gmail.com>

**Description** LP nonparametric high-dimensional K-sample comparison method that includes (i) confirmatory test, (ii) exploratory analysis, and (iii) options to output a data-driven LP-transformed matrix for classification. The primary reference is Mukhopadhyay, S. and Wang, K. (2020, Biometrika); <[arXiv:1810.01724](https://arxiv.org/abs/1810.01724)>.

**Depends** R (>= 2.10), apcluster, igraph, mclust, LPGraph

**License** GPL-2

**NeedsCompilation** no

**Date/Publication** 2020-06-02 00:40:12 UTC

**Repository** <https://kaijunwang19.r-universe.dev>

**RemoteUrl** <https://github.com/cran/LPKsample>

**RemoteRef** HEAD

**RemoteSha** 7bb53e030fc1a7b982cdb5f59fe2dab0c9569ccb

## Contents

LPKsample-package . . . . .	2
GLP . . . . .	2
leukemia . . . . .	4
LP.comean . . . . .	5
LPT . . . . .	6
W.Gen . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

 LPKsample-package

*LP Nonparametric High Dimensional K-Sample Comparison*


---

### Description

This package performs high dimensional K-sample comparison using graph-based LP nonparametric (GLP) method.

### Author(s)

Mukhopadhyay, S. and Wang, K.

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

### References

Mukhopadhyay, S. and Wang, K. (2020), "A Nonparametric Approach to High-dimensional K-sample Comparison Problem", arXiv:1810.01724.

Mukhopadhyay, S. (2017+), "Unified Statistical Theory of Spectral Graph Analysis".

Mukhopadhyay, S. and Parzen, E. (2014), "LP Approach to Statistical Modeling", arXiv:1405.2601.

---

 GLP

*A function to perform K-sample test using GLP algorithm*


---

### Description

This function performs the GLP multivariate K-sample learning.

### Usage

```
GLP(X,y,m.max=4,components=NULL,alpha=0.05,c.poly=0.5,clust.alg='kmeans',perm=0,
  combine.criterion='pvalue',multiple.comparison=TRUE,
  compress.algorithm=FALSE,nbasis=8, return.LPT=FALSE,return.clust=FALSE)
```

### Arguments

X	A $n$ -by- $d$ matrix of the observations, the observations should be grouped by their respective classes.
y	A length $n$ vector indicating the sample class.
m.max	An integer, maximum order of LP component to investigate, default: 4.
components	A vector specifying which components to test. If provided with any value other than NULL, the test will only examine the components mentioned in this argument, ignoring the m.max settings.
alpha	Numeric, confidence level $\alpha$ , default: 0.05.

<code>c.poly</code>	Numeric, parameter for polynomial kernel, default: 0.5.
<code>perm</code>	Number of permutations for approximating p-value, set to 0 to use asymptotic p-value.
<code>combine.criterion</code>	How to obtain the overall testing result based on the component-wise results; 'pvalue' uses Fisher's method to combine the p-values from each component; 'kernel' computes an overall kernel $W$ based on the significant components and run the LP graph test on the $W$ .
<code>multiple.comparison</code>	Set to TRUE to use adjustment for multiple comparisons when determining which components are significant.
<code>compress.algorithm</code>	Use the smooth compression of Laplacian spectra for testing the null hypothesis. Recommended for large $n$ .
<code>nbasis</code>	Number of bases used for approximation when <code>compress.algorithm=TRUE</code> .
<code>clust.alg</code>	"mclust" or "kmeans"; algorithm used for clustering in graph community detection.
<code>return.LPT</code>	logical, whether or not to return the data driven covariate matrix, default: FALSE.
<code>return.clust</code>	logical, whether or not to return the class labels assigned by graph community detection, default: FALSE.

### Value

A list containing the following items:

<code>GLP</code>	Overall GLP statistics.
<code>pval</code>	Overall P-value.
<code>table</code>	The GLP component table indicating the significance of each component.
<code>components</code>	significant eLP components for the data set.
<code>LPT</code>	(optional) matrix of data driven covariates.
<code>clust</code>	(optional) class labels assigned by graph community detection.

### Author(s)

Mukhopadhyay, S. and Wang, K.

### References

- Mukhopadhyay, S. and Wang, K. (2020), "A Nonparametric Approach to High-dimensional K-sample Comparison Problem", arXiv:1810.01724.
- Mukhopadhyay, S. and Wang, K. (2020). "Towards a unified statistical theory of spectralgraph analysis", arXiv:1901.07090,

## Examples

```
##1.multivariate normal distribution with only mean difference:
##generate data, n1=n2=10, dimension 25
X1<-matrix(rnorm(250,mean=0,sd=1),10,25)
X2<-matrix(rnorm(250,mean=0.5,sd=1),10,25)
y<-c(rep(1,10),rep(2,10))
X<-rbind(X1,X2)
##GLP test:
locdiff.test<-GLP(X,y,m.max=4)

## Not run:
##2.Leukemia data example
data(leukemia)
attach(leukemia)
leukemia.test<-GLP(X,class,components=1:4)
##confirmatory results:
leukemia.test$GLP # overall statistic
#[1] 0.2092378
leukemia.test$pval # overall p-value
#[1] 0.0001038647
##exploratory outputs:
leukemia.test$table # rows as shown in Table 3 of reference
#   component  comp.GLP    pvalue
#[1,]      1 0.209237826 0.0001038647
#[2,]      2 0.022145514 0.2066876581
#[3,]      3 0.002025545 0.7025436476
#[4,]      4 0.033361702 0.1211769396

## End(Not run)
```

---

leukemia

*Leukemia cancer gene expression data*

---

## Description

Gene expression data for two classes: Acute lymphoblastic leukemia (ALL) and Acute myeloid leukemia (AML), over  $n=72$  observations, and  $d=7128$  genes.

## Usage

```
data("leukemia")
```

## Format

A list containing the following items:

**class:** a vector of class labels

**X :** 72 by 7128 matrix, gene expressions for each observation

**Source**

<http://statweb.stanford.edu/~ckirby/brad/LSI/datasets-and-programs/datasets.html>

**Examples**

```
data(leukemia)
```

---

LP.comean

*Function to find LP-comeans*

---

**Description**

The function computes the LP comeans between  $x$  and  $y$ .

**Usage**

```
LP.comean(x, y, perm=0)
```

**Arguments**

$x$	vector, observations of an univariate random variable
$y$	vector, observations of another univariate random variable
perm	Number of permutations for approximating p-value, set to 0 to use asymptotic p-value.

**Value**

A list containing:

LPINFOR	The test statistics based on LP comeans
p.val	Test p-value
LP.matrix	LP comean matrix

**Author(s)**

Mukhopadhyay, S. and Wang, K.

**References**

Mukhopadhyay, S. and Wang, K. (2020), "A Nonparametric Approach to High-dimensional K-sample Comparison Problem", arXiv:1810.01724.

Parzen, E. and Mukhopadhyay, S. (2012) "Modeling, Dependence, Classification, United Statistical Science, Many Cultures".

**Examples**

```
#example: LP-comean for two simple vectors:
y<-c(1,2,3,4,5)
z<-c(0,-1,-1,3,4)
comeanYZ=LP.comean(y,z)
#sum square statistics of LP comean:
comeanYZ$LPINFOR
#p-value:
comeanYZ$p.val
#comean matrix:
comeanYZ$LP.matrix
```

LPT

*eLP Transformation***Description**

Empirical LP Transformation on the data

**Usage**

```
LPT(x, k);
LP.Poly(x, m);
```

**Arguments**

<code>x</code>	A column vector of the data
<code>k</code>	An integer, order of LP component for transformation
<code>m</code>	An integer, maximum order of LP component for transformation

**Details**

Given a vector of data  $x$ , the `LPT(x,k)` function computes the vector of eLP component of order specified by  $k$  for  $x$ . While the `LP.Poly(x,m)` function computes all components up until  $m$ .

**Value**

A vector containing the elements of  $k$ -th order component of the eLP transformation on  $x$  (`LPT`);  
Or a matrix with columns of 1 to  $m$ -th order component of the eLP transformation on  $x$  (`LP.Poly`);

**Author(s)**

Mukhopadhyay, S. and Wang, K.

**References**

Mukhopadhyay, S. and Wang, K. (2020), "A Nonparametric Approach to High-dimensional K-sample Comparison Problem", arXiv:1810.01724.  
Mukhopadhyay, S. and Parzen, E. (2014) "LP Approach to Statistical Modeling", arXiv:1405.2601.

**Examples**

```
##  
x<-runif(10)  
LPT(x,1)
```

---

W.Gen

*Similarity matrix based on eLP basis and polynomial kernel*

---

**Description**

Given data matrix  $X$  and eLP order  $k$ , this function generate the similarity matrix  $W$  for graph analysis.

**Usage**

```
W.Gen(X, k, c.poly = 0.5)
```

**Arguments**

$X$	A $n$ -by- $d$ matrix of the observations
$k$	An integer, order of LP component
$c.poly$	Numeric, parameter for polynomial kernel

**Value**

A  $n$ -by- $n$  similarity matrix generated from  $k$ -th order eLP transformation of  $X$

**Author(s)**

Mukhopadhyay, S. and Wang, K.

**References**

Mukhopadhyay, S. and Wang, K. (2020), "A Nonparametric Approach to High-dimensional K-sample Comparison Problem", arXiv:1810.01724.

**See Also**

[GLP](#)

**Examples**

```
#example: 6 observations on 3 features:  
x<-rbind(matrix(runif(9),3,3),matrix(runif(9)+1,3,3))  
#LP similarity matrix:  
simmat<-W.Gen(x,1)$W  
image(simmat)
```

# Index

- \* **GLP main function**

- GLP, [2](#)

- \* **GLP other functions**

- LP.comean, [5](#)

- LPT, [6](#)

- W.Gen, [7](#)

- \* **datasets**

- leukemia, [4](#)

- \* **package**

- LPKsample-package, [2](#)

GLP, [2](#), [7](#)

graph.clust.test (GLP), [2](#)

leukemia, [4](#)

LP.comean, [5](#)

LP.Poly (LPT), [6](#)

LPKsample (LPKsample-package), [2](#)

LPKsample-package, [2](#)

LPT, [6](#)

W.Gen, [7](#)