

Package: hypergraph.sizing (via r-universe)

June 24, 2026

Title Hypergraph-Based Sizing Function for Generalised Linear Step-Up Method

Version 1.0.0

Description Calculates a sizing function based on the number of independent sets in the rejected hypergraph (Organ, Kenney & Gu, 2026, <doi:10.48550/arXiv.2606.20514>). The sizing function is designed to be used with the 'GLSUP' package.

Depends R (>= 3.2.3)

License GPL-3

Encoding UTF-8

Imports utils,stats

Enhances GLSUP

NeedsCompilation yes

Maintainer Toby Kenney <tkenney@mathstat.dal.ca>

Author Toby Kenney [aut, cre]

Repository <https://cran.r-universe.dev>

Date/Publication 2026-06-24 09:30:07 UTC

RemoteUrl <https://github.com/cran/hypergraph.sizing>

RemoteRef HEAD

RemoteSha 656cc99dba557fe532e3c61457b43c971a7abb29

Contents

complete.hypergraph	2
count.indep.hyper	3
get.p.vals.hypergraph	4
get.threshold.hyper	5
make.test	6
sizing.hyper	7

Index	10
--------------	-----------

complete.hypergraph *Complete Hypergraph*

Description

Constructs a complete hypergraph containing all edges of order up to a given value.

Usage

```
complete.hypergraph(vertices,max.order)
```

Arguments

vertices	The number of vertices
max.order	The largest number of vertices in an edge

Details

Constructs a hypergraph containing all possible edges of order up to the parameter value max.order. Uses the combn function in the utils package, which lists edges in lexicographic order.

Value

The hypergraph, represented as a list with two components:

vertices The number of vertices in the hypergraph

edges A list of edges. Each edge is a vector of vertex numbers.

The hypergraph includes all edges with at most max.order vertices. The edges are in increasing order of size. Within each size, the edges are produced by the combn function in the utils package, which currently produces them in lexicographic order.

Author(s)

Toby Kenney

Examples

```
H<-complete.hypergraph(10,3)
H
```

count.indep.hyper *Count the Number of Independent Sets for a Hypergraph*

Description

Counts the number of independent sets (or equivalently vertex covers) in all initial subhypergraphs of the given hypergraph.

Usage

```
count.indep.hyper(H, samp.size=1000)
count.vc.hyper(H, samp.size=1000)
```

Arguments

H	The hypergraph, represented as a list with two components: vertices The number of vertices in the hypergraph edges A list of edges. Each edge is a vector of vertex numbers. The order of the edges is important, as the function returns an approximate number of edges for each initial subhypergraph.
samp.size	The number of samples used to estimate the number of independent sets for each subhypergraph. Larger values take proportionately longer, but produce more accurate values. Because samples from subgraphs are reused for efficiency, this is not the exact number of samples used.

Details

An independent set for a hypergraph is a set that contains no edges. A vertex cover is a set that has non-empty intersection with every edge. The complement of an independent set is a vertex cover, so there are the same number of independent sets as vertex covers. If the edges of a hypergraph are totally ordered, then we have a sequence of nested hypergraphs. These functions count the number of independent sets for each hypergraph in this nested sequence. Since counting the number of independent sets in a hypergraph is NP-hard, this function uses a random importance sampling scheme that sequentially samples independent sets starting from the empty set and adding one element each time. The samp.size parameter is the number of such sequences generated.

Value

A vector of length the number of edges. The *i*th entry in this vector is the approximate number of independent sets in the subhypergraph consisting of the first *i* edges of the hypergraph.

Author(s)

Toby Kenney

References

Hypergraph Variable Selection with False Discovery Rate Control

Sarah Organ, Toby Kenney, Hong Gu

<doi:10.48550/arXiv.2606.20514>

Examples

```
set.seed(1)
edge.orders<-rpois(15,1.6)+1

H<-list("vertices"=10,lapply(edge.orders,function(s){sample(seq_len(10),s)}))

ind.set<-count.indep.hyper(H)
```

get.p.vals.hypergraph *Calculate p-values for set-wise variable selection from a hypergraph*

Description

Performs Hypothesis Tests for the Null Hypothesis that a Set of Predictors Contains no True Predictor.

Usage

```
get.p.vals.hypergraph(x,y,H,test)
```

Arguments

x	The matrix of predictors.
y	A vector containing the response variable.
H	A hypergraph represented as a list with two components: vertices The number of vertices edges A list of subsets of vertices (represented as vectors of elements in the subset).
test	either one of the character strings "gaussian", "binomial" or "poisson", or else a list with two components: model A function for fitting the model based on a formula typically "lm" for gaussian regression, something based on "glm" for other glm models. p.val A test function based on comparison of the two models typically based on the anova function The make.test function creates a standard test of this form for "gaussian", "binomial" or "poisson" GLMs.

Details

Each edge of the hypergraph corresponds to a collection of subsets of the predictor variables. This function, for each of these sets, computes the p-values for the hypotheses that the set contains no true predictors.

Value

A vector of p-values in the same order as the edges of the hypergraph.

Author(s)

Toby Kenney

Examples

```
set.seed(1)
X<-matrix(rnorm(200),20,10)%*(diag(rep(1,10))-c(0.4,0.4,rep(0,8))%*t(c(0.4,0.4,rep(0,8))))
Y<-rnorm(20)+X%*c(0,2,0,2,2,0,0,0,0,0)
H<-complete.hypergraph(10,2)
get.p.vals.hypergraph(X,Y,H,"gaussian")
```

get.threshold.hyper *Calculate a Threshold to Control FDR for Hypergraph-Based GLSUP*

Description

Calculates a threshold to control gFDR at the specified level when using GLSUP for hypotheses corresponding to the edges of a hypergraph, using a conservative bound on the independent set sizing function.

Usage

```
get.threshold.hyper(H,level,assumption="PRDS")
```

Arguments

H	The hypergraph, represented as a list with two components: vertices The number of vertices in the hypergraph. edges A list of edges. Each edge is a vector of vertex numbers. The order of the edges is important, as the sizing function is calculated for each initial subhypergraph.
level	The desired gFDR control level for GLSUP.
assumption	The assumptions for the dependency between null p-values. Options are "PRDS", "independent" or null. "PRDS" and "independent" give the same threshold, while null gives a more conservative threshold.

Details

The GLSUP rejects as many hypotheses as possible such that the sizing function applied to the rejected hypotheses exceeds a threshold multiplied by the p-value cut-off. This threshold is calculated based on the assumptions about dependency between null p-values, and the desired gFDR control level. This function performs these calculations under common cases.

Value

The threshold for FDR control.

Author(s)

Toby Kenney

References

Hypergraph Variable Selection with False Discovery Rate Control

Sarah Organ, Toby Kenney, Hong Gu

<doi:10.48550/arXiv.2606.20514>

Setwise Hierarchical Variable Selection and the Generalized Linear Step-Up Procedure for False Discovery Rate Control

Sarah Organ, Toby Kenney, Hong Gu <doi:10.48550/arXiv.2603.02160>

Examples

```
set.seed(1)
edge.orders<-rpois(15,1.6)+1

H<-list("vertices"=10,"edges"=lapply(edge.orders,function(s){sample(seq_len(10),s)}))
get.threshold.hyper(H,0.05,assumption="PRDS")
```

make.test

Standard hypothesis tests.

Description

Produces a standard hypothesis test for a given null hypothesis that a certain set of variables are null. These tests can be input to the get.p.values.hypergraph functions

Usage

```
make.test(test.name)
```

Arguments

test.name The name of the test being used. Currently, options are "gaussian", "binomial" and "poisson".

Details

Produces two-sided hypothesis tests for the null hypothesis that a set of predictors are all null, for the given GLM model. The models are fitted by the standard glm package, and p-values are calculated using ANOVA - a likelihood ratio test for "binomial" and "poisson", and an F-test for "gaussian".

Value

A test object, which is a list with two components:

model A function taking a formula as input and returning a fitted model.

p.val A function taking two nested fitted models as input and returning a p-value for the hypothesis that the outer model does not fit the data any better than the inner model.

This test object can be used by the `get.p.vals.hypergraph` function to perform any collection of tests.

Author(s)

Toby Kenney

Examples

```
set.seed(1)
X<-matrix(rnorm(200),20,10)%*(diag(rep(1,10))-c(0.4,0.4,rep(0,8))%*%t(c(0.4,0.4,rep(0,8))))
Y<-rnorm(20)+X%*%c(0,2,0,2,2,0,0,0,0,0)

tst<-make.test("gaussian")

m1<-tst$model(Y~X)
X_sub<-X[,-c(1,2)]
m2<-tst$model(Y~X_sub)

tst$p.val(m1,m2)
```

sizing.hyper

Create a Hypergraph-Based Sizing Function for GLSUP

Description

Creates a hypergraph-based sizing function for the GLSUP package.

Usage

```
sizing.hyper(H,samp.size=1000)
```

Arguments

H	The hypergraph, represented as a list with two components: vertices The number of vertices in the hypergraph edges A list of edges. Each edge is a vector of vertex numbers. The order of the edges is important, as the sizing function is calculated for each initial subhypergraph.
samp.size	The number of samples used to estimate the number of independent sets for each subhypergraph. Larger values take proportionately longer, but produce more accurate values. Because samples from subgraphs are reused for efficiency, this is not the exact number of samples used.

Details

An independent set for a hypergraph is a set that contains no edges. A vertex cover is a set that has non-empty intersection with every edge. The complement of an independent set is a vertex cover, so there are the same number of independent sets as vertex covers. If the edges of a hypergraph are totally ordered, then we have a sequence of nested hypergraphs. These functions count the number of independent sets for each hypergraph in this nested sequence. Since counting the number of independent sets in a hypergraph is NP-hard, this function uses a random importance sampling scheme that sequentially samples independent sets starting from the empty set and adding one element each time. The samp.size parameter is the number of such sequences generated.

For a hypergraph H of rejected hypotheses, the sizing function $|V(H)| - \log_2(|IS(H)|)$, where $V(H)$ is the set of vertices of H and $IS(H)$ is the set of independent sets, is suggested. This function returns a sizing function that calculates this for a nested sequence of hypergraphs from the full hypergraph.

Value

A sizing function - that is a function that takes a permutation as input, then computes the function $|V(H')| - \log_2(|IS(H')|)$, where $V(H')$ is the set of vertices of H' and $IS(H')$ is the set of independent sets of H' , for every H' obtained by taking the first i edges of H . This sizing function can be used with the GLSUP package to perform gFDR-controlled hypothesis testing for this sizing function.

Author(s)

Toby Kenney

References

Hypergraph Variable Selection with False Discovery Rate Control
Sarah Organ, Toby Kenney, Hong Gu
<doi:10.48550/arXiv.2606.20514>

Examples

```
set.seed(1)
edge.orders<-rpois(15,1.6)+1

H<-list("vertices"=10,lapply(edge.orders,function(s){sample(seq_len(10),s)}))
```

```
sizing<-sizing.hyper(H)  
sizing(seq_len(15))
```

Index

`complete.hypergraph`, [2](#)
`count.indep.hyper`, [3](#)
`count.vc.hyper (count.indep.hyper)`, [3](#)

`get.p.vals.hypergraph`, [4](#)
`get.threshold.hyper`, [5](#)

`make.test`, [6](#)

`sizing.hyper`, [7](#)