

# Package ‘DmelSGI’

June 27, 2024

**Type** Package

**Title** Experimental data and documented source code for the paper ``A Map of Directional Genetic Interactions in a Metazoan Cell''

**Version** 1.36.0

**Description** The package contains the experimental data and documented source code of the manuscript ``Fischer et al., A Map of Directional Genetic Interactions in a Metazoan Cell, eLife, 2015, in Press.". The vignette code generates all figures in the paper.

**License** Artistic-2.0

**LazyLoad** true

**Imports** grid, TSP, limma, rhdf5, knitr, abind, gplots, igraph, grDevices, graphics, stats

**Depends** R (>= 3.0)

**VignetteBuilder** knitr

**Suggests** BiocStyle, EBImage, RColorBrewer, RNAinteractMAPK, hwriter, xtable, beeswarm

**NeedsCompilation** no

**biocViews** MicrotitrePlateAssayData, CellCulture, HighthroughputImagingData, Drosophila\_melanogaster\_Data, ExperimentData, HighThroughputImagingData, ReproducibleResearch

**git\_url** <https://git.bioconductor.org/packages/DmelSGI>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** ade0469

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-06-27

**Author** Bernd Fischer [aut],  
Wolfgang Huber [ctb],  
Mike Smith [cre]

**Maintainer** Mike Smith <mike.smith@embl.de>

## Contents

DmelSGI-package . . . . .	3
applyDimensionReduction . . . . .	4
callInteractions . . . . .	5
datamatrix . . . . .	6
DPiM . . . . .	7
estimatePairwiseInteractions . . . . .	8
FBgn2anno . . . . .	9
Features . . . . .	10
fitepistasis . . . . .	11
getBaseDir . . . . .	12
grid.spider . . . . .	12
hrNames . . . . .	13
Interactions . . . . .	14
Intogen . . . . .	15
learnCoComplexFct . . . . .	16
mainEffects . . . . .	17
myHeatmap . . . . .	19
mymedpolish . . . . .	20
orderDim . . . . .	21
orderSpiderAxis . . . . .	22
pimatrix . . . . .	23
plot2Phenotypes . . . . .	24
plotHairballLabels . . . . .	25
qualityControlFeature . . . . .	26
qualityControlGene . . . . .	26
RohnEtAl . . . . .	27
selectByStability . . . . .	29
SelectedClusters . . . . .	30
SelectedClustersComplexes . . . . .	30
SKDdata . . . . .	31
stabilitySelection . . . . .	32
subSampleForStabilitySelection . . . . .	33
subSampleForStabilitySelectionFct . . . . .	34
TID2HUGO . . . . .	35
toMatrix . . . . .	35
toRaster . . . . .	36
trsf . . . . .	37

---

DmelSGI-package      *DmelSGI*.

---

## Description

The package contains the data and the source code to reproduce the results and figures from the paper *title TBC*.

## Details

See vignette("DmelSGI") for details.

## Package content

See vignette("DmelSGI") for more detail on how to obtain the data used for specific figures. In addition this vignette contains the complete analysis and the generation of all figures.

The following **datasets** are provided with this package:

- **Features and quality control**
  - [Features](#) Description of the extracted features.
  - [qualityControlFeature](#) Correlation of features between replicates
  - [qualityControlGene](#) Correlation of interaction profiles between independent dsRNA designs
- **Stability selection**
  - [subSampleForStabilitySelection](#) A subsampled dataset used to select features.
  - [stabilitySelection](#) The features selected by stability.
- **Pairwise interaction matrix**
  - [datamatrix](#) Pairwise perturbation screen data
  - [pimatrix](#) Pairwise genetic interaction scores per experiment (no summary per gene pair)
  - [Interactions](#) Pairwise genetic interaction scores and p-values (summary per gene pairs)
  - [mainEffects](#) Main effects (single knock down effects) estimated from the combinatorial data
  - [SKDdata](#) Single knock down screen

Functions in this package:

- **stability selection**
  - [subSampleForStabilitySelectionFct](#)
  - [stabilitySelection](#)
  - [applyDimensionReduction](#)
- **pairwise interactions**
  - [estimatePairwiseInteractions](#)
  - [mymedpolish](#)
  - [callInteractions](#)

**Author(s)**

Bernd Fischer

Maintainer: Bernd Fischer <bernd.fischer@embl.de>

**References**

T. Horn, T. Sandmann, B. Fischer, W. Huber, M. Boutros. Mapping of Signalling Networks through Synthetic Genetic Interaction Analysis by RNAi. Nature Methods, 2011.

**Examples**

```
data(datamatrix, package="Dm1SGI")
```

---

```
applyDimensionReduction
```

*Subsets the features in a genetic interaction dataset in HDF5 format.*

---

**Description**

Subsets the features in a genetic interaction dataset in HDF5 format. The features are selected by [selectByStability](#) beforehand.

**Usage**

```
applyDimensionReduction(fileMatrixData, fileNew, selected,  
                        verbose = TRUE, overwrite = FALSE)
```

**Arguments**

<code>fileMatrixData</code>	(Input) data matrix in HDF5 format.
<code>fileNew</code>	(Output) data matrix in HDF5 format with a subset of features.
<code>selected</code>	The names of the selected features that will be subsetted.
<code>verbose</code>	Prints more output on screen.
<code>overwrite</code>	If TRUE, overwrite an existing output file ( <code>fileNew</code> ), otherwise stops.

**Value**

NULL is returned. As a side effect the HDF5 file 'fileNew' is created and data matrices with subsetted features will be written to it.

**Author(s)**

Bernd Fischer

**See Also**

[selectByStability](#), [stabilitySelection](#), [Dm1SGI-package](#)

**Examples**

```
print(applyDimensionReduction)
```

---

callInteractions	<i>A statistical test to call pairwise interactions from interaction scores.</i>
------------------	--

---

**Description**

Using the four replicates per gene pair (from the two-by-two dsRNA designs) the null hypothesis the the interaction score is zero is tested by a moderated t-test (R-package limma). The p-values are adjusted by the method of Benjamini-Hochberg. The adjusted p-values and the pairwise interaction scores are stored in an HDF5 file.

**Usage**

```
callInteractions(filePI, fileInteractions, verbose = TRUE, overwrite = FALSE)
```

**Arguments**

filePI	Filename of the HDF5 file with interaction scores. (input)
fileInteractions	Filename of the HDF5 file with adjusted p-values and interaction scores per gene pair and feature (output).
verbose	Prints more output on screen.
overwrite	If TRUE, overwrite an existing output file (fileInteractions), otherwise stops.

**Value**

NULL is returned. As a side effect the HDF5 file 'fileInteractions' is created and adjusted p-values and pairwise interaction scores are saved to this file.

**Author(s)**

Bernd Fischer

**References**

Horn T, Sandmann T, Fischer B, Axelsson E, Huber W, Boutros M (2011). *Mapping of signaling networks through synthetic genetic interaction analysis by RNAi*. Nature Methods 8: 341-346.

**See Also**

[DmelSGI-package](#)

**Examples**

```
print(callInteractions)
```

---

datamatrix

*Pairwise perturbation screen data*


---

## Description

An data array (D) with the feature data of the pairwise perturbation screen. A contains the annotation of genes and features

## Usage

```
data(datamatrix)
```

## Format

The format is:

List of 2

```
$ D : num [1:1293, 1:2, 1:72, 1:2, 1:21] 14.8 14.9 14.9 14.9 14.9 ...
```

```
..- attr(*, "dimnames")=List of 5
```

```
.. ..$ target : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" "CG6833" ...
```

```
.. ..$ targetDesign: chr [1:2] "1" "2"
```

```
.. ..$ query : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
```

```
.. ..$ queryDesign: chr [1:2] "1" "2"
```

```
.. ..$ phenotype : chr [1:21] "4x.count" "4x.ratioMitotic" ...
```

```
$ Anno:List of 5
```

```
..$ target :'data.frame': 1293 obs. of 6 variables:
```

```
.. ..$ TID : chr [1:1293(1d)] "FBgn0002441" "FBgn0038760" ...
```

```
.. ..$ TargetPlate : int [1:1293(1d)] 1 1 1 1 1 1 1 1 1 1 ...
```

```
.. ..$ group : chr [1:1293(1d)] "sample" "sample" "sample" ...
```

```
.. ..$ Symbol : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" ...
```

```
.. ..$ Name : chr [1:1293(1d)] "malignant brain tumor" ...
```

```
..$ targetDesign:'data.frame': 2 obs. of 1 variable:
```

```
.. ..$ design: int [1:2(1d)] 1 2
```

```
..$ query :'data.frame': 72 obs. of 5 variables:
```

```
.. ..$ TID : chr [1:72(1d)] "FBgn0051156" "FBgn0041111" ...
```

```
.. ..$ Batch : int [1:72(1d)] 1 1 1 1 1 1 2 2 2 2 ...
```

```
.. ..$ Symbol : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
```

```
.. ..$ Name : chr [1:72(1d)] "-" "lilliputian" "Smg1" "Axin" ...
```

```
..$ queryDesign :'data.frame': 2 obs. of 1 variable:
```

```
.. ..$ design: int [1:2(1d)] 1 2
```

```
..$ phenotype :'data.frame': 21 obs. of 1 variable:
```

```
.. ..$ phenotype: chr [1:21] "4x.count" "4x.ratioMitotic" ...
```

## Value

An array with the phenotypic data.

**See Also**[DmelSGL-package](#)**Examples**

```
data(datamatrix)
str(datamatrix)
```

DPiM

*Drosophila Protein Interaction Map (DPiM)***Description**

The Drosophila Protein Interaction Map (DPiM) dataset contains of two parts: (1.) The experimentally identified protein interaction partners and (2.) inferred protein complexes.

**Usage**

```
data("DPiM")
```

**Format**

The format is: List of 2 \$ interactions: 'data.frame': 10969 obs. of 5 variables: ..\$ Interactor\_1 : chr [1:10969] "FBgn0036918" "FBgn0031143" "FBgn0030086" "FBgn0015019" ... ..\$ Interactor\_2 : chr [1:10969] "FBgn0037893" "FBgn0035102" "FBgn0033342" "FBgn0037632" ... ..\$ HGSCore : num [1:10969] 742 737 733 730 726 ... ..\$ Evidence.in.DroID: chr [1:10969] "human\_orthology; yeast\_orthology" "" "human\_orthology" "human\_orthology; yeast\_orthology" ... ..\$ evidence : logi [1:10969] TRUE FALSE TRUE TRUE TRUE TRUE ... \$ complexes :List of 556 ..\$ : chr [1:40] "FBgn0002284" "FBgn0002787" "FBgn0004066" "FBgn0010590" ... ..\$ : chr [1:43] "FBgn0000212" "FBgn0000499" "FBgn0001276" "FBgn0001324" ... ..\$ : chr [1:28] "FBgn0003660" "FBgn0011288" "FBgn0011708" "FBgn0013343" ... ..\$ : chr [1:11] "FBgn0022023" "FBgn0025582" "FBgn0029629" "FBgn0033902" ... .. [list output truncated]

**Value**

The mass spec data and the inferred protein complexes of the DPiM dataset.

**References**

Guruharsha, K. G., et al. "A protein complex network of Drosophila melanogaster." Cell 147.3 (2011): 690-703.

**See Also**[DmelSGL-package](#)**Examples**

```
data(DPiM)
```

---

estimatePairwiseInteractions

*Estimates pairwise interaction scores.*

---

### Description

Estimates pairwise interaction scores for a large, multi-dimensional combinatorial screen.

### Usage

```
estimatePairwiseInteractions(fileMatrixData,  
                             filePI,  
                             verbose = TRUE,  
                             overwrite = FALSE,  
                             useSKD = TRUE)
```

### Arguments

fileMatrixData	Filename of HDF5 file containing the combinatorial screening data (input).
filePI	Filename of HDF5 file to save the pairwise interaction matrix (output).
verbose	Prints more output on screen.
overwrite	If TRUE, overwrite an existing output file (filePI), otherwise stops.
useSKD	If TRUE, the negative controls are used to estimate the overall effect.

### Details

Estimates the pairwise interaction scores for each feature and each batch by calling the function [mymedpolish](#).

### Value

Returns TRUE. As a sideeffect, the array with interaction scores is stored in the HDF5 file 'filePI'.

### Author(s)

Bernd Fischer

### References

Horn T, Sandmann T, Fischer B, Axelsson E, Huber W, Boutros M (2011). *Mapping of signaling networks through synthetic genetic interaction analysis by RNAi*. Nature Methods 8: 341-346.

### See Also

[mymedpolish](#), [DmelSGI-package](#)



**Examples**

```
print(estimatePairwiseInteractions)
```

---

FBgn2anno

*GO annotation of flybase genes*

---

**Description**

GO annotation of the 1367 genes targeted in this screen.

**Usage**

```
data(FBgn2anno)
```

**Format**

A data frame with 118863 observations on the following 4 variables.

source a character vector

gene\_id a character vector

Category a character vector

Name a character vector

**Value**

GO annotation.

**See Also**

[DmelSGI-package](#)

**Examples**

```
data(FBgn2anno)
```

Features

*Description of the extracted features.***Description**

The data.frame describing all features extracted from the images. Beside the name of the feature given as the row.name of the data.frame, 7 columns describe each extracted feature:

- **mag** is the magnification of the image that is used (either 4x or 10x)
- **summary** Features are extracted for each single cell. Features are summarized by mean, standard deviation, quantiles, and histograms.
- **mask** The segmentation mask used to extract the features, e.g. the nuclei, the nuclei in the pH3 channel, or the cell body extracted from the a-tubulin channel
- **channel** The channel used to extract the feature
- **set** The feature category
- **type** The type of feature (area, intensity, ...)
- **param** Additional parameters of the feature extraction (quantiles, histogram bins, ...)

**Usage**

```
data(Features)
```

**Format**

The head of the data.frame is:

```
mag summary mask channel set type param
4x.count "4x" "nrNuclei" "nucleus" "DAPI" "M" "" ""
4x.countpH3 "4x" "nrNuclei" "mitoticNuclei" "pH3" "M" "" ""
4x.isMitotic "4x" "nrNuclei" "nucleus" "DAPI" "M" "" ""
4x.ratioMitotic "4x" "mitoticRatio" "nucleus" "DAPI" "M" "" ""
4x.areaNuc "4x" "mean" "nucleus" "DAPI" "M" "area" ""
4x.areaNucSD "4x" "stddev" "nucleus" "DAPI" "M" "area" ""
```

**Value**

A data.frame listing all extracted features.

**See Also**

[DmelSGI-package](#)

**Examples**

```
data(Features)
head(Features)
```

---

fitepistasis

*Fit of the pi-score vectors as a function of main effects*


---

## Description

The output of the linear fit of the pi-score vectors as a function of main effects. The list contains four datasets:

- A 6-dimensional array of the original data with pi-scores and main effects,
- the coefficients of the linear fit,
- the p-values from Anova and
- the variance explained by the main effects.

## Usage

```
data(fitepistasis)
```

## Format

The format is:

List of 2

```
$ Coef: num [1:3, 1:1293, 1:2, 1:72, 1:2] -0.0464 0.3702 -0.0475 -0.0794 ...
.. attr(*, "dimnames")=List of 5
.. ..$: chr [1:3] "const" "xt" "xq"
.. ..$: chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" "CG6833" ...
.. ..$: chr [1:2] "1" "2"
.. ..$: chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
.. ..$: chr [1:2] "1" "2"
$ Sq: num [1:3, 1:1293, 1:2, 1:72, 1:2] 0.18515 0.00207 0.45705 0.15047 ...
.. attr(*, "dimnames")=List of 5
.. ..$: chr [1:3] "xt" "xq" "res"
.. ..$: chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" "CG6833" ...
.. ..$: chr [1:2] "1" "2"
.. ..$: chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
.. ..$: chr [1:2] "1" "2"
```

## Value

The output of the epistasis estimation

## See Also

[DmelSGI-package](#)

## Examples

```
data(fitepistasis)
```

---

getBaseDir	Returns the base directory for the vignette.
------------	--

---

**Description**

Returns the base directory for the vignette. When knitr is applied on the main vignette, this function ensures that the subvignettes get knowledge of the base directory.

**Usage**

```
getBaseDir(default = ".")
```

**Arguments**

default      The default base directory.

**Value**

Returns a character with the directory name

**Author(s)**

Bernd Fischer

**See Also**

[DmelSGI-package](#)

**Examples**

```
getBaseDir()
```

---

grid.spider	Spider plot
-------------	-------------

---

**Description**

A grid function to draw a spider plot.

**Usage**

```
grid.spider(v, col, col.arms = "black", dlim = NULL)  
grid.spider.legend(vn, col.arms = "black", dlim = NULL)
```

**Arguments**

v	A vector of numbers to be presented in the spider plot.
vn	A vector of dimension names that are represented by the spider arms. Has the same length as v.
col	The color of the polygon area.
col.arms	The color of the background of the spider arms.
dlim	A vector with two values. Limits of the spider arm axis.

**Details**

These function draw a grid spider plot or a legend for the spider arms.

**Value**

Both functions return an invisible NULL, but they have an side-effect that draws a spider plot using grid.

**Author(s)**

Bernd Fischer

**See Also**

[orderSpiderAxis](#), [DmelSGI-package](#)

**Examples**

```
print(grid.spider)
print(grid.spider.legend)
```

---

hrNames

*Human readable feature names*

---

**Description**

Translate feature names to human readable feature names. Names not known to this function for conversion are returned unchanged.

**Usage**

```
hrNames(names)
```

**Arguments**

names	Original feature names.
-------	-------------------------

**Value**

A vector of translated feature names.

**Author(s)**

Bernd Fischer

**See Also**

[DmelSGI-package](#)

**Examples**

```
hrNames(c("4x.count", "4x.ratioMitotic"))
```

---

Interactions	<i>Pairwise genetic interaction scores and p-values (summary per gene pairs)</i>
--------------	--

---

**Description**

Two arrays are provided in this dataset: The pairwise interaction scores summarized per gene pair and the respective adjusted p-values. p-values are computed by a moderated t-test (limma) and corrected for multiple testing by the method of Benjamini-Hochberg. The list (Anno) contains the annotation of the target genes, query genes, and features. See [pimatrix](#) for interaction scores that are not yet summarized per gene pair.

**Usage**

```
data(Interactions)
```

**Format**

The format is:

List of 3

```
$ pscore: num [1:1293, 1:72, 1:21] -0.0866 -0.0924 -0.0707 -0.0878 -0.0587 ...
```

```
..- attr(*, "dimnames")=List of 3
```

```
.. ..$ target : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" "CG6833" ...
```

```
.. ..$ query : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
```

```
.. ..$ phenotype: chr [1:21] "4x.count" "4x.ratioMitotic" ...
```

```
$ padj : num [1:1293, 1:72, 1:21] 0.232 0.249 0.286 0.281 0.369 ...
```

```
..- attr(*, "dimnames")=List of 3
```

```
.. ..$ target : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" "CG6833" ...
```

```
.. ..$ query : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
```

```
.. ..$ phenotype: chr [1:21] "4x.count" "4x.ratioMitotic" ...
```

```
$ Anno :List of 3
```

```
..$ target :'data.frame': 1293 obs. of 6 variables:
```

```

.. ..$ TID : chr [1:1293(1d)] "FBgn0002441" "FBgn0038760" ...
.. ..$ TargetPlate : int [1:1293(1d)] 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ group : chr [1:1293(1d)] "sample" "sample" "sample" ...
.. ..$ Symbol : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" ...
.. ..$ Name : chr [1:1293(1d)] "lethal (3) malignant brain tumor" ...
..$ query : 'data.frame': 72 obs. of 5 variables:
.. ..$ TID : chr [1:72(1d)] "FBgn0051156" "FBgn0041111" ...
.. ..$ Batch : int [1:72(1d)] 1 1 1 1 1 1 2 2 2 2 ...
.. ..$ Symbol : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
.. ..$ Name : chr [1:72(1d)] "-" "lilliputian" "Smg1" "Axin" ...
..$ phenotype: 'data.frame': 21 obs. of 1 variable:
.. ..$ phenotype: chr [1:21] "4x.count" "4x.ratioMitotic" ...

```

**Value**

An object containing the pi-scores, the adjusted p-values and the annotation of the statistical genetic interactions.

**See Also**

[pimatrix, DmelSGI-package](#)

**Examples**

```

data(Interactions)
str(Interactions)

```

---

Intogen

---

*Intogen: Interactive Onco Genomics*


---

**Description**

A list of recurrently mutated genes from Intogen.

**Usage**

```
data("Intogen")
```

**Format**

A data frame with 2933 observations on the following 10 variables.

```

gene      a character vector
symbol    a character vector
project.name a character vector
mut.freq  a numeric vector
MuSiC     a character vector

```

oncodriveFM a character vector  
 oncodriveCLUST a character vector  
 ActiveDriver a character vector  
 MutSig a character vector  
 driver.category a character vector

### Value

The Intogen dataset.

### References

Tamborero, David, et al. "Comprehensive identification of mutational cancer driver genes across 12 tumor types." *Scientific reports* 3 (2013).

### See Also

[DmelSGI-package](#)

### Examples

```
data(Intogen)
```

---

learnCoComplexFct      *Learning the co-complex function*

---

### Description

Learning the co-complex function from a correlation matrix.

### Usage

```
learnCoComplexFct(C, ProteinComplexes)
convertCorrelations(C, coComplexFct)
```

### Arguments

C	A genes-by-genes correlation matrix (or matrix containing any other pairwise scores).
ProteinComplexes	A list of protein complexes. Each element of the list is one protein complex and contains a <code>data.frame</code> with at least one column <code>gene_id</code> .
coComplexFct	<code>coComplexFct</code> is an object containing the co-complex function as returned by <code>learnCoComplexFct</code> .



**Details**

An empirical density function of the values in `C` is computed once for the gene pairs that are co-member of at least one protein complex and once for all other gene pairs. The definition of protein complexes that is used to learn the co-complex function is taken from `ProteinComplexes`.

**Value**

`learnCoComplexFct` will provide an object that contains the empirical density function of correlation.

`convertCorrelations` will provide a matrix of co-complex scores.

**Author(s)**

Bernd Fischer

**References**

A similar approach is used in Ryan, C.J., et al. (2012). Hierarchical modularity and the evolution of genetic interactomes across species. *Molecular cell* 46, 691-704.

**See Also**

[DmelSGI-package](#)

**Examples**

```
print(learnCoComplexFct)
print(convertCorrelations)
```

---

mainEffects	<i>Main effects (single knock down effects) estimated from the combinatorial data</i>
-------------	---

---

**Description**

The overall effect (effect of the negative control experiments), and the estimated main effects (single knock down effects) for template and query genes. Overall effects and template main effects are estimated separately for each batch (1:12). Query main effects are estimated separately for each template plate (1:4)

**Usage**

```
data(mainEffects)
```

**Format**

The format is:

List of 4

\$ overall: num [1:2, 1:12, 1:21] 14.6 14.6 14 14 14.7 ...

..- attr(\*, "dimnames")=List of 3

.. ..\$ targetDesign: chr [1:2] "1" "2"

.. ..\$ batch : chr [1:12] "1" "2" "3" "4" ...

.. ..\$ phenotype : chr [1:21] "4x.count" "4x.ratioMitotic" ...

\$ query : num [1:72, 1:2, 1:4, 1:21] 0.0409 0.0192 -0.587 -0.0636 -0.4709 ...

..- attr(\*, "dimnames")=List of 4

.. ..\$ query : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...

.. ..\$ queryDesign : chr [1:2] "1" "2"

.. ..\$ templatePlate: chr [1:4] "1" "2" "3" "4"

.. ..\$ phenotype : chr [1:21] "4x.count" "4x.ratioMitotic" ...

\$ target : num [1:1293, 1:2, 1:12, 1:21] 0.0921 0.3444 0.1603 0.2252 0.1293 ...

..- attr(\*, "dimnames")=List of 4

.. ..\$ target : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" "CG6833" ...

.. ..\$ targetDesign: chr [1:2] "1" "2"

.. ..\$ batch : chr [1:12] "1" "2" "3" "4" ...

.. ..\$ phenotype : chr [1:21] "4x.count" "4x.ratioMitotic" ...

\$ Anno :List of 7

..\$ target :'data.frame': 1293 obs. of 6 variables:

.. ..\$ TID : chr [1:1293(1d)] "FBgn0002441" "FBgn0038760" ...

.. ..\$ TargetPlate : int [1:1293(1d)] 1 1 1 1 1 1 1 1 1 1 ...

.. ..\$ group : chr [1:1293(1d)] "sample" "sample" "sample" ...

.. ..\$ Symbol : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" ...

.. ..\$ Name : chr [1:1293(1d)] "lethal (3) malignant brain tumor"...

..\$ targetDesign :'data.frame': 2 obs. of 1 variable:

.. ..\$ design: int [1:2(1d)] 1 2

..\$ query :'data.frame': 72 obs. of 5 variables:

.. ..\$ TID : chr [1:72(1d)] "FBgn0051156" "FBgn0041111" ...

.. ..\$ Batch : int [1:72(1d)] 1 1 1 1 1 1 2 2 2 2 ...

.. ..\$ Symbol : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...

.. ..\$ Name : chr [1:72(1d)] "-" "lilliputian" "Smg1" "Axin" ...

..\$ queryDesign :'data.frame': 2 obs. of 1 variable:

.. ..\$ design: int [1:2(1d)] 1 2

..\$ phenotype :'data.frame': 21 obs. of 1 variable:

.. ..\$ phenotype: chr [1:21] "4x.count" "4x.ratioMitotic" ...

..\$ batch :'data.frame': 12 obs. of 1 variable:

.. ..\$ batch: int [1:12] 1 2 3 4 5 6 7 8 9 10 ...

..\$ templatePlate:'data.frame': 4 obs. of 1 variable:

.. ..\$ templatePlate: int [1:4] 1 2 3 4

**Value**

An object containing the query and target main effects.

**See Also**

[DmelSGI-package](#)

**Examples**

```
data(mainEffects)
str(mainEffects)
```

---

myHeatmap	<i>Draws a heatmap for a three dimensional array</i>
-----------	--

---

**Description**

Draws a heatmap for a three dimensional array as e.g. the three dimensional genetic interaction cube.

**Usage**

```
myHeatmap(x, cuts, col, fontsize = 18, colnames = TRUE, rownames = FALSE)
```

**Arguments**

x	A three dimensional array.
cuts	break points for mapping the values in x to col. The length of cuts is one larger than the length of col.
col	A color bar as returned by <a href="#">colorRampPalette</a> .
fontsize	The size of the text labels.
colnames	Logical. If TRUE, the column names are printed.
rownames	Logical. If TRUE, the row names are printed.

**Details**

This function is used to draw the heatmap of the three dimensional genetic interaction cube.

**Value**

Nothing is returned, but the function plots a heatmap as a site-effect.

**Author(s)**

Bernd Fischer and Wolfgang Huber

**See Also**

[DmelSGI-package](#)

**Examples**

```
print(myHeatmap)
```

---

mymedpolish

*A variant of the medpolish function.*


---

### Description

A variant of the R-implementation of [medpolish](#). Fits an additive model using Tukey's median polish procedure. The variant uses negative controls to estimate the overall effect and it estimates column effects separately for each batch of rows indicated by TP (for template plate). **It is highly recommended to use the original function from the R-package stats.** The function is used by [estimatePairwiseInteractions](#).

### Usage

```
mymedpolish(x, TP, TemplateNeg, QueryNeg, eps = 1e-04, maxiter = 100, na.rm = TRUE)
```

### Arguments

x	a numeric matrix.
TP	integer. Specifying the plate number for each row element (template genes). Counting starts with 1. Column effects (query genes) will be estimated for each template plate separately
TemplateNeg	Index of negative controls on template plate
QueryNeg	Index of negative controls in the columns (queries)
eps	real number greater than 0. A tolerance for convergence: see <a href="#">medpolish</a> .
maxiter	the maximum number of iterations
na.rm	logical. Should missing values be removed?

### Value

neg	the fitted constant term.or overall effect representing the effect of negative controls
templateMainEffect	the fitted row effects representing the single knock down effects of the template genes
queryMainEffect	the fitted column effects representing the single knock down effects of the query genes. It is a matrix with dimensions query genes x template plates
pi	the residuals which are the pairwise interaction scores

### Author(s)

Original implementation in the R package stats (See [medpolish](#)). Changes for estimating pairwise interaction scores by Bernd Fischer.

**References**

Tukey, J. W. (1977). *Exploratory Data Analysis*, Reading Massachusetts: Addison-Wesley.  
Horn T, Sandmann T, Fischer B, Axelsson E, Huber W, Boutros M (2011). *Mapping of signaling networks through synthetic genetic interaction analysis by RNAi*. *Nature Methods* 8: 341-346.

**See Also**

[medpolish](#), [estimatePairwiseInteractions](#), [DmelSGI-package](#)

**Examples**

```
print(mymedpolish)
```

---

orderDim

*Orders a three dimensional array along one dimension*

---

**Description**

Orders a three dimensional array along one dimension.

**Usage**

```
orderDim(x, i)
```

**Arguments**

x                    A three dimensional array.  
i                    The array is ordered along the i-th dimension.

**Details**

The three dimensional array is ordered along the i-th dimension.

**Value**

An integer vector with the ordering.

**Author(s)**

Bernd Fischer

**See Also**

[DmelSGI-package](#)

**Examples**

```
print(orderDim)
```

---

orderSpiderAxis	<i>Orders the axis of a spider plot.</i>
-----------------	--

---

**Description**

Solves a traveling salesperson problem to optimally order the arms of a spider plot.

**Usage**

```
orderSpiderAxis(X)
```

**Arguments**

X	A $d \times n$ matrix of values for $n$ instances that are plotted in the spider plots with $d$ arms.
---	---

**Details**

The arms are ordered such that two neighboring spider arms are similar to each other.

**Value**

An integer vector with an optimal order of the spider arms.

**Author(s)**

Bernd Fischer

**See Also**

[grid.spider](#), [DmeISGI-package](#)

**Examples**

```
print(orderSpiderAxis)
```

---

pimatrix	<i>Pairwise genetic interaction scores per experiment (no summary per gene pair)</i>
----------	--

---

## Description

An array (D) with pairwise interaction scores. The interaction scores are given per experiment are not yet summarized per gene pair. See [Interactions](#) for interactions summarized per gene pair. A contains the annotation of genes and features.

## Usage

```
data(pimatrix)
```

## Format

The format is:

List of 2

```
$ D : num [1:1293, 1:2, 1:72, 1:2, 1:21] -0.1086 -0.1799 -0.045 -0.0664 ...
```

```
..- attr(*, "dimnames")=List of 5
```

```
.. ..$ target : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" "CG6833" ...
```

```
.. ..$ targetDesign: chr [1:2] "1" "2"
```

```
.. ..$ query : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
```

```
.. ..$ queryDesign : chr [1:2] "1" "2"
```

```
.. ..$ phenotype : chr [1:21] "4x.count" "4x.ratioMitotic" ...
```

```
$ Anno:List of 5
```

```
..$ target :'data.frame': 1293 obs. of 6 variables:
```

```
.. ..$ TID : chr [1:1293(1d)] "FBgn0002441" "FBgn0038760" ...
```

```
.. ..$ TargetPlate : int [1:1293(1d)] 1 1 1 1 1 1 1 1 1 1 ...
```

```
.. ..$ group : chr [1:1293(1d)] "sample" "sample" "sample" ...
```

```
.. ..$ Symbol : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" ...
```

```
.. ..$ Name : chr [1:1293(1d)] "lethal (3) malignant brain tumor"...
```

```
..$ targetDesign:'data.frame': 2 obs. of 1 variable:
```

```
.. ..$ design: int [1:2(1d)] 1 2
```

```
..$ query :'data.frame': 72 obs. of 5 variables:
```

```
.. ..$ TID : chr [1:72(1d)] "FBgn0051156" "FBgn0041111" ...
```

```
.. ..$ Batch : int [1:72(1d)] 1 1 1 1 1 1 2 2 2 2 ...
```

```
.. ..$ Symbol : chr [1:72(1d)] "CG31156" "lilli" "Smg1" "Axn" ...
```

```
.. ..$ Name : chr [1:72(1d)] "-" "lilliputian" "Smg1" "Axin" ...
```

```
..$ queryDesign :'data.frame': 2 obs. of 1 variable:
```

```
.. ..$ design: int [1:2(1d)] 1 2
```

```
..$ phenotype :'data.frame': 21 obs. of 1 variable:
```

```
.. ..$ phenotype: chr [1:21] "4x.count" "4x.ratioMitotic" ...
```

## Value

The matrix of pairwise interaction scores per RNAi design.

**See Also**

[Interactions,DmeISGI-package](#)

**Examples**

```
data(pimatrix)
str(pimatrix)
```

---

plot2Phenotypes	<i>Plot directed epistatic interactions.</i>
-----------------	--

---

**Description**

Plots the data to estimate a directed epistatic interactions.

**Usage**

```
plot2Phenotypes(X, gt, gq, f1, f2, length = 1, ...)
plotPIdata(X, gt, gq, show = "summary", ...)
```

**Arguments**

X	A 6-dimensional array (phenotype x [xt, xq, pi] x target genes x targetDesigns x query genes x queryDesigns).
gt	The target gene name.
gq	The query gene name.
f1	The first phenotypic feature.
f2	The second phenotypic feature.
length	Length of arrow head.
show	Either show='summary' to show the mean over all dsRNA designs or it is a vector of length 2 that specifies the two dsRNAs to show.
...	Other arguments passed to plot.

**Details**

plot2Phenotypes shows a plot showing the two phenotypes on the axis. 2 x 2 arrows for the single gene effects of the two dsRNA designs for the two genes are shown in green and purple, the expected double knock-down effects under the non-interacting model for the 4 combinations of dsRNA designs in gray, and the four measured double knock-down effects in black. The black arrows are the genetic interactions.

plotPIdata shows two scatter plots for the fit of the vector of pairwise interaction scores across all phenotypes as a function of the single gene effects. Each dot represents one phenotype.

**Value**

Nothing is returned, but the function draws a plot as a site-effect.



**Author(s)**

Bernd Fischer

**See Also**[DmelSGI-package](#)**Examples**

```
print(plot2Phenotypes)
print(plotPIdata)
```

---

plotHairballLabels     *Adds the cluster labels to a graph*

---

**Description**

Adds the cluster labels to a graph

**Usage**

```
plotHairballLabels(g, co, Labels, Col)
```

**Arguments**

g	An igraph object for the graph.
co	A n x 2 matrix of layout coordinates as returned by the igraph layout algorithms.
Labels	A list with a vector of gene names per cluster as they appear as vertices in the igraph objects. The list element names are printed to the graph plot.
Col	The colors of the cluster names.

**Value**

Nothing is returned, but as a side-effect, the labels for the hairball are added to a plot.

**Author(s)**

Bernd Fischer

**See Also**[DmelSGI-package](#)**Examples**

```
print(plotHairballLabels)
```

---

qualityControlFeature *Correlation of features between replicates*

---

### Description

The quality control of features is described by three vectors: The correlation between two replicates, the fraction of finite values, and a logical vector indicating which feature passed the quality control. The features are described in the dataset [Features](#).

### Usage

```
data(qualityControlFeature)
```

### Format

```
List of 3 $ correlation : num [1:328] 0.933 0.927 0.927 0.922 0.97 ... $ ratioFiniteValues: num [1:328] 0.999 0.999 0.999 0.999 0.999 ... $ passed : logi [1:328] TRUE TRUE TRUE TRUE TRUE TRUE ...
```

### Value

A data.frame containing the output of the feature quality control.

### See Also

[DmelSGI-package](#)

### Examples

```
data(qualityControlFeature, package="DmelSGI")
str(qualityControlFeature)
```

---

qualityControlGene *Correlation of interaction profiles between independent dsRNA designs*

---

### Description

The quality control of dsRNA designs of the template genes is described by the correlation of the interaction profile between two independent dsRNA designs. In addition the annotation of the genes and a logical vector indicating which gene passed the quality control is given.

### Usage

```
data(qualityControlGene)
```

**Format**

The format is:

List of 3

\$ correlation: num [1:1463] 0.876 0.877 0.897 0.889 0.865 ...

\$ Annotation : 'data.frame': 1463 obs. of 6 variables:

..\$ TID : chr [1:1463(1d)] "FBgn0002441" "FBgn0038760" ...

..\$ TemplatePlate : int [1:1463(1d)] 1 1 1 1 1 1 1 1 1 ...

..\$ group : chr [1:1463(1d)] "sample" "sample" "sample" ...

..\$ Symbol : chr [1:1463(1d)] "l(3)mbt" "MED25" "CG31156" ...

..\$ Name : chr [1:1463(1d)] "lethal (3) malignant brain tumor"...

\$ passed : logi [1:1463] TRUE TRUE TRUE TRUE TRUE TRUE ...

**Value**

A dataset with the output of the gene quality control.

**See Also**

[DmeISGI-package](#)

**Examples**

```
data(qualityControlGene, package="DmeISGI")
str(qualityControlGene)
```

---

RohnEtAl

*Fly RNAi phenotype data*

---

**Description**

This dataset is a list of RNAi Phenotypes captures in Drosophila S2R+ cells.

**Usage**

```
data("RohnEtAl")
```

**Format**

A data frame with 556 observations on the following 29 variables.

Primer a character vector

Computed.Target a character vector

Symbol a character vector

Decreased.cell.size a numeric vector

Increased.cell.size a numeric vector

Cell.shape.variable a numeric vector

Cell.shape.round.or.non.adherent a numeric vector  
 Cell.shape.processes.or.spiky.or.stretchy a numeric vector  
 Disorganised.peripheral.actin a numeric vector  
 Increased.number.of.actin.stress.fibres a numeric vector  
 Increased.number.of.actin.puncta.or.dots a numeric vector  
 Asymmetric.lamellae a numeric vector  
 Decreased.level.of.actin a numeric vector  
 Increased.level.of.actin a numeric vector  
 Increased.cytoplasmic.actin a numeric vector  
 Decreased.peripheral.actin a numeric vector  
 Increased.peripheral.actin a numeric vector  
 Increased.nuclear.actin a numeric vector  
 Microtubule.clumps a numeric vector  
 Microtubules.disorganised a numeric vector  
 Microtubule.processes a numeric vector  
 Decreased.level.of.microtubules a numeric vector  
 Increased.level.of.microtubules a numeric vector  
 Increased.number.of.multinucleate.cells a numeric vector  
 Increased.DNA.area a numeric vector  
 No.cells a numeric vector  
 Decreased.cell.number a numeric vector  
 Loss.of.cell.monolayer a numeric vector  
 Multiple.layers.of.cells a numeric vector

### Value

A list of phenotypes. The genes showing the respective phenotype are listed in the vector for each phenotype.

### References

Rohn, Jennifer L., et al. "Comparative RNAi screening identifies a conserved core metazoan actinome by phenotype." *The Journal of cell biology* 194.5 (2011): 789-805.

### See Also

[DmeISGI-package](#)

### Examples

```
data(RohnEtAl)
```

---

selectByStability      *Select features by stability.*

---

### Description

Features are selected in a greedy manner. A linear model is fitted to estimate the remove the contribution of each features that can already be explained by previously selected features. The next features is selected such, that the residuals are maximal correlated between replicates.

### Usage

```
selectByStability(subsample,
                  preselect = c("4x.count",
                                "4x.ratioMitotic",
                                "10x.meanNonmitotic.cell.0.s.area"),
                  Rdim = 40,
                  verbose = TRUE)
```

### Arguments

subsample	A list with subsampled data as produced by <a href="#">subSampleForStabilitySelectionFct.data(subSampleForStabilitySelection, package='Dm1SGI')</a> provides the dataset used in the paper.
preselect	The names of features that are preselected, e.g. the features '4x.count' and '4x.ratioMitotic' should be selected, because they have a special interpretation.
Rdim	The maximum number of selected features.
verbose	If TRUE, more output is provided.

### Value

selected	The names of the selected features. Rdim features will be reported.
correlation	The correlation of the residual features.
ratioPositive	The fraction of positively correlated residual features in each step of the selection process. The features with ratioPositive > 0.5 should be selected.
correlationAll	The correlation of the all residual features in each step of the selection process.

### Author(s)

Bernd Fischer

### See Also

[Dm1SGI-package](#)

### Examples

```
print(selectByStability)
```

---

SelectedClusters	<i>Selected processes displayed on the hairball</i>
------------------	---

---

**Description**

A list of gene sets that are displayed on the hairball.

**Usage**

```
data(SelectedClusters)
```

**Format**

The format is:

List of 24

\$ SWI/SNF : chr [1:6] "Bap60" "brm" "dalao" ...

\$ Condensin/Cohesin : chr [1:5] "SMC2" "Cap-D2" "glu" ...

\$ Cytokinesis : chr [1:12] "pav" "sqh" "rok" ...

...

**Value**

A list with selected clusters for visualization.

**See Also**

[DmelSGI-package](#)

**Examples**

```
data(SelectedClusters)
```

---

SelectedClustersComplexes	<i>Selected complexes</i>
---------------------------	---------------------------

---

**Description**

Manually curated protein complexes.

**Usage**

```
data(SelectedClustersComplexes)
```

**Format**

The format is:

List of 27

\$ DREAM complex : chr [1:5] "Caf1" "mip120" "mip130" "mip40" ...

\$ Condensin/Cohesin : chr [1:5] "SMC2" "Cap-D2" "glu" "eco" ...

\$ Apc/C : chr [1:9] "APC10" "Cdc23" "Cdc16" "shtd" ...

...

**Value**

A list of gene sets that represent well-defined protein complexes.

**See Also**

[DmelSGI-package](#)

**Examples**

```
data(SelectedClustersComplexes)
```

---

SKDdata

*Single knock down screen*

---

**Description**

D is the single knock down screen data for the 12 negative control query genes. The annotation of each dimension of D is provided is the list A.

**Usage**

```
data(SKDdata)
```

**Format**

The format is: List of 2 \$ D : num [1:1293, 1:2, 1:12, 1:21] 14.7 15 14.8 14.8 14.8 ...

..- attr(\*, "dimnames")=List of 4

.. ..\$ target : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" "CG6833" ...

.. ..\$ targetDesign: chr [1:2] "1" "2"

.. ..\$ batch : chr [1:12] "1" "2" "3" "4" ...

.. ..\$ phenotype : chr [1:21] "4x.count" "4x.ratioMitotic" ...

\$ Anno:List of 4

..\$ target :'data.frame': 1293 obs. of 6 variables:

.. ..\$ TID : chr [1:1293(1d)] "FBgn0002441" "FBgn0038760" ...

.. ..\$ TargetPlate : int [1:1293(1d)] 1 1 1 1 1 1 1 1 1 ...

.. ..\$ group : chr [1:1293(1d)] "sample" "sample" "sample" ...

.. ..\$ Symbol : chr [1:1293(1d)] "l(3)mbt" "MED25" "CG31156" ...

.. ..\$ Name : chr [1:1293(1d)] "lethal (3) malignant brain tumor"...

```

..$ targetDesign:'data.frame': 2 obs. of 1 variable:
.. ..$ design: int [1:2(1d)] 1 2
..$ batch :'data.frame': 12 obs. of 1 variable:
.. ..$ batch: int [1:12] 1 2 3 4 5 6 7 8 9 10 ...
..$ phenotype :'data.frame': 21 obs. of 1 variable:
.. ..$ phenotype: chr [1:21] "4x.count" "4x.ratioMitotic" ...

```

### Value

A dataset with the single knockdown data.

### See Also

[DmelSGI-package](#)

### Examples

```

data(SKDdata)
str(SKDdata)

```

---

stabilitySelection      *The features selected by stability.*

---

### Description

The features selected by stability used in the selection process are available in this dataset. Furthermore, it contains the correlation of the residual features and the fraction of positive correlated features that is used as a stop criterion.

### Usage

```
data(stabilitySelection)
```

### Format

The format is:  
List of 4  
\$ selected : chr [1:50] "4x.count" "4x.ratioMitotic" ...  
\$ correlation : num [1:50] 0.912 0.868 0.559 ...  
\$ ratioPositive : num [1:50] 1 0.957 0.95 ...  
\$ correlationAll:List of 50  
..\$ : Named num [1:162] 0.912 0.946 0.948 ...  
.. ..- attr(\*, "names")= chr [1:162] "4x.count" "4x.countpH3" ...  
..\$ : Named num [1:161] 0.946 0.795 0.868 ...  
.. ..- attr(\*, "names")= chr [1:161] "4x.countpH3" "4x.isMitotic" ...  
..\$ : Named num [1:160] 0.946 0.129 0.927 ...  
.. ..- attr(\*, "names")= chr [1:160] "4x.countpH3" "4x.isMitotic" ...  
...



**Value**

An object containing the output of the feature selection.

**See Also**

[DmelSGI-package](#)

**Examples**

```
data(stabilitySelection)
str(stabilitySelection)
```

---

```
subSampleForStabilitySelection
```

*A subsampled dataset for use stability selection.*

---

**Description**

This dataset contains a subsample of the interaction screen for use in the function [selectByStability](#). It contains the data matrix  $DD$  with 3000 experiments x 2 replicates (dsRNA designs) x 162 features.

**Usage**

```
data(subSampleForStabilitySelection)
```

**Format**

The format is:  
 List of 3  
 \$ D : num [1:3000, 1:2, 1:162] -1.447 0.351 0.44 ...  
 \$ Sample : int [1:3000] 77465 94252 95176 ...  
 \$ phenotype: chr [1:162(1d)] "4x.count" "4x.countpH3" ...

**Value**

The dataset of samples used for feature selection.

**See Also**

[selectByStability](#), [subSampleForStabilitySelectionFct](#), [DmelSGI-package](#)

**Examples**

```
data(subSampleForStabilitySelection)
str(subSampleForStabilitySelection)
```

---

`subSampleForStabilitySelectionFct`*Subsampling the data for stability selection.*

---

**Description**

The data is subsampled to reduce computation time and memory demand for stability selection. 10000 experiments are selected by chance to estimate the most stable directions.

**Usage**

```
subSampleForStabilitySelectionFct(fileMatrixData, N = 10000, random.seed = NULL)
```

**Arguments**

<code>fileMatrixData</code>	Filename of the HDF5 file containing the data array
<code>N</code>	Number of experiments used for stability selection
<code>random.seed</code>	If not NULL, the <code>random.seed</code> is set before sampling, to generate a reproducible analysis script

**Details**

For each dsRNA design, query gene, and feature, the median value is subtracted and the data are divided by the median deviation.

**Value**

<code>D</code>	An array of dimension $N \times 2 \times F$ , where $N$ is the given number of experiments used, 2 represents the two query dsRNA designs, and $F$ is the number of features.
<code>Sample</code>	The index of the sampled elements.
<code>phenotype</code>	The name of the features.

**Author(s)**

Bernd Fischer

**See Also**

[selectByStability](#), [subSampleForStabilitySelection](#), [Dm1SGI-package](#)

**Examples**

```
print(subSampleForStabilitySelectionFct)
```

---

`TID2HUGO`*Mapping of flygene names to human.*

---

**Description**

Mapping of flybase gene identifier to their human orthologues. It is a one-to-many mapping.

**Usage**

```
data(TID2HUGO)
```

**Format**

The format is:

List of 1293

```
$ FBgn0002441: chr [1:3] "L3MBTL1" "L3MBTL3" "L3MBTL4"
```

```
$ FBgn0038760: chr "MED25"
```

```
$ FBgn0051156: chr "SRBD1"
```

```
... [list output truncated]
```

```
- attr(*, "dim")= int 1293
```

```
- attr(*, "dimnames")=List of 1
```

```
..$ : chr [1:1293] "FBgn0002441" "FBgn0038760" "FBgn0051156" "FBgn0036405" ...
```

**Value**

A list containing the conversion of gene identifier from fly to human.

**See Also**

[DmelSGI-package](#)

**Examples**

```
data(TID2HUGO)
```

---

`toMatrix`*Flattens a three dimensional array to a two dimensional matrix*

---

**Description**

Flattens a three dimensional array to a two dimensional matrix.

**Usage**

```
toMatrix(x)
```

**Arguments**

x                    A three dimensional array.

**Value**

A matrix.

**See Also**

[DmeISGI-package](#)

**Examples**

```
print(toMatrix)
```

---

toRaster	<i>Converts a real valued matrix in a matrix of color codes printable by <a href="#">grid.raster</a></i>
----------	--

---

**Description**

A matrix of real values in a matrix are converted in a matrix of RGB values that can be printed by [grid.raster](#).

**Usage**

```
toRaster(x, cuts, col)
```

**Arguments**

x                    A real valued matrix.  
cuts                Break points for the color values. Length of cuts has to be the length of col plus one.  
col                 A vector of color values, e.g. as produced by [colorRampPalette](#).

**Value**

Returns a matrix of RGB color values that can be printed by [grid.raster](#).

**Author(s)**

Bernd Fischer

**See Also**

[grid.raster](#), [colorRampPalette](#), [DmeISGI-package](#)

**Examples**

```
print(toRaster)
```

---

**trsf***Transform a correlation to a distance*

---

**Description**

Transforms a correlation to a distance.

**Usage**

```
trsf(x)
```

**Arguments**

**x** A real valued vector with values in  $[-1,1]$ . It is intended to be a correlation.

**Value**

a real valued vector of distances.

**See Also**

[DmelSGI-package](#)

**Examples**

```
print(trsf)
```

# Index

## \* datasets

applyDimensionReduction, 4  
callInteractions, 5  
datamatrix, 6  
DPiM, 7  
estimatePairwiseInteractions, 8  
FBgn2anno, 9  
Features, 10  
fitepistasis, 11  
getBaseDir, 12  
grid.spider, 12  
hrNames, 13  
Interactions, 14  
Intogen, 15  
learnCoComplexFct, 16  
mainEffects, 17  
myHeatmap, 19  
mymedpolish, 20  
orderDim, 21  
orderSpiderAxis, 22  
pimatrix, 23  
plot2Phenotypes, 24  
plotHairballLabels, 25  
qualityControlFeature, 26  
qualityControlGene, 26  
RohnEtAl, 27  
selectByStability, 29  
SelectedClusters, 30  
SelectedClustersComplexes, 30  
SKDdata, 31  
stabilitySelection, 32  
subSampleForStabilitySelection, 33  
subSampleForStabilitySelectionFct, 34  
TID2HUGO, 35  
toMatrix, 35  
toRaster, 36  
trsf, 37

## \* package

DmelSGI-package, 3  
applyDimensionReduction, 3, 4  
callInteractions, 3, 5  
colorRampPalette, 19, 36  
convertCorrelations  
(learnCoComplexFct), 16  
datamatrix, 3, 6  
DmelSGI (DmelSGI-package), 3  
DmelSGI-package, 3, 4, 5, 7–17, 19, 21, 22, 24–37  
DPiM, 7  
estimatePairwiseInteractions, 3, 8, 20, 21  
FBgn2anno, 9  
Features, 3, 10, 26  
fitepistasis, 11  
getBaseDir, 12  
grid.raster, 36  
grid.spider, 12, 22  
hrNames, 13  
Interactions, 3, 14, 23, 24  
Intogen, 15  
learnCoComplexFct, 16  
mainEffects, 3, 17  
medpolish, 20, 21  
myHeatmap, 19  
mymedpolish, 3, 8, 20  
orderDim, 21  
orderSpiderAxis, 13, 22  
pimatrix, 3, 14, 15, 23

plot2Phenotypes, [24](#)  
plotHairballLabels, [25](#)  
plotPIdata (plot2Phenotypes), [24](#)  
  
qualityControlFeature, [3](#), [26](#)  
qualityControlGene, [3](#), [26](#)  
  
RohnEtAl, [27](#)  
  
selectByStability, [4](#), [29](#), [33](#), [34](#)  
SelectedClusters, [30](#)  
SelectedClustersComplexes, [30](#)  
SKDdata, [3](#), [31](#)  
stabilitySelection, [3](#), [4](#), [32](#)  
subSampleForStabilitySelection, [3](#), [33](#),  
[34](#)  
subSampleForStabilitySelectionFct, [3](#),  
[29](#), [33](#), [34](#)  
  
TID2HUGO, [35](#)  
toMatrix, [35](#)  
toRaster, [36](#)  
trsf, [37](#)