

# Package ‘dils’

July 4, 2013

**Type** Package

**Title** Data-Informed Link Strength. Combine multiple-relationship networks into a single weighted network.

**Version** 0.1

**Date** 2013-06-13

**Author** Stephen R. Haptonstahl

**Maintainer** Stephen R. Haptonstahl <srh@haptonstahl.org>

**Suggests** testthat

**Description** Combine multiple-relationship networks into a single weighted network. The approach is similar to factor analysis in that contribution from each constituent network varies so as to maximize the information gleaned from the multimetwork. This implementation uses an Item-Response Theory (IRT) model estimated with a custom Gibbs Sampler.

**License** MIT + file LICENSE

**Depends** Rcpp (>= 0.9.13)

**LinkingTo** Rcpp

**Collate**

‘GetSampleFromDataFrame.R’ ‘rcpp\_hello\_world.R’ ‘RelationStrengthSimilarity.R’ ‘ScalablePCA.R’

**Archs** i386, x64

## R topics documented:

dils-package . . . . .	2
GetSampleFromDataFrame . . . . .	2
rcpp_hello_world . . . . .	3
RelationStrengthSimilarity . . . . .	4
ScalablePCA . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

dils-package	<i>Data-Informed Link Strength. Combine multiple-relationship networks into a single weighted network.</i>
--------------	--

---

### Description

Combine multiple-relationship networks into a single weighted network. The approach is similar to factor analysis in that contribution from each constituent network varies so as to maximize the information gleaned from the multimetwork. This implementation uses Principal Component Analysis calculated using 'prcomp'.

### Details

Package:	dils
Type:	Package
Version:	0.1
Date:	2013-07-05
License:	What license is it under?

Description of how to use and main functions.

### Author(s)

Stephen R. Haptonstahl Maintainer: Stephen R. Haptonstahl <srh@haptonstahl.org>

### References

"Discovering Missing Links in Networks Using Similarity Measures", Hung-Hsuan Chen, Liang Gou, Xiaolong (Luke) Zhang, C. Lee Giles. 2012.

---

GetSampleFromDataFrame	<i>Short description of the function</i>
------------------------	--

---

### Description

A longer description of the function. This can be perhaps a paragraph, perhaps more than one.

### Usage

```
GetSampleFromDataFrame(n, x)
```

### Arguments

n	numeric, size of sample.
x	data.frame, data whose rows will be sampled.

**Value**

data.frame, size n random subset of the rows of x

**Author(s)**

Stephen R. Haptonstahl <srh@haptonstahl.org>

**References**

<http://www.haptonstahl.org/R>

**See Also**

[ScalablePCA](#)

**Examples**

```
data(iris) # provides example data
x <- GetSampleFromDataFrame(10, iris)
```

---

rcpp\_hello\_world

*Simple function using Rcpp*

---

**Description**

Simple function using Rcpp

**Usage**

```
rcpp_hello_world()
```

**Examples**

```
## Not run:
rcpp_hello_world()

## End(Not run)
```

---

**RelationStrengthSimilarity**

*Calculate the RSS from one node to another*

---

**Description**

A longer description of the function. This can be perhaps a paragraph, perhaps more than one.

**Usage**

```
RelationStrengthSimilarity(xadj, v1, v2, radius)
```

**Arguments**

xadj	numeric matrix, then description of arg1.
v1	numeric Object type, then description of arg2.
v2	numeric Object type, then description of arg2.
radius	numeric, length of longest path examined from v1 to v2.

**Value**

numeric, the Relation Strength Similarity score from v1 to v2.

**Author(s)**

Stephen R. Haptonstahl <srh@haptonstahl.org>

**References**

"Discovering Missing Links in Networks Using Similarity Measures", Hung-Hsuan Chen, Liang Gou, Xiaolong (Luke) Zhang, C. Lee Giles. 2012.

**See Also**

[ScalablePCA](#)

**Examples**

```
M.test.1 <- matrix(0, nrow=6, ncol=6)
M.test.1[1,2] <- M.test.1[2,1] <- 1
M.test.1[1,3] <- M.test.1[3,1] <- 1
M.test.1[3,4] <- M.test.1[4,3] <- 1
M.test.1[4,5] <- M.test.1[5,4] <- 1
M.test.1[5,6] <- M.test.1[6,5] <- 1
M.test.1[6,1] <- M.test.1[1,6] <- 1
M.test.1[1,4] <- M.test.1[4,1] <- 1
M.test.1
## Not run: RelationStrengthSimilarity(xadj=M.test.1, v1=5, v2=6, radius=2)
## Not run: RelationStrengthSimilarity(xadj=M.test.1, v1=5, v2=6, radius=3)
## Not run: RelationStrengthSimilarity(xadj=M.test.1, v1=5, v2=6, radius=4)
```

---

ScalablePCA*Perform Principal Component Analysis on a large data set*

---

**Description**

Runs 'prcomp' on subsamples of the data set and compiles the results for the first dimension.

**Usage**

```
ScalablePCA(x, filename = NULL, db = NULL,  
  subsample = 10000, n.subsamples = 1000, ignore.cols,  
  use.cols, progress.bar = FALSE)
```

**Arguments**

x	data.frame, data over which to run PCA
filename	character, name of the file containing the data (NOT IMPLEMENTED)
db	Object type, database connection to table containing the data (NOT IMPLEMENTED)
subsample	numeric or logical, If an integer, size of each subsample. If FALSE, runs PCA on entire data set.
n.subsamples	numeric, number of subsamples.
ignore.cols	numeric, indices of columns not to include
use.cols	numeric, indices of columns to use
progress.bar	logical, if TRUE then progress in running subsamples will be shown.

**Value**

vector, named vector of component weights for first dimension of principal component analysis (see example for comparison to [prcomp](#))

**Author(s)**

Stephen R. Haptonstahl <srh@haptonstahl.org>

**References**

<http://www.haptonstahl.org/R>

**See Also**

[prcomp](#)

**Examples**

```
data(iris)          # provides example data  
prcomp(iris[,1:4], center=FALSE, scale.=FALSE)$rotation[,1]  
ScalablePCA(iris, subsample=10, use.cols=1:4)  
ScalablePCA(iris, subsample=10, ignore.cols=5)
```

# Index

\*Topic **network**

dils-package, [2](#)

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

dils-package, [2](#)

GetSampleFromDataFrame, [2](#)

prcomp, [5](#)

rcpp\_hello\_world, [3](#)

RelationStrengthSimilarity, [4](#)

ScalablePCA, [3](#), [4](#), [5](#)