

# Package ‘LoopAnalyst’

January 2, 2012

**Version** 1.2-2

**Date** 2009-10-12

**Title** A collection of tools to conduct Levins’ Loop Analysis

**Author** Alexis Dinno <adinno@post.harvard.edu>

**Maintainer** Alexis Dinno <adinno@post.harvard.edu>

**Depends** R (>= 1.8.0), nlme

**Description** Loop analysis makes qualitative predictions of variable change in a system of causally interdependent variables, where “qualitative” means sign only (i.e. increases, decreases, non change, and ambiguous). This implementation includes output support for graphs in .dot file format for use with visualization software such as graphviz (graphviz.org). Loop Analyst provides tools for the construction and output of community matrices, computation and output of community effect matrices, tables of correlations, adjoint, absolute feedback, weighted feedback and weighted prediction matrices, change in life expectancy matrices, and feedback, path and loop enumeration tools.

**License** GPL-2

**LazyLoad** true

**URL** <http://www.doyenne.com/LoopAnalyst>

**Repository** CRAN

**Date/Publication** 2009-10-13 08:19:29

## R topics documented:

cem.corr . . . . .	2
cem.levins . . . . .	3
cm.dambacher . . . . .	4
cm.DLR . . . . .	4

cm.lewins . . . . .	5
enumerate.loops . . . . .	5
enumerate.paths . . . . .	6
feedback . . . . .	7
graph.cem . . . . .	8
graph.cm . . . . .	9
make.adjoint . . . . .	10
make.cem . . . . .	11
make.clem . . . . .	12
make.cm . . . . .	13
make.T . . . . .	14
make.wfm . . . . .	15
out.cm . . . . .	17
save.cm . . . . .	18
submatrix . . . . .	18
validate.cem . . . . .	19
validate.cm . . . . .	20
weighted.predictions . . . . .	21
<b>Index</b>	<b>23</b>

---

cem.corr

---

*Compute the Tables of Correlations for a Community Effect matrix*


---

## Description

Validates a community effect matrix and computes its associated tables of correlations.

## Usage

```
cem.corr(CEM)
```

## Arguments

CEM            A valid community effect matrix.

## Details

The supplied matrix is validated as a community effect matrix, and the table of predicted correlations is computed. Output is formatted as per Puccia and Levins, 1986 (see page 57) and correlations are represented as "-", "0", "+", and "?" for negative correlation, no correlation, positive correlation, and ambiguous correlation, respectively.

## Author(s)

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

## References

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press. pp 55-59

## See Also

[make.cem](#).

## Examples

```
## compute tables of correlations
data(cem.levins)
cem <- make.cem(cem.levins)
cem.corr(cem)
```

---

cem.levins

*Community Effect Matrix For A Four-Variable Trophic Model*

---

## Description

This dataset contains the computed community effect matrix of the model in Figure 2 from Levins' 1996 paper (see references).

## Usage

```
data(cem.levins)
```

## Format

A 4 by 4 (-1, 0, 1) matrix. The matrix is interpreted as the *i*th parameter has the indirect causal effect [i,j] on parameter *j* after the system stabilizes in response to a perturbation.

## References

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

Levins, R. and Schultz, B. B. (1996) Effects of density dependence, feedback and environmental sensitivity on correlations among predators, prey and plant resources: Models and practical implications. *Journal of Animal Ecology*, **65**(6),802-812

---

cm.dambacher

*A Ten-Variable Trophic Model Of Danish Shallow Lakes*

---

### Description

This dataset represents the causal relationships amongst the ten variables defined by iDambacher's 2002 in figure 6, page 1381.

### Usage

```
data(cm.dambacher)
```

### Format

A 10 by 10 (-1, 0, 1) matrix with each element having at least one parent and at least one child. The matrix is interpreted as the jth variable has the direct causal effect  $a[i,j]$  on variable i.

### References

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

Dambacher, J. M., Li, H. M., and Rossignol, P. A. (2002) Relevance of community structure in assessing indeterminacy of ecological predictions. *Ecology*, **83**(5),1372-1385

---

cm.DLR

*Model of Two Mosquito Life Stages With Predation and Biocontrol*

---

### Description

This dataset represents the causal relationships amongst the four variables defined by Dambacher et al. in their 2005 publication in figure 2, page 10.

### Usage

```
data(cm.DLR)
```

### Format

A 4 by 4 (-1, 0, 1) matrix with each element having at least one parent and at least one child. The matrix is interpreted as the jth variable has the direct causal effect  $a[i,j]$  on variable i.

### References

Dambacher, J. M., R. Levins and P. A. Rossignol. (2005) Life expectancy change in perturbed communities: Derivation and qualitative analysis. *Mathematical Biosciences* **197**,1-14

---

`cm.levins`*Model of A Resource, An Herbivore, And Two Predators*

---

**Description**

This dataset represents the causal relationships amongst the four variables defined by Levins' 1996 publication in figure 2, page 804.

**Usage**

```
data(cm.levins)
```

**Format**

A 4 by 4 (-1, 0, 1) matrix with each element having at least one parent and at least one child. The matrix is interpreted as the jth variable has the direct causal effect  $a[i,j]$  on variable i.

**References**

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

Levins, R. and Schultz, B. B. (1996) Effects of density dependence, feedback and environmental sensitivity on correlations among predators, prey and plant resources: Models and practical implications. *Journal of Animal Ecology*, **65**(6),802-812

---

`enumerate.loops`*Enumerates all Loops in a (-1, 0, 1) Matrix*

---

**Description**

Enumerates a list of all loops in a (-1, 0, 1) matrix.

**Usage**

```
enumerate.loops(CM)
```

**Arguments**

CM                    A (-1, 0 1) matrix.

**Details**

The returned list of loops contains each loop in CM in breadth-first search order). Each element in a loop is represented by its variable number, and the first element also terminates each loop.

**Value**

A list of loops.

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

**References**

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

**See Also**

[make.cm](#).

**Examples**

```
## assess community matrix
data(submatrix)
enumerate.loops(submatrix)
```

---

enumerate.paths

*Enumerates all Paths from i to j in a Community Matrix*

---

**Description**

Enumerates a list of all paths from variable i to variable j in a community matrix after first validating that CM is a community matrix.

**Usage**

```
enumerate.paths(CM, i, j)
```

**Arguments**

CM	A community matrix.
i, j	variables in the community matrix CM. Variables may be specified by their names or numbers (i.e. the row/column number corresponding to a specific variable).

**Details**

The returned list of paths contains each path from i to j in CM in breadth-first search order). Each element in a path is represented by its variable number.

**Value**

A list of paths.

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

**References**

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

**See Also**

[make.cm](#).

**Examples**

```
## assess community matrix
data(cm.levins)
enumerate.paths(cm.levins,2,4)
```

---

feedback

*Feedback of a Partially Specified System*

---

**Description**

Computes the qualitative feedback of a partially specified system; where *qualitative* means direction: increases, decreases, no change, or ambiguous.

**Usage**

```
feedback(C)
```

**Arguments**

C                    A (-1, 0 1) square matrix.

**Details**

The supplied matrix is validated as a square (-1, 0, 1) matrix, and the sign-corrected product of system-spanning sets of disjunct loops is summed over all levels of feedback, and the result returned. Ambiguous results are returned as [NA](#).

**Value**

-1, 0, 1, or NA

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

## References

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

## See Also

[make.cm](#).

## Examples

```
## compute the feedback of a system
data(submatrix)
feedback(submatrix)
```

---

graph.cem

*Graph a Community Effect Matrix*

---

## Description

Represents a community effect matrix as a graph in the dot language. This representation is sometimes termed a 'prediction scenario'.

## Usage

```
graph.cem(CEM, file, color="bw")
```

## Arguments

CEM	a community effect matrix to be graphed.
file	a connection or a character string giving the name of the dot file (should have a .dot suffix).
color	select which color mode to graph the system: bw, color, or greyscale. Default bw is black and white.

## Details

This function outputs a dot file for use with graphviz or similar graph layout package to visually represent the community effect matrix system. The color options color and greyscale assist in graph readability when there are a large number of nodes and connections between them. Ambiguous effects are represented by dotted edges and tee-style arrowheads.

## comments

The representation of loop analytic predictions in graph form is an emerging practice. Feedback and ideas are welcome, and I am amenable to implementing them in future versions of LoopAnalyst. graph.cem does not currently work with weighted.predictions output.

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

**References**

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

Gansner, E., Koutsofios, E. and North, S. (2002) *Drawing graphs with dot*. <http://www.graphviz.org>

**See Also**

[make.cem](#).

**Examples**

```
## graph a community effect matrix
data(cm.levins)
make.cem(cm.levins) -> cem.levins
graph.cem(cem.levins, file="levins.dot", color="color")

## graph a community effect matrix
data(cm.dambacher)
make.cem(cm.dambacher) -> cem.dambacher
graph.cem(cem.dambacher, file="dambacher.dot", color="color")
```

---

graph.cm

*Graph a Community Matrix*

---

**Description**

Represents a community matrix as a signed digraph in the dot language.

**Usage**

```
graph.cm(CM, file, color="bw")
```

**Arguments**

CM	a community matrix to be graphed.
file	a connection or a character string giving the name of the dot file (should have a .dot suffix).
color	select which color mode to graph the system: bw, color, or greyscale. Default bw is black and white.

## Details

This function outputs a dot file for use with graphviz or similar graph layout package to visually represent the community matrix system. The color options color and greyscale assist in graph readability when there are a large number of nodes and connections between them.

## Author(s)

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

## References

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

Gansner, E., Koutsofios, E. and North, S. (2002) *Drawing graphs with dot*. <http://www.graphviz.org>

## See Also

[make.cm](#).

## Examples

```
## graph a community matrix
data(cm.levins)
graph.cm(cm.levins, file="levins.dot")

## graph a community matrix
data(cm.dambacher)
graph.cm(cm.dambacher, file="dambacher.dot", color="color")
```

---

make.adjoint

*Compute the Adjoint of the Negative Community Matrix*

---

## Description

Validates a community matrix and computes the adjoint matrix of it's negative.

## Usage

```
make.adjoint(CM, status=FALSE)
```

## Arguments

CM	A valid community matrix.
status	Switches on an element-by-element progress indicator when set to TRUE. Set to FALSE by default.

## Details

The supplied matrix is validated as a community matrix, and the adjoint matrix of the community matrix's negative is computed. The value of a given element indicates the sum of feedback at all levels, under the assumption that, in the absence of relative specification of community matrix linkages, each level of feedback ought to account for the same proportion of total feedback contributing to the effect on  $j$  of a press perturbation on  $i$ . Values of a given element may be positive, zero or negative integer values, but of a magnitude no larger than the corresponding value in the absolute feedback matrix  $T$ .

NOTE: weighted feedback, adjoint and absolute feedback matrices are transposed relative to the community effect matrix.

## Value

The adjoint matrix of a negative community matrix

## Author(s)

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

## References

Dambacher, J. M., et al. (2003) Qualitative stability and ambiguity in model ecosystems. *The American Naturalist*, **161**(6),876–888

Dambacher, J. M. and Li, H. W. and Rossignol, P. A. (2002) Relevance of community structure in assessing indeterminacy of ecological predictions. *Ecology*, **83**(5),1372–1385

## See Also

[make.cem](#), [make.wfm](#), [make.T](#), and [weighted.predictions](#).

## Examples

```
## compute adjoint of a negative of a community matrix
data(cm.dambacher)
make.adjoint(cm.dambacher)
```

---

make.cem

*Compute the Community Effect Matrix*

---

## Description

Validates and community matrix and computes its associated community effect matrix.

## Usage

```
make.cem(CM, status=FALSE, out=FALSE)
```

**Arguments**

CM	A valid community matrix.
status	Switches on an element-by-element progress indicator when set to TRUE. Set to FALSE by default.
out	Switches on formatting of community effect matrix element output from "-1", "0", "1", "NA" to "-", "0", "+", and "?" respectively when set to TRUE. Set to FALSE by default.

**Details**

The supplied matrix is validated as a community matrix, and the community effect matrix (i.e. table of predictions) is computed. Ambiguous effects are represented by link{NA}.

**Value**

A community effect matrix

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

**References**

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

**See Also**

[make.cm](#).

**Examples**

```
## compute community effect matrix
data(cm.levins)
make.cem(cm.levins)
```

---

make.clem

*Compute the Change in Life Expectancy Matrix*

---

**Description**

Validates a community matrix and computes its associated change in life expectancy matrix.

**Usage**

```
make.clem(CM, status=FALSE)
```

## Arguments

CM	A valid community matrix.
status	Switches on an element-by-element progress indicator when set to TRUE. Set to FALSE by default.

## Details

The supplied matrix is validated as a community matrix, and a table of predicted changes in life expectancy (i.e. the inverse of turnover) given a press perturbation is computed. Diagonal elements can differ for births and deaths, and these predictions, in that order, are separated by a comma.

## Value

A change in life expectancy matrix

## Author(s)

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

## References

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

Dambacher, J. M., R. Levins and P. A. Rossignol. (2005) Life expectancy change in perturbed communities: Derivation and qualitative analysis. *Mathematical Biosciences* **197**,1-14

## See Also

[imake.cm](#), [make.cm](#).

## Examples

```
## compute change in life expectancy matrix
data(cm.DLR)
make.clem(cm.DLR)
```

---

make.cm

*Interactively Make a Community Matrix*

---

## Description

Make and validate a new community matrix by soliciting the number of variables, variable names and interactions between each.

## Usage

```
make.cm(n)
```

**Arguments**

n                    an integer value denoting the number of component variables modeled by the community matrix. The minimum number of variables is 2. If left unspecified, make.cm will prompt for the number of variables.

**Details**

make.cm prompts the user for the number of variables (if not specified as an option), variable names, and the qualitative signifier (-1, 0, or 1) of direct causal effect from and to each variable.

**Value**

a labeled (-1,0,1) square matrix.

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

**References**

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

**See Also**

[out.cm](#), [graph.cm](#), [make.cm](#).

**Examples**

```
## assign a community matrix
## Not run: CM <- make.cm(4)
```

---

make.T

*Compute the Absolute Feedback Matrix of a Community Matrix*

---

**Description**

Validates a community matrix and computes its absolute feedback matrix.

**Usage**

```
make.T(CM, status=FALSE)
```

**Arguments**

CM                    A valid community matrix.

status                Switches on an element-by-element progress indicator when set to TRUE. Set to FALSE by default.

## Details

The supplied matrix is validated as a community matrix, and its absolute feedback matrix is computed. The value of a given element indicates the total number of disjoint cycles spanning all variables of the complementary subsystem for each path from  $j$  to  $i$ . Values of a given element may be positive or zero, and bounds the magnitude of the associated adjoint matrix of the negative community matrix.

NOTE: weighted feedback, adjoint and absolute feedback matrices are transposed relative to the community effect matrix.

## Value

The absolute feedback matrix for a community matrix

## Author(s)

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

## References

Dambacher, J. M., et al. (2003) Qualitative stability and ambiguity in model ecosystems. *The American Naturalist*, **161**(6),876–888

Dambacher, J. M. and Li, H. W. and Rossignol, P. A. (2002) Relevance of community structure in assessing indeterminacy of ecological predictions. *Ecology*, **83**(5),1372–1385

## See Also

[make.cm](#), [make.wfm](#), [make.adjoint](#) and [weighted.predictions](#).

## Examples

```
## compute T for a community matrix
data(cm.dambacher)
make.T(cm.dambacher)
```

---

make.wfm

*Compute the Weighted Feedback Matrix*

---

## Description

Validates a community matrix and computes its associated weighted feedback matrix.

## Usage

```
make.wfm(CM, status=FALSE, digits=1, sign=FALSE)
```

**Arguments**

CM	A valid community matrix.
status	Switches on an element-by-element progress indicator when set to TRUE. Set to FALSE by default.
digits	Indicates precision for elements in the weighted feedback matrix. By default, this is set to 1 significant digit.
sign	Switch to provide output as the signed value of the adjoint matrix elements divided by the absolute feedback matrix elements. The default value is FALSE

**Details**

The supplied matrix is validated as a community matrix, and the weighted feedback matrix is computed. Each element is equal to the absolute value of the corresponding element of the adjoint of the negative community matrix divided by the corresponding element of the total feedback matrix T. Resulting values range from 0 to 1.0, with values of magnitude of 0.5 or greater indicating that positive or negative feedback is expected to dominate (as per the sign of the adjoint value). Values of 1 indicate unambiguous effects of feedback, regardless of the quantitative magnitude of the system's linkages. Unresolvably ambiguous effects are represented by values between 0 and 0.5. The sign implementation differs from Dambacher's.

NOTE: weighted feedback, adjoint and absolute feedback matrices are transposed relative to the community effect matrix.

**Value**

The weighted feedback matrix for a community matrix

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

**References**

Dambacher, J. M., et al. (2003) Qualitative stability and ambiguity in model ecosystems. *The American Naturalist*, **161(6)**,876-888

Dambacher, J. M. and Li, H. W. and Rossignol, P. A. (2002) Relevance of community structure in assessing indeterminacy of ecological predictions. *Ecology*, **83(5)**,1372-1385

**See Also**

[make.cm](#), [make.adjoint](#), [make.T](#), [weighted.predictions](#).

**Examples**

```
## compute weighted feedback matrix
data(cm.dambacher)
make.wfm(cm.dambacher)
```

---

out.cm	<i>Output a Community Matrix or Community Effect Matrix in Conventional Format</i>
--------	--

---

## Description

Output a community matrix or community effect matrix object in conventional format to the console.

## Usage

```
out.cm(M)
```

## Arguments

M                    A matrix to be output.

## Details

This function outputs a community matrix or community effect matrix object in conventional format to the console where matrix elements are formatted from "-1", "0", "1", and "NA" to "-", "0", "+", and "?" respectively. This command does not validate the input matrix.

## Author(s)

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

## References

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

## See Also

[make.cm](#), [make.cem](#).

## Examples

```
## out a community matrix
data(cem.levins)
out.cm(cem.levins)
```

---

save.cm	<i>Save a Community Matrix</i>
---------	--------------------------------

---

**Description**

Validates and saves a community matrix in compressed binary format.

**Usage**

```
save.cm(CM, file)
```

**Arguments**

CM	A community matrix to be validated and saved.
file	a connection or a character string giving the name of the file to save.

**Details**

This function uses [validate.cm](#) and [save](#) to store a community matrix object.

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

**See Also**

[load](#).

**Examples**

```
## save a community matrix
data(cm.levins)
save.cm(cm.levins, file="Model_1.cm")
```

---

submatrix	<i>A Five-Variable Sub-Matrix of the First Five Variables of Dambacher's Matrix</i>
-----------	---

---

**Description**

This dataset represents a subset of the causal relationships defining Dambacher's five-variable system. Specifically, it contains the first five variables of the dataset `cm.dambacher`.

**Usage**

```
data(submatrix)
```

**Format**

A 5 by 5 (-1, 0, 1) matrix. The matrix is interpreted as the  $j$ th variable has the direct causal effect  $a[i,j]$  on  $i$ .

**References**

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

Dambacher, J. M., Li, H. M., and Rossignol, P. A. (2002) Relevance of community structure in assessing indeterminacy of ecological predictions. *Ecology*, **83**(5),1372-1385

---

validate.cem

*Validate a Community Effect Matrix*

---

**Description**

Validates a community effect matrix, returning descriptive errors if validation fails and no value otherwise.

**Usage**

```
validate.cem(CEM)
```

**Arguments**

CEM                    A potential community effect matrix to be tested.

**Details**

A community effect matrix is deemed valid if it is a square matrix, with no missing values, where each element has the value NA, -1, 0 or 1.

**Author(s)**

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

**References**

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

**See Also**

[make.cem](#).

## Examples

```
## assess community effect matrix
data(cm.levins)
make.cem(cm.levins) -> cem.levins
validate.cem(cem.levins)
```

---

validate.cm

*Validate a Community Matrix*

---

## Description

Validates a community matrix, returning descriptive errors if validation fails and nothing otherwise.

## Usage

```
validate.cm(CM)
```

## Arguments

CM                    A potential community matrix to be tested.

## Details

A community matrix is deemed valid if it is a square matrix, with no missing values, where each element has the value -1, 0 or 1, it is not a fully specified matrix, and there is at least one direct or indirect path from each element to each element.

## Author(s)

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

## References

Puccia, C. J. and Levins, R. (1986) *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge: Harvard University Press.

## See Also

[make.cm](#).

## Examples

```
## assess community matrix
data(cm.levins)
validate.cm(cm.levins)
```

---

weighted.predictions    *Compute the Matrix of Weighted Predictions*

---

### Description

Validates a community matrix and computes its associated weighted predictions matrix.

### Usage

```
weighted.predictions(CM, status=FALSE)
```

### Arguments

CM	A valid community matrix.
status	Switches on an element-by-element progress indicator when set to TRUE. Set to FALSE by default.

### Details

The supplied matrix is validated as a community matrix, and a weighted predictions matrix is computed. This matrix is equivalent to the transposed community effect matrix with some ambiguous elements resolved using the value of the corresponding feedback matrix. Such values are represented enclosed in parentheses. In keeping with the paper by Levins, Dambacher and Rossignol (expression 42 in the paper cited below), the matrix orientation is congruent with the weighted feedback matrix, and transposed to the community effect matrix.

### Value

The weighted prediction matrix for a community matrix

### Author(s)

Alexis Dinno <http://www.doyenne.com/LoopAnalyst/>

### References

Dambacher, J. M., et al. (2003) Qualitative stability and ambiguity in model ecosystems. *The American Naturalist*, **161**(6),876–888

Dambacher, J. M. and Li, H. W. and Rossignol, P. A. (2002) Relevance of community structure in assessing indeterminacy of ecological predictions. *Ecology*, **83**(5),1372–1385

### See Also

[make.cem](#), [make.wfm](#), [make.adjoint](#), and [make.T](#).

**Examples**

```
## compute community effect matrix, and note high prevalence of ambiguous predictions
data(cm.dambacher)
make.cem(cm.dambacher, out=TRUE)
## compute weighted prediction matrix, and note disambiguation of the cem
weighted.predictions(t(cm.dambacher))
```

# Index

## \*Topic **datasets**

- cem.levins, [3](#)
- cm.dambacher, [4](#)
- cm.DLR, [4](#)
- cm.levins, [5](#)
- submatrix, [18](#)

## \*Topic **file**

- graph.cem, [8](#)
- graph.cm, [9](#)
- out.cm, [17](#)
- save.cm, [18](#)

## \*Topic **graphs**

- cem.corr, [2](#)
- enumerate.loops, [5](#)
- enumerate.paths, [6](#)
- feedback, [7](#)
- make.adjoint, [10](#)
- make.cem, [11](#)
- make.clem, [12](#)
- make.cm, [13](#)
- make.T, [14](#)
- make.wfm, [15](#)
- validate.cem, [19](#)
- validate.cm, [20](#)
- weighted.predictions, [21](#)

- cem.corr, [2](#)
- cem.levins, [3](#)
- cm.dambacher, [4](#)
- cm.DLR, [4](#)
- cm.levins, [5](#)

- enumerate.loops, [5](#)
- enumerate.paths, [6](#)

- feedback, [7](#)

- graph.cem, [8](#)
- graph.cm, [9](#), [14](#)

- load, [18](#)

- make.adjoint, [10](#), [15](#), [16](#), [21](#)
- make.cem, [3](#), [9](#), [11](#), [11](#), [13–17](#), [19](#), [21](#)
- make.clem, [12](#)
- make.cm, [6–8](#), [10](#), [12](#), [13](#), [13](#), [17](#), [20](#)
- make.T, [11](#), [14](#), [16](#), [21](#)
- make.wfm, [11](#), [15](#), [15](#), [21](#)

- NA, [7](#)

- out.cm, [14](#), [17](#)

- save, [18](#)
- save.cm, [18](#)
- submatrix, [18](#)

- validate.cem, [19](#)
- validate.cm, [18](#), [20](#)

- weighted.predictions, [11](#), [15](#), [16](#), [21](#)