Package 'FlowCAr'

June 28, 2019

Title Flow Network Construction and Analysis

Type Package

Version 1.0.0

Author Christopher Waspe, Ruchit Mahabir, Ursula Scharler

Depends R (>= 3.1.0)

Imports LIM, limSolve, enaR

Maintainer Christopher Waspe <waspe60@gmail.com>

Description Creates and assesses a list of possible networks solved using 'LIM' (Linear Inverse Modelling) (Soetaert, Karline, and Dick Van Oevelen (2009) <doi:10.1007/s10021-009-9297-6>) and restructuring this list, enabling ENA (Ecological Network Analysis) to be performed on the flow network in R package 'enaR' (Borrett, Stuart R., and Matthew K. Lau (2014) <doi:10.1111/2041-210X.12282>).

License GPL-3

Encoding UTF-8

LazyData true

Suggests knitr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-06-28 12:50:06 UTC

R topics documented:

biomassVector 2
enaListGen 3
exportVector 4
FlowCAr 5
flowCheck 5
flowLimit 6
inputVector 7
internalFlowGen 8

LIMbuild 9
limListGen 10
livingVector 11
N4 12
N4LIM 12
N4LIMv 13
N4list 13
outputVector 14
PackNet 14
plotFlowRange 15
plotNodeFlows 16
plotRange 17
respVector 18

Index **19**

`biomassVector` *The creation of the biomass vector required for packing into the enaR network object*

Description

The generation of the biomass vector required for use in packing a lim file into an enaR network object.

Usage

```
biomassVector(limfile)
```

Arguments

`limfile`	This is the user generated LIM file created for analysis purposes.

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
biomassVecList <- biomassVector(limfile = N4)
```

enaListGen *A one step method which creates a list of the limfile, list of LIM networks and list of packed enaR entwork objects.*

Description

The main function enaListGen produces a list which contains the limfile (created through LIMbuild), the list of LIM networks (created through limListGen) and the list of enaR network objects. The enaR network objects are created from a specified LIM input file, ready for use in enaR.

Usage

```
enaListGen(Rfile, limfile, limList, limName, limListName, enaListName,
           storeAll, flowCheck, ...)
```

Arguments

Rfile	This is the user generated R text file defining the network.
limfile	The created limfile (LIMbuild)
limList	The list of networks solved using LIM (limListGen)
limName	If no limfile exists, the name of the limfile for the function to output
limListName	If no limList exists, the name of the limList for the function to output
enaListName	The name of the enaR network object list for the function to output
storeAll	Boolean. Indicates whether to store each enaR component as a separate list.
flowCheck	Boolean. Indicates whether to run flowCheck function.
...	Parameters avaliable through the use of the xsample function.

Value

A list containing three objects

[[1]] The limfile

[[2]] The list of LIM networks

[[3]] The list of enaR network objects

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
#All steps in one
data(N4list)
N4list <- enaListGen("4node.R", limName = "lim4",
                  limListName = "ll4", enaListName = "ena4",storeAll = FALSE,
                  flowCheck = FALSE, iter = 10000, jmp = NULL)

#creates the following object

###If the limfile and limList has been created
N4LIM <- data(N4LIM)
N4list <- enaListGen(limfile = N4,limList = N4LIM, enaListName = "ena4")
###If only the limfile has been created
N4list <- enaListGen(limfile = N4,limListName = "lim4node",enaListName = "ena4",
                       iter = 10000, jmp = NULL)
```

`exportVector` *The creation of the export vector required for packing into the enaR network object*

Description

The output created will be a single list of vectors that holds the cumulative value for the export elements per node.

Usage

```
exportVector(limfile,limList)
```

Arguments

`limfile`	This is the user generated LIM file created for analysis purposes.
`limList`	The list of networks solved using LIM (`limListGen`)

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
exportVectorList <- exportVector(limfile = N4, limList = N4LIM)
```

FlowCAr *Flow Network Construction and Analysis*

Description

This package allows for the creation of multiple possible network models based on a single FlowCAr input file.

Details

The following table indicates the functions used in the FlowCAr package.

Procedure	FlowCAr Functions
-	-
Network Construction	LIMbuild
	limListGen
	flowCheck
	flowLimit
Restructuring and Packing	inputVector
	exportVector
	respVector
	outputVector
	biomassVector
	livingVector
	PackNet
Main Function	enaListGen
Network Validation and Visualisation	plotNodeFlows
	plotRange
	plotFlowRange

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

References

#TBC

flowCheck *Checks the flow against the original distribution*

Description

Cross-checks the networks generated against the original flow network to ensure that flows are as they should be.

Usage

```
flowCheck(limfile,limList)
```

Arguments

limfile	This is the user generated LIM file created for analysis purposes.
limList	The list of networks solved using LIM (limListGen)

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
CrossCheckedFlows <- flowCheck(limfile = N4, limList = N4LIM)
```

---

flowLimit *Change the sampled ranged of a specific flow*

Description

If a flow seems to be falling outside the scope of your required work, a restriction can be placed on it to allow only required values to be considered in processing. When calling the function, the user is required to enter the list of LIM networks created in the '(limListGen)' function. Following that, the flow name must be specified. Thereafter, a minimum value and a maximum value can be specified to create a new list of matrices that adhere to the restrictions. (step 3) Alternatively, just a minimum value can be entered to create a list of matrices where all values above the minimum value for the specified flow are stored into a new list. (step 1) This works the same for entering just a maximum value, where all values for the specified flow below the maximum value entered are stored into a new list. (step 2) When entering just one value, the other value can be left blank or declared 'NULL' The output created will be a new list of LIM networks according to the contstraints required.

Usage

```
flowLimit(limfile,limList,flow,minVal,maxVal)
```

Arguments

limfile	This is the user generated LIM file created for analysis purposes.
limList	The list of networks solved using LIM (limListGen)
flow	This is the flow which is being restricted/limited
minVal	This is the minimum value to discard all values below
maxVal	This is the maximum value to discard all values above

Details

If certain flows need to be restricated due to some values not being correct, they can be discared here by selecting a range of values to keep for processing.

Value

[1]A matrix of all flow values, which can be used for visualisation

[[2]] A list of LIM networks

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
##------------------------------------------------------------##
## A simple four node network                                 ##
##------------------------------------------------------------##

viableNodesRangeTest <- flowLimit(limfile = N4,limList = N4LIM,
flow = "pp->invertebrate",minVal = 50, maxVal = 300)
viableNodesMinTest <- flowLimit(limfile = N4,limList = N4LIM,
flow = "pp->invertebrate",minVal = 50)
viableNodesMaxTest <- flowLimit(limfile = N4,limList = N4LIM,
flow = "pp->invertebrate",maxVal = 300)
```

---

inputVector — *The creation of the input vector required for packing into the enaR network object*

Description

The generation of an input vector required for use in packing a lim file into an enaR network object. The output created will be a single list of vectors that holds the cumulative value for the input elements per node.

Usage

```
inputVector(limfile, limList)
```

Arguments

limfile	This is the user generated LIM file created for analysis purposes.
limList	The list of networks solved using LIM (limListGen)

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
inputVectorList <-  inputVector(limfile = N4, limList = N4LIM)
```

---

internalFlowGen *Internal flowmatrix generation*

Description

Generates a flow matrix using just the internal nodes of a network for the enaR network object. The output created will be a single list of matrices that holds the internal flows for each interation.

Usage

```
internalFlowGen(limfile, limList)
```

Arguments

limfile	This is the user generated LIM file created for analysis purposes.
limList	The list of networks solved using LIM (limListGen)

Details

Generating the internal flowmatrix of each iteration of all flowmatrices.

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
InternalFlowMatrices <- internalFlowGen(limfile = N4, limList = N4LIM)
```

`LIMbuild` *Setting up and checking the LIM file formatting*

Description

Setting up of the LIM package and the enaR package is done in this function thus there is no need to do it prior e.g. using the functions `Read` or `Setup` from the LIM package. When preparing to use the LIM file, this method will prompt the user to verify that the information output correctly represents what was intened. Respiration node/element is required to be first in the externals list. To correctly identify living or non-living nodes, all non-living nodes are required to have 'NLNode' at the end of the node name i.e. detritus would be detNLNode or detritusNLNode.

Usage

```
LIMbuild(Rfile)
```

Arguments

`Rfile`	This is the user generated R text file defining the network.

Details

Used primarily for the FlowCAr package, this is a method simply for preparation and verification purposes that need not be run if entirely sure the requirements are met. If processing is conducted and results aren't what is expected, run this method to verify the lim file is set up correctly to fulfill the requiremnts.

Displayed in the console will be indications as to what data will be used for which calculation, however, it is up to the user to notice if anything is incorrect.

Value

See "value" in `Setup` Console output depicting information about what processing will be done.

Note

This function does no calculations and is not a necessity in order to proceed with the preparing of ENA network objects in this package.

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

References

–need to reference lim package here

Examples

```
##-------------------------------------------------------------##
## A simple four node network                                  ##
##-------------------------------------------------------------##

  data(N4)
  N4

  #Output generated using the following code

  N4 <- LIMbuild (Rfile = "4node.R")
```

`limListGen` *LIM Networks Generation*

Description

This function creates multiple possible network models from the limfile created from `LIMbuild`. Roughly 1000 networks/second. When calling the function, the user is required to enter the lim file, the number of iterations required and the jump vaue. Using the xsample function, it is applied to the LIM file to create interations(step 1). Using the flowmatrix function, it is applied to each iteration created in step 1 to produce a flowmatrix for the respective iteration(step 2). The output of step 1 will be a matrix of all iterations as and the output of step 2 will be a single list of respective flowmatrices.

Usage

```
limListGen(limfile,...)
```

Arguments

`limfile`	This is the user generated food web LIM file created for analysis purposes.
`...`	Parameters avaliable through the use of the `xsample` function.

Details

By using a food web LIM file that follows the required format,processing can be done to create multiple flowmatrices based on the input. Each flowmatrix is a plausible network model and depicts the flows between internal nodes as well as external nodes and the respiration element. These default amount of iteration created, as defined by `xsample`, is 3000. All matrices are stored into a single list.

Value

Returned is a list of iterations each containing a matrix which is a single network model. Also returned is a dataframe of how the data is read and interpreted.

Note

Generation times can differ depending on parameters applied and processing power.

NULL jmp values usually require more time.

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
##-------------------------------------------------------------------##
## A simple four node network with 150 iterations and default jump size ##
##-------------------------------------------------------------------##

N4LIM <- limListGen(limfile = N4, iter = 150)

##-------------------------------------------------------------------##
## A simple four node network with 10000 iterations and jump size of 1 ##
##-------------------------------------------------------------------##

N4LIM <- limListGen(limfile = N4, iter = 10000, jmp = 1)
```

`livingVector` *Generation of a single logical (TRUE/FALSE) vector based on the naming structure of the LIM file*

Description

Nodes with "NLnode" (not case sensitive) are viewed as non-living nodes and are thus "FALSE" values in the vector. All living nodes have a "TRUE" value.

Usage

```
livingVector(limfile)
```

Arguments

`limfile`	This is the user generated LIM file created for analysis purposes.

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
livingVecList <- livingVector(limfile = N4)
```

N4 *Four Node Network*

Description

Output from LIMbuild run on 4node.R

Format

A FlowCAr output containing a list of 28 variables

Source

FlowCAr package

References

FlowCAr ref

N4LIM *Solved Four Node Network*

Description

Output from limListGen run on N4. 1000 iterations. NULL jumps.

Format

A FlowCAr output containing a list of 2 variables.

Source

FlowCAr package

References

FlowCAr ref

N4LIMv *Restricted Solved Four Node Network*

Description

Output from flowLimit run on N4LIM, restricted Invertebrate flow.

Format

A FlowCAr output containing a list of 2 variables.

Source

FlowCAr package

References

FlowCAr ref

N4list *Solved and packed four node network*

Description

Output from FlowCAr main function, enaListGen run on "4node.R". 1000 iterations. NULL jumps.

Format

A FlowCAr output containing a list of 3 variables.

Source

FlowCAr package

References

FlowCAr ref

`outputVector` *The creation of the output vector required for packing into the enaR network object*

Description

Uses vector addition for the respective export vector and the respiration vector in the vector lists to produce a single vector list. The output created will be a single list of vectors that holds the output value (exports + resp) for each node per interation.

Usage

```
outputVector(limfile,expVec,rspVec)
```

Arguments

`limfile`	This is the user generated LIM file created for analysis purposes.
`expVec`	A list containing all export vectors generated
`rspVec`	A list of all respiration vectors generated

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
outputVecList <- outputVector(limfile = N4,
expVec = exportVectorList, rspVev = respVectorList)
```

`PackNet` *Packing possible LIM networks into a list.*

Description

Through the use of certain functions in this package, a list of enaR network objects are created. The output created is a list of network objects ready to be used in enaR. In order to 'pack' an enaR network object, you require a flow matrix of type 'matrix'. You will require input,respiration,export, and output vectors of type 'vector' Storage consists of the 'components' value from the LIM file called using "'limfile'$Components" where limfile is the name of the lim file read in. Living requires a vector consisting of true and false values to indicate whether the corresponding node is alive or dead. The LIMtoENA functions provide this requirements.

Usage

```
PackNet(internalflows, importsvec, respvec, exportsvec, outputsvec, biomassC, livingvec)
```

Arguments

internalflows	A list containing matrices of the internal flows of a network.
importsvec	A list containing vectors of import values for a network.
respvec	A list containing vectors of respiration values for a network.
exportsvec	A list containing vectors of export values for a network.
outputsvec	A list containing vectors of output values for a network.
biomassC	A vector of biomass values for a network.
livingvec	A vector of true and false values for living and non-living nodes in a network.

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
PackedNetworkObjectList <- PackNet(internalflows,
importsvec, respvec, exportsvec, outputsvec, biomassC, livingvec)
```

---

plotFlowRange — *Generate a plot which illustrates the proportion an individual flow has been sampled*

Description

The plotFlowRange function produces a plot illustrating how an individual flow of the users choosing has been sampled by the MCMC and xsample algorithms

Usage

```
plotFlowRange(limList,flow,...)
```

Arguments

limList	The list of networks solved using LIM (limListGen)
flow	The desired node name
...	Values depicted in plot

Value

A graphical plot

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
plotFlowRange(limList = N4LIM,flow = "pp->invertebrate")
#OR
plotFlowRange(limList = N4list$N4LIM,flow = "pp->invertebrate")
```

---

plotNodeFlows *Generates a plot illustrating the frequency of values for a required flow*

Description

The plotNodeFlows function produces a density plot, histogram and boxplot for each flow leaving the specified node

Usage

```
plotNodeFlows(limfile, limList, flowfrom, bins)
```

Arguments

limfile	This is the user generated LIM file created for analysis purposes.
limList	The list of networks solved using LIM (limListGen)
flowfrom	The node in which out-flows are required to be plotted.
bins	The amount of bins that splits the axis

Value

A graphical plot

Author(s)

Ruchit Mahabir Christopher Waspe Ursula Scharler

Examples

```
plotNodeFlows(limfile = N4, limList = N4LIM, flowfrom = "INVERTEBRATE")
#OR
plotNodeFlows(limfile = N4list$N4, limList = N4list$N4LIM, flowfrom = "INVERTEBRATE")
```

plotRange — *Graphical representation of flow ranges and possible network flow values*

Description

The plotRange function allows the user to generate a plot illustrating the range of each flow, the parsimonious solution for each flow, the average of the flow value calculated using the (limListGen) function, each iteration calculated using the (limListGen) function and the restricted flow values using the function (flowLimit)

Usage

```
plotRange(limfile, limList, limListLimit = NULL, legend, avcol="green",
          allcol="red", limitcol="blue", avpch=16, allpch=3, limitpch=3, ...)
```

Arguments

limfile	This is the user generated LIM file created for analysis purposes.
limList	The list of networks solved using LIM (limListGen)
limListLimit	The new list of restricted networks limited through (flowLimit)
avcol	Chosen colour for the point representing the average of the LIM iterations, default is "green".
allcol	Chosen colour for the points representing each of the LIM iterations, default is "red".
limitcol	Chosen colour for the points representing each of the restricted LIM iterations, default is "blue".
avpch	Chosen point type for the point representing the average of the LIM iterations, default is 16.
allpch	Chosen point type for the points representing the LIM iterations, default is 3.
limitpch	Chosen point type for the points representing the restricted LIM iterations, default is 3.
legend	Boolean. If TRUE will add a legend to the top right of the figure.
...	A list of arguments outlined in Plotranges

Value

A graphical plot

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

See Also

Plotranges

Examples

```
plotRange(limfile = N4,limList = N4LIM,legend = TRUE,xlab ="Flow Range Value")
#OR after enaListGen function has been used
plotRange(limfile = N4list$N4,limList = N4list$N4LIM,legend = TRUE,xlab ="Flow Range Value")
#After Restrictions has been applied (using flowLimit function)
N4LIMv <- data(N4LIMv)
plotRange(limfile = N4,limList = N4LIM,
limListLimit = N4LIMv,legend = TRUE,xlab ="Flow Range Value")

#OR enaListGen has been used
plotRange(limfile = N4list$N4,limList = N4list$N4LIM,
limListLimit = N4LIMv,legend = TRUE,xlab ="Flow Range Value")
```

---

respVector *The creation of the respiration vector required for packing into the enaR network object*

Description

Creates a list of vectors for the presented respiration element. The output created will be a single list of vectors that holds the respiration value for each node per interation.

Usage

```
respVector(limfile,limList)
```

Arguments

limfile	This is the user generated LIM file created for analysis purposes.
limList	The list of networks solved using LIM (limListGen)

Note

If structured correctly, the respiration element is the first item in the Externals list.

Author(s)

Ruchit Mahabir

Christopher Waspe

Ursula Scharler

Examples

```
respVectorList <-  respVector(limfile = N4 , limList = N4LIM)
```

Index

∗Topic **datasets**
N4, 12
N4LIM, 12
N4LIMv, 13
N4list, 13

biomassVector, 2, *5*

enaListGen, 3, *5*
exportVector, 4, *5*

FlowCAr, 5
flowCheck, *5*, 5
flowLimit, *5*, 6, *17*

inputVector, *5*, 7
internalFlowGen, 8

LIMbuild, *3*, *5*, 9, *10*
limListGen, *3–8*, 10, *15–18*
livingVector, *5*, 11

N4, 12
N4LIM, 12
N4LIMv, 13
N4list, 13

outputVector, *5*, 14

PackNet, *5*, 14
plot, *15*
plotFlowRange, *5*, 15
plotNodeFlows, *5*, 16
plotRange, *5*, 17
Plotranges, *17, 18*

Read, *9*
respVector, *5*, 18

Setup, *9*

xsample, *3*, *10*