

# Package ‘VOSONDash’

August 9, 2019

**Version** 0.4.4

**Title** User Interface for Collecting and Analysing Social Networks

**Description** A 'Shiny' application for the interactive visualisation and analysis of networks that also provides a web interface for collecting social media data using 'vosonSML'.

**Type** Package

**Imports** shiny (>= 1.3.2), magrittr, igraph (>= 1.2.2), rtweet (>= 0.6.8), vosonSML (>= 0.27.0), RColorBrewer, tm, wordcloud, syuzhet, httr, utils, graphics, lattice, httpuv

**Depends** R (>= 3.2.0)

**Encoding** UTF-8

**Author** Bryan Gertzel, Robert Ackland

**Maintainer** Bryan Gertzel <bryan.gertzel@anu.edu.au>

**License** GPL (>= 3)

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**URL** <https://github.com/vosonlab/VOSONDash>

**BugReports** <https://github.com/vosonlab/VOSONDash/issues>

**Repository** CRAN

**Date/Publication** 2019-08-09 10:20:03 UTC

## R topics documented:

VOSONDash-package	2
addAdditionalMeasures	2
applyCategoricalFilters	3
applyComponentFilter	3
applyGraphFilters	4
applyPruneFilter	5
getNetworkMetrics	5

getRedditUrlSubreddit . . . . .	6
getRedditUrlThreadId . . . . .	6
getVertexCategories . . . . .	7
getYoutubeVideoId . . . . .	7
loadDemoGraph . . . . .	8
mixmat . . . . .	9
runVOSONDash . . . . .	10
wordCloudPlot . . . . .	10
wordFreqChart . . . . .	11
wordSentChart . . . . .	11
wordSentValenceChart . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

VOSONDash-package	<i>Interface for collection and interactive analysis of social networks</i>
-------------------	---

---

### Description

VOSONDash provides functions and an interface in the form of an interactive R Shiny web application for the visualisation and analysis of network data. The app has sections for visualising and manipulating network graphs, performing text analysis, and displaying network metrics. It also has an interface for the collection of social network data using the vosonSML R package.

### Author(s)

Bryan Gertzel and Robert Ackland.

---

addAdditionalMeasures	<i>Add additional measures to graph as vertex attributes</i>
-----------------------	--

---

### Description

Adds degree, in-degree, out-degree, betweenness and closeness measures to graph as vertex attributes.

### Usage

```
addAdditionalMeasures(g)
```

### Arguments

**g** **igraph** graph object.

### Value

An igraph graph object.

---

`applyCategoricalFilters`*Filter out graph vertices not in selected category*

---

**Description**

This function removes vertices that are not in the selected categories values list or sub-categories.

**Usage**

```
applyCategoricalFilters(g, selected_cat, selected_subcats,  
  cat_prefix = "vosonCA_")
```

**Arguments**

`g` **igraph** graph object.  
`selected_cat` Character string. Selected vertex category without prefix.  
`selected_subcats` List. Selected sub-category values to include in graph.  
`cat_prefix` Character string. Category attribute prefix format to match. Default is "vosonCA\_".

**Value**

An igraph graph object.

**Examples**

```
## Not run:  
# return a graph containing only vertices that have the vertex category  
# attribute "vosonCA_Stance" value "liberal"  
g <- loadDemoGraph("DividedTheyBlog_40A1ist_release.graphml")  
  
g <- applyCategoricalFilters(g, "Stance", c("liberal"))  
  
## End(Not run)
```

---

`applyComponentFilter` *Filter out graph vertices not in component size range*

---

**Description**

This function removes any graph vertices that are in components that fall outside of the specified component size range.

**Usage**

```
applyComponentFilter(g, component_type = "strong", component_range)
```

**Arguments**

**g** **igraph** graph object.

**component\_type** Character string. Use strongly or weakly connected components by specifying "strong" or "weak". Ignored for undirected graphs. Default is "strong".

**component\_range** Numeric vector. Min and max values or size range of component.

**Value**

An igraph graph object.

---

applyGraphFilters	<i>Filter out graph vertices and edges from graph object that are isolates, multi edge or edge loops</i>
-------------------	--

---

**Description**

This function removes isolate vertices, multiple edges between vertices and or vertex edge loops from a graph.

**Usage**

```
applyGraphFilters(g, isolates = TRUE, multi_edge = TRUE,
  loops_edge = TRUE)
```

**Arguments**

**g** **igraph** graph object.

**isolates** Logical. Include isolate vertices in graph. Default is TRUE.

**multi\_edge** Logical. Include multiple edges between vertices in graph. Default is TRUE.

**loops\_edge** Logical. Include vertex edge loops in graph. Default is TRUE.

**Value**

An igraph graph object.

**Note**

Removing multiple edges or edge loops from a graph will simplify it and remove other edge attributes.

---

applyPruneFilter	<i>Prune vertices from graph by vertex id</i>
------------------	---

---

**Description**

This function removes a list of vertices from the graph object by vertex id value.

**Usage**

```
applyPruneFilter(g, selected_prune_verts)
```

**Arguments**

**g** **igraph** graph object.  
**selected\_prune\_verts** List. Selected vertex ids to remove.

**Value**

An igraph graph object.

---

getNetworkMetrics	<i>Get graph network metrics</i>
-------------------	----------------------------------

---

**Description**

Function creates a vector of calculated network metrics for a graph.

**Usage**

```
getNetworkMetrics(g, component_type = "strong")
```

**Arguments**

**g** **igraph** graph object.  
**component\_type** Character string. Use strongly or weakly connected components by specifying "strong" or "weak". Ignored for undirected graphs. Default is "strong".

**Value**

Network metrics as named vector.

---

getRedditUrlSubreddit *Get subreddit name from url*

---

**Description**

This function extracts the subreddit name from a reddit thread url.

**Usage**

```
getRedditUrlSubreddit(url)
```

**Arguments**

url                    Character string. Reddit thread url.

**Value**

Subreddit name as character string.

---

---

getRedditUrlThreadId *Get a reddit thread id from url*

---

**Description**

This function extracts the thread id from a reddit thread url.

**Usage**

```
getRedditUrlThreadId(url)
```

**Arguments**

url                    Character string. Reddit thread url.

**Value**

Reddit thread id as character string.

---

getVertexCategories     *Get a list of vertex category attribute names and values*

---

### Description

This function returns a list of graph vertex attribute names that match a category attribute prefix format and their unique values.

### Usage

```
getVertexCategories(g, cat_prefix = "vosonCA_")
```

### Arguments

**g**                    **igraph** graph object.  
**cat\_prefix**        Character string. Category attribute prefix format to match. Default is "vosonCA\_".

### Value

A named list of vertex category attributes and values.

### Examples

```
## Not run:  
# get a list of voson vertex categories and values  
g <- loadDemoGraph("DividedTheyBlog_40Alist_release.graphml")  
  
vcats <- getVertexCategories(g)  
  
# vcats  
# $Stance  
# [1] "conservative" "liberal"  
  
## End(Not run)
```

---

getYoutubeVideoId     *Get a youtube video id from url*

---

### Description

This function extracts the youtube video id from a youtube video url.

### Usage

```
getYoutubeVideoId(url)
```

**Arguments**

url                    Character string. Youtube video url.

**Value**

Video id as character string.

---

loadDemoGraph                    *Load package demonstration network graph*

---

**Description**

This function loads a demonstration network graph included in the `extdata` directory of the `VOSONDash` package by file name.

**Usage**

```
loadDemoGraph(fname)
```

**Arguments**

fname                    Character string. Name of demonstration graphml file.

**Value**

An `igraph` graph object.

**Examples**

```
## Not run:  
# load the "DividedTheyBlog" demonstration network graph  
g_demo_div <- loadDemoGraph("DividedTheyBlog_40A1ist_release.graphml")  
  
# load the "enviroActivistWebsites" demonstration network graph  
g_demo_env <- loadDemoGraph("enviroActivistWebsites_2006.graphml")  
  
## End(Not run)
```



---

mixmat	<i>Create a mixing matrix</i>
--------	-------------------------------

---

## Description

Function creates a mixing matrix by graph vertex attribute.

## Usage

```
mixmat(g, attrib, use_density = TRUE)
```

## Arguments

<code>g</code>	<b>igraph</b> graph object.
<code>attrib</code>	Character string. Vertex attribute or category.
<code>use_density</code>	Logical. Use edge density. Default is TRUE.

## Value

A mixing matrix.

## Note

Mixing matrix original function written by Gary Weissman. See: <https://gist.github.com/gweissman/2402741>.

## Examples

```
## Not run:  
# create a mixing matrix of the demonstration network based on vertex  
# categorical attribute for political stance "vosonCA_Stance"  
g <- loadDemoGraph("DividedTheyBlog_40Alist_release.graphml")  
  
mm <- mixmat(g, "vosonCA_Stance", use_density = FALSE)  
  
## End(Not run)
```

---

runVOSONDash	<i>Run the VOSON Dashboard Shiny Application</i>
--------------	--

---

### Description

This function launches the **VOSONDash** Shiny app in the default web browser.

### Usage

```
runVOSONDash(pkgStartupMsgs = FALSE, isLocal = NULL)
```

### Arguments

`pkgStartupMsgs` Logical. Display app package loading messages. Default is FALSE.  
`isLocal` Logical. Manually set app local or server mode flag.

### Value

None

---

wordCloudPlot	<i>Create a Word Cloud plot</i>
---------------	---------------------------------

---

### Description

This function creates a wordcloud plot of words in a text corpus.

### Usage

```
wordCloudPlot(corp, seed = NULL, min_freq = 1, max_words = 50,  
              pcolors = NULL)
```

### Arguments

`corp` **tm** package document [Corpus](#) object.  
`seed` Numeric. Seed value can be supplied to reproduce a word cloud layout.  
`min_freq` Numeric. Minimum word frequency to include a word in the word cloud. Default is 1.  
`max_words` Numeric. Maximum number of words to render in the word cloud. Default is 50.  
`pcolors` List. Colors to assign categorical variable in the plot. Default is NULL.

### Value

A wordcloud plot.

---

wordFreqChart	<i>Create a Word Frequency chart</i>
---------------	--------------------------------------

---

**Description**

This function creates a horizontal barchart of word frequencies in a text corpus.

**Usage**

```
wordFreqChart(corp, min_freq = 1, top_count = 20, pcolors = NULL)
```

**Arguments**

corp	<b>tm</b> package document <a href="#">Corpus</a> object.
min_freq	Numeric. Minimum frequency for a word to be included in the chart. Default is 1.
top_count	Numeric. Top count of words to render in word frequency chart. Default is 20.
pcolors	List. Colors to assign categorical variable in the plot. Default is NULL.

**Value**

A barchart plot.

---

wordSentChart	<i>Create an NRC Emotion chart</i>
---------------	------------------------------------

---

**Description**

This function creates a horizontal barchart measuring and sorting the eight NRC lexicon emotions in the text corpus. Emotions are measured as the proportion of the total value of the eight emotions in the text as a percentage.

**Usage**

```
wordSentChart(corp, pcolors = NULL)
```

**Arguments**

corp	<b>tm</b> package document <a href="#">Corpus</a> object.
pcolors	List. Colors to assign categorical variable in the plot. Default is NULL.

**Value**

A barchart plot.

**Note**

Uses the **syuzhet** package implementation of Saif Mohammad's NRC Emotion lexicon.

---

wordSentValenceChart *Create an NRC Positive and Negative Sentiment valence chart*

---

**Description**

This function creates a vertical barchart of negative, positive and the sum of the sentiment values or valence in a text corpus.

**Usage**

```
wordSentValenceChart(corp)
```

**Arguments**

corp **tm** package document [Corpus](#) object.

**Value**

A barchart plot.

**Note**

Uses the **syuzhet** package implementation of Saif Mohammad's NRC Emotion lexicon.

# Index

[addAdditionalMeasures](#), [2](#)  
[applyCategoricalFilters](#), [3](#)  
[applyComponentFilter](#), [3](#)  
[applyGraphFilters](#), [4](#)  
[applyPruneFilter](#), [5](#)

[Corpus](#), [10–12](#)

[getNetworkMetrics](#), [5](#)  
[getRedditUrlSubreddit](#), [6](#)  
[getRedditUrlThreadId](#), [6](#)  
[getVertexCategories](#), [7](#)  
[getYoutubeVideoId](#), [7](#)

[loadDemoGraph](#), [8](#)

[mixmat](#), [9](#)

[runVOSONDash](#), [10](#)

[VOSONDash \(VOSONDash-package\)](#), [2](#)  
[VOSONDash-package](#), [2](#)

[wordCloudPlot](#), [10](#)  
[wordFreqChart](#), [11](#)  
[wordSentChart](#), [11](#)  
[wordSentValenceChart](#), [12](#)