Figure Paper

Valeria Policastro

20 settembre 2019

# Fig.3 Plot of the two curves for the robustness

library("cowplot")

## Loading required package: ggplot2

##   
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggplot2':  
##   
## ggsave

library("robin")

## Loading required package: igraph

##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

library("gprege")

## Loading required package: gptk

## Loading required package: Matrix

## Loading required package: fields

## Loading required package: spam

## Loading required package: dotCall64

## Loading required package: grid

## Spam version 2.2-2 (2019-03-07) is loaded.  
## Type 'help( Spam)' or 'demo( spam)' for a short introduction   
## and overview of this package.  
## Help for individual functions is also obtained by adding the  
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##   
## Attaching package: 'spam'

## The following object is masked from 'package:Matrix':  
##   
## det

## The following objects are masked from 'package:base':  
##   
## backsolve, forwardsolve

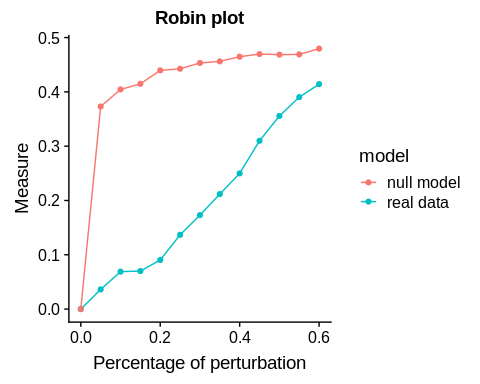
## Loading required package: maps

## See https://github.com/NCAR/Fields for  
## an extensive vignette, other supplements and source code

my\_file <- system.file("example/football.gml", package="robin")  
graph <- prepGraph(file=my\_file, file.format="gml")  
graphRandom <- random(graph=graph)  
proc <- robinRobust(graph=graph, graphRandom=graphRandom, measure="vi",   
 method="louvain", type="independent")

## [1] 31  
## [1] 61  
## [1] 92  
## [1] 123  
## [1] 153  
## [1] 184  
## [1] 215  
## [1] 245  
## [1] 276  
## [1] 306  
## [1] 337  
## [1] 368

plotRobin(graph=graph, model1=proc$Mean, model2=proc$MeanRandom,   
legend=c("real data", "null model"), measure="vi")



# Fig.4 Plot for the comparison of the two algorithms

comp <- robinCompare(graph=graph, method1="fastGreedy",  
 method2="louvain", measure="vi", type="independent")

## [1] 31  
## [1] 61  
## [1] 92  
## [1] 123  
## [1] 153  
## [1] 184  
## [1] 215  
## [1] 245  
## [1] 276  
## [1] 306  
## [1] 337  
## [1] 368

# Fig.5 Functional Data Analysis Test