nature sustainability

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41893-018-0190-1

In the format provided by the authors and unedited.

# An imperfect vision of indivisibility in the Sustainable Development Goals

**Philip J. K. McGowan[1], Gavin B. Stewart[1*], Graham Long[2,3] and Matthew J. Grainger[1,3]**

[1]School of Natural and Environmental Sciences, Newcastle University, Newcastle upon Tyne, UK. [2]School of Geography, Politics and Sociology, Newcastle University, Newcastle upon Tyne, UK. [3]These authors contributed equally to this work: Graham Long, Matthew J. Grainger.
*e-mail: Gavin.stewart@newcastle.ac.uk

# An imperfect vision of indivisibility in the Sustainable Development Goals

Philip J.K. McGowan[1], Gavin B. Stewart*[1], Graham Long[2 +], Matthew J. Grainger[1+].

[1] School of Natural and Environmental Sciences, Newcastle University, NE1 7RU, United Kingdom.
[2] School of Geography, Politics and Sociology, Newcastle University, NE1 7RU, United Kingdom.
[+]Joint last Authors
*corresponding author Gavin.stewart@newcastle.ac.uk

**Supplementary Methods**

We used ICSU-ISSC (2015)[1] which highlights the linkages (identified by 40 experts in the natural and social sciences) between Goals through targets, to develop a graphical model (i.e. a network) of the SDGs. A network consists of nodes (also known as vertices or points) connected by links (known as arcs, edges or lines). Each Goal was considered a node in the network linked by targets. Link weight was determined by summing the number of targets that linked two Goals. The SDGs represent a highly connected network with simple geometry and as such we were able to adopt a traditional approach to understanding node importance via centrality indicators (Table 1)[2,3]. The network was drawn as a directed graph in the igraph package[4] of the R programme[5].

Network metrics

Centrality is a key concept in network analysis, it provides some indication of which nodes are the most "important" in the set. There are a large number of centrality metrics available[6] all of which seek to quantify the position of the node in the network and infer importance from that position. Broadly, centrality metrics can be defined as distance-based, degree-based, eigen-based, neighbourhood-based or path-based[7]. The choice of which metric to use can be dependent on the network structure[8] but there is currently little consensus about which metrics should be preferred[7]. In the light of this uncertainty we use a variety of centrality measures to assess the importance of Goals in the SDGs network.

Network level metrics

A "connected" graph is one where all nodes can be reached from all other nodes[9].The most simple way to measure this connectivity is:

$$L/(N * (N - 1))$$

where L=Links and N=Nodes. This yields a score between 0 and 1 where a totally unconnected network scores 0 and a fully connected network 1. Reciprocity is defined as the proportion of mutual connections (where the opposite counterpart of a directed link is also included in the graph). This is calculated as:

$$\sum(i, j, (A. * A')ij) \,/\, \sum(i, j, Aij)$$

where A.*A' is the element-wise product of matrix A and its transpose. The score is a probability between 0 (no reciprocal links) and 1 (all possible reciprocal links).

Node level metrics

Degree is measured as the number of links going in to or out of a node. The more links a node has the more central it will be to a network. It is calculated as

$$C_D(v_i) = \sum_j a_{ij}$$

Where degree centrality ($C_D$) of each node ($v_i$) is the sum of the i-th row and j-th column in the adjacency matrix. In Supplementary Figure 1a the central node (node 3) has a degree of 4 compared to a degree of 1 for the other four nodes.

Strength is an extension of degree which incorporates the "weight" of a node's connected links combined with that of its nearest neighbours. It is calculated as

$$C_s(v_i) = \sum_j a_{ij} w_{ij}$$

Where the strength centrality ($C_s$) is the degree centrality with the weight (w) of i-th row and j-th column in the adjacency matrix taken in to account. In Supplementary Figure 1b node 7 has the highest degree (5) but when the weight of the links (as indicated by the thickness of the link) is accounted for node 2 is more central. The weight of the links for the SDGs network was determined by the number of targets between the focal Goals.

Closeness calculates the distance from a node to all other nodes in the network. A shorter distance indicates a more central node.

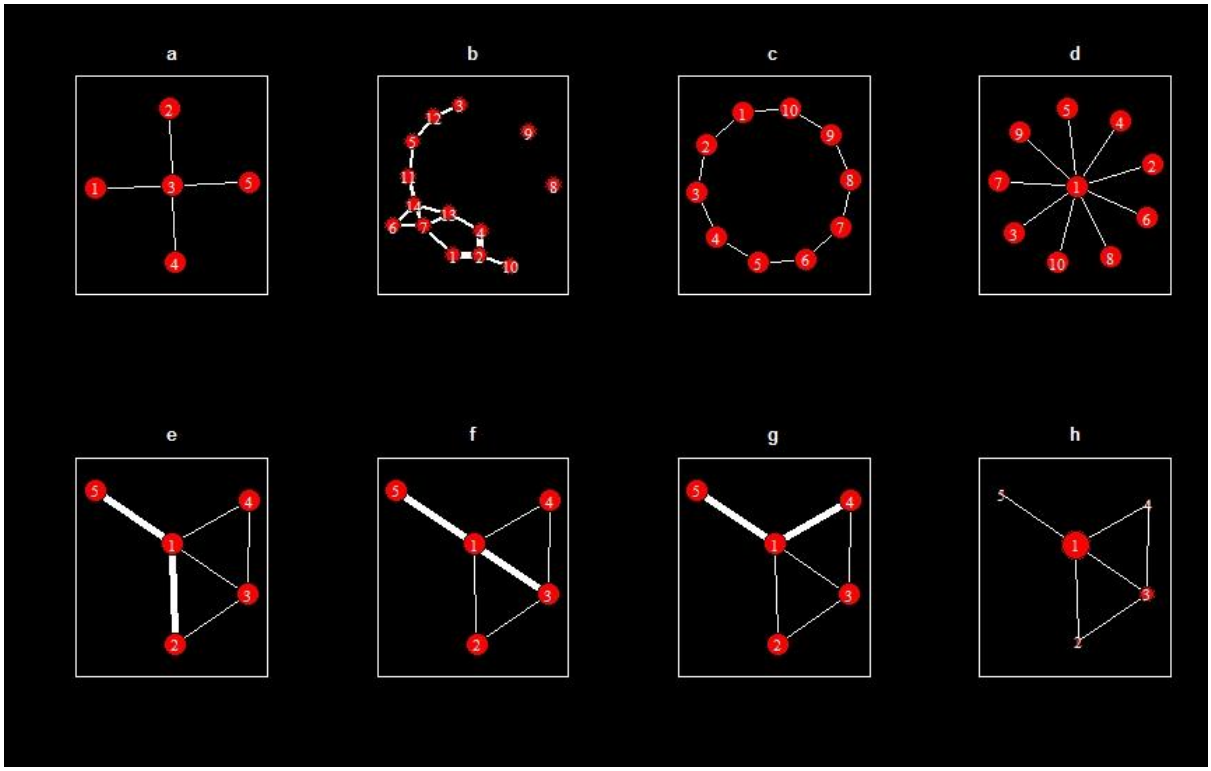
$$C_c(v_i) = \left[\frac{\sum_{j=1}^{N} d(v_i, v_j)}{N-1}\right]^{-1}$$

Closeness centrality ($C_c$) is the sum of the shortest paths (d) from the focal node ($v_i$) to all other nodes ($v_j$) . In the top graph in Supplementary Figure 1c all nodes have equal closeness (i.e. the distance between each node and all other nodes is equal). In Supplementary Figure 1d node 1 has the shortest distance to reach all other nodes and hence the highest closeness score.

Betweenness is the number of shortest paths from all nodes to all others that pass through the focal node. It is calculated as:

$$C_b\ (v_i)\ = \sum_{j \neq k} g_{jk}\ (v_i) / \ g_{jk}$$

Betweenness centrality ($C_b$) of each node ($v_i$) calculates the shortest path (g) between nodes j and k divided by the total number of shortest paths. In the bottom panel of Supplementary Figure 1 the shortest paths between nodes 2 and 5 (Supplementary Figure 1e), 3 and 5 (Supplementary Figure 1f), and 4 and 5 (Supplementary Figure 1g) are highlighted by a thicker white line. Supplementary Figure 1h shows the nodes weighted by their betweenness score. Node 3 has the highest betweenness score because it sits on the shortest pathways between nodes 2 and 5, 3 and 5, 4 and 5, and between 2 and 4. Node 3 also sits between node 2 and 4.




Supplementary Figure 1. Simplified examples of node level centrality measures (see the text for more details). a) Node degree centrality; b) Strength centrality; c) Closeness centrality-circle geometry; d) Closeness centrality-star geometry; e) Betweenness centrality 5-1-2 path; f) Betweenness centrality 5-1-3 path; g) Betweenness centrality 5-1-4 path, h) Betweenness centrality score.

Below is all the R code for generating the data, analysis and figures as presented.

```
##make sure that the global environment is clear
rm(list=ls())
options(warn=-1)
#Load the required libraries####
list.of.packages <- c("igraph", "ggplot2","png", "tidyverse", "RColorBrewe
r", "plotly", "reshape2")
new.packages <- list.of.packages[!(list.of.packages %in% installed.package
s()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
library(igraph)

##
## Attaching package: 'igraph'

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

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

library(ggplot2)
library(png)
library(tidyverse)

## -- Attaching packages ---------------------------------- tidyverse 1.2.
1 --

## v tibble  1.4.2     v purrr   0.2.5
## v tidyr   0.8.1     v dplyr   0.7.7
## v readr   1.1.1     v stringr 1.3.1
## v tibble  1.4.2     v forcats 0.3.0

## -- Conflicts ------------------------------------- tidyverse_conflicts(
) --
## x dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data
_frame()
## x purrr::compose()       masks igraph::compose()
## x tidyr::crossing()      masks igraph::crossing()
## x dplyr::filter()        masks stats::filter()
## x dplyr::groups()        masks igraph::groups()
## x dplyr::lag()           masks stats::lag()
## x purrr::simplify()      masks igraph::simplify()

library(RColorBrewer)
library(plotly)

##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:igraph':
##
##     groups

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout

library(reshape2)

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

##Get the data####
##set working directory##
setwd("~/Current_files/Manuscripts/SDGs")# change to your computer
##We have included the data directly inside the code to reduce the need fo
r extra files##
nodes <- data.frame("id"=c("G01","G02", "G03", "G04",
                           "G05", "G06", "G07", "G08",
                           "G09", "G10", "G11", "G12",
                           "G13", "G14", "G15", "G16"), "label"=c("Goal1",
"Goal2","Goal3","Goal4",
                                                                  "Goal5",
"Goal6",  "Goal7",  "Goal8",
                                                                  "Goal9",
"Goal10", "Goal11", "Goal12",
                                                                  "Goal13"
,"Goal14", "Goal15", "Goal16"))
links<-data.frame("from"=c("G02","G03", "G04", "G05", "G06", "G07", "G08",
"G09", "G10", "G12", "G13", "G14",
                           "G15", "G16", "G01", "G03", "G04", "G05", "G06"
,"G07", "G08","G09","G12","G13","G14","G15","G01","G02","G04","G05",
                           "G06","G07","G10","G12","G13","G14","G15","G01"
,"G02","G03","G05","G06","G07","G08","G09","G10","G12","G15","G16",
                           "G01","G02","G03","G04","G06","G07","G08","G10"
,"G12","G15","G16","G01","G02","G03","G04","G05","G07","G08","G09",
                           "G10","G12","G13","G14","G15","G01","G02","G03"
,"G04","G05","G06","G08","G09","G10","G12","G13","G14","G15","G16",
                           "G01","G02","G03","G04","G06","G07","G09","G10"
,"G12","G13","G14","G15","G02","G03","G04","G05","G06","G07","G08",
                           "G10","G12","G13","G14","G15","G16","G01","G02"
,"G03","G04","G05","G06","G07","G08","G09","G12","G13","G15","G01",
```

```
                            "G02","G03","G04","G05","G06","G07","G08","G09"
,"G12","G13","G14","G15","G02","G03","G04","G05","G06","G07","G08",
                            "G09","G13","G14","G15","G01","G02","G03","G04"
,"G05","G06","G07","G08","G09","G10","G12","G14","G15","G02","G03",
                            "G04","G06","G07","G08","G12","G13","G15","G02"
,"G03","G04","G06","G07","G08","G12","G13","G14","G01","G02","G03",
                            "G04","G05","G06","G07","G08","G10","G12","G13"
,"G14","G15"),
                  "to"=c("G01","G01","G01","G01","G01","G01","G01","G01","
G01","G01","G01","G01","G01","G01","G02","G02","G02","G02","G02",
                           "G02","G02","G02","G02","G02","G02","G02","G03","
G03","G03","G03","G03","G03","G03","G03","G03","G03","G04",
                           "G04","G04","G04","G04","G04","G04","G04","G04","
G04","G04","G04","G05","G05","G05","G05","G05","G05","G05",
                           "G05","G05","G05","G06","G06","G06","G06","G06","
G06","G06","G06","G06","G06","G06","G06","G07","G07","G07",
                           "G07","G07","G07","G07","G07","G07","G07","G07","
G07","G07","G07","G08","G08","G08","G08","G08","G08","G08",
                           "G08","G08","G08","G08","G09","G09","G09","G09","
G09","G09","G09","G09","G09","G09","G09","G09","G10","G10",
                           "G10","G10","G10","G10","G10","G10","G10","G10","
G10","G10","G11","G11","G11","G11","G11","G11","G11","G11",
                           "G11","G11","G11","G11","G12","G12","G12","G12","
G12","G12","G12","G12","G12","G12","G12","G13","G13","G13","G13",
                           "G13","G13","G13","G13","G13","G13","G13","G13","
G13","G14","G14","G14","G14","G14","G14","G14","G14","G14","G15",
                           "G15","G15","G15","G15","G15","G15","G15","G15","
G16","G16","G16","G16","G16","G16","G16","G16","G16","G16","G16",
                           "G16","G16"),"weight"=c(1,7,3,8,3,2,3,2,4,3,4,2,2
,2,1,8,2,4,3,2,3,2,5,2,4,3,1,7,3,9,3,1,2,3,3,1,1,1,6,
                                                   10,10,3,3,3,2,3,1,1,2,1,6
,9,3,3,1,3,3,2,1,6,2,1,8,3,3,4,3,2,1,5,4,3,2,1,15,15,1,3,
                                                   3,4,5,6,14,5,14,1,4,3,2,5
,3,3,4,5,2,5,3,5,1,5,7,4,2,3,5,3,2,4,2,3,1,1,2,8,10,2,4,
                                                   3,1,3,2,1,2,1,2,3,10,3,5,
3,4,3,6,4,5,8,4,9,11,2,4,3,5,3,6,6,7,2,2,3,5,3,3,3,3,3,
                                                   1,3,2,3,2,7,10,3,3,4,3,2,
2,2,9,12,3,3,3,3,5,7,9,1,10,12,1,7,3,5,3,5,3,1,4,2))


##Get SDG image files##
setwd("F:/SDGs/SDG images")# change to your computer - image files are pub
lically available from SDG website
## Get images##
img1 = readPNG("Goal1.png")
img2 = readPNG("Goal2.png")
img3 = readPNG("Goal3.png")
img4 = readPNG("Goal4.png")
img5 = readPNG("Goal5.png")
img6 = readPNG("Goal6.png")
img7 = readPNG("Goal7.png")
img8 = readPNG("Goal8.png")
img9 = readPNG("Goal9.png")
img10 = readPNG("Goal10.png")
```

```r
img11 = readPNG("Goal11.png")
img12 = readPNG("Goal12.png")
img13 = readPNG("Goal13.png")
img14 = readPNG("Goal14.png")
img15 = readPNG("Goal15.png")
img16 = readPNG("Goal16.png")

# create SDGs custom colour palette
SDGcol1 <- rgb(229/255, 36/255, 59/255, 1)
SDGcol2 <- rgb(221/255, 166/255, 58/255, 1)
SDGcol3 <- rgb(76/255, 159/255, 56/255, 1)
SDGcol4 <- rgb(197/255, 25/255, 45/255, 1)
SDGcol5<- rgb(255/255, 58/255, 33/255, 1)
SDGcol6<- rgb(38/255, 189/255, 226/255, 1)
SDGcol7<- rgb(252/255, 195/255, 11/255, 1)
SDGcol8<- rgb(162/255, 25/255, 66/255, 1)
SDGcol9<- rgb(253/255, 105/255, 37/255, 1)
SDGcol10<- rgb(221/255, 19/255, 103/255, 1)
SDGcol11<- rgb(253/255, 157/255, 36/255, 1)
SDGcol12<- rgb(191/255, 139/255, 46/255, 1)
SDGcol13<- rgb(63/255, 126/255, 68/255, 1)
SDGcol14<- rgb(10/255, 151/255, 217/255, 1)
SDGcol15<- rgb(86/255, 192/255, 43/255, 1)
SDGcol16<- rgb(0/255, 104/255, 157/255, 1)
SDGcol17<- rgb(25/255, 72/255, 106/255, 1)


SDG_pal=c(SDGcol1,SDGcol2,SDGcol3,SDGcol4,SDGcol5,SDGcol6,SDGcol7,SDGcol8,
SDGcol9,SDGcol10,
          SDGcol11,SDGcol12,SDGcol13,SDGcol14,SDGcol15,SDGcol16,SDGcol17)

#test the colours
#pie(rep(1, length(SDG_pal)), col = SDG_pal)

###Remove goal 17 (the admin goal)##
####goals only
#######
# Examine the data:
#head(nodes)
#head(links)
net <- graph_from_data_frame(d=links, vertices=nodes, directed=T)
# Examine the resulting object:
#class(net)
#net

# Removing loops from the graph:
net <- simplify(net)
#V(net)#Nodes
#E(net)#Links

##Network level metrics####
##connectence L/(N*(N-1)) where L is links, and N is nodes
N<-nrow(nodes); length(unique(nodes$id))
```

```
## [1] 16

L<-nrow(links); nrow(unique(links[,c("from", "to")]))

## [1] 192

(connectance<-L/(N*(N-1)))

## [1] 0.8

# Reciprocity of the graph
reciprocity(net)

## [1] 0.7916667

#Node level metrics
# Compute degree
simpNet.degree <- degree(net)
simpNet.degreeIN<-degree(net,mode="in")
simpNet.degreeOUT<-degree(net,mode="out")
# Compute strength
simpNet.strength <- strength(net)
# Weighted closeness
simpNet.closeness <- closeness(net)
# Weighted betweenness
simpNet.betweenness <- betweenness(net)
#table
#tab<-rbind(simpNet.degree,simpNet.degreeIN,simpNet.degreeOUT,simpNet.strength,simpNet.closeness,simpNet.betweenness)
#tab
```

## Figure 1

```
##Figure 1####
layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE))
simpNet1 <- set.graph.attribute(net, "layout", layout.circle)
# V(simpNet1)$raster = replicate(vcount(simpNet1), img1, simplify=FALSE)
# V(simpNet1)$raster[[2]] = img2
# V(simpNet1)$raster[[3]] = img3
# V(simpNet1)$raster[[4]] = img4
# V(simpNet1)$raster[[5]] = img5
# V(simpNet1)$raster[[6]] = img6
# V(simpNet1)$raster[[7]] = img7
# V(simpNet1)$raster[[8]] = img8
# V(simpNet1)$raster[[9]] = img9
# V(simpNet1)$raster[[10]] = img10
# V(simpNet1)$raster[[11]] = img11
# V(simpNet1)$raster[[12]] = img12
# V(simpNet1)$raster[[13]] = img13
# V(simpNet1)$raster[[14]] = img14
# V(simpNet1)$raster[[15]] = img15
# V(simpNet1)$raster[[16]] = img16

# plot(simpNet1, vertex.shape="raster", vertex.label=NA, edge.arrow.size=.01,edge.width=0.1, edge.color="black")
#
```

```
#
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal, main="Degree",
#      vertex.size=(30*simpNet.degree)/max(simpNet.degree),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# plot(simpNet1, vertex.label=NA,  vertex.color=SDG_pal,main="Strength",
#      vertex.size=(30*simpNet.strength)/max(simpNet.strength),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal, main="Closeness",
#      vertex.size=(30*simpNet.closeness)/max(simpNet.closeness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal, main="Betweenness"
,
#      vertex.size=(30*simpNet.betweenness)/max(simpNet.betweenness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
#
# dev.off()##
# ##label by number white background##
# radian.rescale <- function(x, start=0, direction=1) {
#   c.rotate <- function(x) (x + start) %% (2 * pi) * direction
#   c.rotate(scales::rescale(x, c(0, 2 * pi), range(x)))
# }
# n=16
# lab.locs <- radian.rescale(x=1:n, direction=-1, start=0)
# par(mfrow=c(2,2), mai = c(0.5,0.5,0.5,0.5),oma = c(.2, .2, 0.2, 0.2))
# plot(simpNet1, vertex.label=V(simpNet1),vertex.label.dist=4,vertex.label
.degree=lab.locs,vertex.color=SDG_pal, main="Degree",
#      vertex.size=(30*simpNet.degree)/max(simpNet.degree),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# plot(simpNet1, vertex.label=V(simpNet1),vertex.label.dist=3,vertex.label
.degree=lab.locs,  vertex.color=SDG_pal,main="Strength",
#      vertex.size=(30*simpNet.strength)/max(simpNet.strength),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# plot(simpNet1, vertex.label=V(simpNet1), vertex.label.dist=3,vertex.labe
l.degree=lab.locs,vertex.color=SDG_pal, main="Closeness",
#      vertex.size=(30*simpNet.closeness)/max(simpNet.closeness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
#
# plot(simpNet1, vertex.label=V(simpNet1), vertex.label.dist=3,vertex.labe
l.degree=lab.locs,vertex.color=SDG_pal, main="Betweenness",
#      vertex.size=(30*simpNet.betweenness)/max(simpNet.betweenness),
```

```
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
#dev.off()
##label by number black background##
#png("Figure1.png")
# or tiff("Figure1.tiff")
#pdf(file = "Figure1.pdf", 7, 7,"landscape", pointsize=7)
par(bg="black", mar=c(0,0,0,0))
radian.rescale <- function(x, start=0, direction=1) {
  c.rotate <- function(x) (x + start) %% (2 * pi) * direction
  c.rotate(scales::rescale(x, c(0, 2 * pi), range(x)))
}
n=16
lab.locs <- radian.rescale(x=1:n, direction=-1, start=0)
par(mfrow=c(2,2), mai = c(0.5,0.5,0.5,0.5),oma = c(.2, .2, 0.2, 0.2))
plot(simpNet1, vertex.label=V(simpNet1),vertex.label.dist=4,vertex.label.d
egree=lab.locs,vertex.label.color="white",vertex.color=SDG_pal, main="Degr
ee",
     vertex.size=(30*simpNet.degree)/max(simpNet.degree),
     edge.arrow.size=.01,
     edge.width=0.1,
     edge.color="white")
title("Degree",cex.main=1.3,col.main="white", line=1.5)
plot(simpNet1, vertex.label=V(simpNet1),vertex.label.dist=3,vertex.label.d
egree=lab.locs, vertex.label.color="white", vertex.color=SDG_pal,main="Str
ength",
     vertex.size=(30*simpNet.strength)/max(simpNet.strength),
     edge.arrow.size=.01,
     edge.width=0.1,
     edge.color="white")
title("Strength",cex.main=1.3,col.main="white", line=1.2)
plot(simpNet1, vertex.label=V(simpNet1), vertex.label.dist=3,vertex.label.
degree=lab.locs,vertex.label.color="white",vertex.color=SDG_pal, main="Clo
seness",
     vertex.size=(30*simpNet.closeness)/max(simpNet.closeness),
     edge.arrow.size=.01,
     edge.width=0.1,
     edge.color="white")
title("Closeness",cex.main=1.3,col.main="white", line=1.2)
plot(simpNet1, vertex.label=V(simpNet1), vertex.label.dist=3,vertex.label.
degree=lab.locs,vertex.label.color="white",vertex.color=SDG_pal, main="Bet
weenness",
     vertex.size=(30*simpNet.betweenness)/max(simpNet.betweenness),
     edge.arrow.size=.01,
     edge.width=0.1,
     edge.color="white")
title("Betweenness",cex.main=1.3,col.main="white", line=1.2)


# ##Label by SDG colour and numbers black background

#pdf(file = "Figure1.pdf", 10, 7,"landscape")
```

```
#png("Figure1.png",width=180, height=125, units="mm", res=500)

# # # or tiff("Figure1.tiff")
# par(bg="black")
# layout(matrix(c(2,3,1,4,5,1), nrow = 2, ncol = 3, byrow = TRUE))
# simpNet1 <- set.graph.attribute(net, "layout", layout.circle)
# V(simpNet1)$raster = replicate(vcount(simpNet1), img1, simplify=FALSE)
# V(simpNet1)$raster[[2]] = img2
# V(simpNet1)$raster[[3]] = img3
# V(simpNet1)$raster[[4]] = img4
# V(simpNet1)$raster[[5]] = img5
# V(simpNet1)$raster[[6]] = img6
# V(simpNet1)$raster[[7]] = img7
# V(simpNet1)$raster[[8]] = img8
# V(simpNet1)$raster[[9]] = img9
# V(simpNet1)$raster[[10]] = img10
# V(simpNet1)$raster[[11]] = img11
# V(simpNet1)$raster[[12]] = img12
# V(simpNet1)$raster[[13]] = img13
# V(simpNet1)$raster[[14]] = img14
# V(simpNet1)$raster[[15]] = img15
# V(simpNet1)$raster[[16]] = img16
#
# plot(simpNet1, vertex.shape="raster", vertex.label=NA, edge.arrow.size=.
01,edge.width=0.1, edge.color="white")
#
# radian.rescale <- function(x, start=0, direction=1) {
#   c.rotate <- function(x) (x + start) %% (2 * pi) * direction
#   c.rotate(scales::rescale(x, c(0, 2 * pi), range(x)))
# }
# n=16
# lab.locs <- radian.rescale(x=1:n, direction=-1, start=0)
# plot(simpNet1, vertex.label=V(simpNet1),vertex.label.dist=4,vertex.label
.degree=lab.locs,vertex.label.color="white",vertex.color=SDG_pal, main="De
gree",
#      vertex.size=(30*simpNet.degree)/max(simpNet.degree),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Degree",cex.main=1.2,col.main="white", line=1.2)
# plot(simpNet1, vertex.label=V(simpNet1),vertex.label.dist=3,vertex.label
.degree=lab.locs, vertex.label.color="white", vertex.color=SDG_pal,main="S
trength",
#      vertex.size=(30*simpNet.strength)/max(simpNet.strength),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Strength",cex.main=1.2,col.main="white", line=1.2)
# plot(simpNet1, vertex.label=V(simpNet1), vertex.label.dist=3,vertex.labe
l.degree=lab.locs,vertex.label.color="white",vertex.color=SDG_pal, main="C
loseness",
#      vertex.size=(30*simpNet.closeness)/max(simpNet.closeness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
```

```
#      edge.color="white")
# title("Closeness",cex.main=1.2,col.main="white", line=1.2)
# plot(simpNet1, vertex.label=V(simpNet1), vertex.label.dist=3,vertex.labe
l.degree=lab.locs,vertex.label.color="white",vertex.color=SDG_pal, main="B
etweenness",
#      vertex.size=(30*simpNet.betweenness)/max(simpNet.betweenness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Betweenness",cex.main=1.2,col.main="white", line=1.2)
# dev.off()
#

# ##Label by SDG colour black background
# dev.off()
# par(bg="black", mar=c(0,0,0,0))
# layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE))
# simpNet1 <- set.graph.attribute(net, "layout", layout.circle)
# # V(simpNet1)$raster = replicate(vcount(simpNet1), img1, simplify=FALSE)
# # V(simpNet1)$raster[[2]] = img2
# # V(simpNet1)$raster[[3]] = img3
# # V(simpNet1)$raster[[4]] = img4
# # V(simpNet1)$raster[[5]] = img5
# # V(simpNet1)$raster[[6]] = img6
# # V(simpNet1)$raster[[7]] = img7
# # V(simpNet1)$raster[[8]] = img8
# # V(simpNet1)$raster[[9]] = img9
# # V(simpNet1)$raster[[10]] = img10
# # V(simpNet1)$raster[[11]] = img11
# # V(simpNet1)$raster[[12]] = img12
# # V(simpNet1)$raster[[13]] = img13
# # V(simpNet1)$raster[[14]] = img14
# # V(simpNet1)$raster[[15]] = img15
# # V(simpNet1)$raster[[16]] = img16
# #
# plot(simpNet1, vertex.shape="raster", vertex.label=NA, edge.arrow.size=.
01,edge.width=0.1, edge.color="white")
#
#
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal, vertex.size=(30*si
mpNet.degree)/max(simpNet.degree),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Degree",cex.main=1.2,col.main="white", line=-1)
# plot(simpNet1, vertex.label=NA,  vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.strength)/max(simpNet.strength),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Strength",cex.main=1.2,col.main="white", line=-1)
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.closeness)/max(simpNet.closeness),
#      edge.arrow.size=.01,
```

```
#      edge.width=0.1,
#      edge.color="white")
# title("Closeness",cex.main=1.2,col.main="white", line=-1.5)
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.betweenness)/max(simpNet.betweenness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Betweenness",cex.main=1.2,col.main="white", line=-1.5)

# ####White background
# dev.off()
# par(bg="white", mar=c(0,0,0,0))
# layout(matrix(c(1,2,3,1,4,5), 2, 3, byrow = TRUE))
# simpNet1 <- set.graph.attribute(net, "layout", layout.circle)
# V(simpNet1)$raster = replicate(vcount(simpNet1), img1, simplify=FALSE)
# V(simpNet1)$raster[[2]] = img2
# V(simpNet1)$raster[[3]] = img3
# V(simpNet1)$raster[[4]] = img4
# V(simpNet1)$raster[[5]] = img5
# V(simpNet1)$raster[[6]] = img6
# V(simpNet1)$raster[[7]] = img7
# V(simpNet1)$raster[[8]] = img8
# V(simpNet1)$raster[[9]] = img9
# V(simpNet1)$raster[[10]] = img10
# V(simpNet1)$raster[[11]] = img11
# V(simpNet1)$raster[[12]] = img12
# V(simpNet1)$raster[[13]] = img13
# V(simpNet1)$raster[[14]] = img14
# V(simpNet1)$raster[[15]] = img15
# V(simpNet1)$raster[[16]] = img16
#
# plot(simpNet1, vertex.shape="raster", vertex.label=NA, edge.arrow.size=.
01,edge.width=0.1, edge.color="black")
#
#
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal, vertex.size=(30*si
mpNet.degree)/max(simpNet.degree),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# title("Degree",cex.main=1.2,col.main="black", line=-1)
# plot(simpNet1, vertex.label=NA,  vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.strength)/max(simpNet.strength),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# title("Strength",cex.main=1.2,col.main="black", line=-1)
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.closeness)/max(simpNet.closeness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# title("Closeness",cex.main=1.2,col.main="black", line=-1.5)
```

```
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.betweenness)/max(simpNet.betweenness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# title("Betweenness",cex.main=1.2,col.main="black", line=-1.5)
#
# ####White background change logo size
# dev.off()
# par(bg="white", mar=c(0,0,0,0))
# layout(matrix(c(1,2,3,1,4,5), 2, 3, byrow = TRUE))
# simpNet1 <- set.graph.attribute(net, "layout", layout.circle)
# V(simpNet1)$raster = replicate(vcount(simpNet1), img1, simplify=FALSE)
# V(simpNet1)$raster[[2]] = img2
# V(simpNet1)$raster[[3]] = img3
# V(simpNet1)$raster[[4]] = img4
# V(simpNet1)$raster[[5]] = img5
# V(simpNet1)$raster[[6]] = img6
# V(simpNet1)$raster[[7]] = img7
# V(simpNet1)$raster[[8]] = img8
# V(simpNet1)$raster[[9]] = img9
# V(simpNet1)$raster[[10]] = img10
# V(simpNet1)$raster[[11]] = img11
# V(simpNet1)$raster[[12]] = img12
# V(simpNet1)$raster[[13]] = img13
# V(simpNet1)$raster[[14]] = img14
# V(simpNet1)$raster[[15]] = img15
# V(simpNet1)$raster[[16]] = img16
#
# plot(simpNet1, vertex.shape="raster", vertex.size=25, vertex.label=NA, e
dge.arrow.size=.01,edge.width=0.1, edge.color="black")
#
#
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal, vertex.size=(30*si
mpNet.degree)/max(simpNet.degree),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# title("Degree",cex.main=1.2,col.main="black", line=-1)
# plot(simpNet1, vertex.label=NA,  vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.strength)/max(simpNet.strength),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# title("Strength",cex.main=1.2,col.main="black", line=-1)
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.closeness)/max(simpNet.closeness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="black")
# title("Closeness",cex.main=1.2,col.main="black", line=-1.5)
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.betweenness)/max(simpNet.betweenness),
#      edge.arrow.size=.01,
```

```
#      edge.width=0.1,
#      edge.color="black")
# title("Betweenness",cex.main=1.2,col.main="black", line=-1.5)

# ##Black background change logo size
# dev.off()
# par(bg="black", mar=c(0,0,0,0))
# layout(matrix(c(1,2,3,1,4,5), 2, 3, byrow = TRUE))
# simpNet1 <- set.graph.attribute(net, "layout", layout.circle)
# V(simpNet1)$raster = replicate(vcount(simpNet1), img1, simplify=FALSE)
# V(simpNet1)$raster[[2]] = img2
# V(simpNet1)$raster[[3]] = img3
# V(simpNet1)$raster[[4]] = img4
# V(simpNet1)$raster[[5]] = img5
# V(simpNet1)$raster[[6]] = img6
# V(simpNet1)$raster[[7]] = img7
# V(simpNet1)$raster[[8]] = img8
# V(simpNet1)$raster[[9]] = img9
# V(simpNet1)$raster[[10]] = img10
# V(simpNet1)$raster[[11]] = img11
# V(simpNet1)$raster[[12]] = img12
# V(simpNet1)$raster[[13]] = img13
# V(simpNet1)$raster[[14]] = img14
# V(simpNet1)$raster[[15]] = img15
# V(simpNet1)$raster[[16]] = img16
#
# plot(simpNet1, vertex.size=30,vertex.shape="raster", vertex.label=NA, ed
ge.arrow.size=.01,edge.width=0.1, edge.color="white")
#
#
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal, vertex.size=(30*si
mpNet.degree)/max(simpNet.degree),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Degree",cex.main=1.2,col.main="white", line=-1)
# plot(simpNet1, vertex.label=NA,  vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.strength)/max(simpNet.strength),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Strength",cex.main=1.2,col.main="white", line=-1)
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.closeness)/max(simpNet.closeness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Closeness",cex.main=1.2,col.main="white", line=-1.5)
# plot(simpNet1, vertex.label=NA, vertex.color=SDG_pal,
#      vertex.size=(30*simpNet.betweenness)/max(simpNet.betweenness),
#      edge.arrow.size=.01,
#      edge.width=0.1,
#      edge.color="white")
# title("Betweenness",cex.main=1.2,col.main="white", line=-1.5)
```

## Figure 2

```
##Figure 2####
##get rank data
SDGS_ranks<-data.frame("Goal (shortened name)"= c("Energy","Hunger","Educa
tion","Production",
                                                  "Poverty","Health","Wate
r","Industry",
                                                  "Climate","Biodiversity
land","Cities",
                                                  "Peace","Gender","Econom
ics","Inequalities",
                                                  "Biodiversity water"),
                       "Goal number"=c("G07", "G02" ,"G04" ,"G12", "G01",
"G03", "G06", "G09" ,
                                       "G13", "G15" ,"G11" ,"G16" ,"G05" ,
"G08", "G10","G14"),
                       "Degree"=c(1,3,3,5,9,5,2,11,5,10,16,15,11,5,13,14),
                       "Strength"=c(2,3,8,4,15,1,7,12,10,11,16,13,5,8,14,6
),
                       "Closeness"=c(6,5,3,4,2,15, 8,  7, 11,  1, 16,  9,
14,  9, 11, 13),
                       "Betweenness"=c(8,9,2,14,1,14,5,3,4,12,14,6,13,10,7
, 11))

#melt the data to rearrange for plotting with ggplot2
SDGS_heat<-melt(SDGS_ranks, id.vars = c("Goal..shortened.name.","Goal.numb
er"))
names(SDGS_heat)<-c("Goal (shortened name)", "Goal number",
                    "variable", "Rank")
#black & white
# ggplot(SDGS_heat, aes(reorder(SDGS_heat$`Goal (shortened name)`,
#                               Rank, sum,order = TRUE),
#                       reorder(variable,Rank,sum,order=TRUE))) +
#   geom_tile(aes(fill = Rank), color = "white")+
#   scale_fill_gradient(high="black",low="white", trans="reverse")+
#   ylab("Centrality measure") +
#   xlab("Goal") +
#   theme(legend.title = element_text(size = 10),
#         legend.text = element_text(size = 10),
#         plot.title = element_text(size=12),
#         axis.title=element_text(size=12,face="bold"),
#         axis.text.x = element_text(angle = 90, hjust = 1)) +
#   labs(fill = "Rank")

# #Blues


#####
SDGS_heat$Goalnum<-paste(SDGS_heat$`Goal number`,SDGS_heat$`Goal (shortene
d name)`)
head(SDGS_heat)
```

```
##   Goal (shortened name) Goal number variable Rank        Goalnum
## 1                Energy         G07   Degree    1     G07 Energy
## 2                Hunger         G02   Degree    3     G02 Hunger
## 3             Education         G04   Degree    3  G04 Education
## 4            Production         G12   Degree    5 G12 Production
## 5               Poverty         G01   Degree    9    G01 Poverty
## 6                Health         G03   Degree    5     G03 Health
```

```
#####Figure 2 single colour
cols<-c("#f7fbff","#deebf7","#c6dbef","#9ecae1","#6baed6", "#4292c6", "#21
71b5", "#084594")#blues

# p <- plot_ly(
#   x = SDGS_heat$Goalnum, y = SDGS_heat$variable,
#   z = SDGS_heat$Rank, colors = cols,type = "heatmap",reversescale=F
# )
# p
#ordered by rank
# ggplot(SDGS_heat, aes(x =reorder(SDGS_heat$Goalnum,Rank, sum,order = TRU
E),
#                       reorder(variable,Rank,sum,order=TRUE))) +
#   geom_tile(aes(fill = Rank)) + theme_bw()+
#   scale_fill_gradientn(colours =cols, trans="reverse")+
#   theme(axis.title.x=element_blank(),
#         axis.title.y=element_blank(),
#         axis.text.x = element_text(angle = 45, hjust = 1),
#         legend.title = element_text(size = 10),
#         legend.text = element_text(size = 10),
#         axis.line = element_blank(),
#         panel.border = element_blank())
#pdf(file = "Figure2.pdf", 7,7,pointsize=7)
#png("Figure2.png")
# or tiff("Figure2.tiff")

ggplot(SDGS_heat, aes(x =SDGS_heat$Goalnum,variable)) +
  geom_tile(aes(fill = Rank)) + theme_bw()+
  scale_fill_gradientn(colours =cols)+
  theme(text = element_text(size=10),
        axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1),
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 10),
        axis.line = element_blank(),
        panel.border = element_blank())


#dev.off()
```

## Supplementary Figure 1

```
##Supplementary Methods####
########################
#pdf(file = "FigureS1.pdf", 7,7,pointsize=7)
```

```r
layout(matrix(c(1,2,3,4,5,6,7,8), 2, 4, byrow = TRUE))
#png("FigureS1.png")
par(bg="black")
col="white"
##degree
##create graph - star shape
el <- matrix( c(1, 3, 2, 3,5,3,4,3), nc = 2, byrow = TRUE)
g1<-graph_from_edgelist(el, directed = F)
plot(g1, vertex.color="red",vertex.size=30 ,edge.color=col, vertex.label.color=col)
title("a",col.main="white",line=1)
box(lty = 1, col = 'white')

#dev.off()
degree(g1)

## [1] 1 1 4 1 1

#strength
#set the seed to be repeatable
#png("FigureS2.png")
#par(bg="black", mar=c(0,0,0,0))
set.seed(10010)
g <- sample_gnm(n=14, m=15)
#plot(g,vertex.color="red",vertex.size=20 ,edge.color=col, vertex.label.color=col)
degree(g)

##  [1] 2 3 1 2 2 2 5 0 0 1 3 2 3 4

#E(g)
E(g)$weight <- c(5,5,2,2,2,2,1,2,2,2,2,2,2,2,2)
plot(g, vertex.color="red",vertex.size=20 ,edge.color=col, vertex.label.color=col, edge.width=E(g)$weight)
title("b",col.main="white",line=1)
box(lty = 1, col = 'white')

strength(g)

##  [1]  7 12  2  7  4  4  9  0  0  2  5  4  6  8

degree(g)

##  [1] 2 3 1 2 2 2 5 0 0 1 3 2 3 4

#dev.off()
#closeness
#example taken from http://igraph.org/r/doc/closeness.html)
#png("FigureS3.png")
# par(bg="black", mar=c(0,0,0,0))
# par(mfrow=c(2,1))
g <-make_ring(10)
plot(g,vertex.color="red",vertex.size=30 ,edge.color=col, vertex.label.color=col)
title("c",col.main="white",line=1)
```

```r
box(lty = 1, col = 'white')

g2 <-make_star(n=10)
plot(g2,vertex.color="red",vertex.size=30 ,edge.color=col, vertex.label.co
lor=col, edge.arrow.mode=0)
title("d",col.main="white",line=1)
box(lty = 1, col = 'white')

closeness(g)

##  [1] 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04

closeness(g2, mode="all")

##  [1] 0.11111111 0.05882353 0.05882353 0.05882353 0.05882353 0.05882353
##  [7] 0.05882353 0.05882353 0.05882353 0.05882353

#dev.off()
#betweeness
#png("FigureS4.png")
# par(bg="black", mar=c(0,0,0,0))
# layout(matrix(c(1,2,3,4,4,4), 2, 3, byrow = TRUE))
# #layout(matrix(c(1,2,3,1,4,5), 2, 3, byrow = TRUE))
set.seed(1234)
el <- matrix( c(2, 1,2,3, 3, 1,3,4,4,1,1,5), nc = 2, byrow = TRUE)
#el
g <- graph_from_edgelist(el)
coords <- layout.davidson.harel(g)
shortest_paths(g,2,5)

## $vpath
## $vpath[[1]]
## + 3/5 vertices, from 8a896a6:
## [1] 2 1 5
##
##
## $epath
## NULL
##
## $predecessors
## NULL
##
## $inbound_edges
## NULL

shortest_paths(g,3,5)

## $vpath
## $vpath[[1]]
## + 3/5 vertices, from 8a896a6:
## [1] 3 1 5
##
##
## $epath
## NULL
```

```
##
## $predecessors
## NULL
##
## $inbound_edges
## NULL
```

```
shortest_paths(g,4,5)
```

```
## $vpath
## $vpath[[1]]
## + 3/5 vertices, from 8a896a6:
## [1] 4 1 5
##
##
## $epath
## NULL
##
## $predecessors
## NULL
##
## $inbound_edges
## NULL
```

```
#E(g)
#path from 2 to 5
E(g)$thickness<-c(5,1,1,1,1,5)
plot(g,layout = coords,vertex.color="red",vertex.size=30 ,edge.color=col,
edge.width= E(g)$thickness,vertex.label.color=col,edge.arrow.mode=0)
title("e",col.main="white",line=1)
box(lty = 1, col = 'white')

#path from 3 to 5
E(g)$thickness<-c(1,1,5,1,1,5)
#path from 4 to 5
plot(g,layout = coords,vertex.color="red",vertex.size=30 ,edge.color=col,
edge.width= E(g)$thickness,vertex.label.color=col,edge.arrow.mode=0)
title("f",col.main="white",line=1)
box(lty = 1, col = 'white')

E(g)$thickness<-c(1,1,1,1,5,5)
plot(g,layout = coords,vertex.color="red",vertex.size=30 ,edge.color=col,
edge.width= E(g)$thickness,vertex.label.color=col,edge.arrow.mode=0)
title("g",col.main="white",line=1)
box(lty = 1, col = 'white')
plot(g,layout = coords,vertex.color="red",vertex.size= 10*betweenness(g)+1
0 ,edge.color=col,vertex.label.color=col,edge.arrow.mode=0)
title("h",col.main="white",line=1)
box(lty = 1, col = 'white')
```

```
#dev.off()
#rm(list=ls())
#options(warn=0)
```

**Supplementary References**


1. ICSU-ISSC. *Review of the Sustainable Development Goals: The Science Perspective* (International Council for Science (ICSU), Paris, 2015).

2. Gibbons, A. *Algorithmic Graph Theory.* (Cambridge University Press, 1985).

3. Borgatti, S. & Everett, M. *Social Networks* **28**, 466–484 (2006).

4. Csardi G. & Nepusz, T. *InterJournal, Complex Systems* **1695** (2006).

5. R Core Development Team. *R: A language and environment for statistical computing* (R Foundation for Statistical Computing, 2018).

6. Jalili, M. et al. *PLoS ONE* **10,** e0143111 (2015).

7. Ashtiani M. et al. *BMC Systems Biology* **12**, 80 (2018).

8. Freeman L. *Sociometry* 40, 35-41 (1977).

9. Kolaczyk, E. *Statistical Analysis of Network Data, Springer Series in Statistics, 79* (Springer Science, 2009).