

# INA smartsurv

K. A. Garrett, University of Florida

## Identifying key sampling nodes, as part of an impact network analysis

Impact network analysis (INA) is designed to address multiple aspects of linked socioeconomic networks (spread of ideas, money, influence, etc.) and biophysical networks (spread of new varieties, certified seed, pathogens, pests, etc.). This document introduces the smartsurv function, and gives results from the first experiment in the following paper:

Garrett, K. A. 2021. Impact network analysis and the INA R package: Decision support for regional management interventions. *Methods in Ecology and Evolution*, in press.

The example given in this vignette addresses identifying the best locations in a network for sampling to quickly detect a spreading species: for example, a pathogen or new plant genotype. A node is better for sampling if the pathogen is likely to be detected at that node before the pathogen has spread very far through the network. For example, an isolated node would be a poor sampling choice because the pathogen would likely spread widely before it reaches the isolated node.

Examples of this type of analysis also appear in the following papers.

Andersen, K. F., C. E. Buddenhagen, P. Rachkara, R. Gibson, S. Kalule, D. Phillips, and K. A. Garrett. 2019. Modeling epidemics in seed systems and landscapes to guide management strategies: The case of sweetpotato in Northern Uganda. *Phytopathology* 109:1519-1532.

Buddenhagen, C. E., J. F. Hernandez Nopsa, K. F. Andersen, J. Andrade-Piedra, G. A. Forbes, P. Kromann, S. Thomas-Sharma, P. Useche, and K. A. Garrett. 2017. Epidemic network analysis for mitigation of invasive pathogens in seed systems: Potato in Ecuador. *Phytopathology* 107:1209-1218.

In the simplest version of the analysis, an invasive species is equally likely to enter the epidemic/spread network at any node, and then moves through the network. In this version of the analysis, it enters the network at only one node (the “introduction node”). Each node in the network (each potential “sampling node”) is then evaluated to determine how early the pathogen can be detected, in terms of how many nodes remain uninfected when the pathogen’s presence in the network is detected at the sampling node. A summary of the performance of a node indicates how early the pathogen is detected at that node, after entering the network at each node in turn. That is, if there are  $N$  nodes, there would be  $N$  potential introduction nodes for the pathogen and  $N$  analyses of how early the pathogen could be detected at the sampling node being evaluated.

An analysis like this, focused on the biophysical network, may be useful even when information is limited about the socioeconomic network. Or information about the socioeconomic network may be incorporated in the form of weights indicating the likelihood that a node is the point of introduction of the species in the network. Nodes associated with less-informed managers may be at higher risk of being the introduction node for an invasive species. Or environmental conditions might make initial establishment of a pathogen more likely at some nodes than others. In this weighted analysis, the performance of a sampling node would be evaluated taking the weights (likelihood of introduction) into account, so that overall performance would be weighted toward the results for more likely introduction nodes.

This vignette starts from the simplest versions of analyses and builds up to a more complete analysis. If you would prefer to jump to a full analysis, you could skip to the description of the function smartsurv, below.

## Getting the INA package from GitHub

An updated version of INA is available on GitHub.

Note that if you do not already have the package devtools, you will need to install it once (using the command commented out below)

Then you will need to install INA from GitHub (using the command commented out below)

```
# use the following three commented commands if you do not yet have  
# INA or need to update the version  
  
#install.packages("devtools") # use this command if you do not yet have devtools  
#installed  
  
#library(devtools)  
  
#devtools::install_github("GarrettLab/INA")  
  
library(INA)  
  
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(viridis)
```

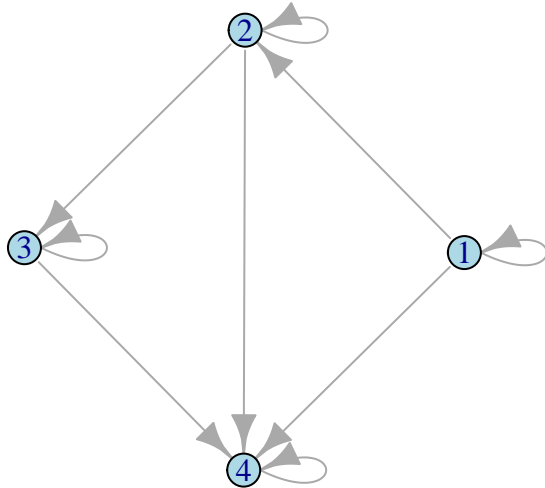
```
## Loading required package: viridisLite
```

## A simple example, illustrating the functions used by the main function `smartsurv`

Consider a simple adjacency matrix, which will be used to illustrate the functions.

Note that this function currently assumes that the diagonal of the adjacency matrix is composed of 1's, meaning that once a node is invaded, it stays invaded

```
Amat <- matrix(c(1,0,0,0,1,1,0,0,0,1,1,0,1,1,1),nrow=4,ncol=4)  
  
#Make an igraph-formatted version of the adjacency matrix  
Amati <- graph.adjacency(Amat)  
  
#Create a layout for the matrix to make comparison of the structure easier in  
#examples throughout this vignette  
layout.ex <- layout.kamada.kawai(Amati)  
  
plot(Amati, layout=layout.ex, vertex.color='lightblue')
```



```
# each potential link has probability 0.7 of existing in one realization
sAmat <- Amat * 0.7
```

### Illustration for a single starting node (onestart function)

For a given introduction node (only one at this point), onestart yields two outputs for one realization.

Output object outmat has rows=time steps, columns = nodes, and entries = invasion status for each node at each time step (1 = invaded, 0 = not invaded).

Output object sampnodes has rows = nodes, first column = time step at which invasion is detected (Inf if node is never reached), second column = the number of nodes invaded at the time of detection, third column = the number of nodes not invaded at the time of detection.

```
# in this case, the second node was selected as the starting point
onestart(adjmat=Amat, start.choice=2, stoch=F)
```

```
## $outmat
##   [,1] [,2] [,3] [,4]
## 1    0    1    0    0
## 2    0    1    1    1
## 3    0    1    1    1
##
## $sampnodes
##      firstt numinvt numnotinvt
## [1,]     Inf        3          1
## [2,]        1        1          3
## [3,]        2        3          1
## [4,]        2        3          1
```

```
onestart(adjmat=sAmat, start.choice=2, stoch=T) # stochastic version
```

```
## $outmat
##   [,1] [,2] [,3] [,4]
## 1    0    1    0    0
## 2    0    1    1    1
```

```
## 3    0    1    1    1
##
## $samplnodes
##      firstt numinvt numnotinvt
## [1,]     Inf      3          1
## [2,]      1      1          3
## [3,]      2      3          1
## [4,]      2      3          1
```

The function `multistart` summarizes this analysis across all potential starting points.

The output is a matrix with rows = starting nodes (introduction nodes), columns = sampling nodes, and entries = number of nodes not invaded by time detected at the sampling node having started at the introduction node.

For the stochastic case, the result is only based on one realization (function `smartsurv`, described below, evaluates multiple realizations).

Note that the diagonal contains threes, because when a node is both the starting point and the sampling point, the invasion will only have reached one node before detection and the remaining three remain free of the invasion.

```
multistart(adjmat=Amat, stoch=F)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   3   1   0   1
## [2,]   1   3   1   1
## [3,]   2   2   3   2
## [4,]   3   3   3   3
```

```
multistart(adjmat=sAmat, stoch=T)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   3   1   0   1
## [2,]   2   3   2   2
## [3,]   2   2   3   2
## [4,]   3   3   3   3
```

```
# For the deterministic version, let's plot the network again with the coloring
# of the nodes indicating how many nodes remain free of the invasion
```

```
# For this we take the mean across all the potential starting nodes (giving all
# the starting nodes equal weight, equal probability of being the introduction
# node, for now)
```

```
temp <- multistart(adjmat=Amat, stoch=F)
temp
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   3   1   0   1
## [2,]   1   3   1   1
## [3,]   2   2   3   2
## [4,]   3   3   3   3
```

```

numinvfree <- colMeans(temp) # trait for coloring nodes
numinvfree

## [1] 2.25 2.25 1.75 1.75

nnodes <- dim(Amat)[1] # the total number of nodes

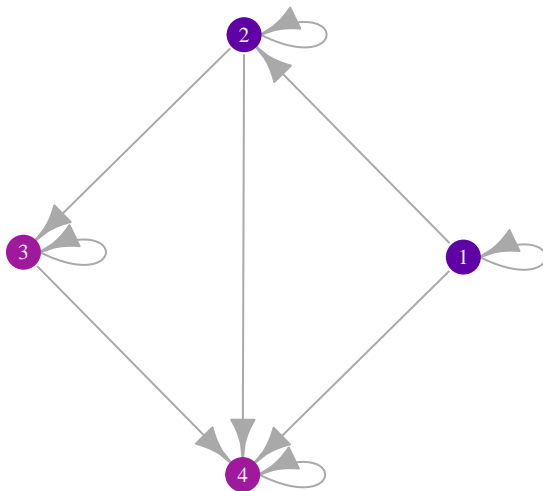
pal1 <- viridis_pal(option="C", direction = -1, end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

plot(
  Amati,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.7,
  vertex.label.color = 'white',
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.ex,
  sub = 'Darker colors indicate better sampling nodes',
  main = 'All starting nodes equally likely (deterministic)'
)

```

## All starting nodes equally likely (deterministic)



Darker colors indicate better sampling nodes

## The function `smartsurv` performs the same analysis, for the specified number of realizations

The first component of the output from `smartsurv` is the response for each simulation - `nrealz` outputs from `multistart` (in this example, 10 realizations). The next component of the output is the mean across the realizations. The final component is the variance across realizations.

Because columns represent potential sampling nodes, the mean of a column (in the second component of the output, `meanarr`) is the mean number of nodes free of invasion by the time the invasion would be detected at that node. The mean of the variance in a column (in the third component of the output, `vararr`) indicates how consistently the mean number of nodes would have been free of invasion.

```
ss.out <- smartsurv(adjmat = sAmat,
                   stoch = T,
                   nrealz = 100)
```

```
# mean across the realizations
ss.out$meanarr
```

```
##      [,1] [,2] [,3] [,4]
## [1,] 3.00 1.86 1.33 1.64
## [2,] 1.77 3.00 1.94 1.77
## [3,] 2.59 2.59 3.00 2.59
## [4,] 3.00 3.00 3.00 3.00
```

```
# variance across the realizations
ss.out$vararr
```

```
##      [,1]      [,2]      [,3]      [,4]
## [1,] 0.0000000 0.5458586 1.2940404 0.8185859
## [2,] 0.6637374 0.0000000 0.5418182 0.6637374
## [3,] 0.2443434 0.2443434 0.0000000 0.2443434
## [4,] 0.0000000 0.0000000 0.0000000 0.0000000
```

```
# a few of the individual realizations
ss.out$outarr[, , 1:3]
```

```
## , , 1
```

```
##
```

```
##      [,1] [,2] [,3] [,4]
## [1,] 3    2    2    2
## [2,] 2    3    2    2
## [3,] 2    2    3    2
## [4,] 3    3    3    3
```

```
##
```

```
## , , 2
```

```
##
```

```
##      [,1] [,2] [,3] [,4]
## [1,] 3    3    3    3
## [2,] 1    3    1    1
## [3,] 2    2    3    2
## [4,] 3    3    3    3
```

```
##
```

```
## , , 3
##
##      [,1] [,2] [,3] [,4]
## [1,]    3    1    0    1
## [2,]    1    3    2    1
## [3,]    3    3    3    3
## [4,]    3    3    3    3
```

```
# the mean number of nodes free of invasion for each sampling node (averaged
# across all the potential starting nodes)
numinvfree <- colMeans(ss.out$meanarr) # trait for coloring nodes
numinvfree
```

```
## [1] 2.5900 2.6125 2.3175 2.2500
```

```
# Coloring nodes based on their role in sampling: the mean number of nodes free
# of invasion when invasion detected at sampling node
```

```
nnodes <- nrow(Amat) # the total number of nodes
```

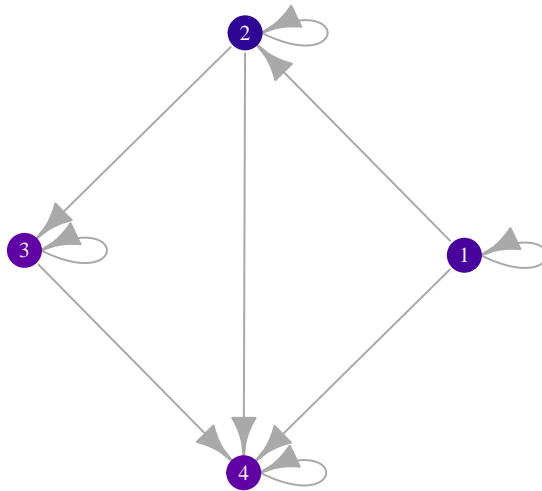
```
pal1 <- viridis_pal(option="C", direction = -1, end = 0.8)
```

```
# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
```

```
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]
```

```
plot(
  Amati,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.7,
  vertex.label.color = 'white',
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.ex,
  sub = 'Darker colors indicate better sampling nodes',
  main = 'All starting nodes equally likely (realz=100)'
)
```

## All starting nodes equally likely (realz=100)



Darker colors indicate better sampling nodes

## Weighted likelihood of a node being an entry point into network

The calculations of means up to this point have been based on the assumption that each node is equally likely to be the starting node.

The function `smartsurv.weight` uses output from `smartsurv` to evaluate the mean number of nodes free of invasion when invasion would be detected at each potential sampling node, where potential starting nodes may have different probabilities of functioning as starting nodes. For example, higher risk of being the starting node might result from a node's role as a port, weather conditions associated with a node, or lack of management information at the node.

Suppose the weights indicating the probability that each of four nodes is the introduction node are as follows.

```
# weights indicating the relative likelihood that each of four nodes is the  
# introduction node for an invasion  
wtvec.ex <- c(0.001, 0.2, 0.798, 0.001)  
  
msf.outex <- smartsurv(adjmat = sAmat,  
                      stoch = T,  
                      nrealz = 10)  
  
temp <-  
  smartsurv.weight(ss.out = msf.outex,  
                  adjmat = sAmat,  
                  wtvec = wtvec.ex)  
  
# Note that smartsurv.weight gives output that includes number of nodes invasion  
# free
```



```

numinvfree <- temp$tsampfree$sampfree # trait for coloring nodes
numinvfree

## [1] 2.5206 2.7594 2.7583 2.5193

nnodes <- dim(Amat)[1] # the total number of nodes

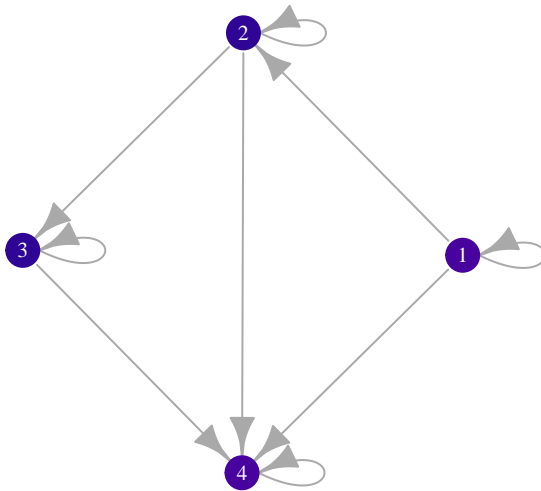
pal1 <- viridis_pal(option = "C",
                    direction = -1,
                    end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

plot(
  Amati,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.7,
  vertex.label.color = 'white',
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.ex,
  sub = 'Darker colors indicate better sampling nodes',
  main = 'Node 3 more likely starting node'
)

```

## Node 3 more likely starting node



Darker colors indicate better sampling nodes

```
# Suppose node 1 is more likely to be the starting node

# weights indicating the relative likelihood that each of four nodes is the
# introduction node for an invasion
wtvec.ex <- c(0.793, 0.2, 0.001, 0.001)

msf.outex <- smartsurv(adjmat = sAmat,
                      stoch = T,
                      nrealz = 10)

temp <-
  smartsurv.weight(ss.out = msf.outex,
                  adjmat = sAmat,
                  wtvec = wtvec.ex)

# Note that startwt gives output that includes number of nodes invasion free
numinvfree <- temp$tsampfree$sampfree # trait for coloring nodes
numinvfree

## [1] 2.6844 2.2707 1.7534 1.9707

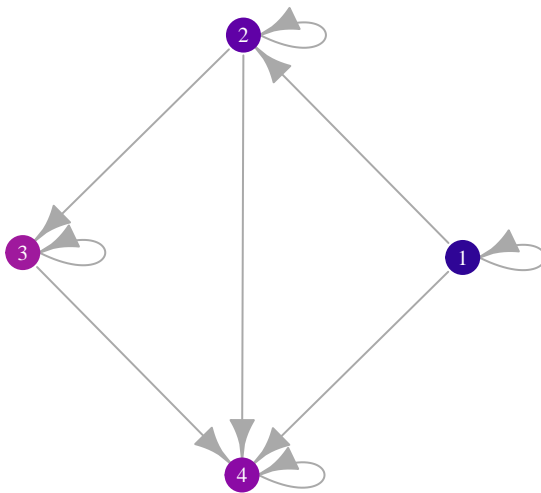
# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]
```

```

plot(
  Amati,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.7,
  vertex.label.color = 'white',
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.ex,
  sub = 'Darker colors indicate better sampling nodes',
  main = 'Node 1 more likely starting node'
)

```

## Node 1 more likely starting node



## Darker colors indicate better sampling nodes

```

#smartsurv.weight(ss.out=msf.outex, adjmat5=sAmat, wtvec=wtvec.ex,
#nodenam=c("KS", "NE", "ND", "SD"))

```

Note that weighting has changed the outcome in this example. When node 3 is much more likely to be the starting location, the importance for sampling shifted.

## Applying the analyses to some larger networks

In this experiment, key nodes for sampling are identified for a set of biophysical network types.

The importance of nodes for sampling by the criteria developed here is evaluated for nine scenarios, each combination of three types of networks and three types of weighting. The three types of networks considered here are random (Erdos-Renyi), small world, and power law (Barabasi). The three types of weighting of potential starting nodes are unweighted, weights proportional to node degree, and weights inversely proportional to node degree.

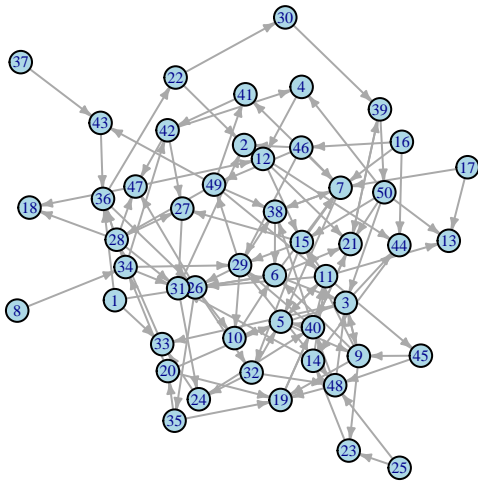
## Random network

Random network: nodes equally likely to be the starting node

```
# generate a directed random network with 50 nodes
random.eq <- sample_gnp(
  n = 50,
  p = 0.05,
  directed = T,
  loops = F
)

layout.random <- layout_with_kk(random.eq)

plot(
  random.eq,
  edge.arrow.size = 0.3,
  vertex.size = 10,
  vertex.label.cex = 0.5,
  layout = layout.random,
  vertex.color = 'lightblue'
)
```



```
# find the value of nodes for sampling based on a deterministic model
```

```
temp1 <- as_adjacency_matrix(random.eq, sparse = F)
diag(temp1) <- 1 # nodes invaded stay invaded
temp2 <- multistart(adjmat = temp1, stoch = F)

numinvfree <- colMeans(temp2) # trait for coloring nodes
numinvfree
```

```
## [1] 9.38 24.34 27.14 18.62 33.04 27.56 30.90 9.38 28.04 31.02 32.18 22.30
## [13] 23.80 22.30 30.76 9.40 9.38 21.40 25.36 16.72 29.48 17.08 19.86 26.60
## [25] 9.38 33.06 27.36 23.00 30.68 14.18 27.68 27.92 26.44 14.20 23.40 21.28
## [37] 9.38 29.32 23.68 32.22 22.94 19.36 16.84 27.10 22.18 10.20 27.88 27.84
## [49] 22.46 17.50
```

```

nnodes <- dim(temp1)[1] # the total number of nodes

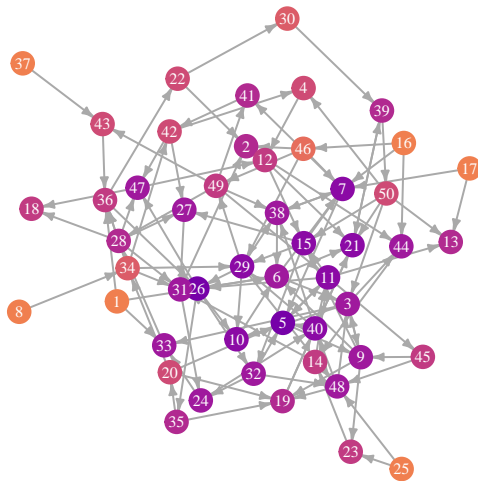
pal1 <- viridis_pal(option = "C",
                    direction = -1,
                    end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

plot(
  random.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.random,
  sub = 'Darker colors indicate better sampling nodes',
  main = 'Random graph: all starting nodes equally likely (deterministic)'
)

```

## Random graph: all starting nodes equally likely (deterministic)



Darker colors indicate better sampling nodes

## Random network: likelihood a node is the starting node is proportional to node degree

As an example, suppose that for the same network, the likelihood that the species enters the network at a given node is proportional to node degree. This might be reasonable if, for example, nodes of high degree in the known network are also more likely to be linked in other, unknown networks.

```
# find the degree for each node and construct a proportional weight from it
wtintermed <- degree(random.eq, mode = "all", loops = F)
wtvec1 <- wtintermed / sum(wtintermed)

temp1 <- as_adjacency_matrix(random.eq, sparse = F)
diag(temp1) <- 1 # nodes invaded stay invaded
msf.outex <- smartsurv(adjmat = temp1,
                      stoch = F,
                      nrealz = 1)

temp <-
  smartsurv.weight(ss.out = msf.outex,
                  adjmat = temp1,
                  wtvec = wtvec1)

# Note that startwt gives output that includes number of nodes invasion free
numinvfree <- temp$tsampfree$sampfree # trait for coloring nodes
numinvfree

## [1] 8.296748 21.422764 27.845528 16.813008 33.182927 28.796748 31.585366
## [8] 7.947154 28.760163 31.552846 32.601626 21.272358 23.500000 20.735772
## [15] 31.203252 8.308943 8.121951 21.186992 25.804878 16.028455 28.886179
## [22] 14.910569 19.833333 25.178862 8.121951 33.813008 27.093496 23.406504
## [29] 30.959350 11.869919 25.837398 28.252033 25.947154 12.756098 23.560976
## [36] 19.552846 7.947154 29.861789 22.073171 31.873984 23.349593 18.369919
## [43] 15.967480 25.918699 22.178862 9.146341 28.191057 27.825203 22.825203
## [50] 16.044715

nnodes <- dim(temp1)[1] # the total number of nodes

pal1 <- viridis_pal(option = "C",
                   direction = -1,
                   end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

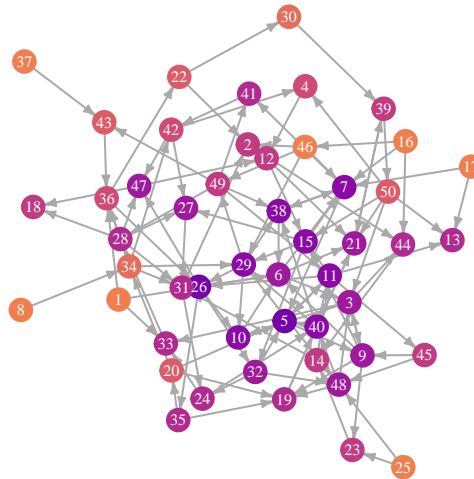
plot(
  random.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
```

```

edge.curved = F,
vertex.color = nodecol,
layout = layout.random,
sub = 'Darker colors indicate better sampling nodes'
)
title(main = 'Random graph: higher degree nodes more likely starting nodes (deterministic links)',
      cex.main = 0.9)

```

## Random graph: higher degree nodes more likely starting nodes (deterministic links)



### Darker colors indicate better sampling nodes

Note: “deterministic links” here refers to how the links generated when the random network was formed are maintained rather than having some probability of disappearing as they would in the stochastic option for the model.

**Random network:** likelihood a node is the starting node is inversely proportional to node degree

In this case, suppose the likelihood that a node is the introduction point for an invasive in the systems is inversely proportional to node degree. This could be logical if higher node degree for potential invasive movement is associated with higher node degree for information flow, for example.

```

# find the degree for each node and construct a weight inversely proportional to
# it
wtintermed <- degree(random.eq, mode = "all", loops = F)
hilo <- sum(range(wtintermed))
wtintermed <- hilo - wtintermed

wtvec1 <- wtintermed / sum(wtintermed)

```

```
temp1 <- as_adjacency_matrix(random.eq, sparse = F)
diag(temp1) <- 1 # nodes invaded stay invaded
msf.outex <- smartsurv(adjmat = temp1,
                      stoch = F,
                      nrealz = 1)
```

```
temp <-
  smartsurv.weight(ss.out = msf.outex,
                  adjmat = temp1,
                  wtvec = wtvec1)
```

```
# Note that startwt gives output that includes number of nodes invasion free
numinvfree <- temp$tsampfree$sampfree # trait for coloring nodes
numinvfree
```

```
## [1] 10.42913 27.16535 26.45669 20.37008 32.90157 26.36220 30.23622 10.76772
## [9] 27.34252 30.50394 31.77165 23.29528 24.09055 23.81496 30.33071 10.45669
## [17] 10.59843 21.60630 24.92913 17.38976 30.05512 19.18110 19.88583 27.97638
## [25] 10.59843 32.33071 27.61811 22.60630 30.40945 16.41732 29.46457 27.59843
## [33] 26.91732 15.59843 23.24409 22.95276 10.76772 28.79528 25.23622 32.55512
## [41] 22.54331 20.31890 17.68504 28.24409 22.18110 11.22047 27.57874 27.85433
## [49] 22.10630 18.90945
```

```
nnodes <- dim(temp1)[1] # the total number of nodes
```

```
pal1 <- viridis_pal(option = "C",
                   direction = -1,
                   end = .8)
```

```
# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
```

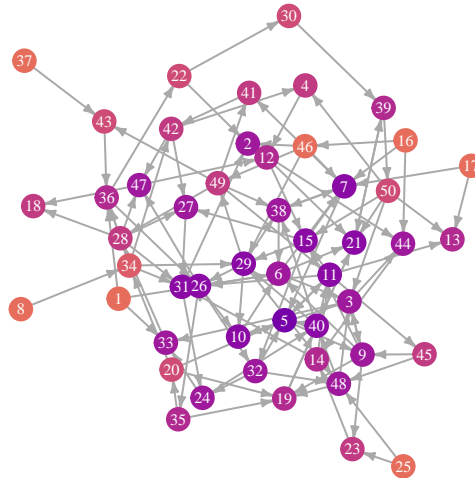
```
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]
```

```
plot(
  random.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.random,
  sub = 'Darker colors indicate better sampling nodes'
)
```

```
title(main = 'Random graph: higher degree nodes less likely starting nodes (deterministic links)',
      cex.main = 0.9)
```



Random graph: higher degree nodes less likely starting nodes (deterministic links)



Darker colors indicate better sampling nodes

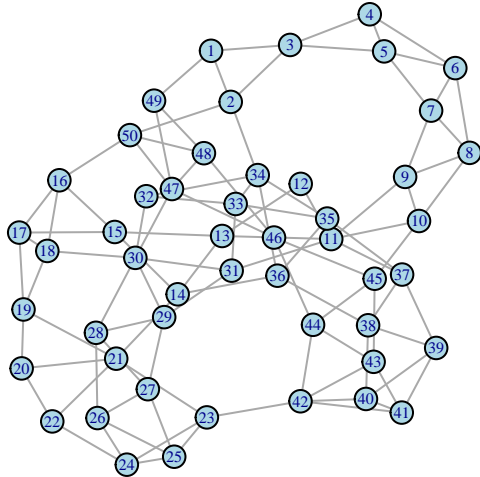
Small world network

Small world network: nodes equally likely to be the starting node

```
# generate a small world network with 50 nodes
small.eq <- sample_smallworld(
  dim = 1,
  size = 50,
  nei = 2,
  p = 0.05,
  loops = F
)

layout.small <- layout_with_kk(small.eq)

plot(
  small.eq,
  edge.arrow.size = 0.3,
  vertex.size = 10,
  vertex.label.cex = 0.5,
  layout = layout.small,
  vertex.color = 'lightblue'
)
```



```

# find the value of nodes for sampling based on a deterministic model

temp1 <- as_adjacency_matrix(small.eq, sparse = F)
diag(temp1) <- 1 # nodes invaded stay invaded
temp2 <- multistart(adjmat = temp1, stoch = F)

numinvfree <- colMeans(temp2) # trait for coloring nodes
numinvfree

## [1] 16.16 21.24 15.28 10.64 11.88 10.92 14.38 15.40 21.10 23.04 28.52 19.52
## [13] 24.06 24.60 22.10 21.20 19.64 22.08 18.32 14.46 21.94 13.92 19.84 14.72
## [25] 15.36 14.20 16.08 18.98 22.76 27.98 27.78 22.46 21.94 23.90 22.38 23.52
## [37] 22.78 20.78 17.76 17.16 16.34 20.28 18.38 20.38 21.14 21.72 24.52 19.94
## [49] 17.34 22.84

nnodes <- dim(temp1)[1] # the total number of nodes

pal1 <- viridis_pal(option = "C",
                    direction = -1,
                    end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

plot(
  small.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout_small,

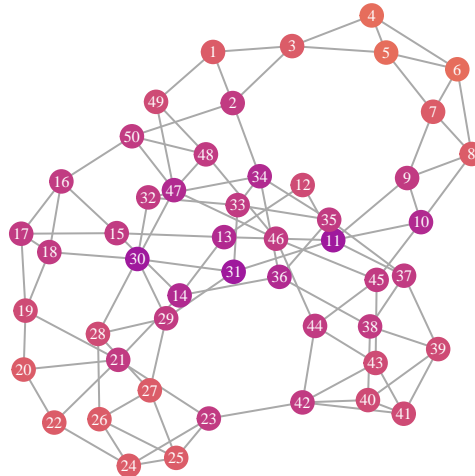
```

```

sub = 'Darker colors indicate better sampling nodes',
main = 'Small world: all starting nodes equally likely (deterministic links)'
)

```

## Small world: all starting nodes equally likely (deterministic links)



Darker colors indicate better sampling nodes

Note that this small world network is not directed, and movement through the network can be more rapid than in the random graph example. There is a path between any two nodes.

**Small world network: likelihood a node is the starting node is proportional to node degree**

```

# find the degree for each node and construct a proportional weight from it
wtintermed <- degree(small.eq, mode = "all", loops = F)
wtvec1 <- wtintermed / sum(wtintermed)

temp1 <- as_adjacency_matrix(small.eq, sparse = F)
diag(temp1) <- 1 # nodes invaded stay invaded
msf.outex <- smartsurv(adjmat = temp1,
                      stoch = F,
                      nrealz = 1)

temp <-
  smartsurv.weight(ss.out = msf.outex,
                  adjmat = temp1,
                  wtvec = wtvec1)

# Note that startwt gives output that includes number of nodes invasion free

```

```
numinvfree <- temp$tsampfree$sampfree # trait for coloring nodes
numinvfree
```

```
## [1] 15.860 20.935 14.835 10.145 11.440 10.550 14.055 15.250 21.025 23.105
## [11] 28.680 19.480 24.050 24.415 22.000 20.955 19.470 22.200 18.340 14.400
## [21] 21.940 13.785 19.875 14.765 15.490 14.295 16.335 19.125 22.955 28.150
## [31] 28.005 22.570 21.995 23.835 22.235 23.200 22.960 20.870 17.995 17.365
## [41] 16.545 20.425 18.545 20.540 21.230 21.930 24.635 19.995 17.235 22.740
```

```
nnodes <- dim(temp1)[1] # the total number of nodes
```

```
pal1 <- viridis_pal(option = "C",
                    direction = -1,
                    end = .8)
```

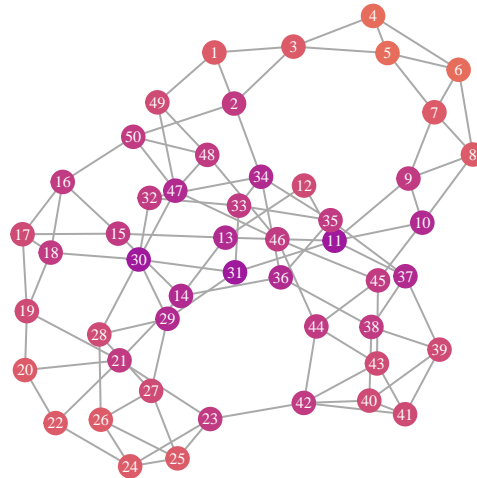
```
# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
```

```
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]
```

```
plot(
  small.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.small,
  sub = 'Darker colors indicate better sampling nodes'
)
```

```
title(main = 'Small world: higher degree nodes more likely starting nodes (deterministic links)',
      cex.main = 0.9)
```

## Small world: higher degree nodes more likely starting nodes (deterministic links)



Darker colors indicate better sampling nodes

This has little effect for the small world model because the degree is so similar across all nodes.

**Small world network: likelihood a node is the starting node is inversely proportional to node degree**

```
# find the degree for each node and construct a weight inversely proportional to it
wtintermed <- degree(small.eq, mode = "all", loops = F)
hilo <- sum(range(wtintermed))
wtintermed <- hilo - wtintermed

wtvec1 <- wtintermed / sum(wtintermed)

temp1 <- as_adjacency_matrix(small.eq, sparse = F)
diag(temp1) <- 1 # nodes invaded stay invaded
msf.outex <- smartsurv(adjmat = temp1,
                      stoch = F,
                      nrealz = 1)

temp <-
  smartsurv.weight(ss.out = msf.outex,
                  adjmat = temp1,
                  wtvec = wtvec1)

# Note that startwt gives output that includes number of nodes invasion free
numinvfree <- temp$sampfree$sampfree # trait for coloring nodes
numinvfree
```

```
## [1] 16.460 21.545 15.725 11.135 12.320 11.290 14.705 15.550 21.175 22.975
## [11] 28.360 19.560 24.070 24.785 22.200 21.445 19.810 21.960 18.300 14.520
## [21] 21.940 14.055 19.805 14.675 15.230 14.105 15.825 18.835 22.565 27.810
## [31] 27.555 22.350 21.885 23.965 22.525 23.840 22.600 20.690 17.525 16.955
## [41] 16.135 20.135 18.215 20.220 21.050 21.510 24.405 19.885 17.445 22.940
```

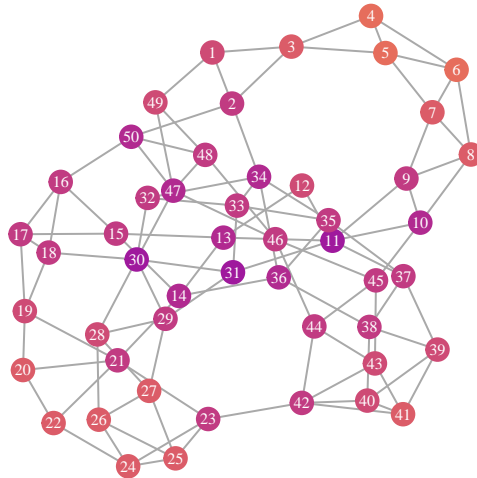
```
nnodes <- dim(temp1)[1] # the total number of nodes

pal1 <- viridis_pal(option = "C",
                    direction = -1,
                    end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

plot(
  small.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.small,
  sub = 'Darker colors indicate better sampling nodes'
)
title(main = 'Small world: higher degree nodes less likely starting nodes (deterministic links)',
      cex.main = 0.9)
```

**Small world: higher degree nodes less likely starting nodes (deterministic links)**



Darker colors indicate better sampling nodes

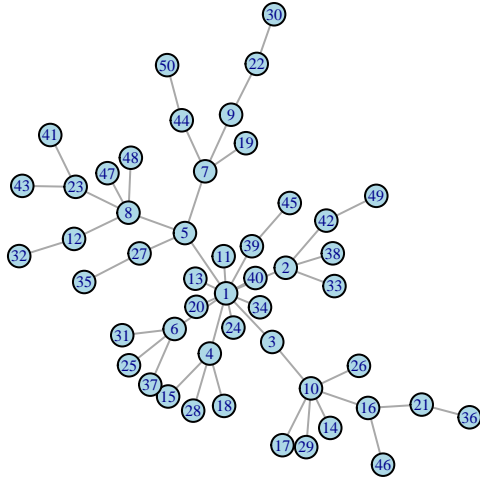
**Scale-free network**

**Scale-free network: nodes equally likely to be the starting node**

```
# generate a scale-free network with 50 nodes
sf.eq <- sample_pa(
  n = 50,
  power = 1,
  m = 1,
  directed = F
)

layout.sf <- layout_with_kk(sf.eq)

plot(
  sf.eq,
  edge.arrow.size = 0.3,
  vertex.size = 10,
  vertex.label.cex = 0.5,
  layout = layout.sf,
  vertex.color = 'lightblue'
)
```



```

# find the value of nodes for sampling based on a deterministic model

temp1 <- as_adjacency_matrix(sf.eq, sparse = F)
diag(temp1) <- 1 # nodes invaded stay invaded
temp2 <- multistart(adjmat = temp1, stoch = F)

numinvfree <- colMeans(temp2) # trait for coloring nodes
numinvfree

## [1] 42.62 30.96 32.30 30.62 36.90 30.62 28.38 28.78 16.82 22.88 29.72 17.04
## [13] 29.72 13.28 18.34 13.78 13.28 18.34 16.62 29.72 9.34 9.36 17.20 29.72
## [25] 18.34 13.28 27.50 18.34 13.28 5.28 18.34 9.42 18.84 29.72 14.50 6.44
## [37] 18.34 18.84 29.98 29.72 9.56 18.94 9.56 16.72 17.50 9.26 16.92 16.92
## [49] 9.24 9.14

nnodes <- dim(temp1)[1] # the total number of nodes

pal1 <- viridis_pal(option = "C",
                    direction = -1,
                    end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

plot(
  sf.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.sf,

```

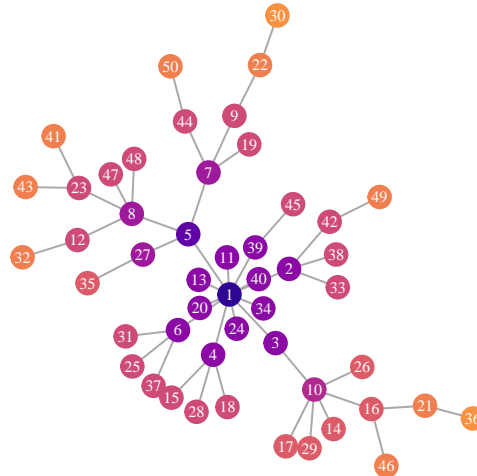


```

sub = 'Darker colors indicate better sampling nodes',
main = 'Scale-free: all starting nodes equally likely (deterministic links)'
)

```

## Scale-free: all starting nodes equally likely (deterministic links)



Darker colors indicate better sampling nodes

Scale-free network: likelihood a node is the starting node is proportional to node degree

```

# find the degree for each node and construct a proportional weight from it
wtintermed <- degree(sf.eq, mode = "all", loops = F)
wtvec1 <- wtintermed / sum(wtintermed)

temp1 <- as_adjacency_matrix(sf.eq, sparse = F)
diag(temp1) <- 1 # nodes invaded stay invaded
msf.outex <- smartsurv(adjmat = temp1,
                      stoch = F,
                      nrealz = 1)

temp <- smartsurv.weight(ss.out = msf.outex,
                        adjmat = temp1,
                        wtvec = wtvec1)

# Note that startwt gives output that includes number of nodes invasion free
numinvfree <- temp$sampfree$sampfree # trait for coloring nodes
numinvfree

```

```
## [1] 42.989796 31.091837 32.459184 30.744898 37.153061 30.744898 28.459184
```

```
## [8] 28.867347 16.663265 22.846939 29.826531 16.887755 29.826531 13.051020
## [15] 18.214286 13.561224 13.051020 18.214286 16.459184 29.826531 9.030612
## [22] 9.051020 17.051020 29.826531 18.214286 13.051020 27.561224 18.214286
## [29] 13.051020 4.887755 18.214286 9.112245 18.724490 29.826531 14.295918
## [36] 6.071429 18.214286 18.724490 30.091837 29.826531 9.255102 18.826531
## [43] 9.255102 16.561224 17.357143 8.948980 16.765306 16.765306 8.928571
## [50] 8.826531
```

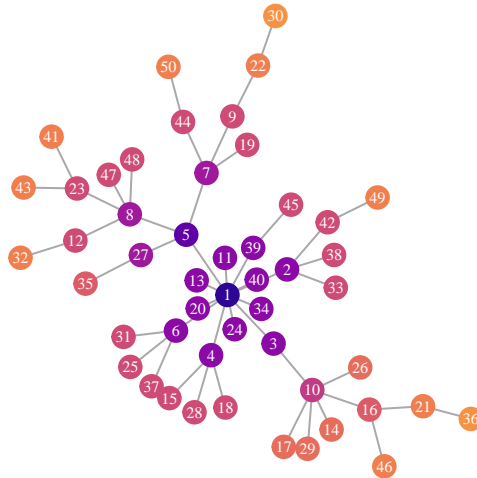
```
nnodes <- dim(temp1)[1] # the total number of nodes

pal1 <- viridis_pal(option = "C",
                    direction = -1,
                    end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

plot(
  sf.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.sf,
  sub = 'Darker colors indicate better sampling nodes'
)
title(main = 'Scale-free: higher degree nodes more likely starting nodes (deterministic links)',
      cex.main = 0.9)
```

## Scale-free: higher degree nodes more likely starting nodes (deterministic links)



Darker colors indicate better sampling nodes

Scale-free network: likelihood a node is the starting node is inversely proportional to node degree

```
# find the degree for each node and construct a weight inversely proportional to  
# it  
wtintermed <- degree(sf.eq, mode = "all", loops = F)  
hilo <- sum(range(wtintermed))  
wtintermed <- hilo - wtintermed  
  
wtvec1 <- wtintermed / sum(wtintermed)  
  
temp1 <- as_adjacency_matrix(sf.eq, sparse = F)  
diag(temp1) <- 1 # nodes invaded stay invaded  
msf.outex <- smartsurv(adjmat = temp1,  
  stoch = F,  
  nrealz = 1)  
  
temp <-  
  smartsurv.weight(ss.out = msf.outex,  
  adjmat = temp1,  
  wtvec = wtvec1)  
  
# Note that startwt gives output that includes number of nodes invasion free  
numinvfree <- temp$tsampfree$sampfree # trait for coloring nodes  
numinvfree
```

```
## [1] 42.554348 30.936594 32.271739 30.597826 36.855072 30.597826 28.365942
## [8] 28.764493 16.847826 22.885870 29.701087 17.067029 29.701087 13.320652
## [15] 18.362319 13.818841 13.320652 18.362319 16.648551 29.701087 9.394928
## [22] 9.414855 17.226449 29.701087 18.362319 13.320652 27.489130 18.362319
## [29] 13.320652 5.349638 18.362319 9.474638 18.860507 29.701087 14.536232
## [36] 6.505435 18.362319 18.860507 29.960145 29.701087 9.614130 18.960145
## [43] 9.614130 16.748188 17.525362 9.315217 16.947464 16.947464 9.295290
## [50] 9.195652
```

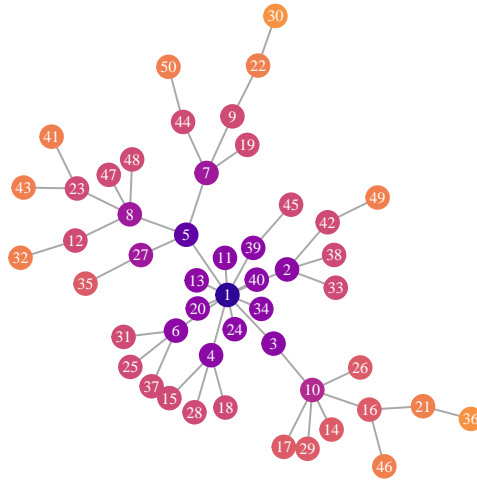
```
nnodes <- dim(temp1)[1] # the total number of nodes

pal1 <- viridis_pal(option = "C",
                    direction = -1,
                    end = .8)

# create palette that incorporates the possible extremes in terms of number of
# nodes free from invasion (0 and nnodes-1) to make comparison across figures
# easier
nodecol = pal1(15)[as.numeric(cut(c(0, numinvfree, (nnodes - 1)), breaks = 15))]
nodecol <- nodecol[2:(nnodes + 1)]

plot(
  sf.eq,
  vertex.frame.color = nodecol,
  vertex.label.cex = 0.5,
  vertex.label.color = 'white',
  vertex.size = 10,
  edge.arrow.size = 0.3,
  edge.curved = F,
  vertex.color = nodecol,
  layout = layout.sf,
  sub = 'Darker colors indicate better sampling nodes'
)
title(main = 'Scale-free: higher degree nodes less likely starting nodes (deterministic links)',
      cex.main = 0.9)
```

## Scale-free: higher degree nodes less likely starting nodes (deterministic links)



### Darker colors indicate better sampling nodes

Using weights based on the number of information sources, or the quality of information sources, is one way to integrate the socioeconomic network and the biophysical network, with an egocentric network focus. There are many other possibilities for linking the two networks, potentially drawing on more information about the network of information communication. Examples of this analysis applied to potential spread of disease through a seed system are available in the following two studies.

### References

More information and references are available at [www.garrettlab.com/ina](http://www.garrettlab.com/ina)

Andersen et al. 2019 and Buddenhagen et al. 2017 have examples of the application of the algorithms in INA function `smartsurv`.

Andersen, K. F., C. E. Buddenhagen, P. Rachkara, R. Gibson, S. Kalule, D. Phillips, and K. A. Garrett. 2019. Modeling epidemics in seed systems and landscapes to guide management strategies: The case of sweetpotato in Northern Uganda. *Phytopathology* 109:1519-1532.

Buddenhagen, C. E., J. F. Hernandez Nopsa, K. F. Andersen, J. Andrade-Piedra, G. A. Forbes, P. Kromann, S. Thomas-Sharma, P. Useche, and K. A. Garrett. 2017. Epidemic network analysis for mitigation of invasive pathogens in seed systems: Potato in Ecuador. *Phytopathology* 107:1209-1218.

Garrett, K. A. 2021. Impact network analysis and the INA R package: Decision support for regional management interventions. *Methods in Ecology and Evolution*, in press.

Csardi, G. and T. Nepusz. 2006. The igraph software package for complex network research. *InterJournal, Complex Systems*: 1695. <http://igraph.org>

Wickham, H., J. Hester, and W. Chang. `devtools`: Tools to Make Developing R Packages Easier. 2020. R package version 2.3.0. <https://CRAN.R-project.org/package=devtools>