Erdos Renyi Simulations of Perturbation Theory

Carlyle Morgan

6/1/2021

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.0.5
```

```
library(dplyr)
library(stringr)
library(readr)
library(stats)
library(expm)
```

```
## Warning: package 'expm' was built under R version 4.0.5
```

Introduction

A simulation of a random perturbation of a network can be conducted using an Erdos-Renyi game. For chosen probability p, generate a random number between 0 and 1 for each two nodes in a network, such that if this number is less than p:

- If these two nodes are connected, disconnect them
- If these two nodes are disconnected, connect them

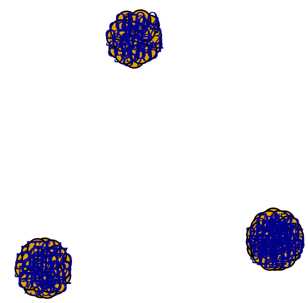
Consider some community structure that is perfectly identified. Not only does each community form a connected component of some graph, but say each member node of a community is connected to every other member of a community.

For the purposes of this simulation, consider a network with the following "ground-truth" community structure, reflecting the true community membership of each node:

- Community A has 50 nodes (Let's call these nodes 1-50)
- Community B has 30 nodes (Let's call these nodes 51-80)
- Community C has 20 nodes (Let's call these nodes 81-100)

We can construct and visualize this network in igraph using the following commands:

```
g.full1<-graph.full(50)
g.full2<-graph.full(30)
g.full3<-graph.full(20)
g.full<-g.full1+g.full2+g.full3
set.seed(123)
plot(g.full)
```



As expected, a simple visualization of this network shows that each node is connected to each other member of its community with no connections between communities.

Erdos-Renyi Simulations

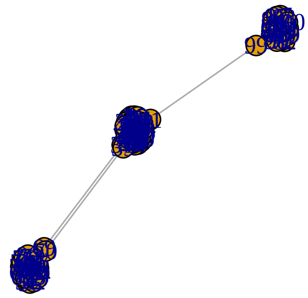
We can construct a generic function to simulate a perturbation via an Erdos-Renyi game for any undirected graph:

```
erdosrenyi_perturb<-function(graph, p, directed){
  edgelist<-as.data.frame(get.edgelist(graph))
  pairlist<-as.data.frame(t(combn(vcount(graph),2)))
  seed_list<<-c(0)
  for (i in 1:nrow(pairlist)){
    seed<-runif(1)
    seed_list<<-append(seed_list,seed)
    if (seed<p){
      if (tail(duplicated(rbind(edgelist,c(pairlist[i,][,1],pairlist[i,][,2]))),1)
          | tail(duplicated(rbind(edgelist,c(pairlist[i,][,2],pairlist[i,][,1]))),1) == TRUE){
        graph<-graph-edges(c(pairlist[i,][,1],pairlist[i,][,2]))
      } else {
        graph<-graph+edges(c(pairlist[i,][,1],pairlist[i,][,2]))
      }
    }
  }
  return(graph)
}
```

Applying this functions to our benchmark graph for different values of p and visualizing the results illustrates the breakdown of implied community structure as p increases:

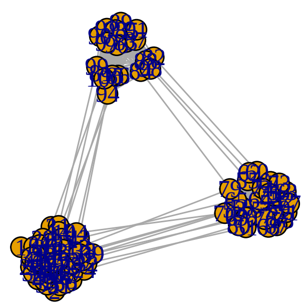
For $p = 0.001$:

```
g.p.001<-erdosrenyi_perturb(g.full, 0.001)
set.seed(123)
plot(g.p.001)
```



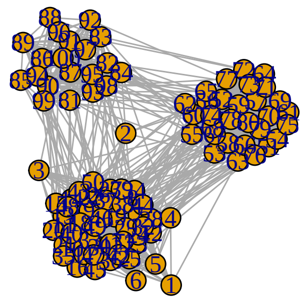
For $p = 0.01$:

```
g.p.01<-erdosrenyi_perturb(g.full, 0.01)
set.seed(123)
plot(g.p.01)
```



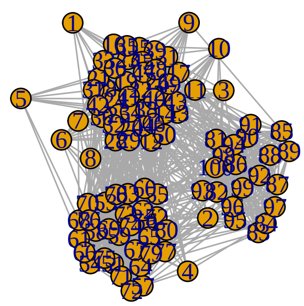
For $p = 0.05$:

```
g.p.05<-erdosrenyi_perturb(g.full, 0.05)
set.seed(123)
plot(g.p.05)
```



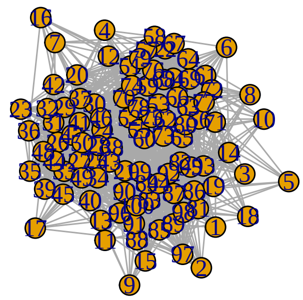
For $p = 0.1$:

```
g.p.1<-erdosrenyi_perturb(g.full, 0.1)
set.seed(123)
plot(g.p.1)
```



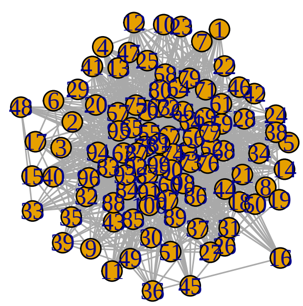
For $p = 0.2$:

```
g.p.2<-erdosrenyi_perturb(g.full, 0.2)
set.seed(123)
plot(g.p.2)
```



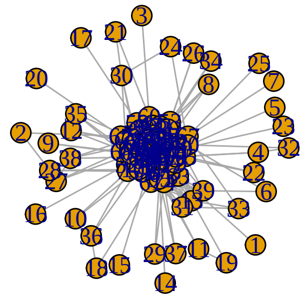
For $p = 0.3$:

```
g.p.3<-erdosrenyi_perturb(g.full, 0.3)
set.seed(123)
plot(g.p.3)
```

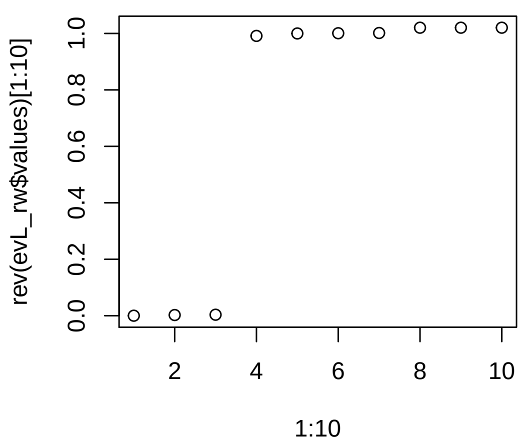


For $p = 1$:

```
g.1<-erdosrenyi_perturb(g.full, 1)
set.seed(123)
plot(g.1)
```

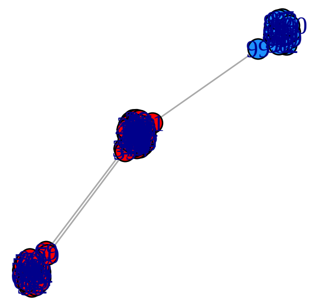


```
A<-get.adjacency(g.p.001)
A<-as.matrix(A)
W<-A
D<-diag(strength(g.p.001))
L_unnorm<-D-W
D_negone<-1/D
D_negone[D_negone==Inf]<-0
L_rw<-D_negone %*% L_unnorm
evL_rw <- eigen(L_rw, symmetric=FALSE)
plot(1:10,rev(evL_rw$values)[1:10])
```

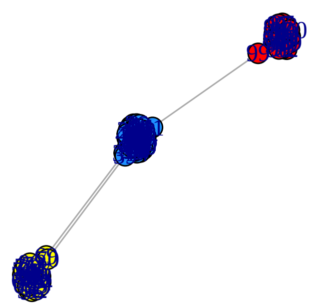



```
k   <- 2
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.001)$cluster<-km$cluster
V(g.p.001)[cluster == 1]$color<-"red"
V(g.p.001)[cluster == 2]$color<-"dodgerblue"
```

```r
set.seed(123)
plot(g.p.001)
```

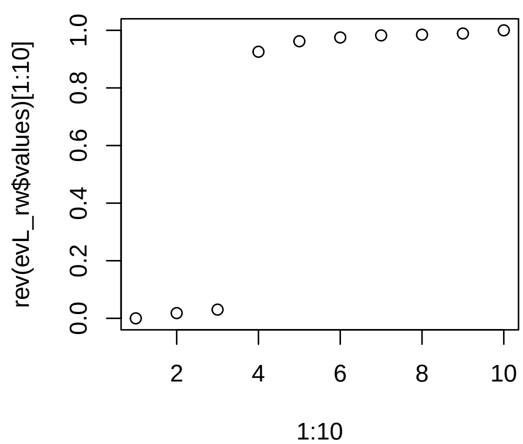


```r
k   <- 3
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.001)$cluster<-km$cluster
V(g.p.001)[cluster == 1]$color<-"red"
V(g.p.001)[cluster == 2]$color<-"dodgerblue"
V(g.p.001)[cluster == 3]$color<-"yellow"
set.seed(123)
plot(g.p.001)
```

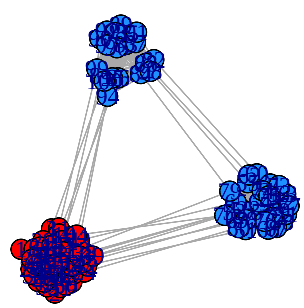


```r
A<-get.adjacency(g.p.01)
A<-as.matrix(A)
W<-A
D<-diag(strength(g.p.01))
```

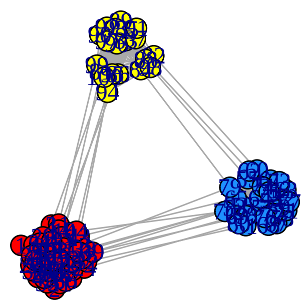
```
L_unnorm<-D-W
D_negone<-1/D
D_negone[D_negone==Inf]<-0
L_rw<-D_negone %*% L_unnorm
evL_rw <- eigen(L_rw, symmetric=FALSE)
plot(1:10,rev(evL_rw$values)[1:10])
```



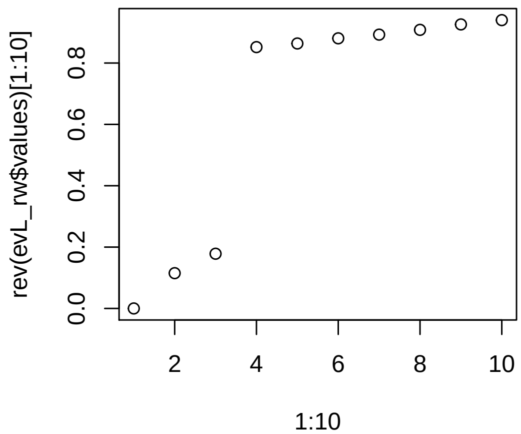

```
k   <- 2
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.01)$cluster<-km$cluster
V(g.p.01)[cluster == 1]$color<-"red"
V(g.p.01)[cluster == 2]$color<-"dodgerblue"
set.seed(123)
plot(g.p.01)
```



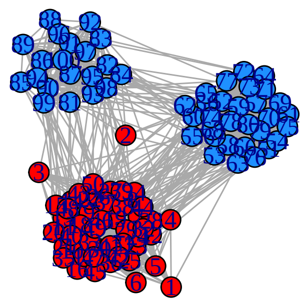
```r
k   <- 3
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.01)$cluster<-km$cluster
V(g.p.01)[cluster == 1]$color<-"red"
V(g.p.01)[cluster == 2]$color<-"dodgerblue"
V(g.p.01)[cluster == 3]$color<-"yellow"
set.seed(123)
plot(g.p.01)
```



```r
A<-get.adjacency(g.p.05)
A<-as.matrix(A)
W<-A
D<-diag(strength(g.p.05))
L_unnorm<-D-W
D_negone<-1/D
D_negone[D_negone==Inf]<-0
L_rw<-D_negone %*% L_unnorm
evL_rw <- eigen(L_rw, symmetric=FALSE)
plot(1:10,rev(evL_rw$values)[1:10])
```

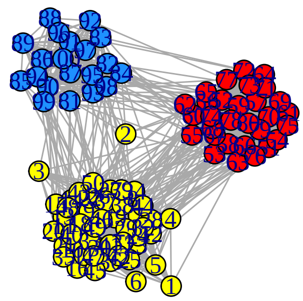



```r
k   <- 2
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.05)$cluster<-km$cluster
V(g.p.05)[cluster == 1]$color<-"red"
V(g.p.05)[cluster == 2]$color<-"dodgerblue"
set.seed(123)
plot(g.p.05)
```

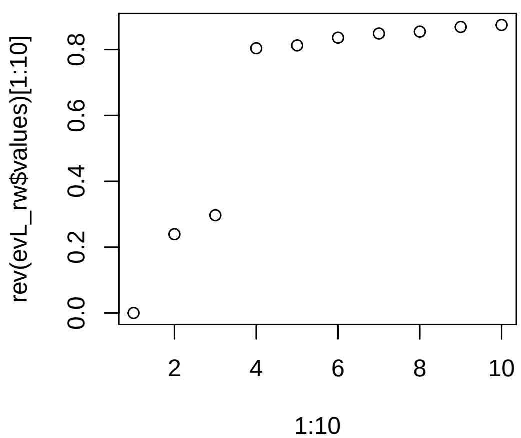


```r
k   <- 3
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.05)$cluster<-km$cluster
V(g.p.05)[cluster == 1]$color<-"red"
V(g.p.05)[cluster == 2]$color<-"dodgerblue"
V(g.p.05)[cluster == 3]$color<-"yellow"
set.seed(123)
```

```
plot(g.p.05)
```

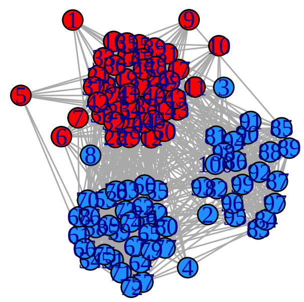


```
A<-get.adjacency(g.p.1)
A<-as.matrix(A)
W<-A
D<-diag(strength(g.p.1))
L_unnorm<-D-W
D_negone<-1/D
D_negone[D_negone==Inf]<-0
L_rw<-D_negone %*% L_unnorm
evL_rw <- eigen(L_rw, symmetric=FALSE)
plot(1:10,rev(evL_rw$values)[1:10])
```

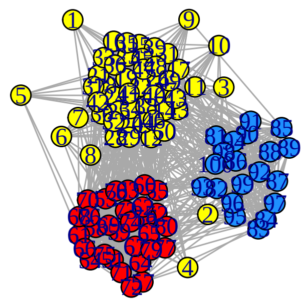



```
k   <- 2
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.1)$cluster<-km$cluster
```

```
V(g.p.1)[cluster == 1]$color<-"red"
V(g.p.1)[cluster == 2]$color<-"dodgerblue"
set.seed(123)
plot(g.p.1)
```

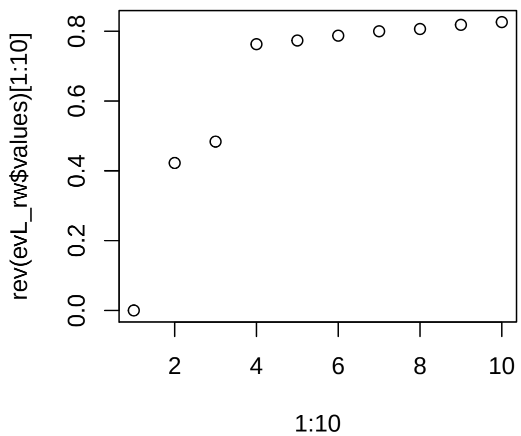


```
k   <- 3
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.1)$cluster<-km$cluster
V(g.p.1)[cluster == 1]$color<-"red"
V(g.p.1)[cluster == 2]$color<-"dodgerblue"
V(g.p.1)[cluster == 3]$color<-"yellow"
set.seed(123)
plot(g.p.1)
```

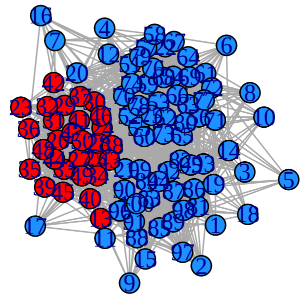


```
A<-get.adjacency(g.p.2)
A<-as.matrix(A)
```

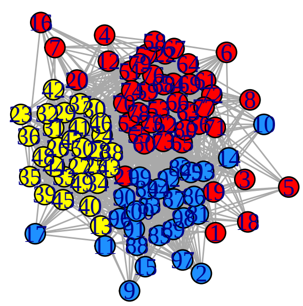
```r
W<-A
D<-diag(strength(g.p.2))
L_unnorm<-D-W
D_negone<-1/D
D_negone[D_negone==Inf]<-0
L_rw<-D_negone %*% L_unnorm
evL_rw <- eigen(L_rw, symmetric=FALSE)
plot(1:10,rev(evL_rw$values)[1:10])
```




```r
k   <- 2
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.2)$cluster<-km$cluster
V(g.p.2)[cluster == 1]$color<-"red"
V(g.p.2)[cluster == 2]$color<-"dodgerblue"
set.seed(123)
plot(g.p.2)
```

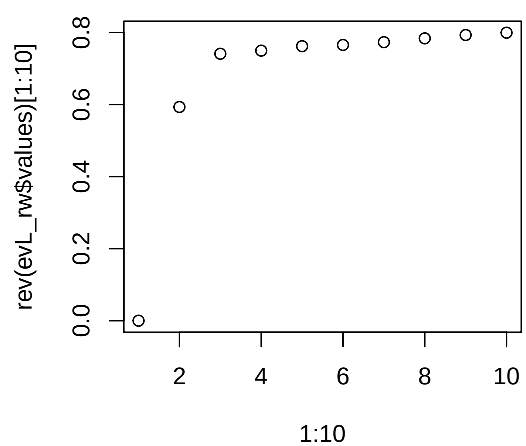


```r
k   <- 3
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.2)$cluster<-km$cluster
V(g.p.2)[cluster == 1]$color<-"red"
V(g.p.2)[cluster == 2]$color<-"dodgerblue"
V(g.p.2)[cluster == 3]$color<-"yellow"
set.seed(123)
plot(g.p.2)
```

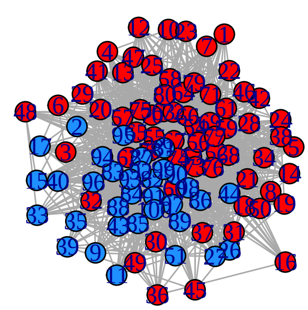


```r
A<-get.adjacency(g.p.3)
A<-as.matrix(A)
W<-A
D<-diag(strength(g.p.3))
L_unnorm<-D-W
D_negone<-1/D
D_negone[D_negone==Inf]<-0
```

```
L_rw<-D_negone %*% L_unnorm
evL_rw <- eigen(L_rw, symmetric=FALSE)
plot(1:10,rev(evL_rw$values)[1:10])
```

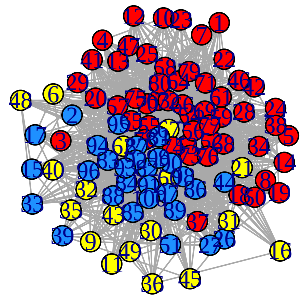



```
k   <- 2
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.3)$cluster<-km$cluster
V(g.p.3)[cluster == 1]$color<-"red"
V(g.p.3)[cluster == 2]$color<-"dodgerblue"
set.seed(123)
plot(g.p.3)
```

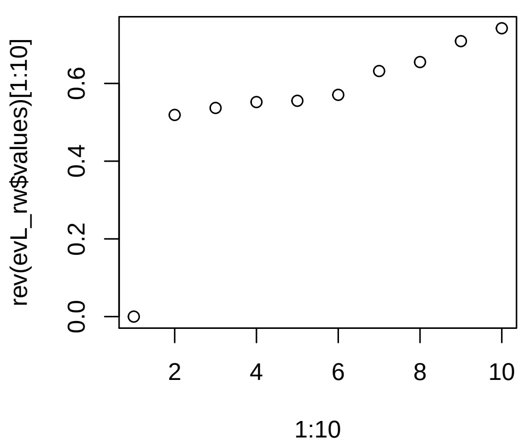


```
k   <- 3
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.p.3)$cluster<-km$cluster
```

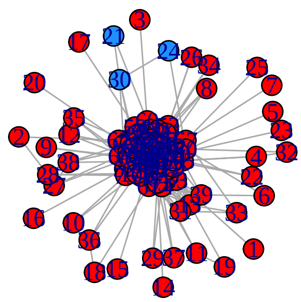
```r
V(g.p.3)[cluster == 1]$color<-"red"
V(g.p.3)[cluster == 2]$color<-"dodgerblue"
V(g.p.3)[cluster == 3]$color<-"yellow"
set.seed(123)
plot(g.p.3)
```



```r
A<-get.adjacency(g.1)
A<-as.matrix(A)
W<-A
D<-diag(strength(g.1))
L_unnorm<-D-W
D_negone<-1/D
D_negone[D_negone==Inf]<-0
L_rw<-D_negone %*% L_unnorm
evL_rw <- eigen(L_rw, symmetric=FALSE)
plot(1:10,rev(evL_rw$values)[1:10])
```



```
k   <- 2
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.1)$cluster<-km$cluster
V(g.1)[cluster == 1]$color<-"red"
V(g.1)[cluster == 2]$color<-"dodgerblue"
set.seed(123)
plot(g.1)
```



```
k   <- 3
Z   <- evL_rw$vectors[,(ncol(evL_rw$vectors)-k+1):ncol(evL_rw$vectors)]
km <- kmeans(Z, centers=k, nstart=5)
V(g.1)$cluster<-km$cluster
V(g.1)[cluster == 1]$color<-"red"
V(g.1)[cluster == 2]$color<-"dodgerblue"
V(g.1)[cluster == 3]$color<-"yellow"
set.seed(123)
plot(g.1)
```