Network measures of community structure

Assortativity (Newman 2003) and modularity (Sah 2014) are common and similar measures of network community structure but there's a lack of understanding about why and how they are similar.

Using equations from various texts, here I compare compare assortativity and modularity for two different networks: one with 2 modules (e.g., men/women) and one with 6 modules (e.g., school classrooms).

Definitions

Assortativity (r) is calculated as a ratio of the number of edges within-group to the number of edges that should occur within-group by random chance.

$$r = \frac{\sum_i e_{ii} - \sum_i (a(i)b(i))}{1 - \sum_i (1 - a(i)b(i))} = \frac{Tr\mathbf{E} - ||E^2||}{1 - ||E^2||}$$

where e_{ij} is the fraction of edges connecting vertices of type i and j, $a(i) = \sum_j e_(ij)$ and $b(j) = \sum_i e_(ij)$.

Network *modularity* (Q) is a similar measure of non-random mixing in networks:

$$Q = \sum_i e_{ii} - a_i^2$$

where e_{ij} is the proportion of edges in the network that link nodes in community i to community j and $a_i = \sum_j e_{ij}$ represents the proportion of edges in the network that link to nodes in subgroup i. The maximum value of Q is $1 - \frac{1}{K}$ where K is the number of modules (Sah 2014).

Two modules

```
### Modular graph (Q~0.4) to compare assortativity and modularity
library(igraph)
```

```
##
## Attaching package: 'igraph'

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

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

```
get_e_type <- function(elRow, types){
  ifelse(diff(types[elRow])==0, "w", "b")
}

setwd("~/Documents/phd/research-projects/miller-tb-assortativity/analysis/simulations-sah/simulations-5
g <- read_graph("G_Q0.4_N2000_rep1.graphml", format = "graphml")

plot(g, vertex.label="", vertex.size=1.5, vertex.color=V(g)$module)
```

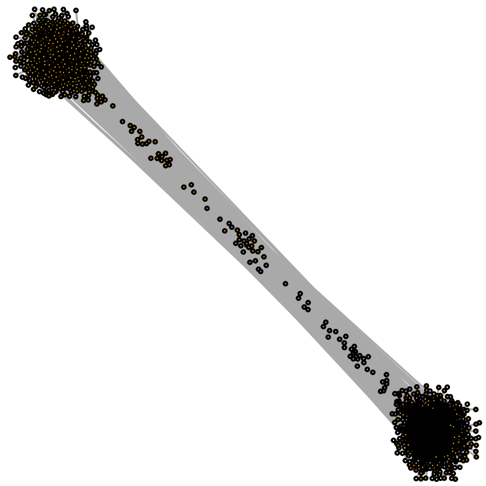

Figure 1: Graph with two modules (Q=0.4) generated by Sah et al. 2014 algorithm.

```
## CALCULATE ASSORTATIVITY
## EXTRACT EDGE TYPES
V(g)$id <- 1:2000
el <- as_edgelist(g) # edgelist
el1 <- V(g)$module[el[, 1]]
el2 <- V(g)$module[el[, 2]]

elj <- cbind(el1, el2)
foo <- apply(elj, 1, function(x){paste0(x[1], x[2])})
table(foo)
```

```
## foo
##   00   01   11
## 4555  882 4559
```

```
## CONSTRUCT EDGE MATRIX
Eij <- matrix(c(table(foo)[1], table(foo)[2]/2, table(foo)[2]/2, table(foo)[3]), nrow=2, byrow = TRUE)
eij <- Eij/sum(Eij)

sum(eij)
```

```
## [1] 1
```

```
ai = rowSums(eij)
ai
```

```
## [1] 0.4997999 0.5002001
```

```
bi = colSums(eij)
bi
```

```
## [1] 0.4997999 0.5002001
```

```
r <- (sum(diag(eij)) - sum(ai * bi))/(1 - sum(ai * bi))
r
```

```
## [1] 0.8235294
```

```r
## CHECK IF CALCULATION EQUALS IGRAPH FUNCITON
assortativity_nominal(g, types=(V(g)$module+1), directed=FALSE)
```

```
## [1] 0.8235294
```

```r
## EQUATION 1 from SAH 2014
V(g)$id <- 1:2000
el <- as_edgelist(g) # edgelist
el1 <- V(g)$module[el[, 1]]
el2 <- V(g)$module[el[, 2]]

elj <- cbind(el1, el2)
foo <- apply(elj, 1, function(x){paste0(x[1], x[2])})
table(foo)
```

```
## foo
##   00   01   11
## 4555  882 4559
```

```r
m <- length(E(g))

# k=0  # group 0 nodes
ekk0 <- table(foo)["00"]/m
ak20 <- sum(degree(g, v=V(g)$id[V(g)$module==0]))/(2*m)
k0 <- ekk0 - ak20^2

# k=1  # group 1 nodes
ekk1 <- table(foo)["11"]/m
ak21 <- sum(degree(g, v=V(g)$id[V(g)$module==1]))/(2*m)
k1 <- ekk1 - ak21^2

# Q is the sum of these values
Q <- k0 + k1
Q
```

```
##        00
## 0.4117646
```

```r
# igraph's built in function very closely matches this value
# perhaps a rounding error in my code?
modularity(g, (V(g)$module + 1))
```

```
## [1] 0.4117646
```

```r
# assortativity = Q / (1 - prop.edges.expected.within.random)
Q/(1-0.5)
```

```
##        00
## 0.8235293
```

Here, it seems there is an exact relationship between assortativity and modularity multiplied by 2. We expected modularity and assortativity to be related such that

modularity = assortativity / (1 - expected.prop.between.edges)

where the expected proportion of between edges is calculated by taking the proportion of edges "touching" each subgroup, squaring it, and summing them all up.

Since groups have equal size, you would expect 1/2 to occur between groups by chance. Thus, assortativity

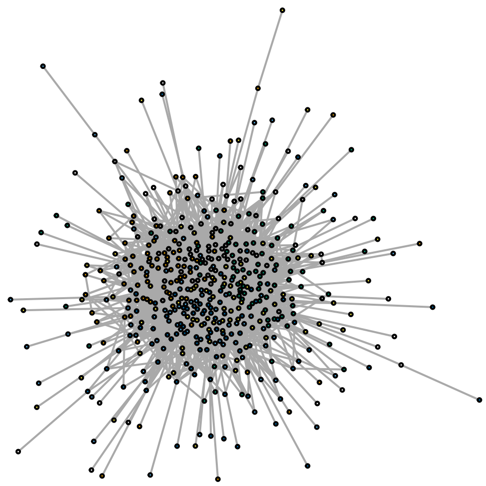

Figure 2: Graph with six modules (Q=0.4) generated by Sah et al. 2014 algorithm.

gets divided (by 1-0.5) by while modularity does not.

More than two modules

```
rm(list=ls())

### Modular graph (Q~0.4) to compare assortativity and modularity
get_e_type <- function(elRow, types){
  ifelse(diff(types[elRow])==0, "w", "b")
}

setwd("~/Documents/phd/research-projects/miller-tb-assortativity/analysis/simulations-sah")
g <- read_graph("scalefree_Q0.4_N500_d10_m6.graphml", format = "graphml")

plot(g, vertex.label="", vertex.size=1.5, vertex.color=V(g)$module)
```

```
## CALCULATE ASSORTATIVITY
## EXTRACT EDGE TYPES
V(g)$id <- 1:vcount(g)
el <- as_edgelist(g) # edgelist
el1 <- V(g)$module[el[, 1]]
el2 <- V(g)$module[el[, 2]]

elj <- cbind(el1, el2)
foo <- apply(elj, 1, function(x){paste0(x[1], x[2])})
table(foo)
```

```
## foo
##  00  01  02  03  04  05  11  12  13  14  15  22  23  24  25  33  34  35
## 240  77  68  64  76  65 237  67  64  66  82 248  73  60  66 241  72  75
##  44  45  55
## 245  66 248
```

```
## CONSTRUCT EDGE MATRIX
es <- table(foo)
```

```r
esg <- as.numeric(c(es["00"]/1, es["01"]/2, es["02"]/2, es["03"]/2, es["04"]/2, es["05"]/2,
                    es["01"]/2, es["11"]/1, es["12"]/2, es["13"]/2, es["14"]/2, es["15"]/2,
                    es["02"]/2, es["12"]/2, es["22"]/1, es["23"]/2, es["24"]/2, es["25"]/2,
                    es["03"]/2, es["13"]/2, es["23"]/2, es["33"]/1, es["34"]/2, es["35"]/2,
                    es["04"]/2, es["14"]/2, es["24"]/2, es["34"]/2, es["44"]/1, es["45"]/2,
                    es["05"]/2, es["15"]/2, es["25"]/2, es["35"]/2, es["45"]/2, es["55"]/1))

Eij <- matrix(esg, nrow=6, byrow = TRUE)

eij <- Eij/sum(Eij)

sum(eij)
```

```
## [1] 1
```

```r
ai = rowSums(eij)
ai
```

```
## [1] 0.166 0.166 0.166 0.166 0.166 0.170
```

```r
bi = colSums(eij)
bi
```

```
## [1] 0.166 0.166 0.166 0.166 0.166 0.170
```

```r
r <- (sum(diag(eij)) - sum(ai * bi))/(1 - sum(ai * bi))
r
```

```
## [1] 0.500312
```

```r
## CHECK IF CALCULATION EQUALS IGRAPH FUNCITON
assortativity_nominal(g, types=(V(g)$module+1), directed=FALSE)
```

```
## [1] 0.500312
```

```r
## EQUATION 1 from SAH 2014
table(foo)
```

```
## foo
##  00  01  02  03  04  05  11  12  13  14  15  22  23  24  25  33  34  35
## 240  77  68  64  76  65 237  67  64  66  82 248  73  60  66 241  72  75
##  44  45  55
## 245  66 248
```

```r
m <- length(E(g))

# k=0  # group 0 nodes
ekk0 <- table(foo)["00"]/m
ak20 <- sum(degree(g, v=V(g)$id[V(g)$module==0]))/(2*m)
k0 <- ekk0 - ak20^2

# k=1  # group 1 nodes
ekk1 <- table(foo)["11"]/m
ak21 <- sum(degree(g, v=V(g)$id[V(g)$module==1]))/(2*m)
k1 <- ekk1 - ak21^2

# k=2  # group 1 nodes
ekk2 <- table(foo)["22"]/m
ak22 <- sum(degree(g, v=V(g)$id[V(g)$module==2]))/(2*m)
```

```r
k2 <- ekk2 - ak22^2

# k=3  # group 1 nodes
ekk3 <- table(foo)["33"]/m
ak23 <- sum(degree(g, v=V(g)$id[V(g)$module==3]))/(2*m)
k3 <- ekk3 - ak23^2

# k=4  # group 1 nodes
ekk4 <- table(foo)["44"]/m
ak24 <- sum(degree(g, v=V(g)$id[V(g)$module==4]))/(2*m)
k4 <- ekk4 - ak24^2

# k=5  # group 1 nodes
ekk5 <- table(foo)["55"]/m
ak25 <- sum(degree(g, v=V(g)$id[V(g)$module==5]))/(2*m)
k5 <- ekk5 - ak25^2

# Q is the sum of these values
Q <- k0 + k1 + k2 + k3 + k4 + k5
as.numeric(Q)
```

```
## [1] 0.41692
```

```r
# igraph's built in function very closely matches this value
# perhaps a rounding error in my code?
modularity(g, (V(g)$module + 1))
```

```
## [1] 0.41692
```

```r
# assortativity = Q / (1 - prop.edges.expected.within.random)
Q/(1-(1/6))
```

```
##       00
## 0.500304
```

Here, we see that there is an (almost) exact relationship between assortativity and modularity divided by (5/6).

Conclusions

- assortativity = Q / (1 - prop.edges.expected.within.random)
- Note that $Q = \sum_i e_{ii} - a_i^2 = Tr\mathbf{E} - ||E^2||$