Modularity-Based Consensus Community Detection

```r
filename <- "management_1619"
```

Modularity-Based Consensus Community Detection

on network management_1619

```r
library(igraph)
source('source_functions_cons_com_det.r')
```

```
## [1] "loading Consensus Community Detection functions ..."
```

```r
path <- "./networks/"
echo = FALSE # FALSE to avoid printing intermediate results

# parameters for consensus community detection
n_trials = 10
alphas = c(0.0, 0.05, 0.1)
res = c(0.9, 1.0, 1.1)
epsilon = 1/100

# pruning
min_p <- 0.50
min_vds <- 5
min_w <- 0.01

# consensus
reps = 10
```

```r
load_network <- function(path, filename) {
  fn <- paste0(path,filename, ".graphml")
  print(paste("Loading", fn, "..."))
  g <- read_graph(fn, format = "graphml")
  E(g)$ww <- E(g)$weight
  V(g)$str <- strength(g)
  V(g)$name <- paste0("V" , V(g)$name)
  degrees <- degree(g)
  kcore <- coreness(g)
  print(paste("Number of nodes:", length(V(g)$name)))
  print(paste("Number of edges:", length(E(g)$weight)))
  print(paste("Mean degree:", mean(degrees)))
  print(paste("Max k-coreness: ", max(kcore)))
  return(g)
}
```

analysis of management 11 15

```
g <- load_network(path, filename)
```

```
## [1] "Loading ./networks/management_1619.graphml ..."
## [1] "Number of nodes: 2367"
## [1] "Number of edges: 3238"
## [1] "Mean degree: 2.73595268272074"
## [1] "Max k-coreness:  6"
```

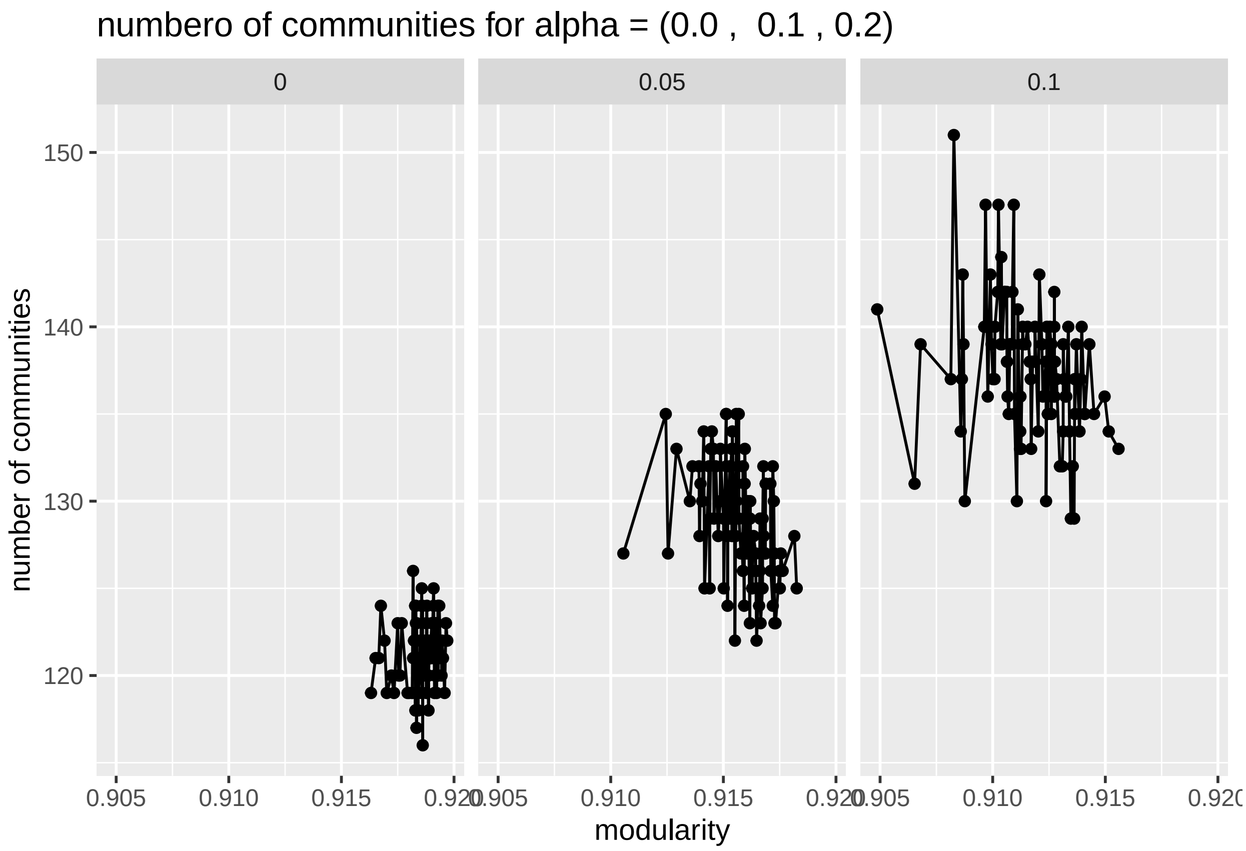
```
dfresults <- consensus_community_detection(g,
                                           alphas, reps, n_trials,
                                           epsilon, res, min_p, min_vds, min_w,
                                           echo = FALSE)
```

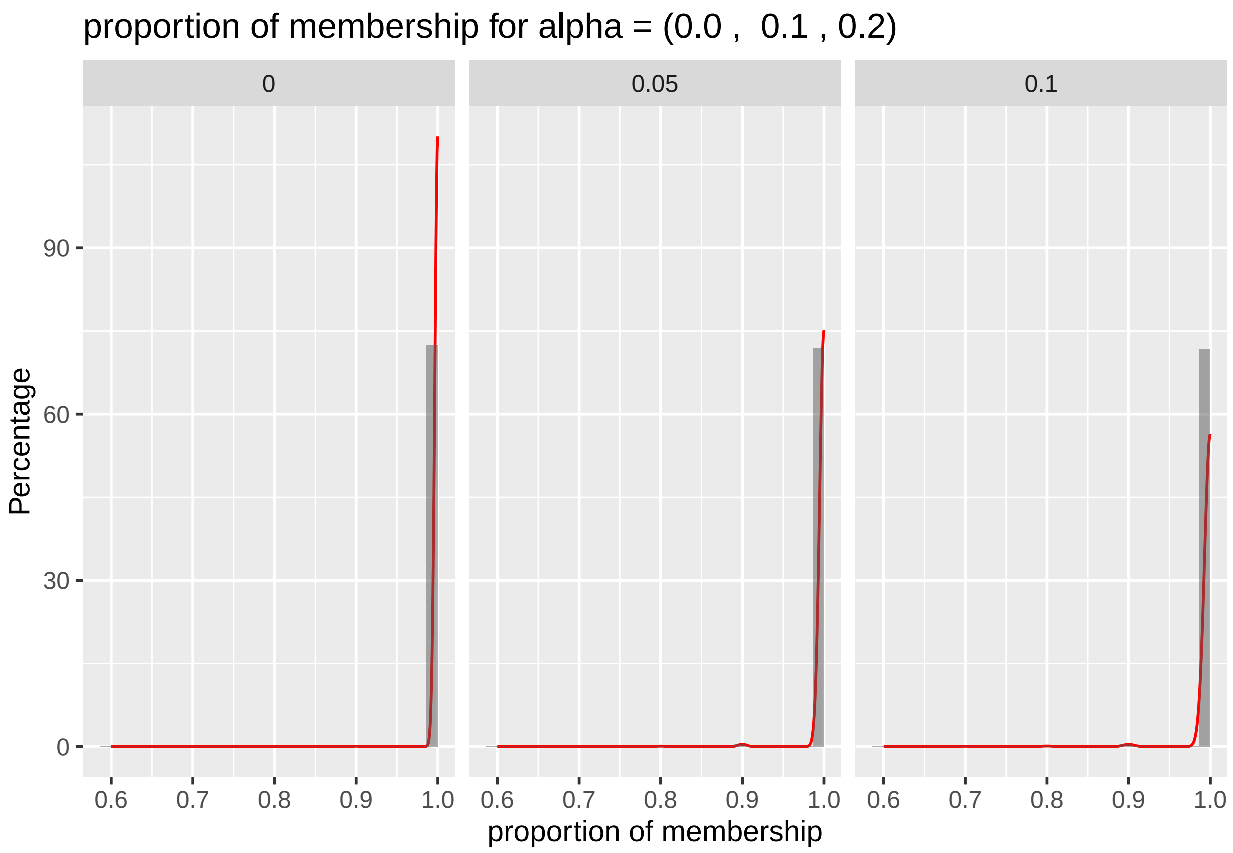
```
## [1] "Alpha 0"
## [1] "Repetition 1"
## [1] "Repetition 2"
## [1] "Repetition 3"
## [1] "Repetition 4"
## [1] "Repetition 5"
## [1] "Repetition 6"
## [1] "Repetition 7"
## [1] "Repetition 8"
## [1] "Repetition 9"
## [1] "Repetition 10"
## [1] "Alpha 0.05"
## [1] "Repetition 1"
## [1] "Repetition 2"
## [1] "Repetition 3"
## [1] "Repetition 4"
## [1] "Repetition 5"
## [1] "Repetition 6"
## [1] "Repetition 7"
## [1] "Repetition 8"
## [1] "Repetition 9"
## [1] "Repetition 10"
## [1] "Alpha 0.1"
## [1] "Repetition 1"
## [1] "Repetition 2"
## [1] "Repetition 3"
## [1] "Repetition 4"
## [1] "Repetition 5"
## [1] "Repetition 6"
## [1] "Repetition 7"
## [1] "Repetition 8"
## [1] "Repetition 9"
## [1] "Repetition 10"
```

```
json_data <- toJSON(dfresults)
write(json_data, file=paste0(filename,"_results.json"))
print("Completed.")
```

```
## [1] "Completed."
```

```
dfresults  %>% filter(method == "LV") %>%
  ggplot( aes(x = modularit, y =  nc , group = a)) +
  facet_grid(cols = vars(a)) +
  geom_line() +
  geom_point() +
   labs(title = "numbero of communities for alpha = (0.0 ,  0.1 , 0.2)",
   x = "modularity",
   y = "number of communities")
```




proportion of membership for alpha = (0.0 , 0.1 , 0.2)
0
0.05
0.1
Percentage
90
60
30
0
0.6
0.7
0.8
0.9
1.0
proportion of membership