CAI Lab, Session 13: Communities in networks

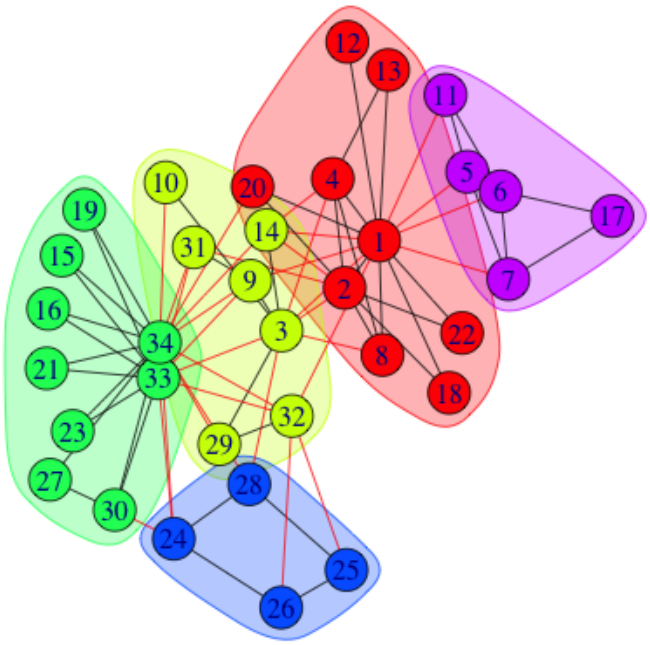
In this session we will explore the community discovery algorithms available in `igraph`.

Here are some, including some we have seen in theory class:

- `edge.betweenness.community` [Newman and Girvan, 2004]
- `fastgreedy.community` [Clauset et al., 2004] (modularity optimization method)
- `label.propagation.community` [Raghavan et al., 2007]
- `leading.eigenvector.community` [Newman, 2006]
- `multilevel.community` [Blondel et al., 2008] (the Louvain method)
- `optimal.community` [Brandes et al., 2008]
- `spinglass.community` [Reichardt and Bornholdt, 2006]
- `walktrap.community` [Pons and Latapy, 2005]
- `infomap.community` [Rosvall and Bergstrom, 2008]

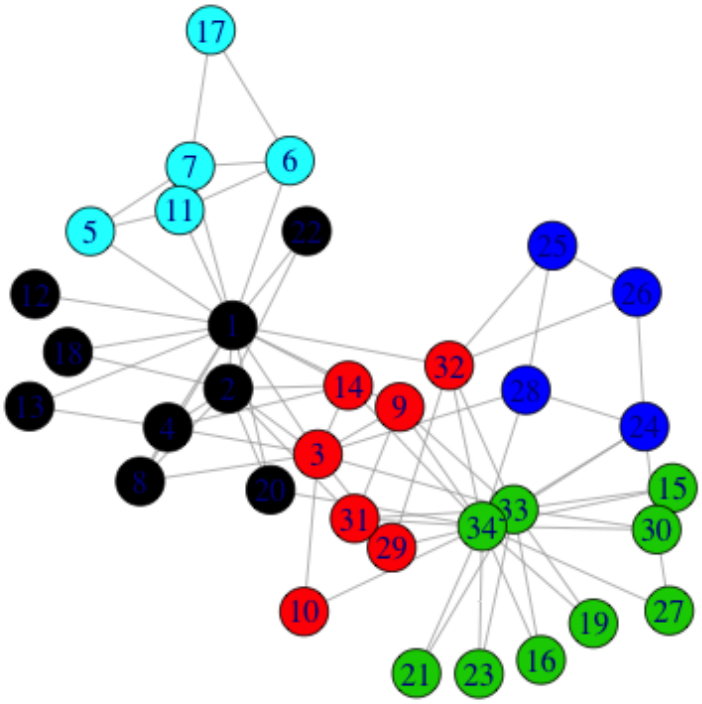
All of these methods return a `commmunities` object, which you can then use to explore, plot, and compute metrics on. As an example, consider the following snippet of code:

```
> karate <- graph.famous("Zachary")
> wc <- walktrap.community(karate)
> modularity(wc)
[1] 0.3532216
> membership(wc)
 [1] 1 1 2 1 5 5 5 1 2 2 5 1 1 2 3 3 5 1 3 1 3 1 3 4 4 4 3 4 2 3 2 2 3 3
> plot(wc, karate)
```



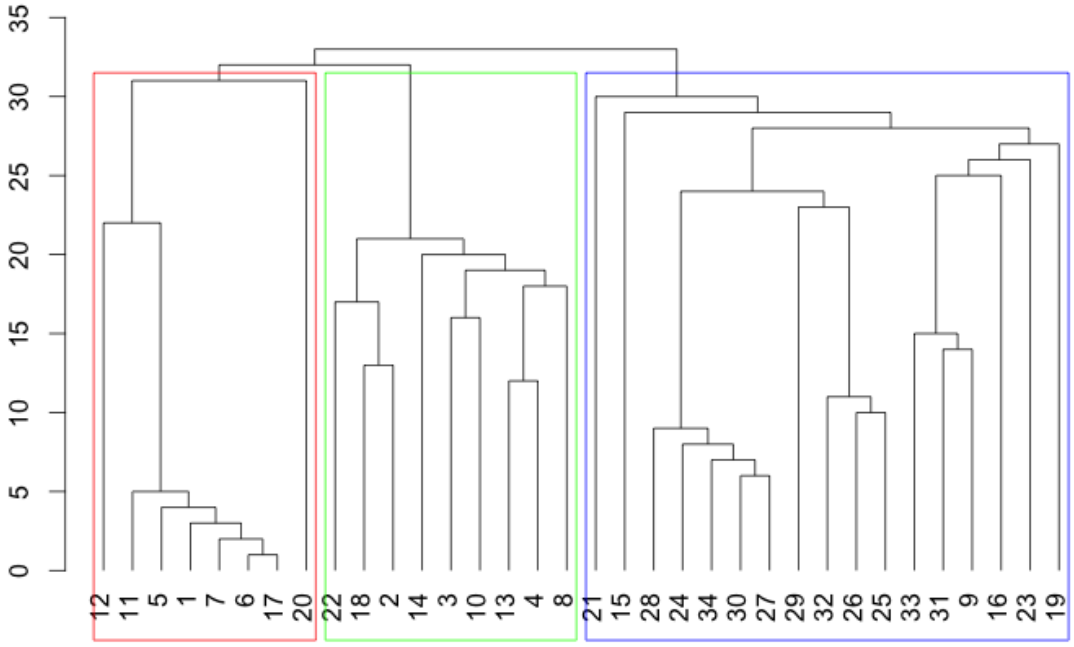
An alternative way of plotting communities without the shaded regions is:

```
> plot(karate, vertex.color=membership(wc))
```



For those algorithms that output communities with hierarchical structure, this information can be visualized using the `dendPlot` function, which displays the corresponding dendogram:

```
> karate <- graph.famous("Zachary")
> fc <- fastgreedy.community(karate)
> dendPlot(fc)
```



`igraph`'s help and documentation[1] should be useful to carry out the following tasks.

[1] http://igraph.sourceforge.net/documentation.html

Load the network from `edges.txt` provided with this session's files. Make sure you read it as an undirected graph.

1. Describe the network a little. How many edges and nodes does it have? What is its diameter? And transitivity? And degree distribution? Does it look like a random network? Visualize the network with node sizes proportional to their pagerank.
2. Now, try a couple of community detection algorithm of your choice from the list provided. How many nodes does the largest community found contain? Plot the histogram of community sizes. Plot the graph with its communities.

If you want to try larger graphs, see `http://networkrepository.com/networks.php` For example, go to `http://networkrepository.com/bio-celegansneural.php`, download the `.zip` file, unzip it and try

```
# download from http://networkrepository.com/ca-AstroPh.php, and unzip
file <- "bio-celegansneural.mtx"
dat <- read.table(file, skip=2, sep=" ")
g <- graph_from_data_frame(dat, directed = FALSE)
length(E(g))
# some graphs may need simplifying, particularly multiple edges
g2 <- simplify(g, remove.multiple = TRUE, remove.loops = TRUE)
length(E(g2))
c <- fastgreedy.community(g2)
modularity(c)
plot(c,g2)
```

This is a smallish file for the repository, but useful to show the differences in running time among algorithms.

Deliverables

Rules: Same rules as in previous labs about working solo/in pairs and plagiarism.

To deliver: You must deliver a pdf file with at most 1 page (title, authors, etc. included) indicating whether you succeeded in running these steps, and if you found anything interesting. No problem if it is only half a page.

Procedure: Submit your work through the Racó.

Deadline: Work must be delivered within **2 weeks** from the lab. Late deliveries risk being penalized or not accepted at all. If you anticipate problems with the deadline, tell me as soon as possible.

References

[Blondel et al., 2008] Blondel, V. D., Guillaume, J.-l., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of community hierarchies in large networks. *Networks*, pages 1–6.

[Brandes et al., 2008] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2008). On Modularity Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20.

[Clauset et al., 2004] Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*.

[Newman, 2006] Newman, M. E. J. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 74:036104.

[Newman and Girvan, 2004] Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69(2 Pt 2):026113.

[Pons and Latapy, 2005] Pons, P. and Latapy, M. (2005). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, 10:191–218.

[Raghavan et al., 2007] Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 76:036106.

[Reichardt and Bornholdt, 2006] Reichardt, J. and Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74.

[Rosvall and Bergstrom, 2008] Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105:1118–1123.