Community detection methods

Blockmodeling is very slow and can be applied to small networks only. When searching for groups in large networks we use community detection methods.

Communities - dense clusters for which there are more lines inside than among clusters (values of lines are taken into account too).

In Pajek two community detection methods are available: *Louvain method* and *VOS Clustering*.

When applying Louvain method we search for partition into clusters with the highest value of *modularity (Q)*. Modularity is defined in the following way:

$$Q = \frac{1}{2m} \sum_{s} (e_s - r * \frac{K_s^2}{2m})$$

- m – total number of lines in network,

- s – cluster (community),

- $e_s = \sum_{ij \in s} A_{ij}$ – 2 times the number of lines in community s

- $K_s = \sum_{i \in s} k_i$ – sum of degrees in community s

- r – *resolution parameter*, default value 1 means modularity as originally defined

Similar method is *VOS Clustering*, where *VOS quality function* is taken into account instead of modularity.

Both methods are available in:
Network/Create Partition/Communities

Several parameters can be changed, but they are important only when we are analysing larger networks.

By changing *resolution parameter - r* we can get larger or smaller communities. By default resolution parameter is set to 1. Setting r larger than 1 means searching for larger number of smaller communities. Setting r smaller than 1 means searching for smaller number of larger communities.

Examples: shr1.net, football.net, import.net.

Communities obtained by both methods are usually very similar. We can check this by
Partitions/Info
which computes *Cramer*, *Adjusted Rand* and some other indices for comparison of two partitions.

Some more examples:
http://mrvar.fdv.uni-lj.si/pajek/community/LouvainVOS.htm

Examples

1. Find communities in networks <u>shr1.net</u>, <u>football.net</u>, <u>import.net</u> using Louvain method and VOS Clustering. Compare results - compute Cramer contingency coefficient.