Supporting Information
Community detection by graph Voronoi diagrams

Dávid Deritei, Zsolt I. Lázár, István Papp, Ferenc Járai-Szabó, Róbert Sumi, Levente Varga, Erzsébet Ravasz Regan, Mária Ercsey-Ravasz

1 Benchmarks

The benchmark software framework provides a great variety of different programs, making us able to generate directed, undirected and weighted graphs with disjoint, hierarchical or overlapping communities. In the first phase of our research, we mostly tested our methods on undirected networks, therefore we used a version of the benchmark software, which can generate large scale undirected networks. The software is very flexible and there is a wide range of variable network parameters including: 

- \(N\) - the number of nodes,
- \(k\) - the average degree,
- \(k_{\text{max}}\) - the maximum degree,
- \(\gamma\) - the degree distribution,
- \(\beta\) - the community size distribution,
- \(\mu\) - the mixing parameter (which sets the rate of edges inside and between clusters),
- \(s_{\text{max}}\) - the maximum size of communities,
- \(s_{\text{min}}\) - the minimum size of communities.

The construction of a realization of the benchmark proceeds through the following steps, according to 1.

1. Each node is given a degree taken from a power law distribution with exponent \(\gamma\). The extremes of the distribution \(k_{\text{min}}\) and \(k_{\text{max}}\) are chosen such that the average degree is \(k\).

2. Each node shares a fraction \(1 - \mu\) of its links with the other nodes of its community and a fraction \(\mu\) with the other nodes of the network; \(\mu\) is the mixing parameter.

3. The sizes of the communities are taken from a power law distribution with exponent \(\beta\), such that the sum of all sizes equals the number \(N\) of nodes of the graph. The minimal and maximal community sizes \(s_{\text{min}}\) and \(s_{\text{max}}\) are chosen so to respect the constraints imposed by our definition of community: \(s_{\text{min}} > k_{\text{min}}\) and \(s_{\text{max}} > k_{\text{max}}\). This ensures that a node of any degree can be included in at least one community.

4. At the beginning, all nodes are homeless, i.e., they are not assigned to any community. In the first iteration, a node is assigned to a randomly chosen community; if the community size exceeds the internal degree of the node i.e., the number of its neighbors inside the community, the node enters the community, otherwise it remains homeless. The procedure stops when there are no more homeless nodes.

5. To enforce the condition on the fraction of internal neighbors expressed by the mixing parameter \(\mu\), several re-wiring steps are performed, such that the degrees of all nodes stay the same and only the split between internal and external degree is affected, when needed. In this way the ratio between external and internal degree of each node in its community can be set to the desired share \(\mu\) with good approximation.
The output files of the benchmark software are an edge list with 'source target' format, a file containing a node list with the community data (node cluster-id format), and a file with statistical information about the network.

The authors of \(^1\) also provided another software, the so-called *mutual information* \(^2\), which returns the rate of efficiency of a method, if given the community structure computed by the tested method, and the community data generated by the benchmark.

## 2 Identification of generator points

As we described in the main text we choose generator points as the nodes with the highest local density in a region within radius \(r\). This way we can avoid selection of multiple generator points within one cluster. In case two nodes have equal density, we choose one randomly. One could ask how frequently does this happen, does it introduce a significant stochastic effect to the method? We measured the relative local density values of all nodes on many graphs and we found that increasing the size of graphs this event becomes less and less frequent. This also holds when increasing the density. Even in those rare cases when it occurs, it affects nodes with small local density (Fig.1). Candidates for generator nodes are always the nodes with high local density, so the situation of choosing randomly between two local maxima is extremely rare.

![Figure 1: Distribution of the relative local density of nodes on a benchmark network with \(N = 1000\) nodes, average degree \(<k> = 15\), mixing \(\mu = 0.2\). The number of different density values is 995, meaning there are only 5 values which are not unique. The histogram bin size is extremely small \(\approx 3 \times 10^{-5}\). We can see that the probability of having equal or (almost equal) density values is very small, especially at high densities where potential generator nodes belong.](image)
3 Community detection methods used for reference

The Louvain method is a community detection algorithm based on local modularity optimization. The method consists of two main phases. Initially all nodes are put in different communities. In the first phase nodes are considered in random order and the modularity gain is calculated if a given node is removed from its community and is added to one of its neighbor’s community. The gain is calculated for all neighbors, and if there is at least one positive gain, the node is placed in the community, where the gain is maximum. If no more modularity gain can be achieved, the algorithm enters its second phase: communities found in the first phase are replaced by a single node, and one can return to phase one. The two phases can be repeated several times.

GANXiS (SLPA) is a heuristic method capable to detect both overlapping and non-overlapping (crisp) communities. It is an extension of the Label Propagation Algorithm. During an iteration step nodes are selected to be listeners in random order. The neighbors of the listener send a label randomly chosen from their label list with a probability proportional to the labels’ frequency. The listener selects the most popular label and adds it to its label list, or if it is already present increases its frequency by one. Initially all label lists contain a single label: the label of the node they belong to. After performing a predefined number of iteration steps, label lists are post-processed in order to define communities. The probability distribution of labels present in label lists defines the strength of association to communities to which the node belongs.

The Infomap method uses probability flow of random walks as information flow and the network is decomposed into modules by describing the path of this flow.

The Link-Community clustering method instead of assuming that a community is a set of nodes with many links between them, this approach considers a community to be a set of closely interrelated links. Moreover, the Link-Community clustering method results a hierarchical clustering. Taking into account the similarity between links a dendrogram is built. Each leaf of the dendrogram represents a link from the network and branches represent link communities. The links stand at unique positions of the dendrogram. Accordingly, the nodes will occupy multiple positions. The link dendrogram represents a hierarchy structure and to obtain community structure it is necessary to realize a horizontal cut on this dendrogram.

4 Real-world networks

We tested our algorithm on several real-world networks with widely different structure and properties, frequently used in the literature: 1) The updated version of the collaboration network of scientists posting preprints on the condensed matter archive at www.arxiv.org (CONDMAT). This is based on preprints posted to the archive between January 1, 1995 and March 31, 2005 and it contains 39576 nodes and 175692 links. 2) Protein interaction network for yeast (YEAST)
including 1845 nodes and 4405 edges. 3) Network of links between american political blogs (POLBLOGS) 12 with 1223 nodes and 19087 edges. 4) Neural network of the nematode C. Ele-gans (CEL) 13, 14 contains 297 nodes and 2359 links. 5) The famous network of friendships between the 34 members of the Zachary karate club linked with 78 edges 15.

In Figure 2 for each of the five networks we show the modularity function when increasing radius \( r \) and cluster size distributions at three different \( r \) values (highlighted with colored dots). It is natural that in all networks as \( r \) increases we have fewer but larger communities. The colormaps show mutual information comparison of partitioning with different radius \( r \). A dot with coordinates \((x, y)\) in the colormap indicates the mutual information of the two clusterings obtained with radius \( x \) and \( y \). The mutual information increases from blue to red and the matrix is symmetric. All mutual information matrixes have the same resolution (50 points) per range, just the ranges are different depending on the size of the network. On the diagonal the mutual information is 1 because it is a self comparison, however the highest mutual information is reached above and beyond the diagonal, too. This happens when the clustering does not change in a certain interval of \( r \).

We notice several distinct behaviors. In CONDMAT and YEAST the modularity seems to converge to a maximum, but the plateau is reached relatively early on. The colormaps indicate that the clustering itself changes smoothly: the mutual info is relatively large all along the plateau and small variations in \( r \) do not cause drastic changes in the community structure. In these networks the algorithm also detects many small communities, not only a few large ones.

In POLBLOGS, C. Elegans and the Karate club the clustering changes more abruptly. Large red squares in the mutual information matrix indicate intervals where the clustering does not change at all. These correspond to plateaus in the modularity curve. When several such large squares exist (several local maxima or plateaus in the modularity curve) it indicates hierarchical structure.

Figure 2: For the five real-world networks we show the modularity as function of radius and the community size distributions at some different highlighted values of $r$. The size of bins on the histograms are the same at each level of highlighted radius, it is varied only for visual clarity. The colormaps show mutual information comparison of partitioning with different radius $r$. A dot with coordinates $(x, y)$ in the colormap indicates the mutual information of the two clusterings obtained with radius $x$ and $y$. Highest mutual info is shown as red and lowest is blue. White indicates that the entire network is considered as one giant community and there is no clustering to be compared.


