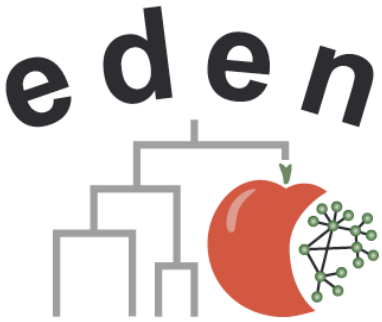


Project no. 043251

EDEN

ECOLOGICAL DIVERSITY AND EVOLUTIONARY NETWORKS

Instrument: Specific Targeted Project (STREP)

Thematic Priority: Integrating and strengthening the European Research Area
NEST Pathfinder initiative Tackling Complexity in Science

D2.3: Report on Community Detection

Due date of deliverable: 31.1.2009
Actual submission date: 31.1.2009

Start date of project: 1 January 2007 Duration: 36 months

Organisation name of lead contractor for this deliverable: TKK/LCE
Other contributors: CCMAR, Bioinf Leipzig, IMEDEA-UIB

Revision 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	

Introduction

This report comprises publications related to identification of communities, produced within the EDEN project. Here, the term “communities” loosely means subgraphs or clusters of the network, which have denser and stronger connections than the network has on average.The main goal has been to assess and develop methods which are applicable to full or very dense weighted networks or matrices of genetic distances, such as those resulting from individual-sample-level analysis of microsatellite data. The EDEN-specific goal has been the identification of genetic groups from individual-level data. Contents and main conclusions of the publications are outlined below.

Detecting modules in dense weighted network, T. Heimo et al, *J. Stat. Mech.* P08007 (2008)
This paper deals with application and generalization of the q-state Potts method (Reichardt & Bornholdt, Phys Rev Lett 93, 218701, 2004) to weighted and dense networks. We introduce a proper null model for optimizing the Hamiltonian of the method and evaluate the behaviour of its resolution limit. The Potts method is based on minimizing the energy of a system of spins, which interact via the network’s links. The ground state, corresponding to the minimum energy, is such that spins within communities are aligned. In addition, a tuning parameter allows changing the resolution from small to large communities. However, as we have shown earlier, the methods suffers from a resolution limit, such that communities below a network-size-dependent limit cannot be resolved. Here, our analysis indicates that the limit behaves fairly well once the networks considered are very dense, and weights are taken into account.

Sequential algorithm for fast clique percolation, J. Kumpula et al, *Phys. Rev. E* 78, 026109 (2008)
In this paper we present an algorithm and a method for applying the clique percolation community detection method (Palla et al, Nature 435, 815 (2005)) on weighted networks. In our method, the weighted networks are progressively thresholded, retaining only the strongest links (corresponding to shortest genetic distances), yielding a hierarchical community structure which can be understood with the help of dendrograms, resembling phylogenetic trees. As a side product, the algorithmic implementation is extraordinarily fast, allowing the use of clique percolation on networks comprising millions of nodes.

Limited resolution and multiresolution methods in complex network community detection, J. Kumpula et al, *Fluctuation and Noise Letters* 7, L209 (2007)
This paper deals with the known issue of resolution limits with global-optimization-based methods, such as the above-mentioned Potts method or the multiresolution method introduced by Arenas et al (Arenas et al, New Journal of Physics 10, 053039 (2008)). The limits of both methods are studied analytically, and the methods are applied to network data sets, with the result that both methods behave fairly similarly.

A network perspective on the genetic population structure of seagrass Posidonia Oceanica, M. Kivelä, Master’s Thesis, Helsinki University of Technology, 2008.
This thesis work deals with analysis of the P. Oceanica microsatellite data. First, proposed measures for defining the genetic distance between two individuals are reviewed critically, and the results are compared, with the outcome that different measures are required for detecting structure resulting from short- or long-time-scale processes. Using the non-shared-alleles distance measure, a matrix of genetic distances between P. Oceanica samples is then constructed and subjected to two community detection methods. Due to biases in sampling, it is seen that the above-mentioned fast clique percolation method cannot resolve the community structure very well, except for the geographic split west-central-east. On the other hand, the block diagonalization method by Sales-Pardo et al (PNAS 104, 15224, 2007) yields a dendrogram whose highest levels are seen to correspond fairly well to the expected geographic division, and where a more detailed community structure is visible. However, the validity of this detailed structure is somewhat questionable, as the method produces structure even if applied on a randomized reference.

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Detecting modules in dense weighted networks with the Potts method

Tapio Heimo[1,2], Jussi M Kumpula[1], Kimmo Kaski[1] and Jari Saramäki[1]

[1] Department of Biomedical Engineering and Computational Science, Helsinki University of Technology, PO Box 9203, FIN-02015 HUT, Finland
[2] Nordea Bank AB, Markets Division, H224, SE-10571 Stockholm, Sweden
E-mail: taheimo@cc.hut.fi, jkumpula@lce.hut.fi, kimmo.kaski@hut.fi and jsaramak@lce.hut.fi

Received 18 June 2008
Accepted 29 July 2008
Published 19 August 2008

Online at stacks.iop.org/JSTAT/2008/P08007
doi:10.1088/1742-5468/2008/08/P08007

Abstract. We address the problem of multiresolution module detection in dense weighted networks, where the modular structure is encoded in the weights rather than topology. We discuss a weighted version of the q-state Potts method, which was originally introduced by Reichardt and Bornholdt. This weighted method can be directly applied to dense networks. We discuss the dependence of the resolution of the method on its tuning parameter and network properties, using sparse and dense weighted networks with built-in modules as example cases. Finally, we apply the method to stock price correlation data, and show that the resulting modules correspond well to known structural properties of this correlation network.

Keywords: socio-economic networks, new applications of statistical mechanics

ArXiv ePrint: 0804.3457

1742-5468/08/P08007+13$30.00

Contents

1. Introduction

During recent years, the network approach has proven to be a very efficient way of investigating a wide range of complex systems [1]–[4]. In this approach, the fundamental elements of the system are represented with nodes and the interactions between them with links. Sometimes it is enough to consider links as 'binary', such that each link either exists or not. In this case, it is assumed that the pure topology carries enough relevant information about the system under study. However, valuable information is often lost if interaction strengths are not taken into account. Because of this, the study of *weighted* networks has recently been receiving a lot of attention. In this framework, a scalar weight representing the associated interaction strength is assigned to each link. It is evident that this additional degree of freedom somewhat complicates the picture; for example generalization of existing measures is not necessarily straightforward (see, e.g., [5]). Thus there is a need for developing new network analysis methods which focus on the weights instead of pure topology.

The study of (weighted) networks has mostly focused on systems whose interaction structure is inherently sparse, such as air transport networks [6, 7] and social networks inferred from electronic communication records [8, 9]. Another approach is to filter out interactions which are considered insignificantly weak, resulting in sparse network representations even for systems where each element interacts with each other, i.e., systems whose 'natural' representation is a full or dense weighted network. For such networks, it is the interaction strengths themselves that carry the most significant information—the networks are constructed on the basis of the assumption that the strongest interactions encode the most significant properties for the system under study. This is the case for instance with correlation-based networks, in which the weights are usually related to correlations between the time series of some relevant activities of the nodes (see, e.g., [10]), or distance-based networks [11], in which the weights are related to distances between the nodes according to some relevant metric. It is evident that in this approach, setting the proper threshold below which interactions are discarded is a non-trivial task.

J. Stat. Mech. (2008) P08007

In addition to weighted networks, the attention of network science has recently been focusing on 'mesoscopic' properties of networks, i.e., structures beyond the scale of single nodes or their immediate neighborhoods. A very important and related problem is the detection and study of *modules* or *communities*[3], i.e., groups of nodes with dense internal connections and sparse connections to the rest of the network [12]–[17]. A number of methods have been introduced, mostly in the context of binary networks. These include various modularity optimization methods building on the work by Newman and Girvan [12], the clique percolation method by Palla *et al* [13], and methods based on statistical inference [18, 19]. Many methods have been generalized to deal with weighted networks [20]–[23]; however, e.g. for the clique percolation method, networks have to be fairly sparse in order for the method to be applicable. Regarding the modularity optimization family of methods, it has been shown that there is an intrinsic resolution limit [20, 24, 25]. However, a lot of attention has recently been given to *multiresolution* methods [15, 21, 23, 25, 26], which allow investigating modular structure at various levels of coarse-graining.

In this work we concentrate on investigating modular structure in dense weighted networks, using a weighted version of the q-state Potts method by Reichardt and Bornholdt (RB) [15, 26]. This method is closely related to modularity optimization methods, and hence there is a resolution limit [20]. However, the method contains a tuning parameter which allows changing this limit. Although the method was originally introduced in the context of sparse, binary networks, edge weights can readily be taken into account [26]. In fact, once this is done, the networks to be analyzed no longer need to be sparse—hence, for example when studying stock market correlations, all correlation matrix elements can be taken into account and no thresholding is necessary.

We begin by discussing the weighted RB method, deriving the required weighted null model, and then investigate the effect of the tuning parameter on the resolution of the method for networks with modular structure encoded in the weights. Then, we apply the method to a correlation-based network of stock return time series, i.e., a full correlation matrix, whose modular structure has been earlier investigated using a wide variety of approaches (see, e.g., [10], [27]–[29]). It should be noted here that the multiresolution method recently introduced by Arenas *et al* [21] bears some similarity with the Potts method (see [25]); thus for comparison we apply it to the same data. Finally, we draw conclusions.

2. The RB method

2.1. Introduction

Let us begin with a short introduction of the community detection method introduced by Reichardt and Bornholdt (RB) [15, 26]. In this method, each node is assigned to exactly one module, and the module indices of nodes are considered as spins of a q-state Potts model. The goal is to assign nodes to modules in such a way that the energy of the system is minimized. In the global optimum, groups of nodes with dense internal connections should end up having parallel spins. The Hamiltonian for the system is defined as

$$\mathcal{H}_u = -\sum_m (l_{mm} - \gamma[l_{mm}]_{p_{ij}}), \tag{1}$$

[3] In this paper, these two terms will be used interchangeably.

where l_{mm} is the number of links inside module m, $[l_{mm}]_{p_{ij}}$ is the expected number of links inside module m given the null model p_{ij}, and $\gamma > 0$ is an adjustable parameter. The summation is over all modules. The null model p_{ij} denotes the probability that a link would exist between nodes i and j if the network was entirely random, i.e., in the absence of modular structure. Essentially, there are two possible choices for the null model: constant $p_{ij} = p$, which corresponds to Erdös–Rényi networks [30], and the configuration model [3], in which the degree sequence of the original network is retained but all links are randomly rewired, such that all correlations are lost to the extent allowed by the degree sequence.

Next we briefly review the derivation of $[l_{mm}]_{p_{ij}}$ for the configuration model. Imagine that all the links in the network are cut in half, such that nodes have stubs (i.e., half-links) connected to them. Then these stubs are to be randomly reconnected to form full links. If two such stubs are randomly picked, the probability that both connect to nodes in module m is simply K_m^2/K^2, where K is the degree sum of the network[4] and K_m the degree sum of nodes in module m. Since there are $K/2$ pairs of stubs, we get

$$[l_{mm}] = \frac{K_m^2}{2K}. \tag{2}$$

Correspondingly, the probability that the two stubs to be connected belong to different modules, say m and n, is $2K_mK_n/K^2$. Thus, the expected number of links between modules m and n reads

$$[l_{mn}] = \frac{K_mK_n}{K}. \tag{3}$$

Let us now address the question of weighted networks. It seems natural that equation (1) transforms to

$$\mathcal{H}_w = -\sum_m (w_{mm} - \gamma[w_{mm}]_{p_{ij}}), \tag{4}$$

where w_{mm} and $[w_{mm}]_{p_{ij}}$ denote the sum of weights and expected sum of weights of links inside module m, respectively. Again, there are essentially two ways to define $[w_{mm}]_{p_{ij}}$. The approach taken in [20] is to calculate the expected number of links using the configuration model and to assume that each link has average weight, that is, $[w_{mm}] = \langle w\rangle[l_{mm}]$. However, here we take another approach, which is analogous to the above derivation for the unweighted case and based on the ideas presented in [31]. In weighted networks, the *strength* s_i of node i is defined as the sum of the weights of the links attached to it. Consider dividing the strength of each node into small 'stubs' of weight ds such that node i has $s_i/\mathrm{d}s$ stubs emerging from it and start randomly connecting pairs of these stubs. This process is analogous to the above unweighted case, and as a result the expected sums of weights of the links inside module m and between modules m and n are

$$[w_{mm}] = \frac{S_m^2}{2S}, \qquad \text{and} \qquad [w_{mn}] = \frac{S_mS_n}{S}, \tag{5}$$

respectively, where $S = \sum_{i=1}^N s_i$ is the strength sum of the network and S_q the strength sum of module q. When all links have weight $w_{ij} = 1$, the above equations reduce to equations (2) and (3).

[4] The degree sum of the network is defined by $K = \sum_{i=1}^N k_i$, where k_i is the degree of node i.

J. Stat. Mech. (2008) P08007

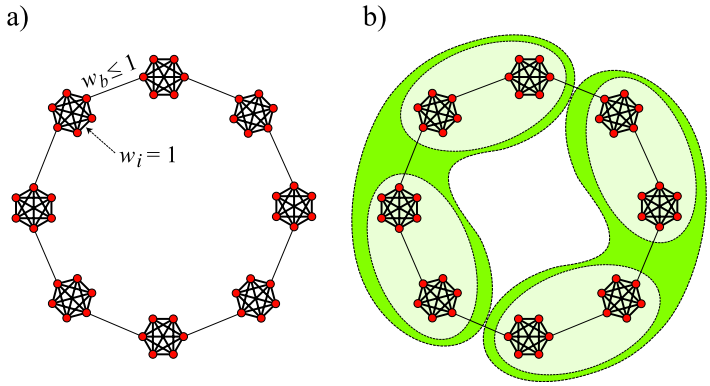


Figure 1. (a) A ring-like network, consisting of N_b cliques, each containing of N_c nodes. Link weights w_i within modules equal unity, whereas modules are joined by links of weight $w_b \leq 1$. (b) The weighted RB method can merge consecutive cliques to larger modules, depending on values of the network parameters and the tuning parameter γ. The hierarchical structure is for illustrative purposes only. In general, the RB method does not yield hierarchical modules.

2.2. Resolution of the weighted RB method for sparse and dense networks

The RB method can be viewed as a general framework for community detection [26], which for the unweighted case includes the modularity optimization method as a special case ($\gamma = 1$ and the configuration model as the null model). Recently, it was shown that the resolution of modularity optimization methods is intrinsically limited [24]. In particular, in large networks small 'physical' communities cannot be resolved and thus there is a lower limit to the size of communities which can be detected by the method. This limit depends on the number of links in the network and is also inherited by the more general RB method [20]. However, by changing the parameter γ, the resolution of the method can be tuned such that small values yield large modules and vice versa. This provides a clear advantage over 'traditional' modularity optimization, which is restricted to a single resolution.

We now address the issue of resolution of the weighted RB method, beginning with a weighted modular network which is sparse, that is, whose average degree $\langle k \rangle \ll N$. Consider a simple case, where the N nodes are arranged into modules of constant size N_c, so that the number of such modules is $N_b = N/N_c$. Let the modules form a ring-like structure, as illustrated in figure 1, and let each module be a fully connected clique. Let the internal links within cliques have weight $w_i = 1$, and successive modules be connected by a single link of weight w_b, where $w_b \leq 1$. This presents perhaps the simplest possible modular structure for a weighted connected network.

The community structure found by the weighted RB method corresponds to the global minimum of the Hamiltonian (or energy) defined in equation (4). Depending of the network parameters N_b, N_c, and w_b as well as the tuning parameter γ, this structure may or may not correspond to the built-in modules. Let us consider two ways to group the built-in modules into communities: the first one is the 'natural' grouping in which each built-in module is identified as a single community. In the second case, we take two successive built-in modules and consider them merged, that is, identified as one community. Other

J. Stat. Mech. (2008) P08007

built-in modules are still considered as separate communities exactly as in the first case. Clearly, if the second grouping has smaller energy (4) than the first one, the resolution of the method is limited. A straightforward calculation shows that this is equal to the requirement

$$w_{mn} > \gamma[w_{mn}] = \gamma \frac{S_m S_n}{S}, \tag{6}$$

where m and n are the built-in modules to be merged, $S = \sum_{i=1}^{N} s_i$ is again the strength sum of the network, and S_q the strength sum of module q. Now, $w_{mn} = w_b$, $S_m = S_n = N_c(N_c - 1) + 2w_b$, and $S = N_b S_m$. Plugging these into equation (6) yields the merging condition for the example network:

$$w_b > \gamma \frac{1}{N_b}(N_c^2 - N_c + 2w_b). \tag{7}$$

Now, let the network size N increase while the module size N_c remains constant. Then, as $N_b = N/N_c$ increases, larger and larger values of γ are needed for obtaining the built-in modules. Increasing w_b makes merging easier, as expected. For $w_b = 1$, equation (7) yields the resolution limit for the similar unweighted network studied in [20].

Let us now move on to a more interesting case where the network in question is fully connected, i.e., links exist between each node, and the modular structure is purely encoded in the weights. Perhaps the simplest possible structure for a fully connected network with modules is the case where N_b modules each consisting of N_c nodes are constructed such that inside the modules the links have weight $w_i = 1$ and links between nodes in different modules have weight w_b ($0 < w_b \leq 1$); see figure 2. Like in the above analysis for the sparse weighted network, we again consider two ways to group the built-in modules to communities: the 'natural' grouping and the one in which two built-in modules are considered as a single module. Again, the method prefers the second grouping over the natural one if it yields smaller energy (equation (4)). The condition for this is again given by equation (6), but now we have $w_{mn} = N_c^2 w_b$ and $S_q = N_c s_i$, where $s_i = N_c - 1 + (N_b - 1)N_c w_b$ denotes the (constant) strength of the nodes. Thus, equation (6) is equivalent to

$$N_c^2 w_b > \gamma N_c^2 \left[\frac{1 - 1/N_c}{N_b} + \left(1 - \frac{1}{N_b}\right) w_b \right] \approx \gamma N_c^2 w_b, \tag{8}$$

where the approximation is valid when N_b is large. In this case, equation (8) further simplifies to $\gamma < 1$, where it should be understood that the specific merging value $\gamma = 1$ appears as a result of the simple structure of the example case. With a more general scope, the expected weight between modules $[w_{mn}] \approx N_c^2 w_b$ is independent of the number of modules N_b, i.e., network size. Thus, merging is solely controlled by γ. This is different from the sparse network case discussed above, where increasing system size eventually triggers merging as the expected number and the total weight of links between modules decreases.

Finally, we analyze the effects of a single strong link between the modules in the latter example case. On the basis of the above analysis, merging happens if the total weight between the two modules exceeds $\gamma[w_{mn}]$, which is again of the order of $\gamma N_c^2 w_b$. For sufficiently large N_c, the expected weight is so large that adding one strong link is not enough for merging to occur. Smaller modules are merged more easily. However, the

J. Stat. Mech. (2008) P08007

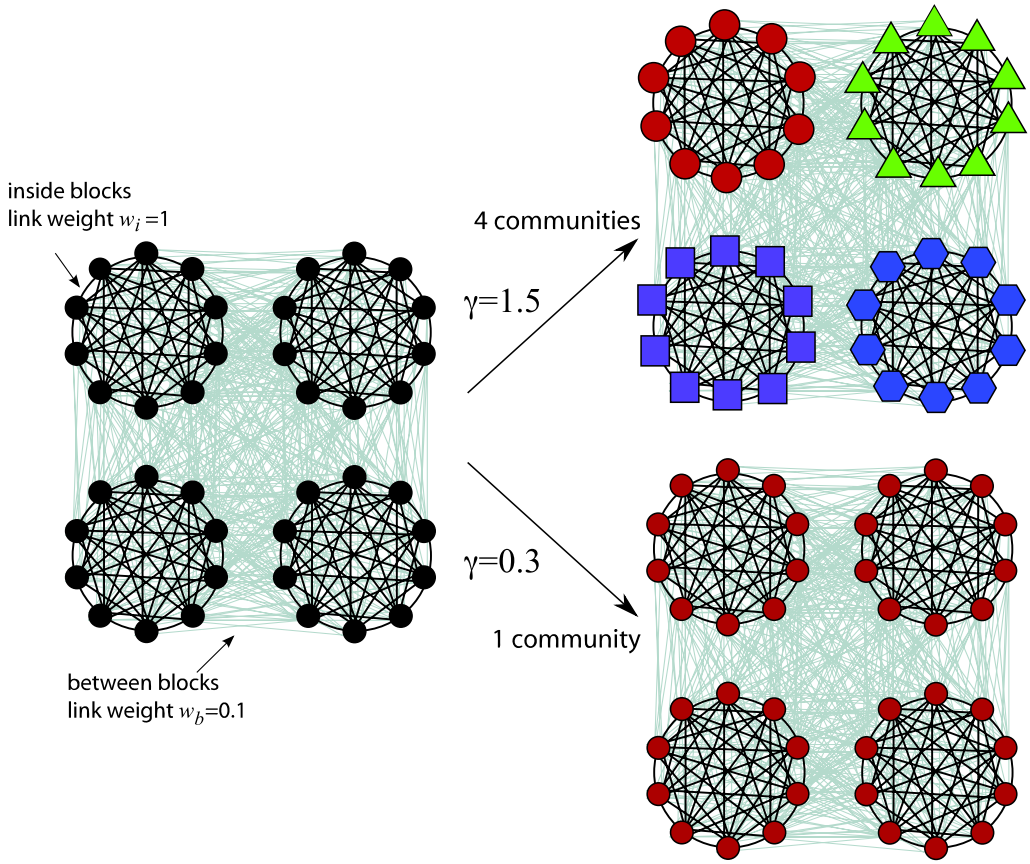


Figure 2. Left: a network consisting of $N_b = 4$ blocks each having $N_c = 10$ nodes. Links inside blocks have weight $w_i = 1$ and nodes in different blocks are connected with links of weight $w_b = 0.1$. On the right is illustrated the effect of γ on the modular structure found. Large values yield the physical communities while for small values the communities appear as one large module. If the number of blocks N_b is large enough, the network size does not affect the γ values where merging happens.

resolution limit still depends only weakly on the number of modules, i.e., system size. This means that sweeping γ can be used to probe communities of different sizes, and the suitable range of γ values is practically independent of the system size.

These considerations show that the resolution of the weighted RB method does not necessarily decrease when dense networks grow in size, unlike for sparse networks. However, for practical purposes, issues such as the distribution of weights both within and between the blocks is expected to affect the actual resolution, and the above examples should be viewed as illustrative only.

3. Example application: modules in a stock correlation network

As a real-world example, we apply the weighted RB method to a correlation-based network of stock return time series. Networks of this type are of special interest as the correlations between asset returns are the main input in the classical and still widely

J. Stat. Mech. (2008) P08007

used Markowitz portfolio optimization theory [32]. Correlations of stock returns were first studied from the network point of view by Mantegna [27], who defined a correlation-based metric and was consequently able to identify modules that make sense also from the economic point of view by using the *maximal spanning tree*. This work has been extended by Bonanno *et al* [28, 33, 34] and Onnela *et al* [35, 36], with the overall conclusion that there is cluster structure which corresponds well to economic sectors. Recently, the structure of correlation-based stock interaction networks has also been studied with the weighted version of the clique percolation method [22] and by spectral and thresholding analyses [10, 29, 37, 38].

To construct our network, we use a data set consisting of the daily closing prices of $N = 116$ NYSE-traded stocks from the time period from 13 January 1997 to 29 January 2000.[5] We estimate the equal time correlation matrix of logarithmic returns by

$$C_{ij} = \frac{\langle \mathbf{r}_i \mathbf{r}_j \rangle - \langle \mathbf{r}_i \rangle \langle \mathbf{r}_j \rangle}{\sqrt{[\langle \mathbf{r}_i{}^2 \rangle - \langle \mathbf{r}_i \rangle^2][\langle \mathbf{r}_j{}^2 \rangle - \langle \mathbf{r}_j \rangle^2]}}, \tag{9}$$

where $\mathbf{r_i}$ is a vector containing the logarithmic returns of stock i. Since there is a small number of elements of $\mathbf{C}$ which are slightly negative, we define the weights of our network by

$$W_{ij} = |C_{ij}| - \delta_{ij}, \tag{10}$$

which can be justified by interpreting the absolute values of correlations as measures of interaction strength without considering whether the interaction is positive or negative.

Here, we take a multiresolution approach to the problem of detecting modules in the above matrix, and sweep the value of γ to obtain the modules of W at multiple levels of resolution. For each value of γ, we assign nodes into modules such that the energy (4) is minimized. Evidently, exploring all possible configurations is computationally impossible, so some approximative method has to be employed. The choice of method naturally depends on the system size, and for very large systems, greedy optimization methods [39, 40] which directly look for local minima might be the only solution. For our case, the system is not very large, and we have chosen the simulated annealing approach, using single-spin flips as well as block flipping as the elementary Monte Carlo operations. It should be noted, however, that it cannot be guaranteed that the energy minimum obtained is a global one. For the RB method, there is no way around this problem.

First, we have investigated the number of modules as a function of γ (see figure 3(a)). For $\gamma \lessapprox 0.8$, all nodes are assigned to a single module. When γ is further increased, the number of modules starts to rapidly increase, until finally each module corresponds to a single node. It is worth noting that no plateaus are seen in the graph, except for the trivial case of $\gamma \lessapprox 0.8$. In [21], using a related multiresolution method, such plateaus were shown to exist for test-case networks, corresponding to built-in hierarchical modules. Plateaus would hence yield 'natural' choices of the tuning parameter. Their absence in figure 3(a) means that there is no range of γ which would correspond to a stable module configuration. However, stability of the number of modules only gives partial insight into the stability of the modular structure. Especially for real-world networks with modules of different sizes and internal weights, changes in this number may only reflect e.g. splitting

[5] The length of the time series is 1000 trading days.

J. Stat. Mech. (2008) P08007

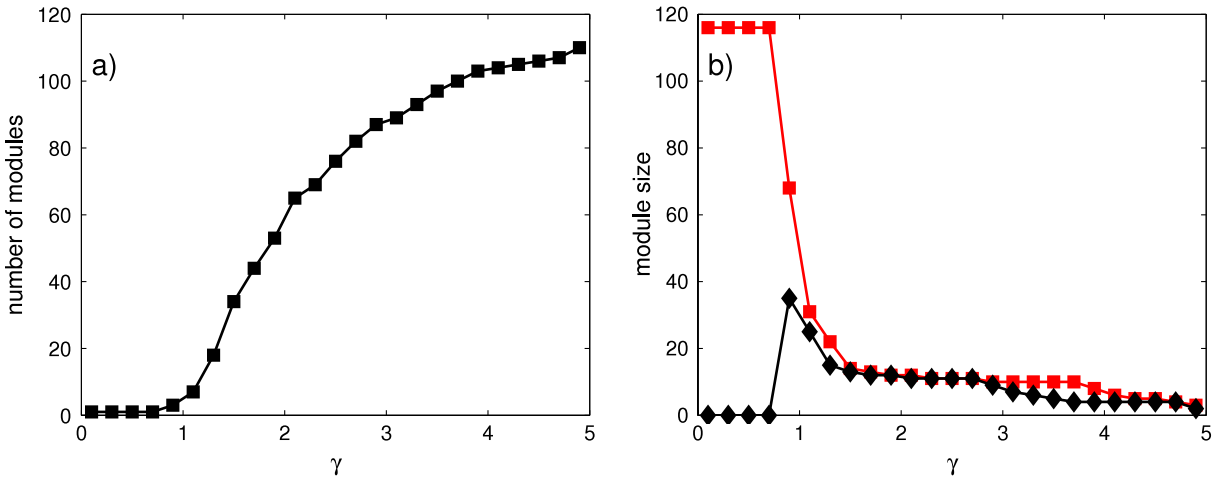


Figure 3. The number of modules (a) and the sizes of the two largest modules (b) as a function of γ.

of small, weak modules, while the strongest modules remain more or less stable when γ is increased. This appears to be the case for our stock interaction network. Panel (b) of figure 3 depicts the sizes of the two largest modules as a function of γ. The sizes remain almost constant for an interval of approximately $\gamma \in [1.4, 3]$, and thus the increase in the module number can be attributed to splitting of smaller modules.

Next, we turn to the modules themselves. In order to visually compare the detected modules with known structural features of the correlation matrix investigated we have utilized the maximal spanning tree (MST) method. The MST of a network or a matrix is a tree connecting all the N nodes with $N-1$ links, such that the sum of the link weights is maximized. Earlier, it was shown that for stock correlation matrices, branches of the MST correspond well to business sectors or industries for the NYSE [27], [33]–[36] as well as FTSE [41]. The typical way to categorize stocks into business sectors is to use the Forbes classification [42]. Panel (a) in figure 4 displays the MST for the stock network, together with the Forbes classification. For comparison, we first set $\gamma = 1$ (figure 4(b)), and color the nodes according to modules detected by the RB method for the full correlation matrix as above. The value $\gamma = 1$ is of particular interest, as in this case the Hamiltonian of equation (1) is equivalent with the weighted version of modularity [12]. For this value, four modules of sizes 13, 34, 34 and 35 are found. For each module, the majority of member nodes are also connected in the MST, and there is a correspondence between the MST branches and the modules. The smallest module corresponds very well to the 'Energy' sector in the Forbes classification, and the other modules roughly to combinations of different sectors. It should be noted here that the Forbes classification is an external one, i.e., it is not based on empirical observations on stock correlations, and thus some Forbes sectors are also relatively disjoint in the MST of figure 4(a).

Let us now change the resolution of the RB method by moving towards larger values of γ. Panel (c) of figure 4 displays the modular structure obtained with $\gamma = 1.4$, i.e., at the onset of the 'plateau' regime of the two largest module sizes. Only modules of size larger than two are depicted by different colors, while the rest of the nodes are indicated by open symbols. An immediate observation is that the modules correspond remarkably well to

J. Stat. Mech. (2008) P08007

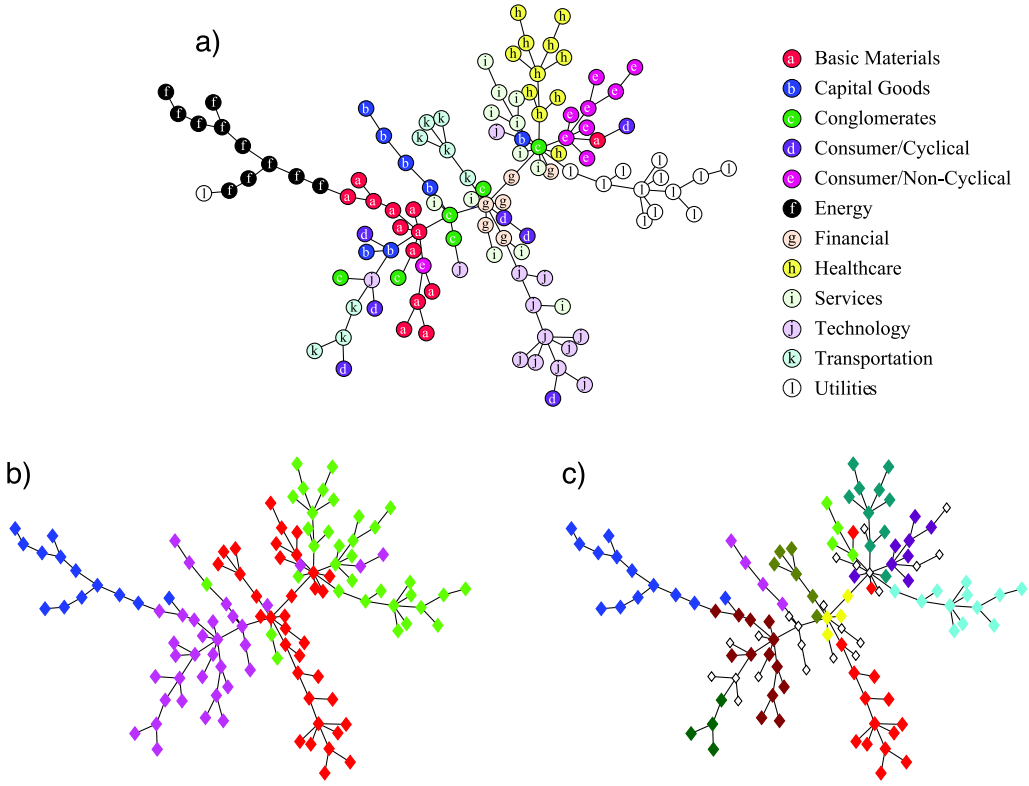


Figure 4. (a) The maximal spanning tree and business sectors according to Forbes [42]. (b) The maximal spanning tree and the modular structure for $\gamma = 1$. Each color corresponds to a module. (c) The maximal spanning tree and the modular structure for $\gamma = 1.4$. Modules of size larger than 2 are depicted in different colors and the rest of the nodes by empty symbols.

the different branches of the MST and very well to the Forbes classification. Increasing γ further splits the modules into smaller ones: for $\gamma = 2$ the number of modules is already 58 and thus their average size is only 2. The largest modules, corresponding to the 'Energy' sector and the 'Electric Utilities' industry, are the last ones to break at around $\gamma \approx 3$ and $\gamma \approx 4$, respectively. Interestingly, the 'Energy' module seems to contain a strong submodule of four nodes. This is also seen as a plateau in the graph depicting the size of the second-largest component (figure 3(b)), which indicates that large values of γ can also yield useful information on the modules.

Finally, we study the correspondence between the modular structure obtained with the RB method and the Forbes classification into business sectors in a more quantitative way. We use two measures defined in [15]: the *sensitivity* defined as the fraction of pairs of nodes classified into the same Forbes sector that are assigned to the same module by the RB method and, correspondingly, *specificity* as the fraction of pairs of nodes belonging to different sectors that are assigned to different modules by the RB method. Sensitivity and specificity are depicted in figures 5(a) and (b), respectively. The sensitivity curve shows a sudden increase in the interval $\gamma \in [0.8, 1.8]$. The reason for its low initial value is the assignment of all nodes to a single module, as discussed above, and the increase corresponds to modules splitting into smaller units which correspond well to the Forbes

J. Stat. Mech. (2008) P08007

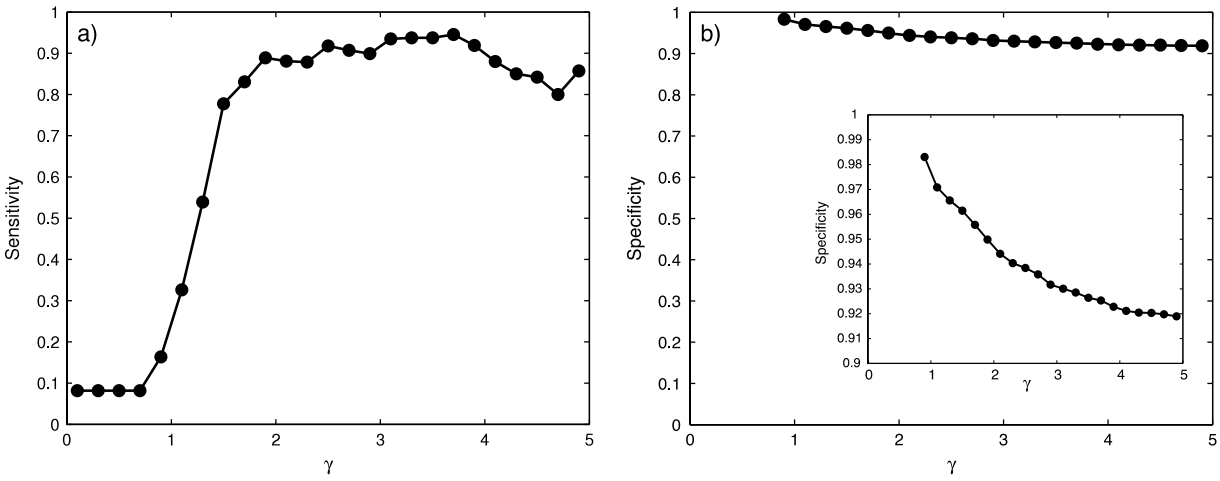


Figure 5. The sensitivity (a) and the specificity (b) of the modular structure with respect to the Forbes classification of business sectors [42] as a function of γ. The solid line is a guide to the eye.

classification. The high value of sensitivity for large γ means that the relatively small modules given by the RB method are proper subsets of the Forbes business sectors. The specificity curve shows a decreasing trend, but its values still remain relatively high. This trend is explained by an increasing number of small modules (including modules consisting of one node only), such that nodes which belong to a common sector appear in different modules. Overall, the above results indicate that the modular structure detected by the weighted RB method corresponds well to the Forbes classification for a wide range of γ, and the small modules obtained at large γ seem to be valid submodules of larger ones.

For comparison, we have also carried out the above analysis using the recently introduced weighted multiresolution method of Arenas *et al* [21]. This method resembles the Potts approach; however, the tuning parameter γ is replaced by the parameter r, which can be interpreted as representing the weight of a self-link added to each node. The number of modules, the sizes of the two largest modules, the sensitivity and the specificity as functions of the tuning parameter r are depicted in figure 6. Comparison with figures 3 and 5, in which the same results for the RB method are shown, suggests that for the correlation matrix analyzed here, the AFG and RB methods behave in a very similar manner.

4. Conclusions

Here we have presented, analyzed, and applied a weighted version of the q-state Potts model approach by Reichardt and Bornholdt [15], introducing a well-motivated null model for expected weights within modules. Our target has been to investigate the modular structure of dense weighted networks such that instead of the topology, the link weights determine the modules. In contrast to conventional approaches, where weights considered insignificant are filtered out, ours has the target of utilizing all information contained in the weight matrix. The weighted RB model fulfills this criterion, as it can equally well be applied to sparse and to dense networks. In addition, it contains a parameter that allows

J. Stat. Mech. (2008) P08007

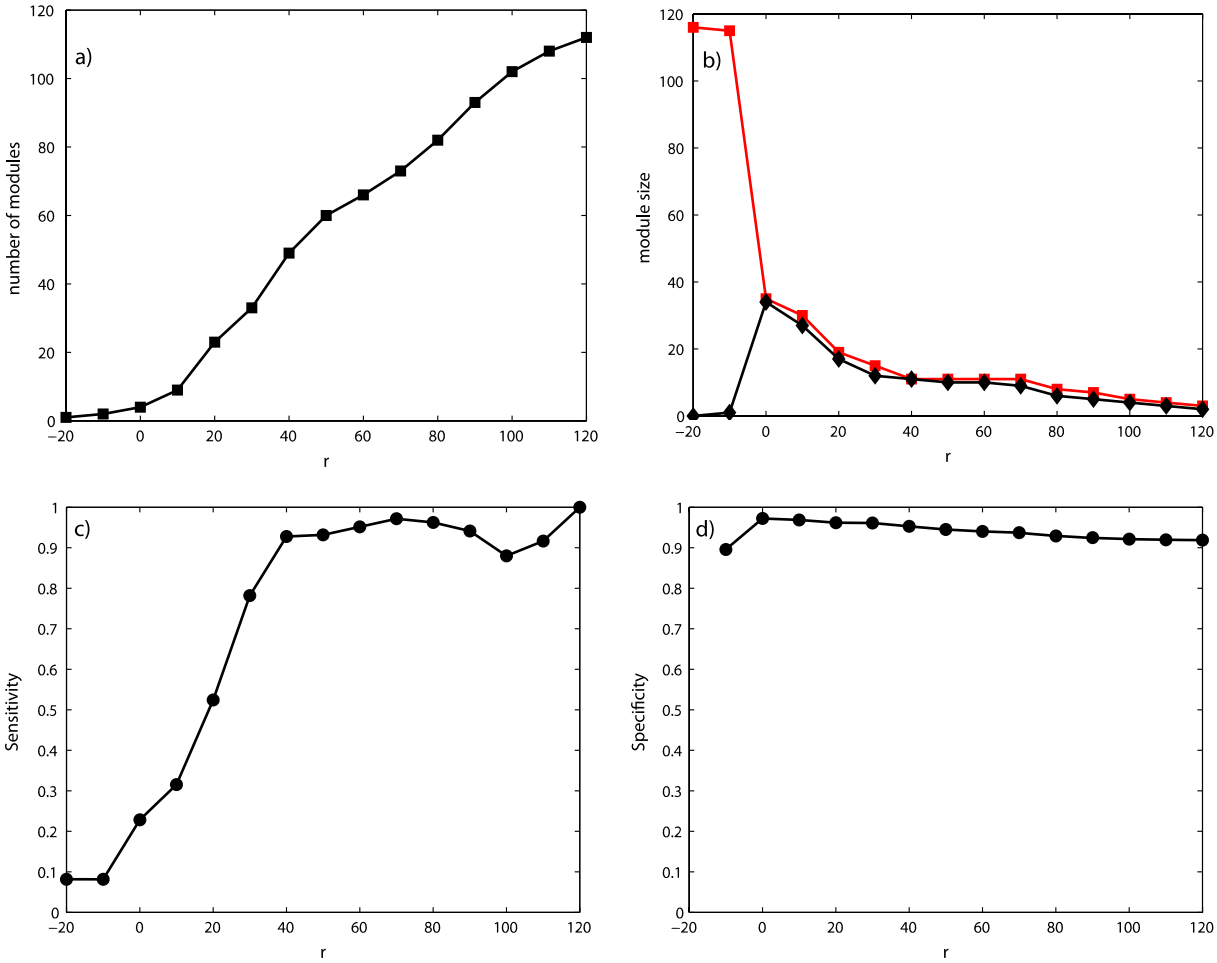


Figure 6. The number of modules (a), the sizes of the two largest modules (b), the sensitivity (c) and the specificity (d) as functions of r with the AFG method. The solid line is a guide to the eye.

tuning its resolution, which is useful for studies of nested community structures. Analysis of the resolution limit of the method has shown that for simple example cases, dense modular networks behave differently from sparse ones, as the resolution is only weakly dependent on the network size. As a practical application, we have used the method in analysis of the modular structure of a stock correlation matrix. Our results indicate that on varying the tuning parameter value, the method is able to detect modules which correspond to relevant business sectors, as well as substructure inside these modules. Thus it turns out that the weighted Potts method provides a feasible approach to community detection in dense networks.

Acknowledgments

Support by the Academy of Finland, the Finnish Center of Excellence Program 2006-2011, Project No. 213470, is acknowledged. JK is partly supported by the GETA graduate school. JS acknowledges support by the European Commission NEST Pathfinder initiative on Complexity through project EDEN (Contract 043251).

References

[1] Newman M E J, Barabási A L and Watts D J, 2006 *The Structure and Dynamics of Networks* (Princeton, NJ: Princeton University Press)
[2] Dorogovtsev S N and Mendes J F F, 2003 *Evolution of Networks: from Biological Nets to the Internet and WWW* (New York: Oxford University Press)
[3] Newman M E J, 2003 *SIAM Rev.* **45** 167
[4] Caldarelli G, 2007 *Scale-Free Networks* (Oxford: Oxford University Press)
[5] Saramäki J, Kivelä M, Onnela J P, Kaski K and Kertész J, 2007 *Phys. Rev.* E **75** 027105
[6] Barrat A, Barthélémy M, Pastor-Satorras R and Vespignani A, 2004 *Proc. Nat. Acad. Sci.* **101** 3747
[7] Colizza V, Barrat A, Barthélémy M and Vespignani A, 2006 *Proc. Nat. Acad. Sci.* **103** 2015
[8] Onnela J P, Saramäki J, Hyvönen J, Szabó G, Lazer D, Kaski K, Kertész J and Barabási A L, 2007 *Proc. Nat. Acad. Sci.* **104** 7332
[9] Lambiotte R, Blondel V D, de Kerchove C, Huens E, Prieur C, Smoreda Z and Van Dooren P, 2008 *Preprint* 0802.2178v1
[10] Heimo T, Tibély G, Saramäki J, Kaski K and Kertész J, 2008 *Physica* A **387** 5930
[11] Rozenfeld A, Arnaud-Haond S, Hernández-García E, Eguíluz V, Matias M, Serrão E and Duarte C, 2007 *J. R. Soc. Interface* **4** 1093
[12] Newman M E J and Girvan M, 2004 *Phys. Rev.* E **69** 026113
[13] Palla G, Derényi I, Farkas I and Vicsek T, 2005 *Nature* **435** 814
[14] Guimerá R and Amaral L A N, 2005 *Nature* **433** 895
[15] Reichardt J and Bornholdt S, 2004 *Phys. Rev. Lett.* **93** 218701
[16] Kumpula J M, Onnela J P, Saramäki J, Kaski K and Kertész J, 2007 *Phys. Rev. Lett.* **99** 228701
[17] Fortunato S and Castellano C, 2007 *Preprint* 0712.2716
[18] Hastings M B, 2006 *Phys. Rev.* E **74** 035102(R)
[19] Hofman J M and Wiggins C H, 2008 *Phys. Rev. Lett.* **100** 258701
[20] Kumpula J M, Saramäki J, Kaski K and Kertész J, 2007 *Eur. Phys. J.* B **56** 41
[21] Arenas A, Fernández A and Gómez S, 2008 *New J. Phys.* **10** 053039
[22] Farkas I, Ábel D, Palla G and Vicsek T, 2007 *New J. Phys.* **9** 180
[23] Lancichinetti A, Fortunato S and Kertesz J, 2008 *Preprint* 0802.1218
[24] Fortunato S and Barthélémy M, 2007 *Proc. Nat. Acad. Sci.* **104** 36
[25] Kumpula J M, Saramäki J, Kaski K and Kertész J, 2007 *Fluct. Noise Lett.* **7** 209
[26] Reichardt J and Bornholdt S, 2006 *Phys. Rev.* E **74** 016110
[27] Mantegna R N, 1999 *Eur. Phys. J.* B **11** 193
[28] Bonanno G, Vandewalle N and Mantegna R N, 2000 *Phys. Rev.* E **62** 7615
[29] Onnela J P, Kaski K and Kertész J, 2004 *Eur. Phys. J.* B **38** 353
[30] Erdös P and Rényi A, 1959 *Publ. Math. Debrecen* **6** 290
[31] Newman M E J, 2004 *Phys. Rev.* E **70** 56131
[32] Markowitz H, 1952 *J. Finance* **7** 77
[33] Bonanno G, Caldarelli G, Lillo F and Mantegna R N, 2003 *Phys. Rev.* E **68** 046130
[34] Bonanno G, Caldarelli G, Lillo F, Micciché S, Vandewalle N and Mantegna R N, 2004 *Eur. Phys. J.* B **38** 363
[35] Onnela J P, Chakraborti A, Kaski K, Kertész J and Kanto A, 2003 *Phys. Rev.* E **68** 056110
[36] Onnela J P, Chakraborti A, Kaski K and Kertész J, 2002 *Eur. Phys. J.* B **30** 285
[37] Heimo T, Saramäki J, Onnela J P and Kaski K, 2007 *Physica* A **383** 147
[38] Onnela J P, Chakraborti A, Kaski K, Kertész J and Kanto A, 2003 *Phys. Scr.* **106** 48
[39] Blondel V D, Guillaume J L, Lambiotte R and Lefebvre E, 2008 *Preprint* 0803.0476
[40] Ronhovde P and Nussinov Z, 2008 *Preprint* 0803.2548
[41] Coelho R, Hutzler S, Repetowicz P and Richmond P, 2007 *Physica* A **373** 615
[42] Forbes http://www.forbes.com (referenced in March–April 2002)

J. Stat. Mech. (2008) P08007

Sequential algorithm for fast clique percolation

Jussi M. Kumpula,* Mikko Kivelä, Kimmo Kaski, and Jari Saramäki
Department of Biomedical Engineering and Computational Science, Helsinki University of Technology, P.O. Box 9203, FIN-02015 HUT, Finland

(Received 12 May 2008; revised manuscript received 27 June 2008; published 15 August 2008)

In complex network research clique percolation, introduced by Palla, Derényi, and Vicsek [Nature (London) **435**, 814 (2005)], is a deterministic community detection method which allows for overlapping communities and is purely based on local topological properties of a network. Here we present a sequential clique percolation algorithm (SCP) to do fast community detection in weighted and unweighted networks, for cliques of a chosen size. This method is based on sequentially inserting the constituent links to the network and simultaneously keeping track of the emerging community structure. Unlike existing algorithms, the SCP method allows for detecting k-clique communities at multiple weight thresholds in a single run, and can simultaneously produce a dendrogram representation of hierarchical community structure. In sparse weighted networks, the SCP algorithm can also be used for implementing the weighted clique percolation method recently introduced by Farkas *et al.* [New J. Phys. **9**, 180 (2007)]. The computational time of the SCP algorithm scales linearly with the number of k-cliques in the network. As an example, the method is applied to a product association network, revealing its nested community structure.

DOI: 10.1103/PhysRevE.78.026109 PACS number(s): 89.75.Fb, 89.75.Hc

I. INTRODUCTION

Over the last decade, complex networks have become a standard framework in the study of complex systems [1,2]. The simplicity of the network representation, where the interactions and interacting elements are mapped to links and nodes, respectively, facilitates its use on a number of systems, ranging from human societies to biological systems. One prominent feature of complex networks is related to their mesoscopic properties. Networks often display modular structure, i.e., are structured in terms of modules or communities, which are, in general, sets of densely interconnected nodes. Such communities are often closely related to functional units of the system, for example, groups of individuals interacting with each other in society [3–6] or functional modules in metabolic networks [7–9].

The problem of detecting communities in complex networks has received a lot of attention over the last few years. This problem is twofold: first, there is no unique way to rigorously define what constitutes a community. For any definition, several choices have to be made: whether communities are defined using local or global network properties, whether nodes can participate in several communities, and whether the definition allows for weighted networks and nested hierarchy of communities. Second, any definition is useful in practice only if it can be reformulated as an algorithm which scales well enough to allow processing networks of large enough size. As a result, a large number of community definitions and their algorithmic implementations have been proposed over the recent years [10–15]; for a review see Ref. [16].

In this paper we focus on a fast algorithmic implementation of the clique percolation (CP) method, originally introduced by Palla *et al.* [9]. The CP method is deterministic and it is based solely on local topological properties, defining a k-clique community as a set of nodes belonging to adjacent k-cliques. This allows for overlapping communities, i.e., nodes having multiple community memberships. The CP method has earlier been successfully applied to various community detection problems: detection of protein communities related to cancer metastasis [17], analysis of communities in coauthorship, word association, and protein-interaction networks [9], and time evolution of social groups [6]. Contrary to existing implementations [18], which detect k-clique communities for all values of k by first finding the maximal cliques by an exponentially scaling algorithm [9], we focus on rapid detection of communities for a chosen value of k. Our sequential clique percolation (SCP) algorithm is based on sequentially inserting links to the network and keeping track of the emerging community structure. It has specifically been designed for weighted networks containing hierarchical communities which are reflected in the link weights. When links are inserted in decreasing order of weight, the algorithm allows for detecting k-clique communities at chosen threshold levels in a single run and simultaneously produces a dendrogram representation of hierarchical community structure. In addition, the algorithm can be used for very fast community detection for unweighted networks.

This paper is structured as follows. First, we present our algorithm for the simplest, unweighted case, and discuss its scaling properties. We then move on to detecting nested communities in weighted networks, applying the algorithm to a product association network generated from data on sellers and products on an online auction site. Finally, we discuss a variation of the algorithm which is based on ordering k-cliques according to their weighted properties, and present our conclusions.

II. THE SCP ALGORITHM

Let us begin by defining k-cliques and k-clique communities [9,19]: A k-clique is a set of k nodes which are all

*jkumpula@lce.hut.fi

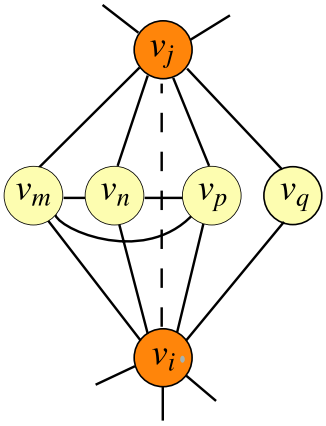


FIG. 1. (Color online) Schematic illustration of the process for detecting the k-cliques a newly inserted link completes. The dashed line depicts the new link, inserted between nodes v_i and v_j. The common neighbors of nodes v_i and v_j are $\mathcal{N}_{ij}=\{v_m,v_n,v_p,v_q\}$. For detecting newly formed 4-cliques, all pairs of nodes in $\mathcal{N}_{ij}$ are checked to see if they are connected, that is, if they form a 2-clique. Each 2-clique in the set gives rise to a 4-clique, so in total the link l_{ij} will generate three 4-cliques. In the case $k=5$, only one 3-clique is found, which contains the nodes v_m, v_n, and v_p. It will give rise to a single 5-clique including these nodes in addition to v_i and v_j.

connected to each other. A k-clique community, or k-community, is a set of nodes which can be reached by a series of overlapping k-cliques, where overlap means that the k-cliques share $k-1$ nodes.

It should be noted that 2-cliques correspond to pairs of nodes connected by single links and 1-cliques to single nodes. Given a network Γ, the goal is then to find the k-communities defined as above. In our case, we restrict ourselves to some specific values of k. Usually choosing $k=3$ or $k=4$ yields useful information, and currently these values of k have yielded, to our knowledge, the most relevant communities in practical applications [6,9,17,20]. Our algorithm is based on detecting and storing k-communities as they emerge and consolidate when links are sequentially inserted into the network. One can think of the process as first "removing" each link l from the network Γ, and then inserting them back one by one. For unweighted networks, the links can be inserted in any order, whereas for weighted networks, it may be desirable to sort the links by weight.

Our algorithm for detecting k-communities consists of two phases: The first phase of the algorithm detects k-cliques which form when a link is inserted. These are then fed to the second phase, which keeps track of formation and merging of k-communities by processing the k-cliques found. The two parts of the algorithm are described in detail below.

A. Phase I: Detecting the k-cliques

The first part of the algorithm involves detecting k-cliques which are formed when a link is inserted into the network. Suppose now that the inserted link connects nodes v_i and v_j (see Fig. 1). The minimum requirement for a new k-clique to form is that nodes v_i and v_j both have degree of at least $k-1$. If this is the case, the algorithm proceeds by collecting all nodes that are neighbors of both nodes $\mathcal{N}_{ij}=\mathcal{N}_i\cap\mathcal{N}_j$, where $\mathcal{N}$ denotes neighborhood. Now, when the link l_{ij} is added, each $(k-2)$-clique contained in the set $\mathcal{N}_{ij}$ will give rise to a new k-clique. Therefore, all newly formed k-cliques are found by detecting all the $(k-2)$-cliques in the $\mathcal{N}_{ij}$. For commonly used small clique sizes, this is very fast: for 3-cliques, $(k-2)$-cliques are single nodes, while for $k=4$, all connected pairs of nodes in $\mathcal{N}_{ij}$ give rise to a new 4-clique. Next the k-cliques detected as above are fed one by one into the second phase of the algorithm.

B. Phase II: Detecting the k-communities

The second phase of the algorithm detects and keeps track of k-communities which form and merge when new k-cliques are input from the first phase. Because a k-community is defined as a set of nodes which all can be reached by a series of overlapping k-cliques, the crucial issue here is the efficient detection of overlap between k-cliques. A naive approach would be to search for shared sets of $k-1$ nodes between the newly input clique and all existing cliques. However, the required computational effort makes this approach unpractical. Instead, we take advantage of the sequential nature of the process by "locally" detecting possible overlap of each new k-clique with existing k-communities and by updating the community structure accordingly.

Let us begin by noting that the k-community structure of a network can be represented by a bipartite network, where the two types of nodes represent k-cliques and $(k-1)$-cliques. In this network, a link exists between two nodes of different type if the k-clique contains the $(k-1)$-clique as a subclique. This is illustrated in Fig. 2. The usefulness of this representation becomes apparent in the following: each connected component in this bipartite network corresponds to a k-clique community, because by definition k-cliques belonging to the same community are connected through shared $(k-1)$-cliques. Furthermore, connected components of the unipartite projections of the bipartite network [21] similarly correspond to k-clique communities. In the following, we focus on the $(k-1)$-clique projection of this bipartite network. We denote the network resulting from this projection by Γ^*. In this unipartite network, nodes v^* represent the $(k-1)$-cliques of Γ, and links l^* exist between nodes which are subcliques of the same k-clique.

For the sake of clarity, we will first present a "physical" interpretation of phase II of the algorithm, and then discuss the algorithmic implementation where certain shortcuts can be made. Similarly to phase I, where the original network Γ is reconstructed link by link, phase II of the SCP algorithm sequentially builds up Γ^* from the k-cliques brought forward from phase I. At the same time, it keeps track of the connected components of Γ^* [see Fig. 2, panels (c) and (d)]. These correspond to k-clique communities. When a new k-clique is input from phase I, its constituent $(k-1)$-cliques are first extracted; obviously there are always k of such subcliques. Each of these $(k-1)$ cliques corresponds to a node in Γ^*. Some of these nodes may already be present, if the corresponding $(k-1)$-cliques have been handled earlier as part of another k-clique; if not, they are created at this stage.

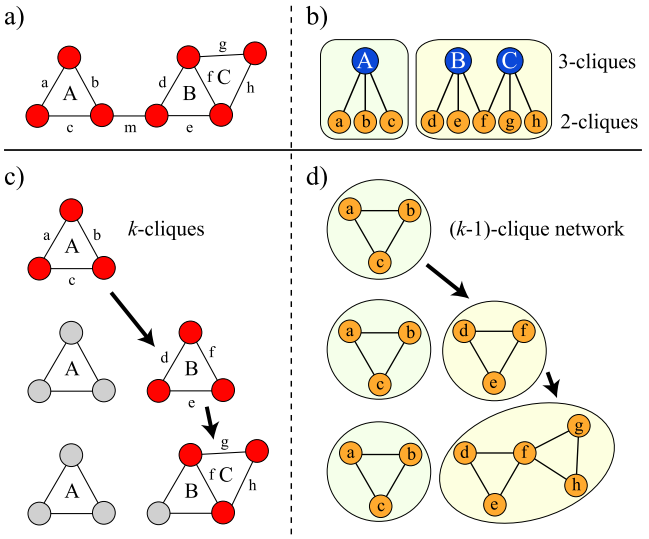


FIG. 2. (Color online) Illustration of the algorithm for detecting k-clique communities in a simple example network. Here, $k=3$. (a) The original network Γ consists of three 3-cliques labeled A, B, and C. 2-cliques, i.e., nodes connected by single links, are labeled with lower case letters. (b) Bipartite network presentation of the clique structure. Note that in the bipartite network, the 3-cliques B and C, which form a 3-clique community, are connected by the shared 2-clique f. Clique A forms another 3-clique community. (c) 3-cliques detected by the first part of the algorithm as links are sequentially inserted into the network. Each new k-clique is denoted by dark nodes whereas nodes associated with existing k-cliques appear gray. (d) Corresponding updates to the $(k-1)$-clique network Γ^* as a result of the second part of the algorithm. k-clique communities correspond to connected components of this network (shaded areas).

Finally, links are created between members of this set of k nodes, and resulting changes in the connected component structure of Γ^* are recorded.

In the algorithmic implementation, things can be done somewhat more efficiently, resembling techniques used in link percolation. The actual network Γ^* does not need to be constructed, as it is enough to keep track of its connected components, i.e., the component indices of its nodes v^*. This is equal to link percolation in Γ^*, which can be implemented for example with disjoint-set forests [22]. At this stage it is enough to ensure that all $(k-1)$-clique-nodes corresponding to the new k clique are marked to belong to the same component [the new $(k-1)$-cliques and their links may either form a new connected component, merge with an existing component, or join together at most k existing components].

The above process is then repeated for each k-clique input from phase I. Finally, once all links have been inserted (phase I) and the subsequently formed k-cliques handled (phase II), the k-communities of the original network Γ can be read from the component indices of v^*, assigning nodes of Γ to their corresponding communities.

In theory, it would also be possible to keep track of the connected components of the whole bipartite network or alternatively project the bipartite network to k-cliques instead of $(k-1)$-cliques. Both representations contain the same connected components and would thus yield the same k-clique communities. However, the former alternative is unnecessarily complicated as it involves nodes of two types. The latter implementation is not as computationally effective as the current choice in cases where a newly inserted k-clique overlaps with a large number of existing k-cliques.

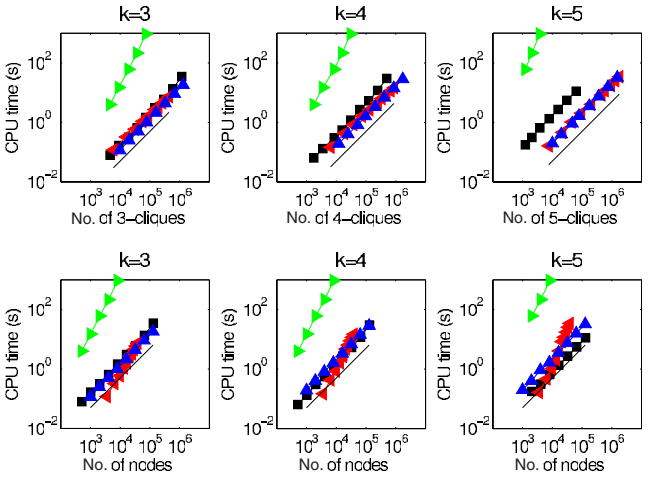


FIG. 3. (Color online) Computation time of the algorithm for three values of k, as a function of the number of k-cliques (upper row) and network size (lower row). Symbols denote different test networks: GN (■), WSN (▲), and CM (▲), see text for details. The solid line is a linear reference. For comparison, we have also plotted the computational time of the CFINDER 1.21 algorithm for the GN networks (▶). Note that CFINDER always processes all values of k.

C. Scaling of the algorithm

Let us next discuss the performance of the SCP algorithm, before moving on to its application to weighted network analysis. Obviously, the computational time required to process a network depends on its properties; here, we wish to investigate the performance as a function of network size and the number of k-cliques contained in the network. To do this, we have applied the SCP algorithm on three types of networks with adjustable sizes. The first test case, introduced by Girvan and Newman [3] (GN), contains built-in communities and has often been used for similar purposes. The GN networks used here consist of groups of 32 nodes, where each node has on the average 12 links to nodes of the same group and 4 links to other groups. The network size N is varied by changing the number of such groups. The second type of networks (WSN) is generated using a recently published model of weighted social networks with communities [23], using parameter values similar to the original reference. As the third type, we have used coauthorship networks based on the cond-mat (CM) archive, constructed similarly to, e.g., Ref. [24]. However, in order to vary the network size, we have used time windows of varying length, such that two authors are connected if they have published a joint paper during the time window. It should be noted that although the WSN networks are inherently weighted, and the CM networks can also be considered such, here we consider binary versions of both types for the performance analysis.

Results in Fig. 3 show that the computational time of the

SCP algorithm grows practically linearly as a function of the number of k-cliques for all networks. This is as expected, because the computational time of the algorithm is dominated by the process of detecting k-cliques and processing them for overlap, such that each k-clique is processed exactly twice. This is also reflected in the network size dependence of the required computational time for both types of model networks (GN, WSN). For these networks the local structure remains essentially unchanged as the network grows and it appears that the number of k-cliques grows linearly with N. However, for the CM networks, the computational time grows faster than linearly as a function of network size. This is because the CM network is a projection of a bipartite author-publication network containing large cliques that grow in size when N increases. The problem is, as pointed out by Palla *et al.* [9], that the number of subcliques of size k within a clique of size s is $\binom{s}{k}$. In the limit $s \gg k$ this leads to

$$\binom{s}{k} \approx \frac{s^k}{k!}. \tag{1}$$

Hence for large s, the number of k-cliques grows as kth power of s, meaning that for networks containing large cliques the SCP method performs best for rather small values of k. For example, when $k > 10$ the analysis of the largest CM networks becomes extremely slow with the SCP method. However, when very large cliques are not abundant in the network under investigation, the SCP algorithm is very fast even for networks of large size. For example, detecting 4-clique communities in a mobile phone call network having approximately four million nodes and six million links [25] takes approximately one minute on a standard desktop computer. Thus, for networks where cliques are on the average fairly small, the main practical limitations of this algorithm seem to be related to the memory consumption as it requires keeping all $(k-1)$-cliques of the network in memory.

Finally, let us compare the performance of the SCP algorithm and the existing method (CFINDER 1.21, [9]). Evidently, this comparison is somewhat complicated, as CFINDER simultaneously processes all clique sizes, whereas the SCP algorithm is by construction limited to a single value of k. Nevertheless, summing up the processing times for all values of k, we have observed that for the GN network, the processing time of the SCP algorithm scales linearly with network size, whereas CFINDER 1.21 appears to scale as N^2 (see Fig. 3). However, for denser networks, such as the CM network, the comparison becomes somewhat meaningless as both methods become extraordinarily slow. This is due to the very large number of k-cliques as discussed above. It should be noted here that the unpublished beta version, CFINDER 2.0b, appears to scale far better than CFINDER 1.21 and seems to be able to deal with very large cliques. However, the key strength of the SCP algorithm is its speed in weighted network analysis: it is able to process multiple weight thresholds in a single run (see Sec. III A below). With the earlier method, this quickly becomes unfeasible, as the networks corresponding to each threshold have to be separately input and analyzed. Thus, even if the processing time of both methods would be exactly the same for a single network, obtaining the k-community structure for 100 weight thresholds would be 100 times faster with the SCP algorithm. Another important difference is the inherent ability of the SCP method to produce a dendrogram of nested k-communities; this feature does not exist in earlier implementations (again, see Sec. III A below).

III. SCP FOR WEIGHTED NETWORKS

A. Thresholding and nested communities

Let us move on to weighted networks, where the concept of community structure becomes somewhat more complicated. Perhaps only for the very simplest cases, where the networks are sparse, weights can be disregarded, such that communities are associated with the pure topology of the network. However, this is usually not feasible, as weighted networks can be rather dense, even to such an extent that the topology no longer matters, as any modular structure is encoded in the link weights only. This is the case, for example, in stock interaction networks [26], whose natural representation is a weight matrix with only nonzero elements.

For such networks, one is essentially left with two choices: the first is to threshold the network, such that links whose weights are considered insignificantly small are removed and communities in the resulting sparse network are detected. It is evident that choosing the right threshold is a nontrivial task; in fact, for many cases it may be better to take a multiresolution approach, by investigating the resulting community structure for a range of thresholds. Another option is to consider the weights directly when defining what constitutes a community, and to apply a method which is based on this definition [20,26].

In the original formulation of the clique percolation algorithm, Palla *et al.* suggested a rule for choosing a weight threshold w^* for the network, such that the resulting k-community structure would be as diverse as possible [9]. More specifically, w^* is chosen such that the largest community is twice the size of the second largest one, i.e., below the percolation threshold where a giant k-clique community appears. For the original implementation, the algorithm had to be run from the beginning for each threshold level. One of the benefits of our approach is that it allows for obtaining k-communities at any point of the process of adding links, which is just thresholding done in reverse: If the links of the original network Γ are sorted and processed in descending order of weight, the algorithm yields for each link the k-community structure of Γ thresholded by the weight of the link. This is very useful for selecting the threshold, as all threshold values can be processed in a single run. Note that for dense networks, sweeping through the entire range of weights is not needed: the algorithm can be stopped before (or immediately after) communities are entirely "smeared out" by a giant community. Stopping the algorithm in time can greatly reduce the workload in dense networks as usually only a small fraction of all links need to be added before the percolating component is found, after which adding more links does not increase the number of nodes in the communities, but only makes the community denser in cliques.

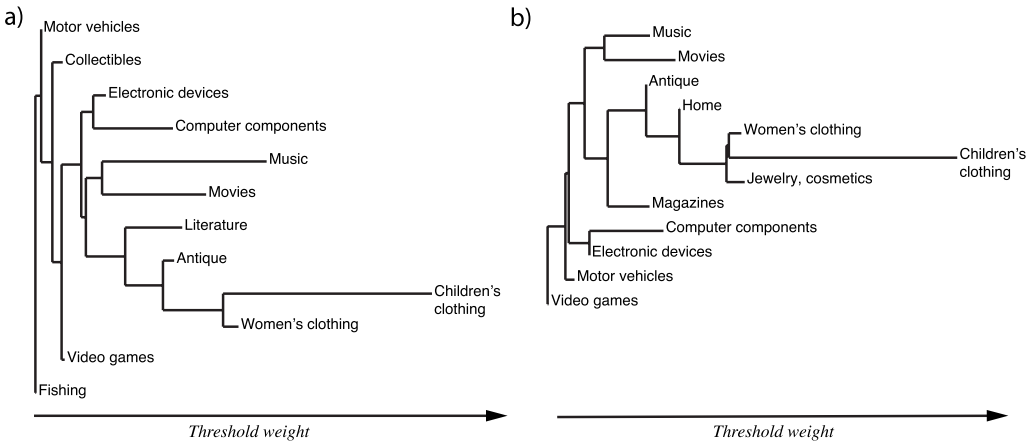


FIG. 4. Dendrogram visualization of the nested k-community structure of the trading categories of the Finnish online auction site Huuto.net for $k=3$ (a) and $k=4$ (b).

However, by focusing on a single threshold weight, valuable information of the community structure contained in the correlations between weights can be lost. Often, the modular structure of networks is inherently hierarchical—denser and stronger communities are nested inside weaker ones, which may further be embedded inside even weaker ones [15,27–29]. It is then natural to investigate this nestedness by considering the development of the community structure when the weight threshold is swept through the range of interest. Evidently, this requires book-keeping of the emergence and merging of communities as the threshold is progressively lowered. For the SCP algorithm, this book-keeping is inbuilt: all necessary information can directly be recorded in phase II of the algorithm. In particular, it is easy to store when a k-community appears, which nodes belong to it, how its size grows as new k-cliques join it, and when it merges with other k-communities. It should be stressed here that this is a genuine advantage: separately detecting the community structure for each threshold and then tracking the formation and merging of communities would be very difficult and time consuming.

This information on the nested community structure is best visualized with a dendrogram, which is a common presentation format in agglomerative community detection (see, e.g., Ref. [29]). In a dendrogram, horizontal lines correspond to communities, and a branching of the lines denotes communities merging. Choosing a single weight threshold would correspond to taking a vertical slice of the dendrogram. Figure 4 shows two examples of the nested community structure within a product category network, for $k=3$ and $k=4$. This network is constructed from online trading data, downloaded from the Finnish auction website Huuto.net. In this network, nodes correspond to product categories ($N=345$), and the weights of links connecting two categories to the number of individuals who have been trading in both of them. This network is very dense, the number of links is 52536, corresponding to a link density $\rho=0.89$, and thus the network can be considered as a suitable test case for the evolution of community structure while sweeping the threshold weight. In Fig. 4 the labels associated with each community describe their dominant product categories. Although the dendrograms formed by using $k=3$ and $k=4$ are not identical, several similar communities appear for both values. From the commonsensical point of view, these appear natural: electronic devices and computer components merge to a single community, as do music and movies, and children's and women's clothing.

Often it is not possible nor meaningful to include all k-communities in such a visualization: the outcome would be too complicated to be interpreted by visual inspection. The main problem are the numerous single k-cliques, which merge to larger k-communities. For any analysis of the dendrogram structure the entire data should be used but for visualization purposes it is useful to threshold the dendrogram such that only k-communities which are larger than a threshold size N_{th} appear in the plot. In Fig. 4 k-communities of sizes larger than k are displayed, i.e., $N_{\mathrm{th}}=k$.

B. Weighted k-clique percolation

As pointed out above, considering the weights in the definition of what constitutes a community is an alternative to simply discarding low-weight links. Such an extension for clique percolation has recently been introduced by Farkas *et al.* in Ref. [20]. In this method, each k-clique is assigned a "weight," which equals the intensity [30] of its edge weights. The intensity is defined as the geometric mean of the link weights in the k-clique. The community structure is then obtained by choosing an intensity threshold I^* and taking into account only those k-cliques whose intensity is above I^*.

For our SCP algorithm, a simple modification allows for weighted clique percolation according to the above scheme. To achieve this, instead of building the k-communities simultaneously as the k-cliques emerge, all links are first inserted to the network and the resulting k-cliques are stored. Then, the intensity of each of these k-cliques is calculated, and the cliques are sorted with respect to the intensity. Finally, the sorted k-cliques are processed one by one by the second part of the algorithm, until the intensity threshold is reached. Multiple thresholding levels are obtained as before, but now with respect to k-clique intensities, and a dendrogram can be constructed similarly. Note that in addition to intensity, any

other measure describing the "weight" of the cliques can be used, e.g., if homogeneous cliques are sought for, one could also take the clique coherence [30] into account. Sorting cliques according to their intensities was briefly described by Farkas *et al.* in Ref. [20]; their construction appears somewhat similar to ours as the intensity-sorted cliques are handled in succession, and the method for obtaining overlapping k-communities seems to correspond to building the whole bipartite network between k- and $(k-1)$-cliques.

The above procedure requires keeping all k-cliques in the memory in addition to the $(k-1)$-cliques. In most cases the loss of speed is minimal, as the additional computational load is related to the memory consumption and sorting of cliques, which can be done in log-linear time. However, a possible problem related to the SCP algorithm—and the weighted clique percolation method, in general—is that all k-cliques have to be processed individually, and their number can be very large in dense networks as discussed in Sec. II C. When the link weight thresholding procedure of Sec. III A is applied, this problem can be somewhat circumvented by simply stopping the algorithm as soon as enough links have been inserted for obtaining the community structure at the desired "resolution." However, for intensity-based clique percolation this cannot be done, as all k-cliques have to be detected and sorted first.

IV. CONCLUSIONS

We have introduced a sequential clique percolation algorithm for detecting k-clique communities in a network by sequentially inserting its edges and keeping track of the emerging community structure [31]. This algorithm has specifically been designed for (dense) weighted networks, where weight-based thresholding of either the links or the cliques formed by them is necessary for obtaining meaningful information on the structure. By applying the algorithm on test networks, we have shown that the computational time required to process a network scales linearly with the number of k-cliques in the network. The sequential nature of the algorithm allows run-time construction of a dendrogram presentation of the nested hierarchical k-community structure, which we have illustrated using a product category network.

The main tradeoff for our algorithm is that it detects the k-communities for a chosen value of k with multiple weight thresholds in a single run, instead of obtaining k-communities for all values of k with a single weight threshold as is done in the maximal clique algorithms. Hence the SCP algorithm can be considered complementary to earlier presented solutions [9]. Neither of these algorithms can be argued to be strictly better or faster than the other as their performance depends heavily on the network topology and other aspects of the problem they are solving. The SCP algorithm is particularly useful when a small clique size k is used and when multiple weight threshold levels need to be studied, or no prior knowledge of the proper threshold level of a dense weighted network is at hand. The algorithm can also be considered as a reasonable choice for very large sparse networks as suggested by the short computation times of the community structure of a mobile telephony network having millions of nodes and links.

ACKNOWLEDGMENTS

We thank J. Hyvönen and J. Kertész for useful discussions, and acknowledge programming assistance by J. Hyvönen. We acknowledge support by the Academy of Finland, the Finnish Center of Excellence program 2006-2011, Project No. 213470. J.M.K. is partly supported by the GETA graduate school. J.S. and M.K. acknowledge support by the European Commission NEST Pathfinder initiative on Complexity through project EDEN (Contract No. 043251).

[1] G. Caldarelli, *Scale-Free Networks: Complex Webs in Nature and Technology* (Oxford University Press, New York, 2007).
[2] M. E. J. Newman, A. L. Barabási, and D. J. Watts, *The Structure and Dynamics of Networks* (Princeton University Press, Princeton, 2006).
[3] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).
[4] D. Lusseau and M. E. J. Newman, Proc. R. Soc. London, Ser. B **271**, 477 (2004).
[5] A. Arenas, L. Danon, A. Díaz-Guilera, P. M. Gleiser, and R. Guimerá, Eur. Phys. J. B **38**, 373 (2004).
[6] G. Palla, A.-L. Barabási, and T. Vicsek, Nature (London) **446**, 664 (2007).
[7] P. Holme, M. Huss, and H. Jeong, Bioinformatics **19**, 532 (2003).
[8] R. Guimerá and L. A. N. Amaral, Nature (London) **433**, 895 (2005).
[9] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005).
[10] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[11] M. E. J. Newman, Eur. Phys. J. B **38**, 321 (2004).
[12] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Natl. Acad. Sci. U.S.A. **101**, 2658 (2004).
[13] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. U.S.A. **104**, 7327 (2007).
[14] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, e-print arXiv:0803.0476.
[15] A. Lancichinetti, S. Fortunato, and J. Kertész, e-print arXiv:0802.1218.
[16] S. Fortunato and C. Castellano, e-print arXiv:0712.2716.
[17] P. F. Jonsson, T. Cavanna, D. Zicha, and P. A. Bates, BMC Bioinf. **7**, 2 (2006).
[18] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, Bioinformatics **22**, 1021 (2006).
[19] I. Derényi, G. Palla, and T. Vicsek, Phys. Rev. Lett. **94**, 160202 (2005).
[20] I. Farkas, D. Ábel, G. Palla, and T. Vicsek, New J. Phys. **9**, 180 (2007).

[21] In a unipartite projection, the bipartite network is collapsed such that only nodes of one type are left, each pair connected by a link if they are both connected to the same node(s) of the other type in the original bipartite network.
[22] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms* (McGraw-Hill, New York, 1990).
[23] J. M. Kumpula, J. P. Onnela, J. Saramäki, K. Kaski, and J. Kertész, Phys. Rev. Lett. **99**, 228701 (2007).
[24] M. E. J. Newman, Phys. Rev. E **64**, 016131 (2001).
[25] J. P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A. L. Barabási, Proc. Natl. Acad. Sci. U.S.A. **104**, 7332 (2007).
[26] T. Heimo, J. M. Kumpula, K. Kaski, and J. Saramäki, e-print arXiv:0804.3457.
[27] A. Clauset, C. Moore, and M. Newman, Lect. Notes Comput. Sci. **4503**, 1 (2007).
[28] M. Sales-Pardo, R. Guimerá, A. A. Moreira, and L. A. N. Amaral, Proc. Natl. Acad. Sci. U.S.A. **104**, 15224 (2007).
[29] A. Clauset, C. Moore, and M. E. J. Newman, Nature (London) **453**, 98 (2008).
[30] J. P. Onnela, J. Saramäki, J. Kertész, and K. Kaski, Phys. Rev. E **71**, 065103 (2005).
[31] A Python implementation of the algorithm can be found online at http://www.lce.hut.fi/~mtkivela/kclique.html

Fluctuation and Noise Letters
Vol. 0, No. 0 (2001) 000–000

Limited resolution and multiresolution methods in complex network community detection

Jussi M. Kumpula[1,*], Jari Saramäki[1], Kimmo Kaski[1], and János Kertész[1,2]
[1] *Laboratory of Computational Engineering, Helsinki University of Technology, P.O. Box 9203, FIN-02015 HUT, Finland;*
[2] *Department of Theoretical Physics, Budapest University of Technology and Economics, Budapest, Hungary*
**e-mail: jkumpula@lce.hut.fi*

Received (received date)
Revised (revised date)
Accepted (accepted date)

Detecting community structure in real-world networks is a challenging problem. Recently, it has been shown that the resolution of methods based on optimizing a modularity measure or a corresponding energy function is limited; communities with sizes below some threshold remain unresolved. One possibility to go around this problem is to vary the threshold by using a tuning parameter, and investigate the community structure at variable resolutions. Here, we analyze the resolution limit and multiresolution behavior for two different methods: a q-state Potts method proposed by Reichardt and Bornholdt, and a recent multiresolution method by Arenas, Fernández, and Gómez. These methods are studied analytically, and applied to three test networks using simulated annealing.

Keywords: Complex networks, Community detection, Limited resolution

Networks consisting of nodes and links are an efficient way to represent and study a large variety of technological, biological and social complex systems [1, 2]. Usually the functionality of these systems is of central interest, which, on turn, is closely related to the structure of the corresponding networks. In particular, substructures called *modules* or *communities* are abundant in networks. These communities are, loosely speaking, groups of nodes that are densely interconnected but only sparsely connected with the rest of the network [3, 4, 5, 6] – consider, *e.g.*, groups of individuals interacting with each other in social networks, or functional modules in metabolic networks. As communities are supposed to play a special role in the often stochastic dynamics of the systems under consideration, their identification is crucial. Thus, reliable and computationally tractable methods for detecting them in empirical networks are required.

Several methods and algorithms have been developed for community detection [7, 8]. One popular class of methods is based on optimizing a global quality

function called modularity [9], or a closely related Hamiltonian [10], which contains the modularity as a special case. The related methods are computationally demanding, especially for large networks, but various approximative algorithms exist [11, 12, 9, 13, 14]. For many test networks, these methods have been shown to perform well [7, 15]. However, it has recently been shown that the resolution of the modularity based methods is intrinsically limited, that is, modularity optimization fails to find small communities in large networks – instead, small groups of connected nodes turn out merged as larger communities [16]. For the Hamiltonian-based method, there is also a resolution limit due to similar underlying reasons [17] though this method contains a tuning parameter which can be used to study communities of different sizes. Recently, Arenas *et al.* proposed a modification of the modularity optimization method which also provides a parameter that can be used to probe the community structure at different resolutions. Here, we compare these two methods and their resolutions analytically, pointing out similarities and differences. Subsequently we apply them to several test networks using optimization by simulated annealing.

We start by briefly reviewing the concept of modularity, introduced by Newman and Girvan [9]. The modularity Q is defined as follows

$$Q = \frac{1}{L}\sum_{s=1}^{m}(l_{ss} - [l_{ss}]), \tag{1}$$

where L is the number of links in the network, l_{ss} is the number of links in community s, $[l_{ss}] \equiv K_s^2/4L$ is the *expected* number of links inside community s, given that the network is random, and K_s is the sum of the degrees of nodes in community s. In modularity optimization, the goal is to assign all nodes into communities such that Q is maximized.

The Hamiltonian-based method introduced by Reichardt and Bornholdt (RB) is based on considering the community indices of nodes as spins in a q-state Potts model, such that if the energy of such as system is minimized, groups of nodes with dense internal connections should end up having parallel spins [10]. The Hamiltonian for the system is defined as follows:

$$\mathcal{H} = -\sum_{s=1}^{m}\left(l_{ss} - \gamma[l_{ss}]_{p_{ij}}\right), \tag{2}$$

where $[l_{ss}]_{p_{ij}}$ is the expected number of links in community s, given the null model p_{ij}, and γ is a tunable parameter. Minimizing $\mathcal{H}$ defines the community structure. When $\gamma = 1$, Eq. (2) becomes Eq. (1) apart from a constant factor. Hence the RB method contains the modularity optimization as a special case, and can be viewed in a more general framework. Changing γ allows to explore the community structure at different resolutions, but communities with large differences in size cannot be simultaneously detected using a single value of γ [17].

Recently Arenas, Fernández and Gómez (AFG) proposed a method [18] for augmenting modularity optimization with a parameter, which similarly to γ above allows tuning the resolution of the method. This approach considers the network to be weighted. The trick introduced by Arenas *et al.* [18] is to add a self-link of

weight r to each node, in which case the modularity becomes

$$Q_w(r) = \frac{1}{W(r)} \sum_{s=1}^{m} \left(w_{ss}(r) - [w_{ss}(r)] \right), \tag{3}$$

where $W(r)$ is total link weight in the network (including self-links), $w_{ss}(r)$ is total link weight inside community s and $[w_{ss}(r)]$ is its expected value. Parameter r adjusts the total weight in the network, which in turn changes the community detection resolution [18]. Sweeping r and observing which communities are most stable with respect to changes in r should reveal the community structure.

Eqs. (2) and (3) suggest that RB and AFG methods are somewhat related, although not equal. The tuning parameters, γ and r, behave qualitatively in the same way: large parameter values allow finding small communities, and small values yield large communities. In fact, in the RB method, the effect of γ in Eq.(2) can be interpreted such that the "effective" number of links in the network equals L/γ, whereas the parameter r in Eq. (3) changes the total weight in the network. However, there is a difference: r also increases the sum of weights within a community, whereas γ has no effect on the number of links within a community. In order to illustrate the connection between these methods, we next derive the "resolution limit" intrinsic for Eq. (3) in the AFG method.

Now suppose that a network consists of "physical" communities, which are somehow known to us. We consider two of these communities, s and t, such that the sum of weights of edges connecting them is w_{st}. If these "physical" communities are merged by the detection method, the modularity $Q_w(r)$ changes by $\Delta Q_w(r) = \frac{1}{W(r)} \left(w_{st} - [w_{st}(r)] \right)$. The optimization of modularity should merge these communities if $\Delta Q_w(r) > 0$, which yields

$$S_s(r) S_t(r) < 2W(r) w_{st}, \tag{4}$$

where $S_s(r)$ is the total node strength in community s. An analogous result for RB method is $\gamma K_s K_t < 2L l_{st}$, where K_s is total node degree in community s. Hence the tuning parameters γ and r are not identical, and they affect the optimization outcome differently. However, if we assume that $S_s = S_t \approx n_s \langle s \rangle$, $n_s = n_t$ and $K_s \approx n_s \langle k \rangle$ Eq. (4) reduces to

$$n_s < \sqrt{\frac{N w_{st}}{\langle s \rangle + r}}, \tag{5}$$

which bears resemblance to the corresponding RB result: $n_s < \sqrt{N l_{st} / (\gamma \langle k \rangle)}$.

Next, we present some numerical results obtained by sweeping the tuning parameters γ and r of the RB and AFG methods across a range of values, and optimizing the respective energy functions using simulated annealing. Three different test networks are used. We show the behavior of the number of communities detected by the methods as a function of the tuning parameter, and look for "stable" regions where this number remains constant [18]. Earlier, community structures detected using several values of γ in the RB method have been reported in [10], but to our knowledge complete sweeps and stability analysis have not been reported earlier. We have used simulated annealing for optimizing the community structure.

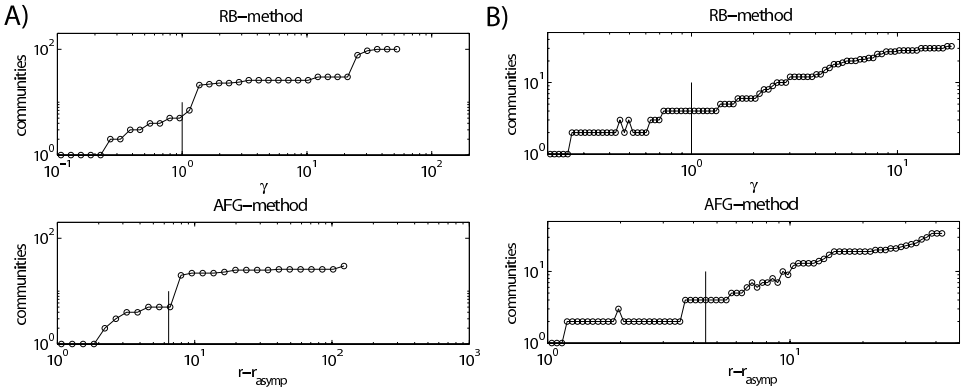
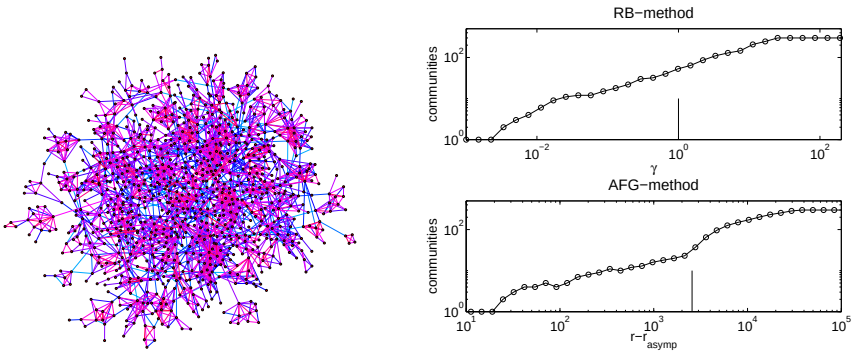


Figure 1. Number of communities as detected with simulated annealing using the RB (upper) and AFG (lower) methods. A: hierarchical scale-free network [19] of 125 nodes, B: Zachary's karate club. The vertical line denotes the traditional modularity optimization case.

Our first test network is a synthetic, hierarchical scale-free network of $N = 125$ nodes [19]. This unweighted network can be viewed to consist of 5 communities of 25 nodes each, which can be further divided into five-node cliques (for a visualization of this network, see [19] or [18]). Figure 1(A) shows the number of communities detected using the RB and AFG methods. Both methods are able to reveal the large communities at small values of sweeping parameter, although the AFG method seems to perform slightly better. One should note that this might be a feature of the numerical optimization, and not the method itself. We remind the reader that the "traditional" modularity optimization corresponds to $\gamma = 1$ and $r = 0$. These points are shown in the figures as vertical lines. Our results for the AFG method are consistent with those reported in [18].

Our second test network is a small, unweighted network representing Zachary's karate club [20], which has often been used as a "testbed" for community detection. Modularity optimization is known to yield four communities, whereas this club was observed to split into two communities. In [18], the authors demonstrated that AFG method is able to find exactly those communities (by using the weighted version of this network). Results for the unweighted network in Fig. 1(B) show that both methods give similar results and are able to detect the two communities. A closer inspection shows that the communities correspond to the split which eventually happened (except for one individual classified differently by the RB method).

Our third test network is weighted, being larger than the previous examples (986 nodes), and has a more complex community structure, Fig. 2(a). The average degree of this network is $\langle k \rangle = 6$ and it has been generated with a model designed to resemble real, weighted social networks. Visually, the communities are less apparent than in the previous test networks, although it can be seen that there are dense groups of nodes with strong internal links, connected by weaker links. Applying the clique percolation method [6, 21, 22] to this network using clique size 4 yields communities whose sizes vary from 4 nodes (20 communities) to 43 nodes (1 community). Because the network is weighted, we have used the a weighted Hamiltonian



(a) (b)

Figure 2. (Color online) A weighted test network having 986 nodes. Link colors vary from blue (weak) to red (strong), Number of communities for the network as a function of the tuning parameters. Note that we have limited the number of communities to 300.

instead of (2) for the RB method. Results in Fig. 2(b) show that no clear "stable" regions of the tuning parameters with a constant number of communities are apparent. One possible explanation is that this is due to quite non-uniform distribution of community sizes, which may result in large communities continuously being split into smaller ones as the tuning parameters are increased. A similar situation could occur for many large real-world networks. However, by using small values of γ and r it might be possible to study the large-scale community structure, such that the network is split into a small number of large communities.

We have discussed the limited resolution of community detection methods where a global energy-like quantity is optimized, by focusing especially on two methods (RB and AFG) where the resolution can be adjusted using a tuning parameter. Although the tuning parameters of these two methods give rise to qualitatively similar changes in resolution, analytic derivations show that their effect on the resolution limit is somewhat different. These two methods have also been numerically tested by using simulated annealing, with the result that in small test networks, stable regions of tuning parameter values, where the number of communities is constant, can easily be found. These can be viewed to reflect "optimal" communities. However, on a large, weighted test network, where the clique percolation method indicates a broader distribution of community sizes, no such regions are apparent.

Acknowledgments: This work was partially supported by the Academy of Finland (Center of Excellence program 2006-2011). JS acknowledges support by the European Commission NEST Pathfinder initiative on Complexity through project EDEN (Contract 043251). JK is partly supported by OTKA K60456.

References

[1] R. Albert and A. L Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47, 2002.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U Hwang. Complex networks: Structure and dynamics. *Phys. Rep.*, 424(4-5):175–308, 2006.

[3] R. Guimera, S. Mossa, A. Turtschi, and L. A. N. Amaral. From the cover: The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *PNAS*, 102(22):7794–7799, 2005.

[4] A. Arenas, L. Danon, A. Diazaz-Guilera, P. M. Gleiser, and R. Guimera. Community analysis in social networks. *Eur. Phys. J. B*, 38, 2004.

[5] R. Guimera and L. A. N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005.

[6] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[7] L. Danon, A. Diaaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 9, 2005.

[8] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

[9] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E.*, 69(2):026113, 2004.

[10] Jorg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E*, 74(1):016110, 2006.

[11] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, 2006.

[12] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):66111, 2004.

[13] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Phys. Rev. E*, 72(2):027104, 2005.

[14] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69(6):066133, June 2004 2004.

[15] M. Gustafsson, M. Hornquist, and A. Lombardi. Comparison and validation of community structures in complex networks. *Physica A*, 367:559–576, 7/15 2006.

[16] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *PNAS*, 104(1):36–41, January 2 2007.

[17] J. M. Kumpula, J. Saramäki, K. Kaski, and J. Kertesz. Limited resolution in complex network community detection with potts model approach. *Eur. Phys. J. B*, 56,41–45, 2007.

[18] A. Arenas, A. Fernandez, and S. Gomez. Multiple resolution of the modular structure of complex networks. *arXiv:physics/0703218v1*, 2007.

[19] E. Ravasz and A. L. Barabasi. Hierarchical organization in complex networks. *Phys. Rev. E*, 67(2):26112, 2003.

[20] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.

[21] I. Derényi, G. Palla, and T. Vicsek. Clique percolation in random networks. *Phys. Rev. Lett.*, 94(16):160202, 2005.

[22] Gergely Palla, Albert-Laszlo Barabasi, and Tamas Vicsek. Quantifying social group evolution. *Nature*, 446(713):664–667, 2007.

HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information and Natural Sciences

Mikko Kivelä

A network perspective on the genetic population structure of seagrass Posidonia oceanica

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology in the Degree Programme for Engineering Physics.

Espoo, 12.01.2009

Supervisor : Professor Kimmo Kaski
Instructor : Dr.Tech. Jari Saramäki

Informaatio- ja luonnontieteiden tiedekunta

Tekijä:	*Mikko Kivelä*
Koulutusohjelma:	*Teknillisen fysiikan koulutusohjelma*
Pääaine:	*Matematiikka*
Sivuaine:	*Laskennallinen tekniikka*
Työn nimi:	*Verkostonäkökulma Posidonia oceanican geneettiseen populaatiorakenteeseen*
Title in English:	*A network perspective on the genetic population structure of seagrass Posidonia oceanica*
Professuurin koodi ja nimi:	*S-114 Laskennallinen tekniikka*
Työn valvoja:	*Professori Kimmo Kaski*
Työn ohjaaja:	*TkT Jari Saramäki*

Tiivistelmä: *Populaatiobiologiassa käytettyjen perinteisten mallipohjaisten menetelmien tiedetään suoriutuvan huonosti, jos käytettävä data ei toteuta niissä tehtyjä oletuksia. Tässä työssä tutkitaan mahdollisuutta käyttää verkostotieteen uusimpia yhteisönhakumenetelmiä populaatiorakenteen löytämiseen geneettisestä samankaltaisuusverkosta, joka on muodostettu 1468 meriheinä Posidonia oceanican yksilön geenisekvenssien perusteella. Käytetyistä mikrosatelliittisekvensseistä voidaan rakentaa geneettinen samankaltaisuusverkko usealla eri tavalla. Työssä kokeiltiin useita tällaisia tapoja, joista sopivin valittiin käyttöön.*

Useimpia verkostotieteen menetelmiä ei ole suunniteltu sovellettaviksi täysiin painotetuihin verkoihin, joita tässä työssä käytetyt samankaltaisuusverkot ovat. Tästä johtuen osa verkostomenetelmistä jouduttiin muokkaamaan samankaltaisuusverkoille sopiviksi sekä laskennalliselta toteutukseltaan että yleiseltä toiminnaltaan. Työssä tutkitaan yhteisönhakumenetelmien löytämien geneettisten ryhmien rakennetta sekä vertaillaan näiden suhdetta maantieteeseen ja samoille yksilöille rakennettuun fylogeniapuuhun. Käytetyistä verkostomenetelmistä löydetään useita puutteita ja rajoituksia, mutta näihin ongelmiin ehdotetaan ratkaisuja ja viitoitetaan samalla tietä mahdolliselle lisätutkimukselle.

Sivumäärä: *87* Avainsanat: *Kompleksiset systeemit, kompleksiset verkot, hierarkkinen yhteisöhaku, painotetut verkot, geneettiset verkot*

Täytetään tiedekunnassa

Hyväksytty: Kirjasto:

Faculty of Information and Natural Sciences

Author:	*Mikko Kivelä*
Degree programme:	*Degree Programme for Engineering Physics*
Major subject:	*Mathematics*
Minor subject:	*Computational Engineering*
Title:	*A network perspective on the genetic population structure of seagrass Posidonia oceanica*
Title in Finnish:	*Verkostonäkökulma Posidonia oceanican geneettiseen populaatiorakenteeseen*
Chair:	*S-114 Computational engineering*
Supervisor:	*Professor Kimmo Kaski*
Instructor:	*Dr. Tech. Jari Saramäki*

Abstract: *In this Thesis, the objective is to study the possibility of applying state-of-the-art community detection methods of network science to genetic networks built of closely related individuals, as the traditional model based methods in population genetics are known make too restrictive assumptions and to perform poorly for data which does not fullfil these assumptions. A data set of 1468 sequenced specimens of the Mediterranean seagrass Posidonia oceanica is used to test the suggested methods. As there is no unique, all-purpose measure for the genetic distance between individuals, several such measures are investigated and the most appropriate is selected for constructing the genetic distance network used in this Thesis.*

Most methods in network theory are not designed for weighted, full networks used in this Thesis, and thus some computational limitations are encountered, which are solved by using different algorithmic approaches. Results of different hierarchical community detection methods are examined, analyzed with respect to the underlying geography, and finally compared to results obtained from phylogeny methods, which also allow hierarchical clustering. Several limitations are identified in the network methods used, and possible solutions and future research directions are suggested.

Number of pages: *87*

Keywords: *complex systems, complex networks, hierarchical community detection, weighted networks, genetic networks*

Faculty fills

Approved: Library code:

Preface

This work was carried out in the Department of Biomedical Engineering and Computational Science at the Helsinki University of Technology as a part of the EDEN (Ecological Diversity and Evolutionary Networks) project supported by the New and Emerging Science and Technology programme (NEST) of the 6th Framework Programme of the European Commission.

I am grateful to my supervisor Professor Kimmo Kaski for providing a great environment for this work. I wish to express my gratitude to my instructor Dr. Tech. Jari Saramäki for his good advice and sharing his insight. I would also like to thank Jari on behalf of the readers of this Thesis, as he has given numerous suggestions how to make this Thesis easier to read. In addition, I would like to thank Jenni Hulkkonen for collaborating with me on this project, and the whole Complex Networks research group for useful discussions and collaboration.

Finally, I want to thank my family, friends and girlfriend Essi for support.

Otaniemi, January 12$^{\text{th}}$, 2009

Mikko Kivelä

Contents

List of Abbreviations

LM Linear Manhattan, see Eq. (2.1)

NSA Non-Shared Alleles, see Eq. (2.2)

MST Minimum/Maximum Spanning Tree

GDS Genetic Diversity Spectrum

ROC Receiver Operating Characteristic

BIC Bayesian Information Criterion, see Eq. (3.4)

NMI Normalized Mutual Information, see Eq. (3.11)

UPGMA Unweighted Pair Group Method with Arithmetic mean

Chapter 1

Introduction

The beginning of modern genetics and the science of inheritance can be traced back to Mendel and his famous cross-breeding experiments in the mid-nineteenth century. He found out that inheritance is controlled by discrete units, which are nowadays called genes [1]. This idea was later combined with the Darwinian theory of evolution into population genetics and the modern synthesis theory. Evolution could now be explained with small changes in genome which lead to genetic diversity in distinct populations, and speciation in separated populations [2]. The discovery of the physical representation of genes as sequences of nucleotides in DNA molecules and the continuous advances in sequencing those genes has since made it possible to directly observe the genes even for a large number of individuals.

Understanding of the importance of genetic variation in combination with modern techniques for measuring and quantifying such variation can nowadays be used to direct the conservation of endangered species. One such species is the Mediterranean seagrass, Posidonia oceanica. It is an important part of the local ecosystem [3]; however, its growth is very slow and thus it is difficult to conserve. A better understanding of Posidonias genetic population structure and the genetic flows shaping it might allow focusing conservation attempts such that the genetic variation is properly preserved.

The problem in the case of Posidonia [3], but also more generally [4], is that the models used for inferring population structure or historical evolutionary events giving rise to the structure are too restrictive. Traditional methods can be mostly divided into two categories: Population genetics studies a large number of genet-

ically similar individuals by using summary statistics of allele distributions in those populations. Phylogenetic trees [5, 6] are mostly built for studying evolutionary relationships of a smaller number of sampled organisms, which are usually of different species. Both of these approaches are well established, but work only when strict requirements for the data are fulfilled. Loosening these requirements would not only allow researchers to combine the two levels of genetic structure of the sequenced individuals, the population-genetic view and phylogenetic trees, but also to study the regions between these levels. However, models taking into account all possible scenarios would have to be extremely complex. In addition, such models should be tailor-made for each species, taking into account their special features for example in reproduction patterns.

The biological system of evolving populations is a typical example of a *complex system*. Complex systems contain a large number of interacting components, which can be simple when isolated from the system, but as a whole exhibit complex emergent behavior. The abstraction of complex systems to networks has proven itself as a successful approach in fields ranging from sociology [7] and linguistics [8] to stock market [9, 10] and epidemiology [11]. Network methods have been useful tools [13–15] for example extracting hierarchical structure, modeling evolving systems and investigating collective behavior, all of which are typical features of living systems. Food webs [16] and protein interaction networks [17] are only some examples of biological systems which have been studied, and networks have a potential for serving as a general framework for the study of other complex biological phenomena, which cannot be described with simple models. In addition, network science has already developed tools that resemble those of phylogenetics and population genetics, such as methods for hierarchical community detection.

In this Thesis, the possibility of using network-based methods for analyzing phylogenetic relationships between individuals is explored. Networks built from genetic distances between specimens of Posidonia oceanica collected from multiple locations in the Mediterranean sea are utilized as a test case. Recent results based on the same data set, obtained by using both traditional and network methods, are also reviewed. Those results are compared to ones produced with methods developed in this Thesis. The main focus in this Thesis is on extracting large and small scale structure from the genetic network of individuals by using hierarchical community detection. This Thesis presents the first results of community detection studies on genetic distance data; to the best of the author's knowledge,

no results of similar studies have been published earlier.

The data set consists of the lengths of microsatellite repetitions in seven loci of the genome of each individual specimen. As the Posidonia oceanica populations evolve, these lengths are altered by two overlapping mechanisms: mutations and sexual reproduction. Due to this, any distance measure defined between two individuals is bound to lose some information and is a compromise between the two mechanisms, making the choice of the distance measure ambiguous. By choosing a distance measure, biological assumptions are made about the data, which will reflect to any network studies made later. Two plausible distance measures are compared in detail.

After selecting the distance measure, the data is ready for network abstraction, and the according methodology can in theory be straightforwardly applied. In reality, however, there are some algorithmic and practical complications caused by the fact that most existing methods are developed for sparse unweighted networks, and we are here dealing with a dense weighted network. Because of this, the aim of this Thesis is to solve some of these initial problems, and try out different methods on the data. Some of the tried methods appear to produce meaningful results, whereas others fail.

This Thesis is organized as follows: Chapter 2 begins with a short introduction to the species Posidonia oceanica, and describes the data set acquired from collected samples. Special emphasis is given to genetic distance methods, as they are the basis of the network analysis in Chapter 3. Before this, traditional methods for studying genetic relationships of data are briefly introduced and results from applying such methods are reviewed.

Chapter 3 deals with the network methods used for studying the genetic population structure of Posidonia. It begins by introducing the basic concepts and ideas of network methods and continues by reviewing previous network studies of the same data. After this, the problem of community detection is discussed in detail, and two community detection methods suitable for analysis of the genetic networks of Posidonia are then introduced. Results given by these methods are then compared to each other, geospatial information on the sampling sites, and to a corresponding phylogenetic tree. Chapter 4 presents conclusions on the results and comments on the usefulness of network methods as tools for studying population genetic data. It also suggest solutions to some of the encountered problems and paths for future research.

This Thesis has three Appendices. The first defines some basic concepts and quantities. The second introduces a sequential clique percolation algorithm developed by the author and his coworkers. The algorithm is an important part of the Thesis, as it is required for carrying out the community detection analysis in a reasonable time. The last Appendix introduces a software toolbox for network analysis, which was designed and implemented during making of this Thesis, and was used for all the computations, excluding the block diagonalization approach.

Chapter 2

Biological background

2.1 The data set and basic statistics

2.1.1 Posidonia oceanica

Posidonia oceanica is an endangered seagrass living only in the Mediterranean area. It forms large meadows in coastal areas at depths from 5 to 50 meters, depending on clarity of water and nutrient availability. Posidonia is a long-living organism, known to live over 1000 years, and it grows horizontally by 1 to 6 centimeters each year. The very slow growth makes it vulnerable to outside influences. The main reasons for Posidonia's endangerment are polluted waters, especially due to nutrients released into water, and fishing-related local damages. Posidonia is an important part of the Mediterranean ecosystem and its meadows work as carbon dioxide sinks. It is thus an important target for conservation efforts.

Posidonia oceanica is an angiosperm mainly reproducing asexually by cloning and self-pollination. Sexual reproduction is known to be sporadic and even unsuccessful in the Western Mediterranean basin [26]. The asexual reproduction combined with low rate of success in pollination can lead to large populations with little genetic variability. It is thus especially important to focus conservation efforts of Posidonia on preserving its genetic variability. Finding out which populations are the most important ones with respect to genetic variability can be problematic. It is hard to assess the importance of meadows to the overall genetic diversity

based only on geographical observations, because local environmental forces and ocean currents heavily affect the gene flow.

A better knowledge of the genetic population structure and diversity of Posidonia could be used to identify the important geographical regions for gene flow. This information could then guide the conservation attempts on Posidonia. However, it is difficult to study Posidonia with traditional phylogeny and population inference methods, as they are usually based on models which assume too simple reproduction dynamics. Posidonia is thus a good candidate for utilizing network based methods, where assumptions on the genetic structure are not as limiting.

Figure 2.1: A photo of Posidonia oceanica taken in Portofine, Italy [21].

2.1.2 Sampling locations

The data set studied in this work consists of 1468 samples of seagrass Posidonia oceanica, which were collected by diving from 37 different locations in the Mediterranean sea. The sampling locations were not chosen uniformly, but with large differences in density at different parts of the sea: The Western Mediterranean sea was more densely sampled than the Eastern, and the West also contains areas with large differences in sampling density. This heterogeneity of sampling locations allows the data to be used to study spatial aspects of the genetic structure

on many scales ranging from few hundreds of meters to thousands of kilometers. The heterogeneity can also cause problems for some analysis methods that assume homogeneous sampling. Such assumptions are often implicit, and can in this case lead to overestimating the importance of some of the western sampling locations. Throughout this Thesis, the sampling locations are divided to three groups: western, central and eastern, to allow a rough assessment of results from different methods with respect to sampling locations. For more details on this division, see Figure 2.2. The article by Rozenfeld *et al.* [3] which contains more detailed information on the sampling locations.

Although the sampling location density varies a lot, the sampling scheme inside each of those locations is similar. Approximately 40 shoots were collected from randomly drawn coordinates from a sampling area 20 meters in width and 80 meters long [3]. From each shoot, the meristem portion was collected for desiccation and preservation in silica crystal [3]. After the collection of the 1468 samples, part of the genome of each sample was sequenced for further studies on the population-genetic structure. A genome wide sequencing would be far too expensive, and thus the sequencing was limited to a number of microsatellite markers. These markers and the sequencing procedure are discussed in the following subsections.

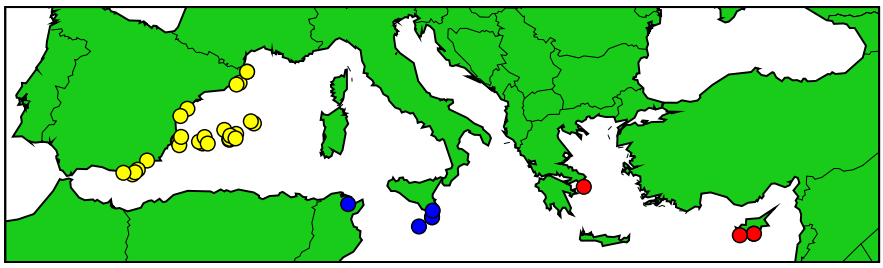


Figure 2.2: The sampling locations of the meadows of Posidonia oceanica are marked with circles. The locations are divided to three geographical groups: west (yellow), central (blue) and east (red).

2.1.3 Microsatellites

Microsatellites are a special class of hypervariable sequences of non-coding DNA, which are widely used for comparing the extent of genetic differences in two organisms [22]. The hypervarialibility of microsatellites, *i.e.* their high mutation rate, makes them ideal for comparing closely related organisms, such as two

samples of the same population or the same species. This property combined with the fact that microsatellites are mostly not under any selection pressure has made them increasingly popular for example in crime investigation, disease studies and structural population analysis.

Structurally, microsatellites are small motifs 1 to 6 nucleotides long, repeated up to 60 times. The structure of the microsatellite sequence makes it prone to special copying errors, which causes the mutation rate to be considerably larger than for example in coding genes. Normally, an error in the DNA copying process would cause a mismatch between the copied DNA strand and the template strand and trigger a repair process, but offsets in the number of repeats are not as easily noticed. This DNA slippage is the main mechanism behind microsatellite mutations. It mostly causes errors that delete or insert one repetition. The process does not seem to depend on the number of repeats, when the number is limited to a certain range usually from few dozens to few hundreds of repeats. Beyond this range, however, there seem to be some mechanisms limiting the length of the sequence [22]. It is fairly straightforward to model the mutation process, as it can roughly be described as a random walk.

2.1.4 Sequencing

The first stage of the sequencing process was isolating the genomic DNA by following a standard CTAB extraction procedure [3, 23]. It would be too expensive to sequence all known microsatellites from all the samples. Because of this, samples from eight locations were fully genotyped for eight dinucleotides, four trinucleotides and one 7-nucleotide, and 7 microsatellite markers where then chosen by using the conditions discussed by Arnaud-Haond *et al.* [24] to achieve the most efficient combination of the markers for separating clones from genetically different specimens [3, 25]. Errors in sequencing typically generate very small dissimilarities among clonal ramets, and all specimen with a distinct genotype for only two or one alleles were re-genotyped for those loci [3]. As Posidonia is a diploid organism, the final genetic data set consists of pairs numbers of repetitions in each seven loci for each of the 1468 samples, which thus constitutes a $(2 \times 7 \times 1468)$ matrix. 834 of the 1468 samples were unique with respect to the chosen microsatellite markers, the rest being clones or not distinguished as genetically different by the resolution given by the 14 markers in 7 loci.

2.2 Choosing the distance measure

In the context of studying the genetic structure of a large, geographically widely-spread population, the individual genomes of the samples are not of much interest. Instead, the focus is on the genetic relationships between these samples. As large-scale genetic relationships can be highly complex, the simplest way to approach the problem seems to be to concentrate on pairwise genetic relationships. The genetic relationship of two samples is naturally described by their similarity. This approach thus leads to defining a pairwise distance between all samples, in the hope that the more complex properties of the whole data set can be inferred from these distances.

The genetic distance approach can be used, for example, to find genetically distinct populations in the data, as individuals belonging to the same population should be genetically close. Likewise, a gene flow between two genetically distinct populations would result in short cross-population genetic distances. Distance-based methods are also the starting point of many model-based phylogenetic [28] and population structure inference methods [18], which have become more popular than purely distance-based methods [5,6,18].

The genetic data of each sample consist of microsatellite repetition numbers of the two alleles in each of the seven loci. Transformation of this microsatellite marker data into pairwise distances turns out to be a non-trivial task. This transformation is discussed in detail below.

2.2.1 Defining the distances

The microsatellite data for each sample consist of unordered pairs of allele lengths for each locus. Genets can be distinguished in the data, but there is no unambiguous measure of the genetic distance between different genotypes, although the mechanisms for their evolution are fairly well known. This is because two such mechanisms have an effect on the distance: mutations and genetic mixing. Both mechanisms implicate a way to define the distance measure. These distance measures are calculated here for the Posidonia oceanica data set and their properties are studied.

Mutations in the microsatellites usually alter the length of the allele by deleting

or inserting one repeat, whose length in our case is two nucleotides. The overall process of genome evolution by mutations only can be described by a random walk in single-allele length. The corresponding distance measure between two genomes can thus either be the minimum number of single-repeat mutations required to transform one genome to another, or the expected time it would take for one genome to transform to another. The first can also be viewed as the maximum parsimony measure, and it is not as sensitive to the definition of the underlying process as the latter one is. The expected-time measure would, for example, have to take into account the mechanism restricting the number of the microsatellite repetitions. The parsimony distance measure has been previously used in network-based studies of Posidonia oceanica [3, 29], and it was thus chosen for closer inspection. Rozenfeld *et al.* named the parsimony distance *linear Manhattan distance* (LM), and defined it as follows:

$$d_i(A, B) = \sum_{i=1}^{k} (|A_i - B_i| + |a_i - b_i|), \tag{2.1}$$

where A_i and B_i denote the lengths of the longer of the two alleles at locus i for samples A and B, and a_i and b_i denote the shorter lengths, respectively. The summation runs over sampled loci.

In the sexual reproduction process only genetic recombination takes place, and it is not affected by the number of repetitions in the alleles. Hence a distance measure which takes the allele lengths into account would possibly produce misleading results when applied to a system with a higher rate of sexual reproduction than the mutation rate. The *non-shared alleles distance* (NSA) counts the number of non-shared alleles at each locus of the two individuals, and as such it effectively discards all information on the differences of allele lengths. Thus the NSA distance is a suitable measure when sexual reproduction is the dominant mechanism of genetic variation. The NSA distance is defined as follows:

$$d_i(A, B) = \sum_{i=1}^{k} \sum_{x \in \{A_i, a_i, B_i, b_i\}} (1 - |\{x\} \cap \{A_i, a_i\} \cap \{B_i, b_i\}|), \tag{2.2}$$

where A_i, B_i, a_i and b_i are defined as in Equation (2.1), and the first summation again runs over sampled loci. Instead of counting non-shared alleles, a binary measure for genetic mixing could be defined as a parsimony measure, as the linear Manhattan measure was defined. The *allele parsimony distance* (AP) counts the minimum number of allele replacements between two genomes for one to

transform to another:

$$d_i(A,B) = \sum_{i=1}^{k} \min(|\{A_i, B_i\}| + |\{a_i, b_i\}|, |\{A_i, b_i\}| + |\{a_i, B_i\}|). \tag{2.3}$$

The NSA and AP distances are similar to each other up to a constant multiplier, and the only differences are the cases where one of the samples is homozygous and other is heterozygous in a locus. In the following, we will use the NSA distance measure.

2.2.2 Comparisons between the distance measures

The scatter plot of Figure 2.3, displaying values of the two distances for each pair of ramets, illustrates the relationship between the NSA and LM distances. Judging from this plot, the relationship between these two seems rather linear, but with reasonably high variance. A more quantitative measure of the relationship is the correlation coefficient which takes a value of approximately 0.71, in agreement with the above conclusion.

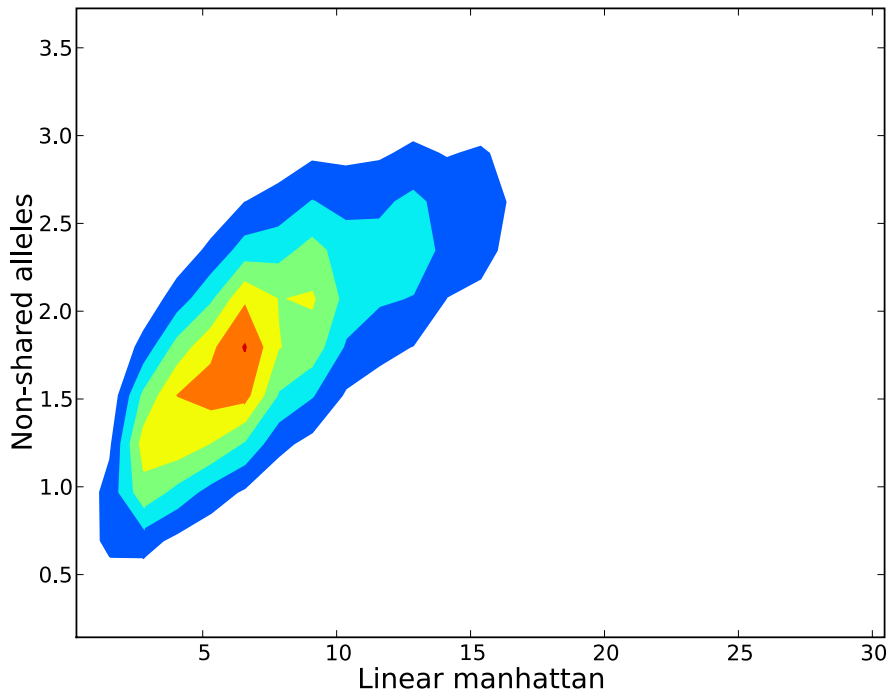


Figure 2.3: Scatter plot of the linear Manhattan distance (2.1) and the non-shared alleles distance (2.2) for every pair of ramets in the Posidonia oceanica data. The value of the correlation coefficient is approximately 0.71.

The two distance measures assume different underlying evolutionary processes. Genetic population structures produced by these processes can be distinguished from each other, but the real process behind the genetic population structure of Posidonia is a priori known to be a complex combination of the two processes. Thus, instead of inferring the probabilities of the processes producing the data, we must resort to a more qualitative comparison of the genetic population structure, and look for genetic structure characteristic of the two evolutionary processes. We begin by looking at the allele length distributions of all seven loci (Figure 2.4). A high mutation rate would yield a high degree of polymorphism, which is the case only for two loci (1 and 3). Instead, the number of repeats in loci 2, 5, 6 and 7 are mostly confined to a small number of clearly distinct values. This is indicative of a slow rate of mutation, implying that the use of the LM distance in the analysis of this data set might not be well justified.

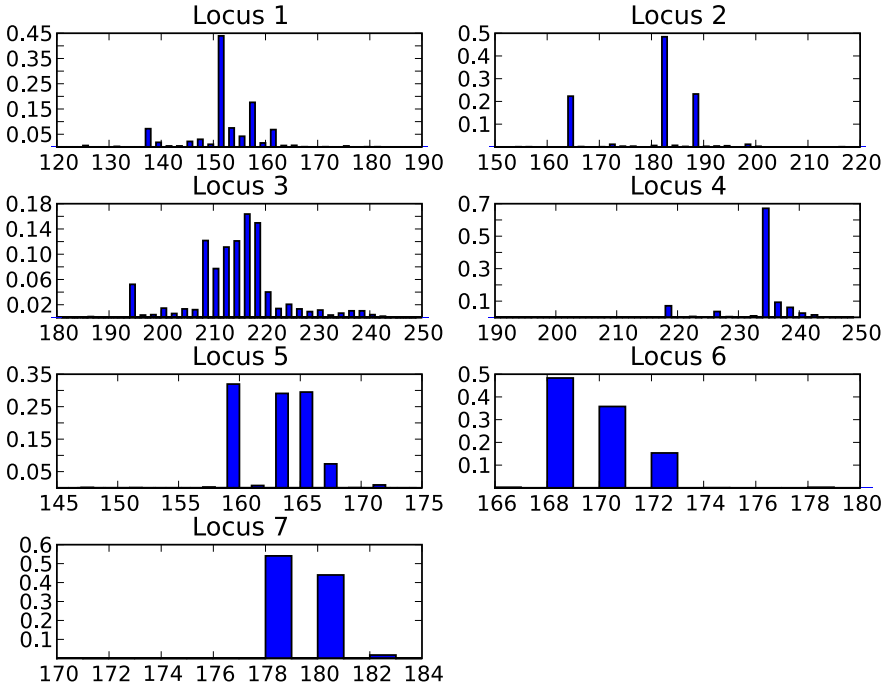


Figure 2.4: The allele frequencies in the 7 sequenced loci of the Posidonia. The horizontal axis represents the number of repetitions defining an allele, and the vertical axis the allele frequency. The distributions for loci 1 and 3 are fairly diffuse with no clear gaps, implying a high rate of mutations. On the contrary, in the other loci the repetition numbers are mostly limited to a small number of distinct values, indicating that sexual reproduction is the dominant mechanism. For locus 2, the difference between these values is large, which causes substantial differences between the NSA and LM measures.

Let us now discuss the effects of these findings on the values of the distance measures. Note that for locus 2, there are only 3 different common allele lengths,

which are spread far apart separated by wide gaps where almost no alleles can be found. Loci 6 and 7 are also confined to 3 distinct values, but with the difference that the values are not separated by gaps. The NSA distance measure discards all information on the allele lengths, and thus the overall contributions of loci 2, 6 and 7 on the distances should be roughly equal. This is not the case with the LM distance, for which the length of the gaps is important information. Figure 2.5 displays the mean contribution of each locus for both distance measures. It is apparent that loci 2, 6 and 7 contribute roughly equally to the NSA distance; however, for the LM distance, the differences are very high. This difference arises from the ambiguity of the distance measures.

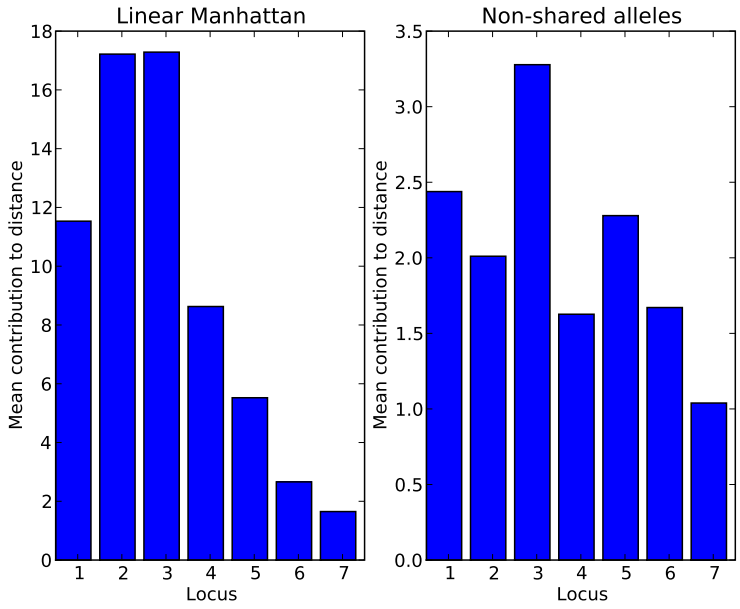


Figure 2.5: Mean contribution over all distances of each locus to the linear Manhattan distance and the non-shared alleles distance. There are substantial differences between the NSA and LM measures, especially for the loci 2, 6 and 7. The NSA measure seems to give similar weights to each loci, as opposed to the LM distance, where the differences can be large.

The problem at hand can be roughly divided into two limiting example cases. If two (or several) alleles with a large difference in the number of repeats coexist in a population of closely related individuals, applying the LM measure can produce erroneous results. As an example, two ramets heterozygous with respect to this allele could have two homozygous descendants. The LM measure would yield a high genetic distance between these, due to the large difference in allele lengths. Thus, NSA would appear as the proper distance measure for such cases. However,

if there are two distinct populations such that all shorter alleles are found within one and longer ones within the other, the LM measure clearly provides a more accurate view. One can thus interpret the LM distance as a measure of genetic differences over long, evolutionary time scales, and the NSA as a measure related to shorter time scales.

In reality, however, both long and short time scales are reflected in the genetic composition of populations. For the case of P. oceanica, this can be clearly seen in Figure 2.6, displaying the geospatial distribution of the ramets homozygous and heterozygous with respect to locus 2. 121 ramets with alleles $(164, 164)$ are located in the eastern and central areas of the Mediterranean sea, whereas 225 ramets with alleles $(182, 182)$ are located in the western and central areas. This causes the LM distance to differentiate well between the east and the west, but causes noise in the distance measure in the central areas. On the other hand, the 82 heterozygous ramets with alleles $(164, 182)$ located in the west and center can also cause large errors between the ramets from the west as discussed in the previous paragraph. This claim is studied more closely in the following Section.

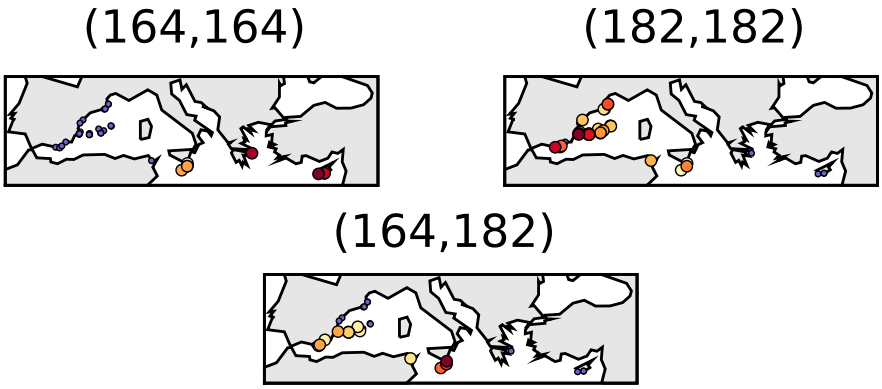


Figure 2.6: Geographical distribution of sampling locations where the two major alleles of locus 2 can be found. Distributions for the homozygous [(164,164), (182,182)] ramets are shown separately. Colors indicate frequency: red for locations with the highest frequency, yellow for lowest frequency, with the intensity of the color reflecting the frequency. Small blue circles denote locations where these alleles are not found. It is clearly seen that the allele 164 is associated with the east of Mediterranean and 182 with west; however there are also some heterozygous $(164, 182)$ samples near Spain.

Correlations with locations

The above results clearly indicate that there are correlations between genetic distances and the locations of the sampling sites of ramets. Furthermore, the analysis discussed in the previous Section suggested that the LM measure differentiates well ramets sampled from geographically distant locations, while the NSA seems to be appropriate measure for analyzing populations which are spatially and genetically close. Here, we test this hypothesis using ROC curves [30, 31], as introduced in this specific context by Klemm [32]. The ROC curves quantify the extent to which the genetic distances can be used to classify samples into clusters, using the sampling locations as a reference. The results of the ROC analysis for comparing different distances to the two geographical divisions can be seen in Figure 2.7, which seems to support the claim, as the LM is better for coarse classification than NSA, but NSA is better when all locations are taken into account.

Each point on the ROC curve corresponds to a threshold value θ. For each threshold value θ, the pairs of nodes are divided into two sets: those having a distance smaller than the threshold and those who have a larger distance. The pairs with the small distances are predicted to belong to same the class, and the ones with large distance are predicted to belong to different classes. Using the sampling sites as the true classes, two rates of success are calculated for each θ: the true negative rate, *i.e.*, the fraction of samples predicted to belong to different classes, which actually belong to different classes, and similarly, the true positive rate. These rates are then plotted against another. Hence a distance which would not correlate at all with the classes would yield a straight line.

To see how much the ROC curves for ML and NSA are affected by the different mean contributions of different loci to the two distance measures illustrated by the Figure 2.5, a renormalized distance measure was constructed. This measure is based on the LM distance, where the contribution of each locus is renormalized such that their means correspond to those of the NSA distance. Figure 2.7 shows that the normalization has a substantial effect, but does not fully explain the difference between the NSA and LM distances.

Locus 3 was seen to have a high degree of polymorphism in Figure 2.4 and thus it should have a high mutation rate. This implies that the LM measure could perform better or as well as NSA in the locus 3. To test this, a hybrid distance

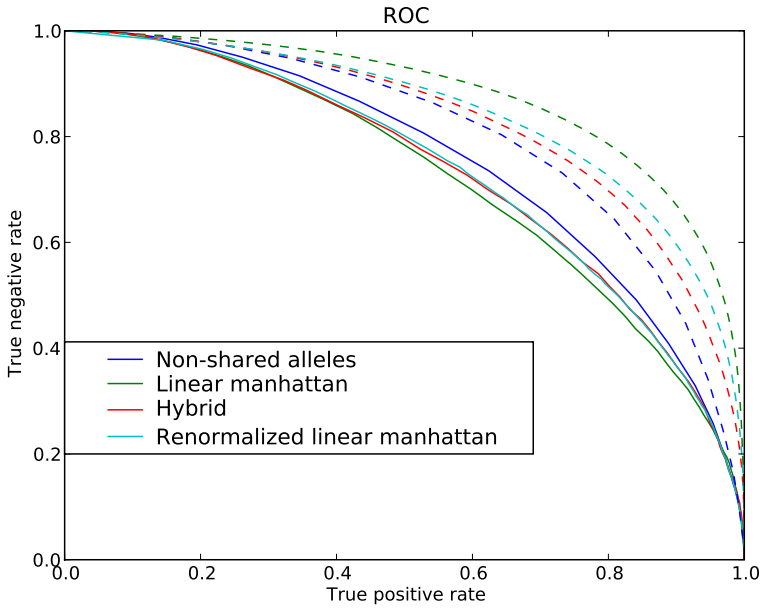


Figure 2.7: ROC curves for the prediction of the sampling locations with different distance measures. The solid curves indicates the classification made with all the 37 locations, and the dashed curves show the results for more coarser classification of the data to western, central and eastern regions.

was constructed such that in the NSA measure, the third term in the distance sum corresponding to locus 3 was replaced by the corresponding term of the LM distance. In addition, this replacement term was normalized such that its overall contribution was similar to the original NSA term. In Figure 2.7, this hybrid model is seen to have a negative effect on the prediction ability of the NSA distance for locations, and a positive effect when only the coarser division to east, center and west is considered.

2.2.3 Conclusions

The ambiguities of measures for microsatellite-based genetic distances between individuals are seen to cause problems when the measure needs to perform well on multiple evolutionary time scales. For the Posidonia oceanica data, this could indicate that for example studying only ramets located on the coast of Spain, where the sampling is most frequent, the NSA distance could be a reasonable starting point. On the other hand, the LM distance could be used to estimate long-range effects, however it could cause substantial noise in the distances when

improperly used.

An interesting solution to the problem would be a hybrid distance measure taking into account population-level information that cannot be otherwise incorporated to the distance measure between two single genomes. However, such a measure would be harder to interpret, and at least is not trivial to construct a measure that would perform better than both NSA and LM in all test cases. A substantial part of this problem is norming the contributions to the distance measures made by each locus, as the variance in the contributions between the two measures is large, which has a substantial effect to the overall performance of the distance when predicting locations.

2.3 Basic statistics

The above-described data set on Posidonia oceanica has been studied with commonly used biological summary statistics methods in Refs. [24,34,35]. Results of these studies can be used as a basis in assessing any results produced with the new methods discussed later. They also give a general idea of how the data is organized, what the limitations of these methods are, and what they are good for. In this Section, those results are briefly reviewed, and some basic statistics of the data are discussed. Before that, the general ideas behind these methods are briefly described.

2.3.1 Phylogenetics and population genetics

Summary statistics of population genetics and phylogenetic trees are both good candidates for studying large data sets of individuals, whose genome is represented by a small number of microsatellite markers. Phylogenetic trees can be used for clonal species or for individuals sampled from distinct populations. Summary statistics are better suited for closely related individuals, which are preferably sampled from same population. However, the use of these methods is often limited by the underlying assumptions. These limitations are discussed briefly in this subsection, to give an idea of what kind of data can be studied with traditional methods, without having to resort to the network methods introduced later.

A phylogenetic tree is a representation of lineages and history of evolutionary events separating them for a set of individual organisms [5, 6]. Phylogenetic trees are also commonly built between genes or species, but these cases are not considered here. A common way of presenting evolutionary relationships is a rooted tree, where the leafs of the tree depict the sampled organisms. The inner nodes represent the ancestors of the leaves such that a common parent of two nodes is the last common ancestor of those nodes. This means that the root of the tree is the most recent common ancestor of all the nodes. This hierarchical branching pattern is called the topology of the tree. Most methods for building phylogenetic trees define branch lengths in addition to the topology. The lengths can represent the period of time covered by the branch or the amount of genetic divergence.

Phylogenetic trees are traditionally used in systematics by comparing morphological differences of species. As the amount of molecular data has exploded in the last decades, phylogenetic analysis has entered the genomic age. This, combined with the development of statistical methods of phylogeny inference, has made it possible to analyze data sets of hundreds of species. Despite this success, phylogenetic trees have limitations that restrict their use, as an example, for the data set of Posidonia oceanica discussed in this Thesis. If the studied set of nodes or samples are from the same population or even from the same species, sexual reproduction can limit the usage of phylogenetic trees as the different lineages can now merge. Thus the tree structure cannot correctly represent the histories of all lineages, as mixing would cause the appearance of cycles in the tree. Such effects can also be caused by horizontal gene transfer, which is thought to play a role in bacterial evolution [33].

Traditional population genetic studies rely on simplified models of genetic evolution and reproduction mechanisms. The aim is to fit the observed data to the models, and calculate summary statistics based on the fitted models. A typical problem with this approach is the often unrealistic assumptions made by the models, such as non-overlapping generations, random mating and equilibrium state, which are in many cases known to be violated in the studied populations [3]. In some cases, the use of the models is limited even more by the choice of the studied organism. As an example, in the case of Posidonia oceanica, clonal reproduction severely limits the number of models that can be used. This means that many of the commonly used summary statistics, such as the effective population size and the generation time, contain assumptions that are not compatible with clonal

organisms such as Posidonia.

2.3.2 Results of the summary statistic studies

The Posidonia oceanica data set described here has been studied with population biology summary statistic methods in at least three articles. The main results of these articles are briefly presented here. The first article by Arnaud-Haond *et al.* [24] optimized the number of microsatellite markers needed to distinguish clones from genetically different genotypes, and found a combination of seven dinucleotide markers, which are also used in this Thesis as discussed earlier. Diaz-Almela *et al.* [34], on the other hand, used the seven markers to study the effect of four Mediterranean fish-farms on Posidonia oceanica. Arnaud-Haond *et al.* [35] again studied spatial correlations in the genetic data and found a strong west-east cleavage. In addition, they found a putative secondary contact zone at Siculo-Tunisian Strait, high genetic structure between meadows, and high spatial autocorrelation in some of the locations.

The strong genetic separation between west and east has also been observed earlier with other data sets [20, 27] of Posidonia. On the basis of this strong evidence, the large scale geographical correlation of the genomic distances is used as a first benchmark for the new methods introduced later. More specifically, the sampling locations and the samples are divided into western, central and eastern locations as discussed earlier, and the division is compared in various ways to any new results.

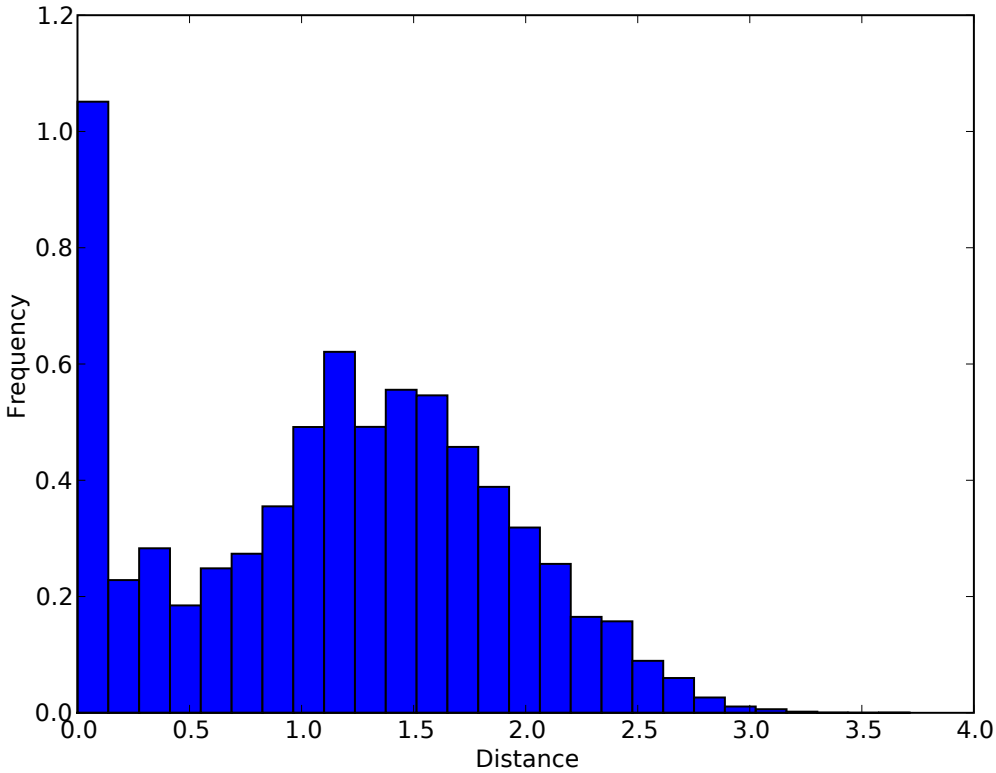
2.3.3 Distance based statistics and studies

Rozenfeld *et al.* [3] studied the linear Manhattan distance distributions by modeling the different reproduction processes and observing typical distances produced by them. Such genetic diversity spectra (GDS) averaged over all within-location distances are shown in Figure 2.8. The shapes of the spectra for the non-shared alleles distance in panel a) and the linear Manhattan distance in panel b) look fairly similar. The only difference seems to be related to the fact that the number of distinct distances is larger for the LM measure. Rozenfeld *et al.* [3] found out that most of the observed distances in the GDS in panel b) were typical for clonal reproduction and outcrossing, and deduced that these are the main mechanisms

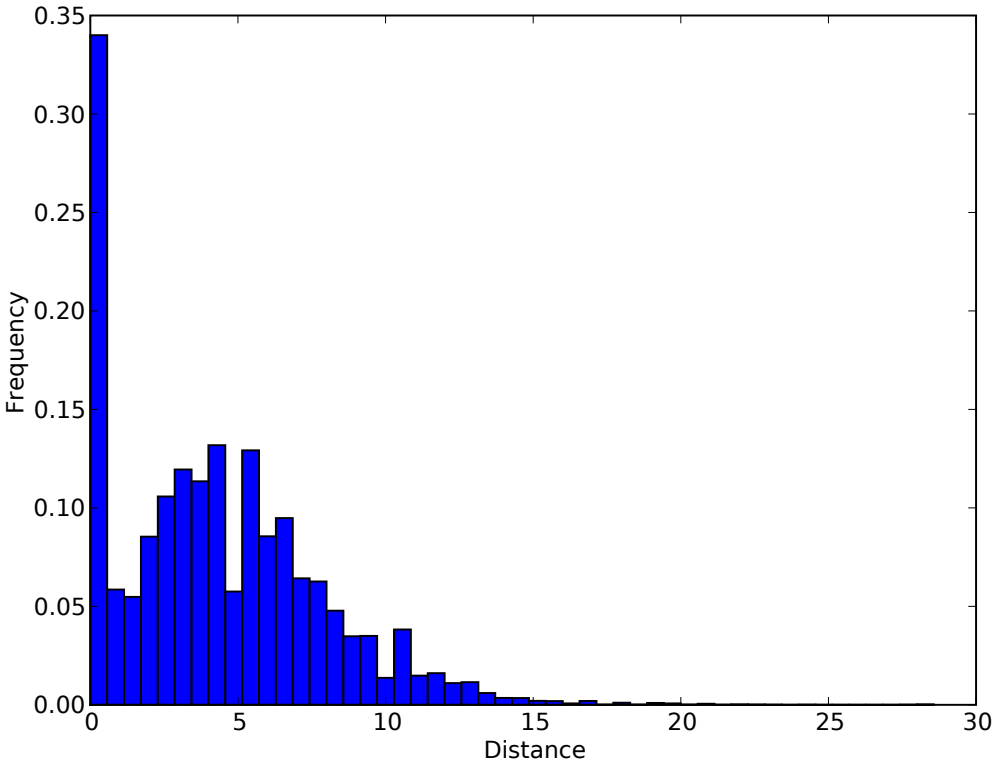
influencing the genetic structure of Posidonia oceanica. In Figure 2.9, the same GDS plots for NSA and LM distances are shown for the whole Mediterranean-wide data set. There, the clonal peaks at distance of zero are almost absent. However, interestingly, the GDS for LM distance shown in panel b) seems to be a combination of multiple normal distributions. This could be due to the strong west-east cleavage. The large peak would then correspond to the distances inside the geographical areas and the small peak to the distances between the west and the east. This effect cannot be seen in the GDS for NSA distance in panel a), which might indicate that the differences inside each of the three geographical areas are already so big for the NSA distance that the larger distances between the areas cannot be distinguished from them. This means that the resolution of the NSA distance for large geographical distances is not as good as the resolution of the LM distance on the same scale.

To verify the hypothesis about the large scale geographical effects on the global GDS, the genetic distances were plotted against the corresponding geographical distance in Figure 2.10. A clear correlation with the geographical and genetic distances can be observed for both genetic distance measures. Notice that the density plots of Figure 2.10 are essentially joint distributions of genetic distance and geographical distance, and the GDS distributions in Figure 2.9 are the marginal distributions of those joint distributions when the geographical distances are integrated out. Figure 2.10 thus confirms that the peak corresponding to the large genetical distances in GDS of LM distance is indeed due to node pairs with large geographical distance. Although similar correlations to geography can be observed for the NSA distance, distinct peaks are not visible in the GDS plots for NSA distance.

Thus the LM and NSA distance distributions appear to support the conclusion made on the basis of ROC curves of Figure 2.7, namely that the LM measure seems to better reflect large-scale variations, whereas the NSA distance performs better in analysis of local populations. This doesn't come as a surprise, because the recombination process measured by the NSA distance has physical constraints with respect to geographical distance, whereas the mutation process measured by the LM distance does not depend on geography.

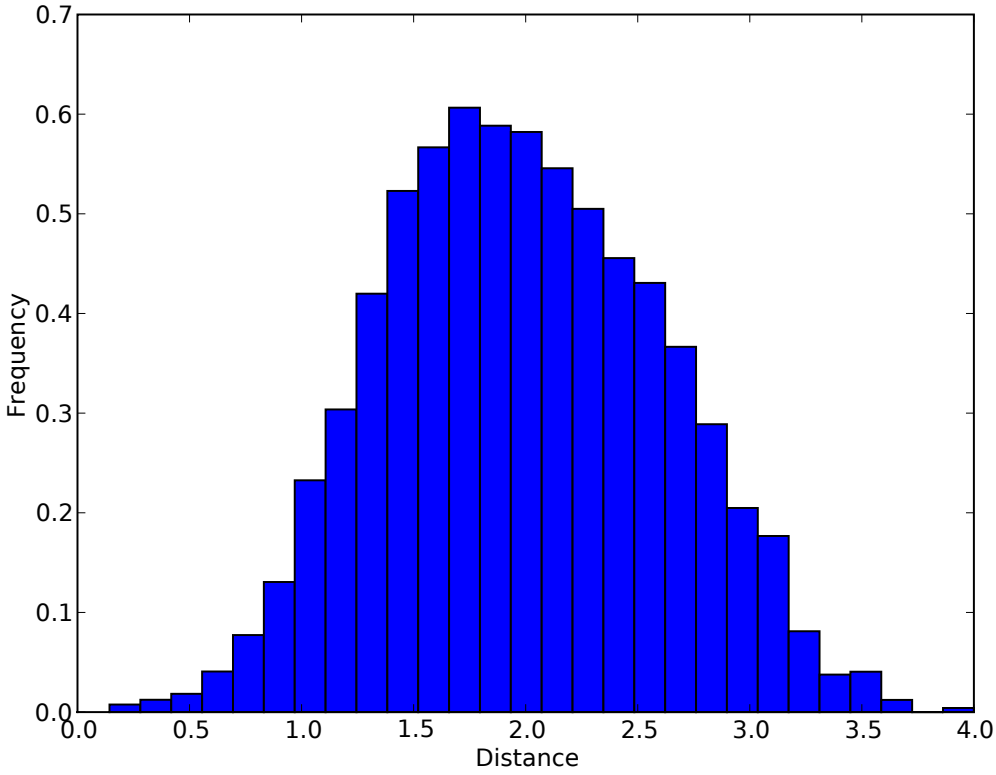


(a) NSA

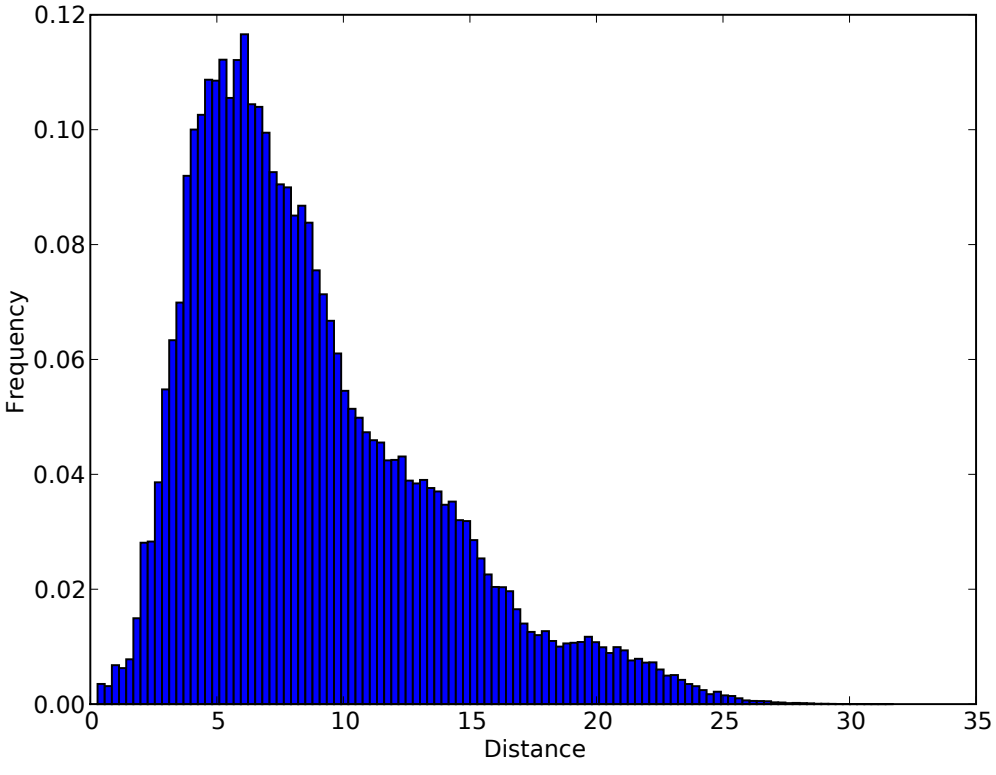


(b) LM

Figure 2.8: Genetic diversity spectrum (GDS) averaged over all the distances in each sampling location for a) the non-shared alleles distance (NSA) and b) the linear Manhattan distance.

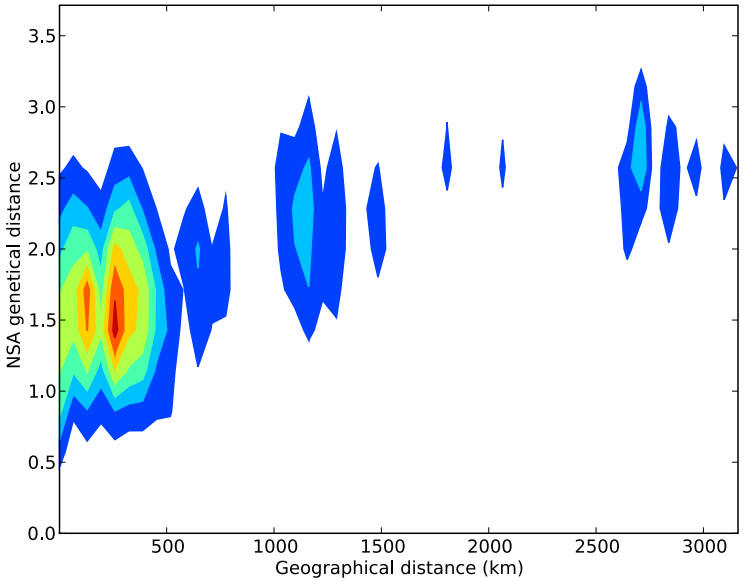


(a) NSA

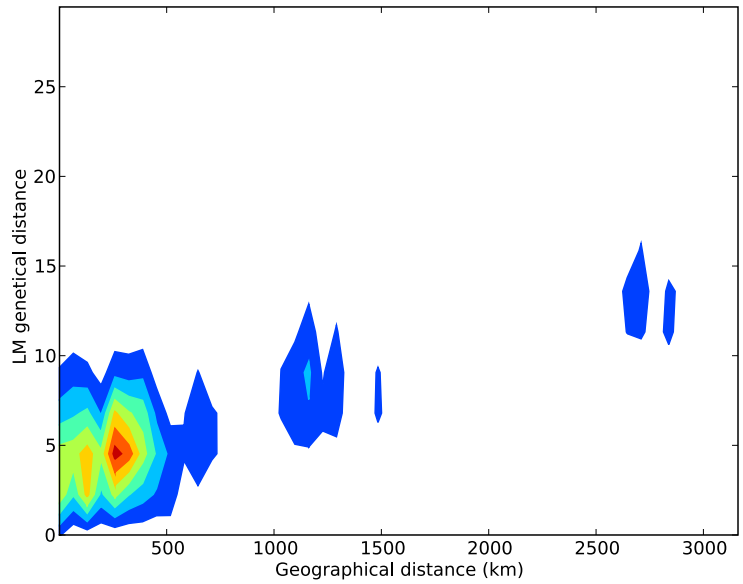


(b) LM

Figure 2.9: Genetic diversity spectrum (GDS) of the whole data for a) the non-shared alleles distance (NSA) and b) the linear Manhattan distance.



(a) NSA



(b) LM

Figure 2.10: Two genetical distances plotted against geographical distance: a) non-shared alleles (NSA) and b) linear Manhattan (LM). Some correlation between geographical and genetic distances is visible, when large and small geographical distances are compared, but the effect is more apparent for the LM distance.

Chapter 3

Network analysis of the data

In the previous Chapter, the Posidonia oceanica data set was seen to contain complexity that could not be fully captured by traditional model-based methods and summary statistics. The reason for this was seen to be the reproduction system of Posidonia consisting of three different mechanisms. On one hand, clonal reproduction violates the most basic assumption of non-overlapping generations made by classical population genetic methods [3], and on the other hand, Posidonia's sexual reproduction severely limits the usage of phylogeny methods. The data was also seen to capture multiple evolutionary scales as the distances between the samples vary from few meters to thousands of kilometers.

Network science has been successfully used to capture both small scale structure and large scale phenomena on complex systems ranging from social [7] to technological [19] systems. The main idea of the network approach is to study systems with a large number of interacting elements by representing them as graphs, where edges represent interactions between the elements. This abstraction step allows the use of generic network-based data analysis methods to be applied to a range of different systems.

In this Chapter, network-based methods are used to tackle the complexity inherent in the population structure of Posidonia oceanica. Earlier network-based studies of the Posidonia oceanica data set are first reviewed. These studies are mostly based on using methods such as minimum spanning trees and thresholding, and computing local node-based topological statistics from the networks resulting from the application of those methods. This line of work is continued in this Thesis by studying the large-scale structure by *hierarchical community*

detection methods. Communities are sets of nodes which are more densely connected to each other than to the rest of the network, and are usually related to the functional units of the system. Nested communities, or communities inside communities, form hierarchical community structures. Community detection is thus clustering of nodes in a graph.

Although it has become increasingly popular to take interaction strengths into account in the form of edge weights, most community detection methods still only use the topological properties of the network. This is a problem, as the genetic similarity network of Posidonia is a full weighted network, and discarding the weights would thus lead to a trivial topology. Here, one community detection method, k-clique percolation, is modified in a way that allows it to produce hierarchical community structure. The other community detection method used in this Thesis is block diagonalization, which is a general method for clustering distance-based data. Results of application of these methods are first compared by visual inspection and then by using the mutual information framework. Geospatial information on the sampling locations and phylogeny-tree-based clusters are also utilized in the comparisons.

3.1 Converting the genetic distance matrix to a network

To apply network methods to any dataset, an abstraction step is needed for interpreting the data at hand as a graph. In our case, the natural way of doing this is interpreting each specimen as a node and adding an edge between each node with a weight representing the genetic distance between them [3]. This is equivalent to using the distance matrix D between the specimens as a weight matrix W. Similar approaches have been used in past e.g. in interpreting stock price correlations as a network [9, 10].

As our data contains clones, the first preparatory step is to collapse each set of clones to one node by removing all but one instance of each clone. This leaves 834 unique nodes.

The usual interpretation of the weight of an edge between two nodes is that the larger the weight, the stronger the interaction between them. However, the

distances defined here behave in exactly the opposite way. It is also useful to normalize the weight matrix such that the maximum weight equals unity. These requirements do not define any unique way to transform distances to weights. Here, one of the simplest ways is chosen:

$$W_{ij} = 1 - \frac{d_{ij}}{\max d_{ij}}. \tag{3.1}$$

3.2 Earlier network studies of the data

In addition to traditional population biology studies discussed earlier, the P. oceanica data set has also been studied using network methods. All of the four articles published about these studies are related to the same EDEN project as is this Thesis, but the author of this Thesis has not participated in any of them. In this Section those articles are briefly reviewed and some of their ideas are adopted as a part of the community studies discussed later.

The first of the four papers, written by Hernández-García *et al.* [29], is a short introduction to the network perspective for tackling the complexity related to this kind of biological data in multiple resolutions. The second, written by Rozenfeld *et al.* [3], goes somewhat deeper into the individual level genetic networks and reproduction systems of the Posidonia oceanica. Contrary to this, in Ref. [36], Rozenfeld *et al.* try to infer large scale genetic flows from the network of genetic distances between populations defined by the sampling locations by using methods such as percolation and betweenness centrality. As this starting point is somewhat different from the one used in this Thesis, further discussion of this paper is omitted here. The last article discussed here, by Hernández-García *et al.* [37], observes and models the size distributions of clonal samples.

3.2.1 Minimum spanning trees

In Ref. [29], Hernández-García *et al.* introduce some ideas for analyzing the microsatellite data of the Posidonia oceanica with weighted complex networks, where the weights are defined with the linear Manhattan distance (Equation 2.1). They start by building a Minimum Spanning Tree (MST, see Appendix A) of the distance network of sampled ramets of Posidonia oceanica, and visual-

ize the resulting tree, where nodes are colored according to sampling locations. Minimum/maximum spanning trees have also been used earlier to study structural properties of e.g. stock price correlation networks [10]. MST visualization is a good and flexible tool for roughly representing the correlations between the genetic structure and the geographical locations of the nodes. MST-based visualizations are used in this Thesis for comparing the results of community detection methods, as well as their relationship to geography.

One should be somewhat cautious when making a priori claims about the meaning of the structure of the MST for any particular type of network. Hernández-García *et al.* [29] interpret the MST as the main path of gene flow among the plant populations, on the basis that the edges represent closest relations between the nodes. This seems to be a straight-forward interpretation, although it might be a little problematic to define the main path of gene flow to go through single organisms as their genomes are of course stationary. The real problems with the MST are more general and subtle, and are related to the fact that the MST is not necessarily unique if multiple links have same weights. Even if this is not the case, the MST is likely to discard many important links due to small differences in their weights, and is thus highly vulnerable to small perturbations. As an extreme example of the non-unique nature of the MST, for a case where all nodes are clones, any tree connecting those nodes is their MST. If only a single MST is given for this case, its topology thus depends solely on the algorithm that was used for constructing the tree or is completely random. Keeping this in mind, it would be advisable to use only one node to represent all of its clones in a MST. The choice of distance measure will also have an effect to the topology of the MST as even small changes in the distances can cause large deviations in the tree. This must be taken into account especially when using a distance measure which can produce large random errors in estimating short distances, because the minimum spanning tree is based on small distances. This was seen to be the case with the linear Manhattan distance of Eq. (2.1) in the previous Chapter where the distance measures were compared.

3.2.2 Thresholding

Another approach for studying full weight matrices with network-related methods is thresholding, where the network is constructed of those matrix elements whose weights are above some threshold value, w_{th}. This approach was adopted by

Rozenfeld *et al.* [3] and Hernández-García *et al.* [37]. Networks resulting from the thresholding procedure can be studied with tools of unweighted network analysis, such as analysis of degree distribution or clustering coefficient (see Appendix A). This approach is better than using the MST in the sense that it discards less information, but on the other hand the choice of the threshold w_{th} can be problematic. In the case of the network of samples of Posidonia Oceanica, the threshold can be chosen based on biological arguments or topological arguments. For example, the average genetic distance between parents in simulated data has been used as a threshold for meadow-wide networks [3]. The network topology itself can be used to determine the threshold by setting the threshold weight w_{th} to be equal to the critical point in percolation, which is roughly the minimum threshold weight giving rise to networks where almost all nodes belong to the largest connected component.

The problem with choosing a single threshold is that many of the network statistics depend only on the chosen value, as for example the average degree and the clustering coefficient go from zero for the maximum threshold to their largest values for zero threshold. One solution to this is to use a range of threshold values instead of a single one, and study the chosen measures as a function of the threshold. This approach is used in later this Thesis for the case of community detection using the k-clique percolation method. This method is strictly topological; however it can be applied to weighted networks by thresholding them first.

In the case of the Posidonia oceanica data set, choosing a threshold is problematic also because of the heterogeneity of the sampling locations, as the genetic distances between samples from the densely sampled Spanish coastal area are short compared to the most of the other distances. Hence, a threshold giving rise to a network where the eastern nodes form a sparse network with visible structure would contain a large, almost fully connected clique of western nodes. Furthermore, a threshold providing some resolution on the Spanish data would leave the eastern nodes disconnected from each other and from the western nodes. Thus the proper thresholds for analyzing western, eastern or the whole data are dramatically different.

This problem of choosing a global threshold is addressed in the article by Rozenfeld *et al.* [3] by simply looking the local networks formed of each sampling locations. Apart from visual inspections of the resulting genetic networks in sampling

locations, the main network-based result seemed to be the small world property [56] of the networks. This means that path lengths of the network remain at the level of Erdős-Rényi random networks [53], while the mean clustering coefficient is notably higher than in random networks. As genetic networks are based on distance measures, the triangle inequality implies a large clustering, and even a small number of random links ensures the small path lengths. As Rozenfeld *et al.* noted, the small world property is a very common feature of complex networks.

In Ref. [37], Hernández-García *et al.* use the thresholding approach and plot the resulting networks for a single sampling location for four values of threshold. In addition, they show the degree distribution averaged over the networks of each location at threshold levels zero and 30 of the linear Manhattan distance. Most of their paper after this is devoted to modeling and studying the clone size distributions, which can also be interpreted as thresholding with value zero. This is of course a trivial threshold level in the network sense as the resulting networks only contain disconnected cliques each corresponding to a set of clonal samples.

3.3 Community detection

3.3.1 Overview of the problem

Initially, research on complex networks focused on studies of distributions of node and edge based statistics, such as the clustering coefficient and betweenness centrality. Since then, the focus has been shifting to more mesoscopic quantities. One of the most fundamental large-scale problems is community detection. Community detection is an important problem as in many cases communities correspond to functional entities in networks, or are otherwise relevant in context of the underlying system. The problems in community detection are not as much related to difficulties with computation or algorithmic performance as they are to the exact mathematical definition of a community. A community is in most cases loosely defined as a set of nodes that are internally more densely connected than externally. The definition of a community is not always based on network topology, and sometimes communities are defined based on a specific underlying problem. For example, in social networks groups of friends can act as communities, or in genetic networks a group of genes performing some well defined function can be considered a community. In spite of this intuitive knowledge of what communities

or clusters in graphs should be, a generally acceptable all-purpose definition is still to be found.

An alternative view to the community detection problem is that a universal definition of a community cannot be found. This view is supported by the existence of dozens of different community detection methods built on clearly different and incompatible underlying assumptions. The lack of a unique formal definition of a community makes the problem of finding the best community detection method ill-posed, and the problem becomes choosing which community detection method is best suited for the task at hand. When selecting a community detection method, at least the following questions should be answered:

1. Should it be possible for a node to belong to more than one community?
2. Should it be possible for a node not to belong to any community, or to form its own single-node community?
3. Should the method assign nodes to communities of roughly the same size, or is a large variation of sizes expected in the data?
4. Should the method be hierarchical, or should it produce just a single division of nodes to communities? That is, should the method be able to detect communities nested inside larger communities?
5. In weighted networks, how are the weights taken into account when detecting communities?

Despite the ambiguities in community detection, several properties are clearly desirable for any method. For example, adding a new component to a network should not affect the community structure of the existing components inferred by any method. This implies that the community structure should be determined using only the local topology of the network. Lack of such locality has proven to be a common pitfall in community detection, as it is not always apparent from the description of a method whether it produces such unwanted global effects in the community structure. Community detection methods can be divided into *global methods*, having a network-level fitness function whose maximum yields the desired classification of nodes to communities, and *local methods*, which only take the local topology of the network into account. Both types of methods have their advantages and problems, which are discussed in more detail in the following subsections.

One local and one global community detection method were chosen for closer inspection in this Thesis. *The clique percolation method* is a local, topological community detection method, and *the block diagonalization method* is a global clustering method for distance matrices. Only a small fraction of all available community detection methods are discussed in this Thesis, and to get a broader view of the field, the reader in encouraged to read the recent review article on the matter [39].

3.3.2 Local methods: Percolation and k-clique percolation

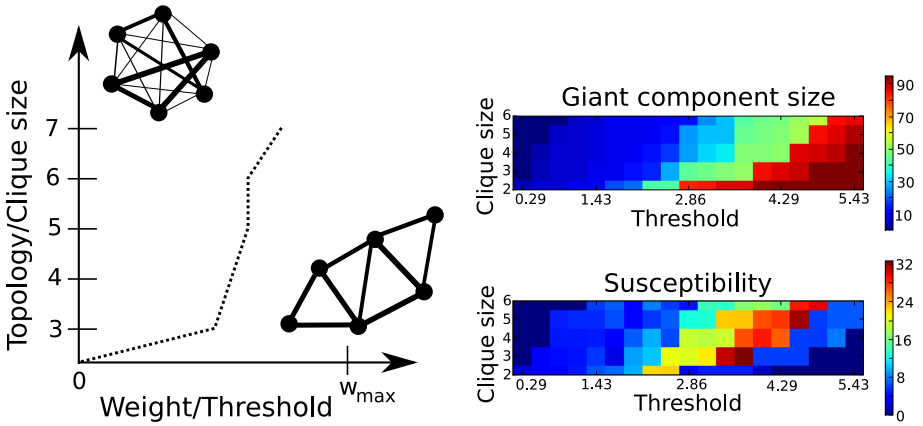
Discarding a fraction of the edges of a network based on some criterion and then interpreting the remaining components as communities is a straightforward and commonly used method for community detection for example in social networks [40]. Such edge percolation methods can be divided into two categories: in agglomerative methods, edges are added to an initially empty network, and in divisive methods, edges are removed from the original network. In both cases, the fraction of edges added or removed acts as a control parameter. A variety of criteria exists for the order of edge removal or addition. For a weighted network, the given edge weights can be used to order the edges. Also topological properties, such as edge betweenness, can be used to determine the order in which the edges are removed or added to the network. Topological properties can be calculated only for the original network, or dynamically after each addition or removal.

The goal is to remove just the right fraction of edges giving rise to a network whose disconnected components correspond to the community structure. If the fraction of removed edges is too high, (almost) all nodes belong to a single connected component (the *giant component*), whereas removing too many edges leads to a severely fragmented and ultimately to an empty network. Such processes have been extensively studied in percolation theory [41], and it has been noticed that in many systems the transition from the situation where almost all nodes belong to a single component to a situation where there are a large number of small components is very rapid. In this context, the number of edges needed to be removed to arrive at the point of transition between the two phases is called the critical point of the system. If such a point exists for a percolation process, it can serve as a good candidate for determining the proper fraction of edges to be removed.

The clique percolation method [42] can be considered a modification of the edge percolation method. For the edge percolation method, two nodes are assigned to the same community, if they are connected via any path along the edges. However, for clique percolation, there has to be a path of cliques. Formally, two k-cliques, *i.e.* cliques of k nodes, are defined to be adjacent if they share a common $k-1$-clique. The k-cliques are thus nodes of a new k-clique network, where links represent these adjacency relations. Then two nodes in the original network are in the same community if they participate in k-cliques which are in the same component of the k-clique network.

The k-clique percolation method has some desirable properties: The number of communities a node can belong to is not predetermined, but a node can belong to any number of communities, or even to no community, depending on the network topology. The method is based on local network topology only, and far-away nodes or edges do not have an effect on the local community structure. A community is explicitly defined, which makes the resulting community structure easy to interpret. In addition, the method is deterministic, which ensures that k-clique percolation algorithms always find the same communities in a network.

The less desirable properties of the clique percolation method include exponentially scaling computational cost as function of the network size, in the worst case when communities of all clique sizes k are sought for. Also, small perturbations in the network can cause large changes in the community structure. For instance, if there is a single k-clique between two large communities, removal of a single link in that clique will cause the two communities to split. The scaling is not a problem in most cases as it is usually enough to use cliques of size three to five [43], and extremely large highly connected subgraphs are not very common in networks. Scaling issues could be solved by choosing the right value for the clique size k, but the choice of the k-clique threshold imposes more problems. First, the clique size k must be integer-valued, which may lead to a situation where a suitable value cannot be found. Second, Palla *et al.* [42] suggested a heuristics for finding a global value of the clique size k. However if the network is highly heterogeneous, this might yield a compromise value only, whereas different values of k could be more suitable for different parts of the network. Choosing a single clique size k depending on the network also violates the property of communities being local in the sense discussed earlier.



(a) Schematic view (b) Data

Figure 3.1: Clique percolation in weighted networks is a compromise between topological coherence and weighted structure of the communities. A schematic view of this with respect to the optimal threshold is represented in *a)*. The topological coherence (vertical axis) is increased as the clique size increases. Increasing the threshold level (horizontal axis), *i.e.* the smallest accepted weight, increases the relative importance of weights. The richest community structure for each clique size can be found by using the optimal threshold, which is the critical point of the clique percolation process. In *b)*, the same shape of the optimal threshold curve is observed in a data of east coast ramets of Posidonia oceanica with linear Manhattan distance. The susceptibility in *b)* is defined as the mean size of the components excluding the largest one, and it is expected to peak and the giant component size to saturate near the optimal threshold.

3.3.3 K-clique percolation and the Posidonia oceanica data

When viewed as networks, the genetic distance matrices of samples of Posidonia oceanica represent weighted, full networks. It is clear that these networks need to be thresholded before the clique percolation method can be used. An alternative is to use the weighted version [44] of the method, but this approach is computationally extremely demanding for full networks. As the network is very likely to contain nested community structure, choosing a single threshold and disregarding of the rest of the weights seems problematic. Also the heterogeneous locations of the sampling sites result in networks with domains of different weight scales. That is, the nodes from western Mediterranean were sampled with much higher resolution than those from the eastern parts of the sea, and thus the weights between western nodes are of different scale than the weights between eastern nodes. These aspects render the use of a single threshold for the clique percolation method useless for our data set of Posidonia oceanica. To solve this problem, a hierarchical version of the clique percolation method using threshold

sweeps [43] was developed by the author and coworkers. In our method, the k-clique communities are computed for each weight threshold level of the network starting from the full network and removing edges one by one, starting from the smallest weights. The clique size k is selected beforehand and kept constant in this process. Before removing any edges, the whole network constitutes a single large community. When edges are removed, communities begin to split, and the splitting process can be interpreted as hierarchical community structure. Such hierarchical structures are presented in this Thesis as rooted trees, where the root is the community consisting of all the nodes in the network, and the leaves are the smallest communities in the hierarchy. As illustrated in Figure 3.1, the clique size k corresponds to the required structural integrity of communities. The weight threshold corresponds to the smallest weight which has an effect on the community structure. The combination of the clique size and the threshold thus determines the relative importance between topology and weights in community detection.

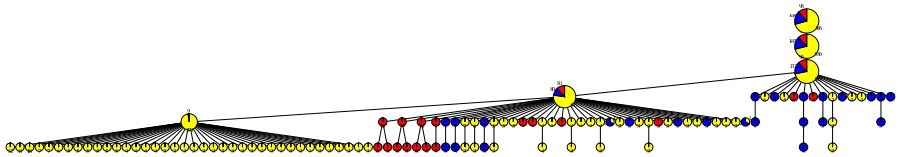
Although the algorithm suggested for clique percolation by Palla et al [42] could be, in theory, used for the hierarchical the clique percolation method, it is not a suitable algorithm for the task. Computation of the largest cliques of a graph, which is required when using the algorithm for clique percolation suggested by Palla *et al.* [42], can be a problem in dense networks, as it is known to be a NP-hard problem. In addition, community structure should be calculated separately for each threshold level, if this algorithm was to be used for the hierarchical clique percolation method. These complications were avoided here by developing a complementary algorithm for clique percolation which is able to produce the community structure at each threshold level in a single run. The trade-off is that only a single clique size can be used at a time. However, the use of single clique size instead of finding the largest ones lowers the theoretical scaling of the computational time when small cliques are used. This new algorithm is discussed in detail in Appendix B and in Ref. [43].

Results

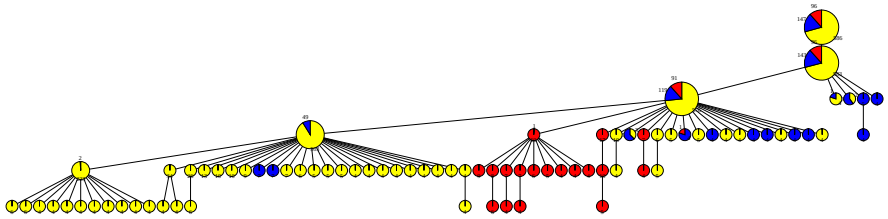
As the hierarchical clique percolation method employed in this Thesis is based on removing edges in order of weight, results are evidently sensitive to the distribution of edge weights. In particular, if the distribution is heavily discrete instead of continuous, such that for each weight there is a large number of edges,

the resolution of the method can be somewhat compromised. In such a case, it is sensible to use the threshold weight as the control parameter instead of the fraction of edges removed, such that the community structure is evaluated at points where all edges below the threshold have been removed. Thus, there may be drastic jumps in the community structure if there is a large number of edges with the same weight, especially near the critical point.

The lack of resolution is a problem for the genetic similarity network of Posidonia oceanica as the distances between samples are highly degenerate when the NSA distance measure is used. The resulting tree thus consists of only a few levels of division steps, which can be clearly seen for clique sizes $k = 3$ and $k = 4$ in Figure 3.2. Using a larger clique size will not help considerably, although this gives more weight to the topology of communities, as it will most likely only shift the region of interest instead of widening it. In addition, finding large cliques is computationally demanding for the genetic similarity network of P. oceanica. Despite the low resolution of the clique percolation method apparent in Figure 3.2, divisions made by the clique percolation method are reasonable when compared to geospatial information on the sampling sites. Clique percolation might be a suitable community detection method for genetic similarity networks for cases where high-definition weights not giving rise to resolution problems would be available, or if the resolution problem could be solved by modifying the method.



(a) 3-clique percolation



(b) 4-clique percolation

Figure 3.2: Splitting process of 3-clique and 4-clique communities when the weight threshold is varied for the genetic similarity network of Posidonia oceanica. The similarity of nodes is based on the NSA distance. The nodes are divided to three geographical groups: west (yellow), center (blue) and east (red) and the frequencies of the groups in each community are shown as a pie chart.

Using the LM distance instead of the NSA distance leads to genetic similarity networks with more distinct weights, and thus increases the number of hierarchy

levels in the hierarchical clique community tree. However, the LM measure might be prone to large random errors in small-scale genetical distances, although it can be used to predict large time scales very accurately. Thus, using the LM distance matrix as a basis of clique percolation might result in more random errors in the low levels of community hierarchy than using the NSA distance with higher resolution that the number of different weights would allow. The increased number of hierarchy levels is visible in the hierarchical community tree of Figure 3.3, which uses a genetic similarity network based on the LM distance. The rough division of the node locations to three classes correspond very well to the results of 3-clique percolation based on the LM distance matrix, which was expected as the LM distance is known to reflect the division better than the NSA distance (see Figure 2.7). It is worth noticing that the shape of the hierarchical community tree produced by clique percolation based on the LM distance matrix is very unbalanced. This is due to the heterogeneous density of the sampling locations in the Mediterranean sea.

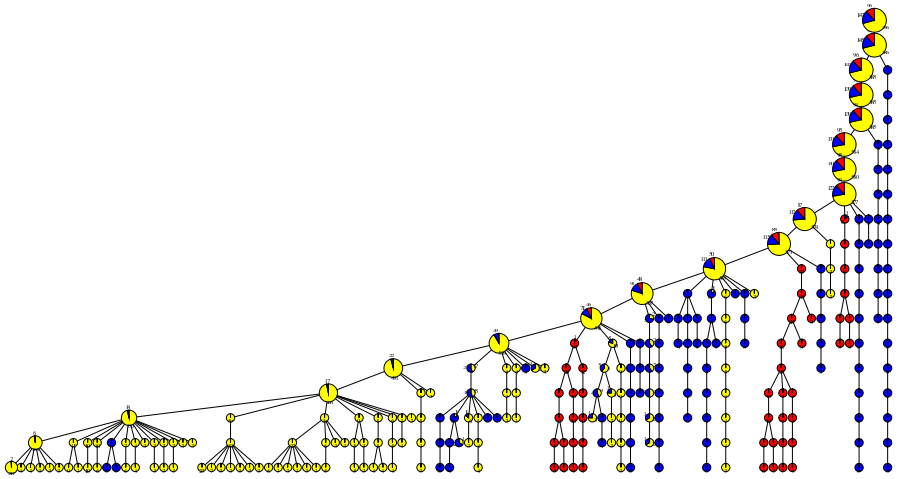


Figure 3.3: Splitting process of the 3-clique communities when the weight threshold is varied for the genetic similarity network of Posidonia oceanica. The similarity of the nodes is based on the LM distance. The nodes are divided to three geographical locations: west (yellow), center (blue) and east (red) and the frequencies of the groups in each community is shown as a pie chart.

3.3.4 Global methods: block diagonalization

Global community detection methods are based on optimizing a value of energy or fitness function computed for each division of nodes to communities. This approach can, in the worst case, lead to unwanted behavior, such as the community

structure correlating with global properties of the network, such as the number of nodes. This has been recently proven [45, 46] to be the case with popular modularity-based methods [40], thus rendering these methods highly unreliable or even completely useless.

After the shortcomings of modularity-based methods were proven, there have been attempts to correct these faults. Some of these attempts have turned out to be just cosmetic changes and have failed to correct the real underlying problem [46]. One of the remedies has been proposed by Sales-Pardo *et al.* [47]. They try to correct the resolution limit problem of the modularity by not limiting their method to one optimal community division. Instead, they try to find a hierarchical community structure based on the modularity measure. This is achieved by calculating multiple greedy modularity optimizations and saving the results in a weighted network, where each link weight correspond to the number of shared communities between two nodes in each of the optimization outcomes.

This affinity matrix is then used for hierarchical clustering by using the block diagonalization method. In this method, the indexing of the nodes in the matrix is reordered in such a way that large-weight elements are as close to the diagonal as possible. This is achieved by optimizing a global energy function assigned for each node indexing:

$$C = \frac{1}{N} \sum_{i,j=1}^{N} W_{ij}|i - j|. \tag{3.2}$$

The optimization is done in Ref. [47] by simulated annealing, but other heuristic methods could also be used. Simulated annealing is a general approach for global optimization problems which tries to mimic the physical process of controlled annealing for reducing the number of crystal defects. The simulated annealing procedure is a greedy optimization method combined with random steps. The ratio of random steps is controlled by the temperature parameter, which is gradually decreased to achieve convergence of the energy to an optimal value. At each step of the process, parameters of the energy function are randomly perturbed and the resulting change in the energy is observed. If the energy is decreased, the perturbation is kept. If the energy increases, the change is accepted with a probability which is determined by the amount of change in the energy and the temperature parameter. Increases in energy are more likely to be accepted at high temperatures than at low temperatures. The whole process starts from very high temperatures to avoid local minima, and the temperature is decreased as a

function of perturbation steps. This process can be repeated multiple times using different starting conditions and schemes for decreasing the temperature.

After the optimization of indices, possible communities should show in the reordered matrix as blocks along the diagonal, as is the case in Figure 3.4. The next step is detecting the blocks, which is done as follows: A similar simulated annealing procedure is used to fit k blocks to the matrix, such that the variance of weights is minimized inside each block and the surrounding area. This is equivalent to minimizing the following residual sum of the squares E for each value of k by moving the block boundaries:

$$E = \sum_{m=1}^{k} \sum_{i,j \in block(m)} (W(i,j) - \langle W_m \rangle)^2 + \sum_{i,j \notin block} (W(i,j) - \langle W_{outside} \rangle)^2, \qquad (3.3)$$

where $\langle W_m \rangle$ is the mean weight inside the block m, and $\langle W_{outside} \rangle$ is the mean of all weights that are not inside any block. This procedure is repeated for a range of values of k, $k = 1..K$.

Increasing the number of blocks k will not ever cause the residual sum of the squares E to increase, because only the number of free parameters in the optimization problem is increased. Finally, the Bayesian information criterion (BIC) [52] is used for selecting a value of k which corresponds to a good compromise between the value of the residual sum E and the number of free parameters. The BIC is defined as

$$BIC = N \ln(\frac{E}{N}) + k \ln(N), \qquad (3.4)$$

where the parameter N is the number of elements on the diagonal of the weight matrix W. The smaller the BIC value is, the better the compromise is, and thus the optimal value of k corresponds to the lowest BIC value.

The hierarchical community structure can be built by repeating the block diagonalization procedure for submatrices corresponding to each community. This means that if the optimal number of blocks k is larger than one, the block diagonalization procedure is used recursively k times. This recursion is further continued, until all blocks have $k = 1$ as their optimal value and splitting is no longer necessary. The results of this recursive splitting process can be represented as a hierarchical community tree, where the root of the tree is the set of all nodes in the network, and the rest of the nodes in the tree correspond to a block found by the block diagonalization procedure. Notice that branch lengths

are not defined for the hierarchy tree.

The block diagonalization method described by Sales-Pardo *et al.* in Ref. [47] uses modularity for unweighted networks to calculate the affinity matrix. Although modularity can be generalized for weighted networks [48], it might be more reasonable to use the weight matrix itself as the affinity matrix if the network is dense. For cases where the weight matrix is derived from a distance matrix it is even more sensible to directly use the distances as affinities. Thus, the block diagonalization procedure was used to detect communities in the Posidonia oceanica genetic similarity network, with the modification that the modularity optimization part was left out, and instead the distance matrix was used as an affinity matrix.

3.3.5 Block diagonalization and the Posidonia oceanica data

The block diagonalization procedure, as described above, was repeated for the Posidonia oceanica data set using both NSA and LM distances. The LM distance was previously seen to contain more noise in the small distances, which radically reflected to the hierarchical community structure detected by the block diagonalization method. The 834 nodes of the network were divided into 61 communities, when the LM distance was used, and only to 24 communities when using the NSA distance. This might suggest that the block diagonalization is fitting the communities to noise caused by the shortcomings of the LM distance measure.

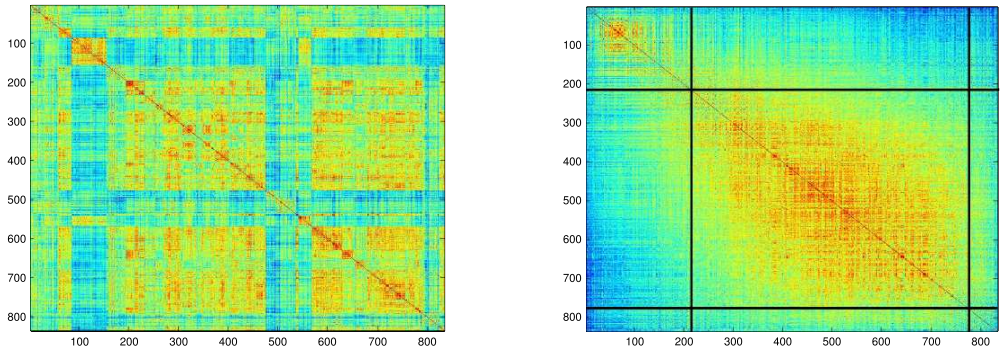
The resulting hierarchical community tree produced by the hierarchical block diagonalization method is shown in Figure 3.7 for the NSA distance and in Figure 3.6 for the LM distance. The nodes of the tree are displayed as pie charts, which represent the west-central-east division of nodes in the corresponding block. The size of nodes indicates block size. The first splits in both community trees correspond well to the large-scale geography, although the first splits might be more accurate in this sense when using the LM distance, which predicts the large-scale divisions better. The more densely sampled west is separated from the east in the beginning of the splitting process, and the two parts are thereafter independent of each other. Thus, it is clear that the heterogeneous sampling does not cause problems for the block diagonalization, as it did for the k-clique percolation method.

The fact that the block diagonalization of the LM distance matrix produced 61 communities, and the same procedure produced only 24 communities when the NSA distance was used, raises questions about the reliability of the communities at the lower levels of hierarchy. The block diagonalization method seems to be finding communities in noise, although the Bayesian information criterion which is used for determining the number of communities should be able to prevent such overfitting. To test the reliability of the hierarchical community structure predicted by the block diagonalization for the LM distance matrix, the data was randomized and the block diagonalization procedure applied to it. Block diagonalization was tested by randomizing the genetic data of ramets in western Mediterranean in two ways: The first way was to randomize the genomes in a way that the pairwise correlations of the alleles in each locus were preserved. This was done by collecting all pairs of alleles into seven vectors, each corresponding to one locus. The elements in each vector were then randomly permutated. The second way was to discard also pairwise correlations and only keep the distributions of alleles in each locus. After this procedure, the data would correspond to a randomly mating population. The NSA distance measure was used to calculated the distance matrices, which are shown in Figure 3.5 for the first randomization scheme. In both of the cases the block diagonalization found a hierarchical community tree with multiple hierarchy levels. This suggests that either the Bayesian information criterion is too loose a condition for choosing the number of splits, or the energy function used for evaluating each division of nodes to blocks is not suitable for this problem. Either way, more research is needed to verify that the lower levels of the hierarchical community trees inferred by block diagonalization are not just artefacts of the method.

3.3.6 Hierarchical community detection vs. phylogenetic trees

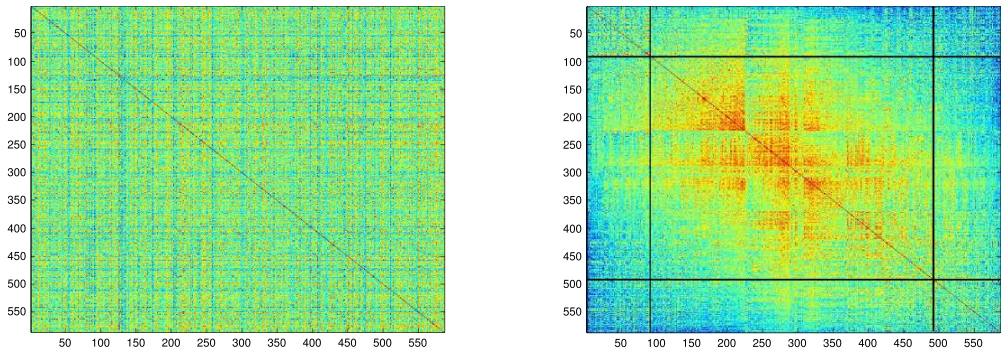
A phylogenetic tree is a representation of the evolutionary relationships between samples having a common ancestor. The sampled organisms are represented by leaves of the tree and inner nodes of the tree represent the most recent common ancestors of their descendant nodes in the tree.

Algorithms used for building phylogenetic trees can be divided into character-based methods and distance-based methods. The first class uses genomes of two nodes to construct the genome of their last common ancestor, which is the parent



(a) Before (b) After

Figure 3.4: The NSA distance matrix of 834 samples. Colors indicate the distance: smallest distances are denoted by red and largest distances by blue. Panel a) shows the distance matrix before reindexing. The visible structure is due to the fact that the indexing is not random, but it follows the sampling locations. The panel b) shows the distance matrix, which is reindexed to maximize Equation 3.2 by moving the small-distance elements close to diagonal.



(a) Before (b) After

Figure 3.5: The randomized (preserving the pairwise allele correlations) NSA distance matrix of the 586 samples from western Mediterranean. Colors indicate the distance: smallest distances are denoted by red and largest distances by blue. In panel a), the indices are random and in panel b), the matrix is reindexed to maximize Equation 3.2 by moving the small distances close to diagonal.

node of those two nodes in the phylogenetic tree. Distance based methods, on the other hand, use genomes only to calculate the distances between all leaves of the tree. After that, the distances from ancestral nodes to other nodes are inferred from the original distance matrix of the samples, and the genomes of ancestral nodes are not explicitly constructed. The character-based framework often leads to better phylogeny trees, but also to complex combinatorial problems. Distance-based methods are mostly more simple clustering algorithms, whose results are used as starting points of heuristic algorithms for character-based methods.

Phylogenetic distance-based methods are similar to community detection meth-

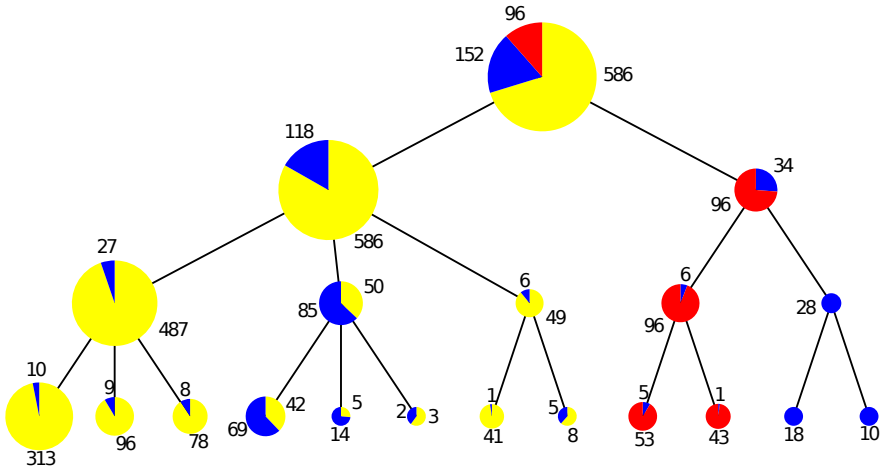


Figure 3.6: Hierarchical community tree produced by block diagonalization using the LM distance matrix. The size of each node is relative to the size of the corresponding community, and the coloring of the pie charts corresponds to geographical division of the nodes to west (yellow), center (blue) and east (red). Only three first branching events are shown (the maximum is 9).

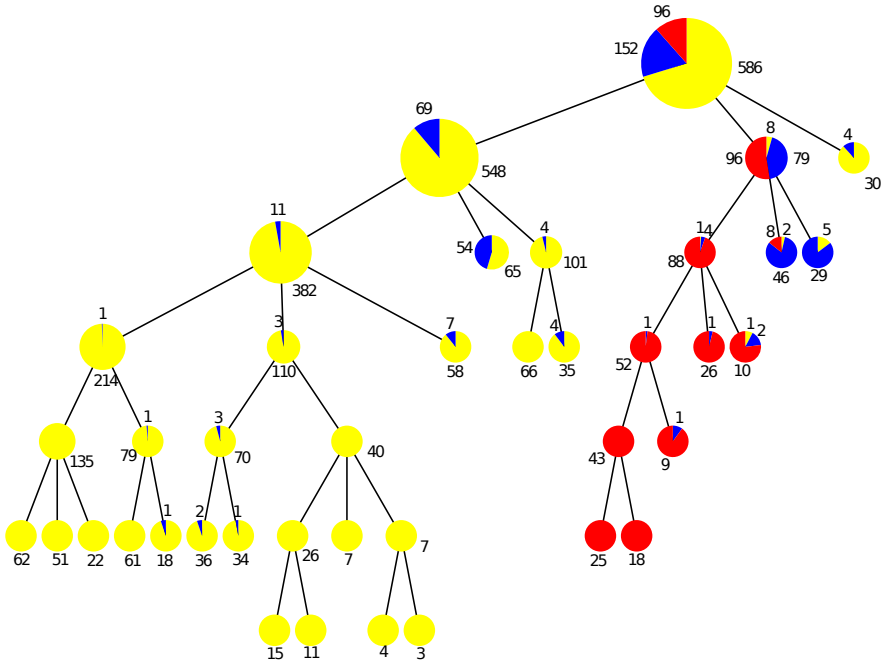


Figure 3.7: Hierarchical community tree produced by block diagonalization using the NSA distance matrix. The size of each node is relative to the size of the corresponding community, and the coloring of the pie charts corresponds to geographical division of the nodes to west (yellow), center (blue) and east (red).

ods for dense weighted networks in the way that both try to find hierarchical structure in distance matrices. In addition, distance-based phylogenetic methods

are basically centroid-clustering methods, and in that sense resemble general clustering methods used for weighted networks. However, due to the different starting points of phylogeny tree and community detection methods, there are some differences: Phylogenetic tree construction methods mostly try to fit a biologically motivated evolution model to the distance matrix, whereas network clustering methods are more generic, and typically do not explicitly assume that the data is based on any model. In addition, phylogeny clustering methods assume that the clustered matrix is a distance matrix, but network clustering methods can use any sparse matrix as a starting point. Lastly, the nodes in the phylogeny trees are splitted until no further divisions are possible, which is not the case for most of the hierarchical community detection methods.

As an example of a distance-based phylogeny tree building method, a simple, but fairly popular method, UPGMA [28], is used for comparing the results of network-based hierarchical community detection methods to a phylogenetic tree. The aim of this comparison is to see if the community detection methods really detect more accurate or meaningful structures. This should be the case, as the objectives behind community detection methods are different of those of phylogeny methods. Note that the choice of the phylogeny tree method is motivated by the simplicity of the UPGMA algorithm, and other popular algorithms such as neighbor-joining [49] could have been used as well.

UPGMA

UPGMA (Unweighted Pair Group Method with Arithmetic mean) is a hierarchical agglomerative clustering method commonly used for building phylogenetic trees using distance-based genetic data. It starts by creating a cluster C_i for each sample i, which are the leaves of the phylogeny tree:

$$C_i = \{i\}. \tag{3.5}$$

The algorithm continues by combining two clusters having the smallest distance between them. This is repeated until there is only one cluster left. The distance between clusters is defined as the mean distance between their constituent nodes:

$$d(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} d(i, j), \tag{3.6}$$

where $d(i, j)$ is the genetic distance between nodes i and j.

Edge lengths in the phylogenetic tree produced by UPGMA are calculated by assuming that the species have developed with the same speed after each division, which is know also as the molecular clock assumption. This leads to defining the edge lengths in such a way that the sum of edge lengths along the path from a leaf to any inner node does not depend on the leaf. This means that every inner node has a height in the tree which equals half of the distance between the two children of that node. The height also corresponds to the evolutionary age of the nodes. The leaves, which have been observed in current time, have a height of zero, and the root is the oldest node.

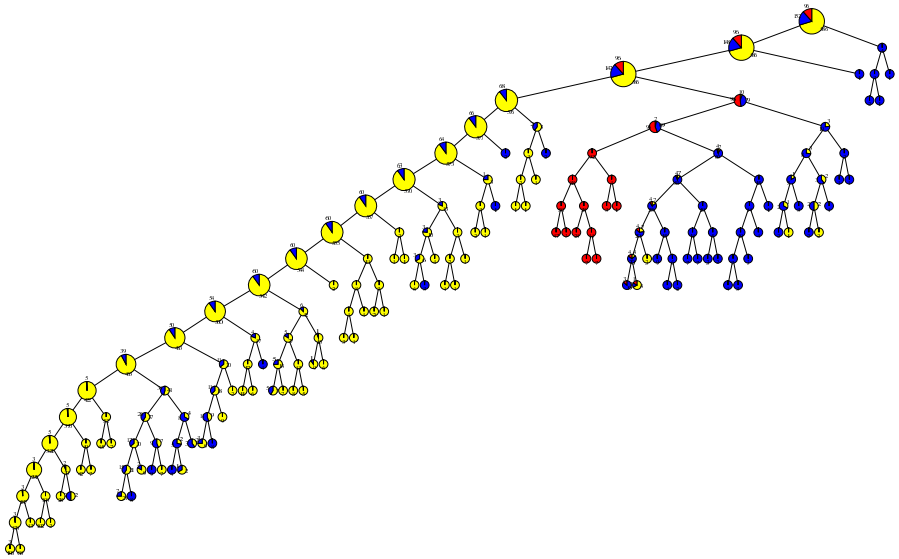


Figure 3.8: The phylogeny tree produced by UPGMA and the NSA distance matrix. The size of each node is relative to the size of the corresponding community, and the coloring of the pie charts corresponds to geographical division of the nodes to west (yellow), center (blue) and east (red).

Figure 3.8 shows the resulting phylogenetic tree when the UPGMA is applied to the NSA distance matrix. Nodes of the tree have been colored with respect to the large-scale geographical divisions. The west-east cleavage is clearly visible in the UPGMA tree, and the eastern and central nodes are perfectly divided into their own clusters. The phylogenetic tree resembles more the k-clique percolation trees of Figures 3.2 and 3.3 than the block diagonalization tree of Figure 3.7, as the shape of the UPGMA tree is unbalanced. This shape is caused by the large western cluster breaking apart by splitting into one small and one large cluster at each hierarchy level. Hence this method does not provide any meaningful information on the cluster structure within the western nodes, although it accurately

detects the east-west cleavage.

3.4 Comparing community detection methods

The problem of comparing community detection methods is twofold: comparing the performance of community detection methods and comparing the similarity of community structures. As discussed earlier, detecting communities is not a straightforward problem mainly because no universal definition of a community exists. The same problem is encountered in a slightly different form when community detection methods are compared. Comparing the performance of two community detection methods is just a variation of the community detection problem, and comparing the similarity of two community structures is only possible if the underlying definitions of a community are compatible. Otherwise, the whole question of similarity of two community structures can be ill-posed.

Comparing the performance of two community detection heuristics is straightforward if the heuristics share the same definition of a community. However, it is impossible to compare the performance of two community detection algorithms without defining communities a priori, because solving the problem of performance comparison would lead to a solution of the problem of community detection. A comparison method which is able to select the better one of any two community structures with respect to performance will immediately induce a definition of a community, because the method can be used to select the best community structure given any network. As the number of possible community structures in any finite network is also finite, the best community structure with respect to the comparison method always exists. An alternative approach to comparing the performance of community detection methods is to use networks for which community structures are defined beforehand. This way, the similarity of the community structure produced by any community detection method and the predefined community structure can be used as a benchmark of performance of the method.

Comparing similarity and finding differences in community structures are nontrivial tasks, and the difficulties encountered in those tasks can be traced back to the list of ambiguities in defining communities. Take the clique percolation method and the block diagonalization method as an example. The block diago-

nalization method always assigns each node to a single community, whereas the clique percolation method allows overlapping communities. Thus, it is impossible for a community structure detected by the block diagonalization method to be exactly similar to one containing overlapping communities detected by the clique percolation method, and the whole concept of similarity between the structures detected by these two methods becomes ambiguous.

The community detection methods used in this Thesis all produce trees representing hierarchical community structures. A simple comparison scheme for two hierarchical community structures is to compare single levels of hierarchy at a time. However, two otherwise very similar hierarchical community structures can appear to be very different to this naive comparison method, if the two levels chosen for comparison correspond to different structural scales in the network. In addition, trying out all the possible ways of choosing a level in a hierarchy tree might not be computationally feasible. A better approach would be to incorporate the whole hierarchical community structure to the comparison.

Despite the problems in comparing community detection methods, some comparisons are made in this Thesis for the community structures produced with the methods introduced in the previous Section. We begin by comparing single levels of hierarchical community structures using visual comparison methods. This allows us to roughly compare the the otherwise incompatible methods such as the clique percolation and block diagonalization. After that, the performance of the block diagonalization method is evaluated by comparing similarity of the community structure detected by it to the large-scale geographical division of nodes using the mutual information framework. Finally, a phylogenetic tree produced by the UPGMA is compared to the community structure produced by the block diagonalization method.

3.4.1 Visualization using MST

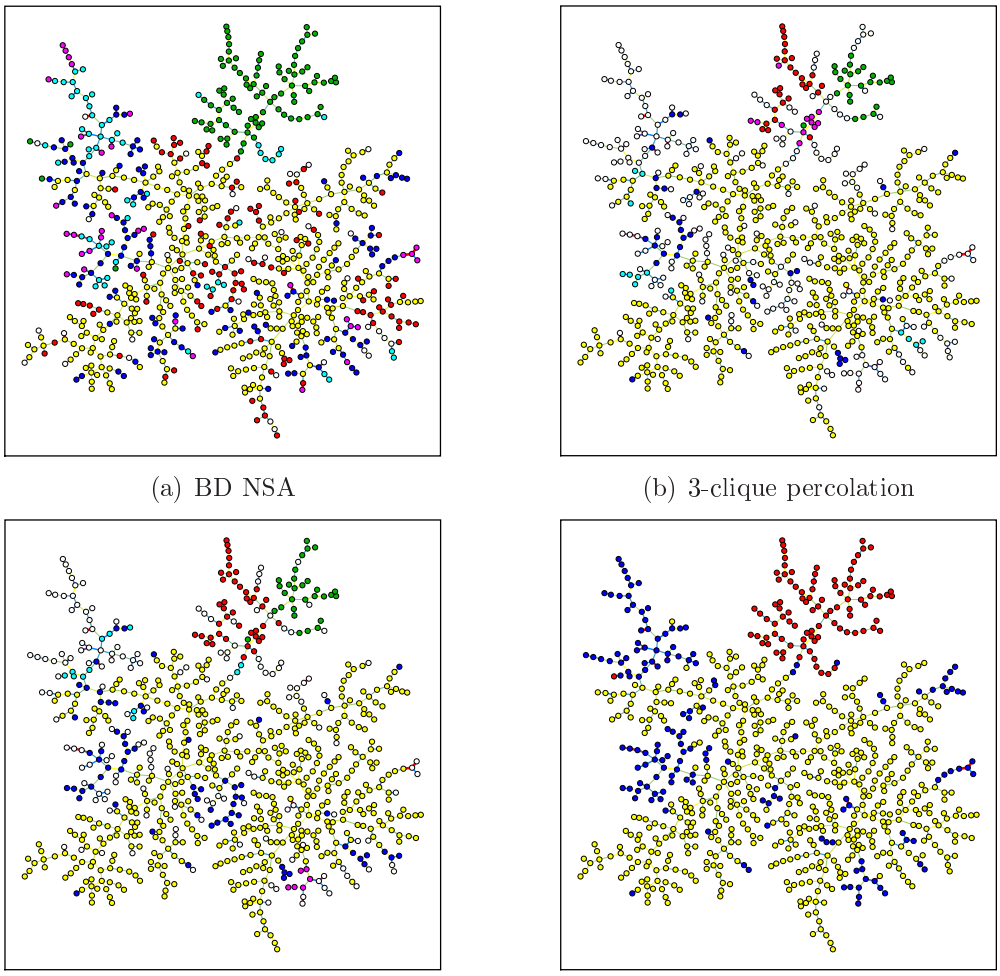
A straightforward way to visualize community structure is to visualize the network such that the color of each node corresponds to its community. Using this approach, Hernández-García *et al.* [29] visualized the genetic similarity network of Posidonia oceanica using the 37 different geographical locations as communities. They used the maximum spanning tree for calculating the layout for the nodes. In the resulting plots, nodes from same locations formed groups, illustrating that

smallest distances are mostly found inside the sampling locations and only rarely between two nodes from separate locations.

Plotting multiple community structures side by side using the same layout of nodes for all plots can be used to visually compare the similarity of the structures, and to identify where the communities differ. In this Thesis, MST-based visualization is used to compare communities formed by different community detection methods. Although this visual comparison is mostly free of assumptions made by the community detection method, and thus avoids some of the problems related to comparing communities, other problems still remain.

The first problem is the limited color scale. Only a few communities can be presented in a way still visible to the eye, which limits the choice of hierarchy level in the community structure. This leads to a bigger problem, where choosing single hierarchy levels from two hierarchical divisions might produce two similar divisions or two very different divisions depending on how the choice is made. This problem is also encountered later when using mutual information to compare community structures, and it is discussed in detail in that context. The number of different colors, and thus the maximum number of different communities visible at the same time, was set to six. The MST of Figure 3.9 displays a division of nodes based on geography and communities detected with the block diagonalization method, the k-clique percolation method, and the edge percolation method. The large-scale geographical correlations are clearly visible in the figure, as the western, central and eastern nodes form distinct groups. The western and eastern parts are well separated, but the central nodes seem to be somewhat mixed with the western nodes. The fact that some of the central nodes are far away from each other on the MST does not necessarily mean that they are far away from each other genetically, but could just be an indication that the central nodes are very close to the western nodes and the MST is somewhat random for that area. The west-east cleavage is visible for all of the three community detection methods, although the percolation methods have already splitted the east to two parts at the chosen level of hierarchy.

MST is a useful visualization tool for networks having a clear structure, like the west-east cleavage observed in genetic population structure of P. oceanica. In such clear cases, MST visualization can give an overview of the community divisions. However, if the MST is unstable or not unique, visualizations might become hard to interpret or even misleading. More quantitative methods for



(a) BD NSA

(b) 3-clique percolation

(c) Edge percolation

(d) Geographical

Figure 3.9: Visualizations of maximal spanning trees of the NSA distance network of Posidonia oceanica, where nodes have been colored according to various divisions: a) block diagonalization, b) k-clique percolation, c) edge percolation and d) geographical communities. For the percolation methods, only nodes belonging to the six largest communities are colored, whereas white nodes belong to smaller or no communities.

comparing similarity of community structures are clearly needed.

3.4.2 Normalized mutual information

Mutual information is an information-theoretic tool [50] which can be used for comparing similarity of two divisions of nodes into communities [51]. In order to use the mutual information framework, community divisions must first be transformed to random variables having discrete probability distributions. A division of nodes to communities can be transformed to a probability distribution

of hitting each community when a node is chosen with uniform probability. For two divisions, the underlying set of nodes is assumed to be the same, and the mutual information answers the following question: if a node is chosen at random and the community of that node in one division is known, how much information, or entropy, does that knowledge carry about the community of the same node in the other division of nodes to communities?

Mutual information can be formalized for a graph $G(V, E)$ and two of its partitions A and B by defining the *confusion matrix* N such that

$$N_{ij} = |\{v | v \in V_i^A \wedge V_j^B\}|, \tag{3.7}$$

where V_i^A is the community i in partition A and V_j^B is the community j in partition B. Let us denote by $\mathcal{A}$ a random variable depicting choosing a community from partition A, if a node is uniformly randomly chosen from V. With these notations, the probability mass function of $\mathcal{A}$ becomes

$$p_{\mathcal{A}}(i) = \frac{\sum_j N_{ij}}{|V|}. \tag{3.8}$$

The respective joint probability mass function is

$$p(i, j) = \frac{|V| N_{ij}}{(\sum_i N_i)(\sum_j N_j)}. \tag{3.9}$$

The joint probability mass function is used in the definition of the mutual information of the two community structures A and B:

$$I(A; B) = \sum_{i \in A} \sum_{j \in B} \frac{p(i, j)}{p_A(i) p(j)}. \tag{3.10}$$

The problem with using mutual information is that two mutual information values are not necessarily comparable, as they are not normalized, and the result depends heavily on the entropies of the two divisions. The normalized mutual information can be defined as

$$I_n(A; B) = \frac{I(A; B)}{0.5(H(A) + H(B))}, \tag{3.11}$$

where $H(A)$ and $H(B)$ denote entropies of partitions A and B.

The mutual information of two random variables tells how much one random variable's entropy is reduced if the other random variable is known. Mutual

information is thus relative to the entropies of the two random variables. Normalized mutual information tells how much this change in the amount of entropy is relative to the mean of the entropies of the two random variables. This ensures that the normalized mutual information is always between 0 and 1, and makes it easier to compare cases where the underlying entropies differ. Comparing unnormalized values of mutual information would only show the overall difference in entropies. This would be the case, *e.g.*, for different community hierarchy levels where the entropies at the lower levels would always be bigger than at the upper levels, and the partitions of the upper levels would have smaller values of mutual information than those of the lower levels.

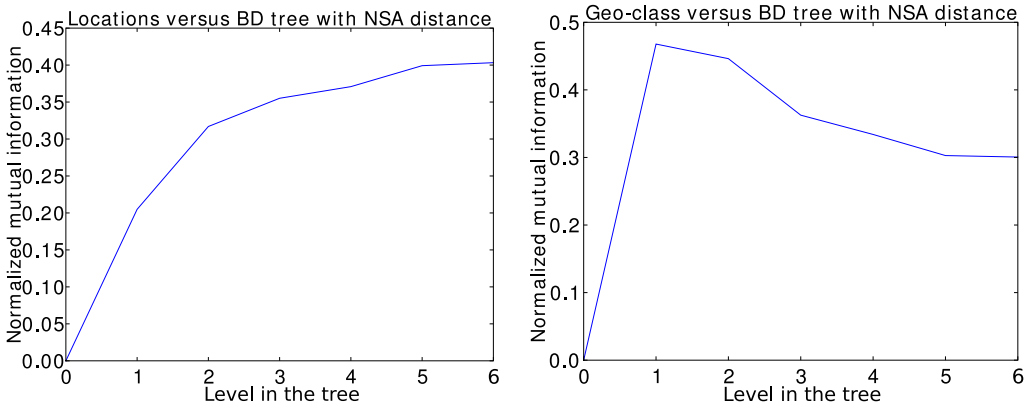
3.4.3 Comparing community detection methods and geography with NMI

Communities detected with various methods were seen to correlate with the geographical divisions of nodes to west, center and east when visualized using the MST of Figure 3.9. The division of nodes to locations was also seen to correlate with the genetic LM distance [29]. Although these correlations were clearly visible in MST visualizations, it was also clear that the correspondence was not perfect. The NMI framework is now used to quantify these correlations in both cases.

Calculating the mutual information between results of a hierarchical community detection method and the large-scale geographical division is not straightforward for two reasons. First of all, the mutual information approach requires division of the nodes into groups, which can be done for hierarchical community structure tree by looking at one level of the hierarchy at the time. This is done for the block diagonalization of the NSA distance matrix by defining hierarchy levels with respect to the number of splitting events. For example, at the third level of hierarchy, all communities are three links away from the root node. This is of course not a unique nor necessarily the best way to define the hierarchy levels. One could, for example, define a distance between the nodes in the tree. The distances could be related to the block diagonalization process, or use some extra biological information. Trying out all the ways of dividing the tree into hierarchy levels would lead to a very large number of different combinations of communities, and would not be a feasible solution. The second problem is a variation of the first: geography can also be hierarchically divided to different regions, subregions

and so on, and the number of such combinations is even larger for geographical data than it is for a hierarchical tree. Multiple hierarchy levels are not used here for the geographical locations. Instead, only two divisions are used: the first is the most accurate geographical division feasible, that is, the division of nodes according to individual sampling locations. The second is the crude division of nodes to the three areas discussed earlier: west, center and east.

The normalized mutual information for each hierarchy level of community structure detected with the block diagonalization method is shown in Figure 3.10. The normalized mutual information of node locations and hierarchy levels is seen to increase as function of hierarchy level in panel a). This means that the last divisions made by the block diagonalization method are not completely random with respect to the locations. However, the block diagonalization method was seen to find communities in randomized data, which might suggest that the last levels of the tree might be noisy also for the real data.



(a) all locations (b) 3 division

Figure 3.10: Normalized mutual information of communities produced with the block diagonalization method using the NSA distance matrix and a) the locations of the specimen and b) division of the nodes to west, center and east. The hierarchy level in the community structure is on the horizontal axis and the NMI on the vertical axis. The NMI between the two geographic division (west-center-east, sampling locations) is approximately 0.37.

The NMI of the large-scale geographical division and the hierarchical community structure, on the other hand, is at its maximum after the first split to blocks, and is seen to slowly decrease thereafter, almost saturating at the last levels of hierarchy. This behavior can be explained with the help of Figure 3.7 illustrating the branching process. The first split separates west from center and east, and the center and east separate only at the next level. As the west is more densely sampled, it has more weight on the value of NMI, and thus the second branching

event separating the east and the center is not enough to raise the overall value of NMI, as it also further divides the western component. It is worth noticing that, as discussed earlier, the number of branching events might not be the optimal way of defining different hierarchy levels. This is highlighted by the fact that if a community division is chosen from the hierarchy tree in such a way that communities with mainly western nodes are chosen from the first level and the rest from the second level, as illustrated in Figure 3.11, the overall NMI improves from the original first level value of 0.468 to a value of 0.527.

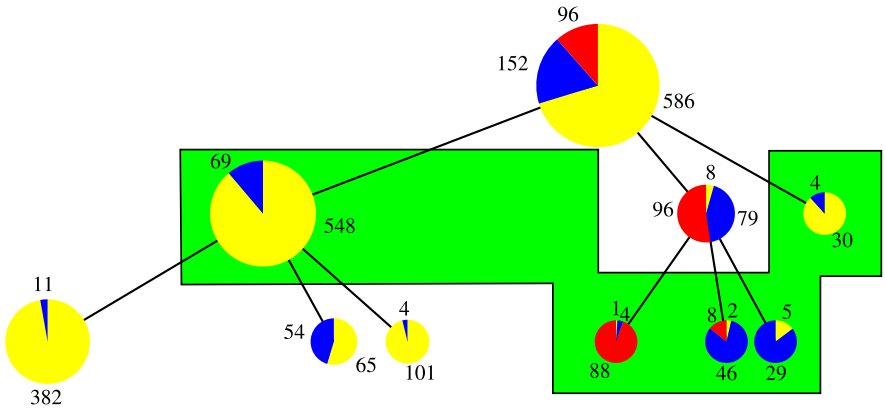


Figure 3.11: The first two levels of the community hierarchy tree produced by detection using the block diagonalization method using the NSA distance measure. The green shading represents an alternative way of choosing a division of nodes to communities from the tree. This particular choice reflects better the division of nodes to the three geographical groups: west, center and east. The NMI of this division and the large-scale geographical division is 0.527, whereas if the original levels of the tree would be used, the corresponding values would be 0.468 for the first and 0.446 for the second level.

3.4.4 Comparison to UPGMA

A phylogeny tree constructed with the UPGMA was compared to the division of the nodes to communities at the final level of block diagonalization using NMI. A height was defined for each node in the phylogenetic tree computed with the UPGMA as a number relative to the age of the nodes. The leaves are the youngest and have height (age) of zero, and the root of the tree is the oldest node thus having the largest height (age). In the hierarchical clustering framework, the height can be interpreted as the hierarchy level of the tree. Figure 3.12 shows the NMI of the block diagonalization community structure and the community structure extracted from the last level of the phylogeny tree as a function of

minimum accepted height. This means that nodes below the minimum accepted height are discarded and the leaves of the tree are considered as the community division at the respective threshold level.

The NMI for the block diagonalization communities and the UPGMA tree is seen to slightly increase when the threshold is increased from its minimum value. Its maximal value of 0.61 is attained at a threshold of approximately 0.25. Thereafter the NMI values begin to decrease. This means that the block diagonalization communities and the UPGMA tree explain approximately 61 percent of each other's entropies at the maximum. The UPGMA tree corresponds better to the communities produced with block diagonalization than any geographical divisions tested here, but the correspondence is still far from perfect.

There are two possible reasons for the imperfect correspondence between the communities produced with block diagonalization and the UPGMA tree. The first reason is that the topology of the tree produced by the block diagonalization method corresponds well to the one produced with the UPGMA, but the thresholding scheme for the UPGMA tree fails to produce suitable community divisions. This explanation is supported by the fact that the value of NMI in Figure 3.12 remains practically unchanged for UPGMA threshold values ranging from 0 to 0.3, which might suggest that the best level of communities could be found as a combination of different thresholds in that range for different branches of the UPGMA tree. The second possible reason is that the biologically motivated assumptions behind the UPGMA and the more general assumptions about the communities behind the block diagonalization method lead to genuinely different community structures.

3.4.5 Summary

As community detection methods can in general be divided into two categories, local and global, methods from both categories were chosen to study clusters in the genetic structure of Posidonia oceanica. The clique percolation method was chosen as representative of local methods and the block diagonalization method was chosen as the global method. Although both methods have earlier been used for unweighted networks, they had to be modified to allow community detection in dense, weighted networks, and a completely different approach to the algorithmic implementation of the clique percolation method had to be developed.

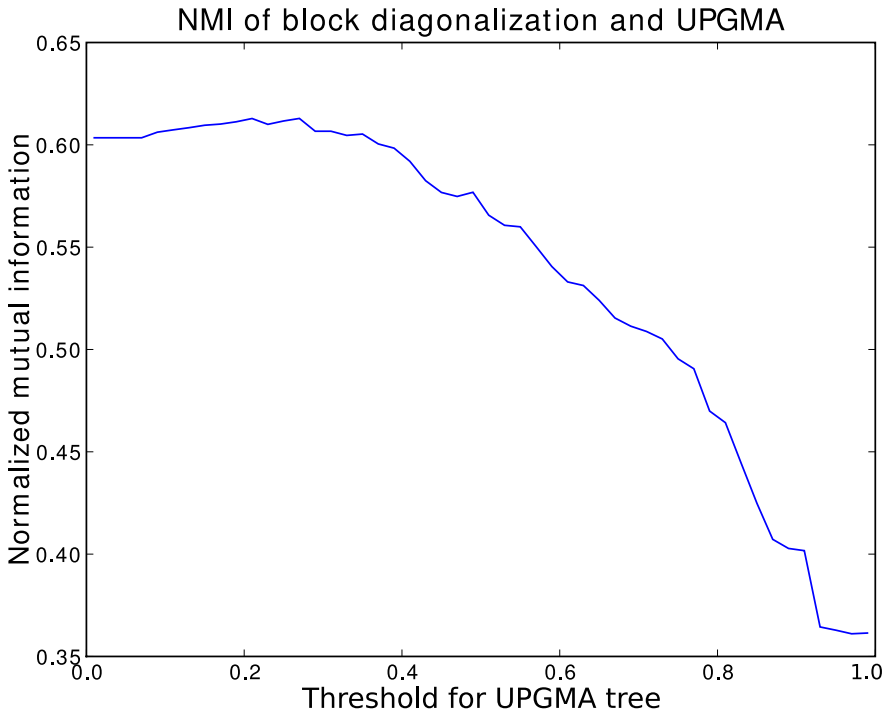


Figure 3.12: Comparing the last level of communities inferred with the block diagonalization method using the NSA distance to the last level of UPGMA tree with different height thresholds. Thresholding is done in such a way that nodes which are at distance larger than the threshold level from the root are not splitted any further.

Results obtained using the clique percolation method seemed to be discouraging as the resolution of the edge weights seemed to be too low, whereas the block diagonalization method produced sensible hierarchical community structure. However, both of the methods performed well when compared to the overall west-central-east geography. A visualization approach using maximum spanning trees and a more quantitative approach using the normalized mutual information were used to compare the different community detection methods, a phylogeny tree method and underlying geography. The comparison methods seemed to suffer from the fact that a single community division needed to be used instead of comparing whole hierarchical trees. Despite of this, NMI was successful in quantifying how much communities detected with the block diagonalization method correlate with the geography, and showing that the UPGMA phylogeny tree performs better in explaining these communities than the geographical divisions.

Based on these results one can argue that the overall geography is clearly reflected in the genetic structure of the sampled P. oceanica populations, as the west-central-east division was detect by all methods. However, when comparing

community divisions by any used method to the higher resolution geography (the 37 sampling locations), the NMI values indicate that there is no clear one-to-one mapping: the detected clusters correlate with geography, but do not localize within well-defined small geographic areas.

Chapter 4

Conclusions and future research

In this Thesis, the possibility to use the network framework and network-based methods to unveil the genetic population structure of Posidonia oceanica has been critically assessed using several techniques, such as the minimum spanning tree, k-clique percolation, block diagonalization and normalized mutual information. The choice of the proper genetic distance measure was seen to be ambiguous and acknowledged to play an important role, as it serves as a link between biological data and the network abstraction. In this Section, the main results and conclusions of this Thesis are collected together, and suggestions for solving some of the encountered problems are given.

Two genetic distance measures with different background assumptions were tested and their effects on the following network analysis discussed. The non-shared alleles distance (NSA) assumes that variation between two individuals is due to recombination of predefined alleles, and the linear Manhattan distance (LM) assumes that variations are due to mutations in the lengths of the microsatellite repetitions. The NSA measure is closer to the population dynamics view and works well on local, population-level scales. The LM distance tends to be more accurate for longer-timescale changes, and is closer to the phylogenetic tree perspective as no recombinations are assumed to happen. This effect can also be seen in the ROC curve comparing the two distances to geographical divisions at different scales in Figure 2.7. The NSA distance was chosen to be the main distance used in this Thesis, but it might be worth the effort to pursue a more efficient distance measure working at multiple hierarchy levels. The distance measure could for example be a combination of NSA and LM type distances giving

more weight to the NSA part at small distances and more weight to the LM part at large distances.

The k-clique percolation method with thresholding was seen to fail to deliver good results, mainly due to the lack of resolution in edge weights. However, correlations with large-scale geography were clearly present in the resulting communities. A new algorithm was developed for the the k-clique percolation method in order to detect hierarchical communities in dense weighted networks. Simultaneously, this algorithm was proven to be very fast for sparse networks and low values of k. Nevertheless, results of community detection with the new algorithm suffer from the biased sampling scheme of the data, as all the edges inside the western parts of the Mediterranean have far higher weights than almost any edges between the large-scale geographical areas. The k-clique percolation method thus seems to be almost useless for such cases, and there are no trivial solutions to this problem. Increasing the clique size k would not improve the resolution problems, and would severely harm algorithmic performance. One way to increase the resolution would be to accept the randomness caused by evaluating the community structure when an arbitrary number of edges is added. The resulting tree could be then sampled multiple times with the same number of added edges, but with different permutations of edges with the same weights. Consensus tree methods [54] could then be applied to the set of found trees.

The block diagonalization method, modified from the one introduced by Sales-Pardo *et al.* [47] for detecting communities, seemed to work better than the k-clique percolation method for genetic similarity networks of Posidonia oceanica. Heterogeneous sampling did not cause trouble for the block diagonalization method, and the resulting hierarchical tree was more balanced than the ones produced by the k-clique percolation method and the UPGMA. The genetic structure predicted by the block diagonalization method seemed to correlate with geography on both large and small scales (see Figures 3.7 and 3.10). Despite this success, the block diagonalization method has some problems: The method does not produce any branch lengths for the hierarchy tree, which severely limits the use of the hierarchy, or can even cause misleading results if the absence of branch lengths is solved by assigning a same length to each branch. Another problem with the method is that it produced hierarchical structure even for randomized null models, which might indicate that the lowest communities in hierarchy trees might not be robust or reliable. This overfitting might be solved by using a better model for selecting the blocks after the reindexing procedure, as it seems clear to hu-

man eye that there are no distinct blocks in panel b) of Figure 3.5. Furthermore, the method is computationally expensive, as methods such as simulated annealing needs to be used for both reindexing the distance matrix and for detecting blocks in the matrix. More work on these problems is clearly needed before the block diagonalization method can be reliably used for community detection in this context.

Mutual information can be used as a tool for comparing similarity of two division of same set of elements, and has a solid basis in information theory. Normalized mutual information (NMI) has been earlier used to compare community structures found by different community detection methods [51]. The NMI was used in this Thesis to compare similarity of community structures, but hierarchical structures caused problems, as a single hierarchy level or division of nodes to communities needs to be chosen in order to use the NMI framework. This is problematic specially for the block diagonalization trees, where there are no branch lengths, and the choice of a proper hierarchy level is ambiguous. Topological measures for the similarity of trees are used in phylogeny and could possibly be also used in the community detection context with some alterations. Similar problems were encountered when communities were compared to geographical divisions. A ROC curve of geographical distances and communities might be more illustrating than calculating NMI for the two geographical divisions used here.

Although the network perspective for studying complex phenomena related to biological systems at the borderlines between population biology and phylogenetics seems promising, some caution is needed. Network studies are not completely free of biological assumptions that seem to restrict the usage of traditional methods. Network-based methods have both explicitly defined assumptions, such as the choice of genetic distance measure, and implicit assumptions, such as the ones made by community detection algorithms. The network perspective seems to be best suited for exploring new data, but results can be somewhat unreliable or even misleading if generic network methods are used without caution. It would be advisable to use multiple network methods or some traditional methods to verify any results produced with network-based methods.

Appendix A

Networks: definitions and basic measures

The terms "network" and "graph" can be used interchangeably, although it is typical to speak of complex networks instead of complex graphs when referring to networks/graphs with non-trivial structure. In this Thesis "network" refers to an undirected graph that doesn't have multiple or self edges. Mathematically such a graph $G(V, E)$ consists of a finite set of vertices V and a set of edges $E \subset VxV \setminus \{(v, v)|v \in V\}$. When an index is given for each vertex in V, the graph $G(V, E)$ can be presented as an adjacency matrix A, with $A_{ij} = 1$ if $(v_i, v_j) \in E$, and $A_{ij} = 0$ otherwise.

In many cases it is useful to assign some weight to each edge of the network. This defines a weighted graph or weighted network $G(V, E, w)$, where w is some function from the set of edges E to positive real numbers: $w : E \to \Re_+$. This can also be represented as a weight matrix where the absence of an edge is interpreted as a zero weight: $W_{ij} = w((v_i, v_j))$ if $(v_i, v_j) \in E$ and $W_{ij} = 0$ otherwise.

The degree k for each node i is defined as the number of neighbors it has:

$$k_i = \sum_{j=1}^{|V|} A_{ij}. \tag{A.1}$$

In a weighted network an analogous measure, the strength, is defined as

$$s_i = \sum_{j=1}^{|V|} W_{ij}. \tag{A.2}$$

Another useful single-node characteristic is the *clustering coefficient* [56]. It is a measure originating from social sciences where it measures the probability of one's friends also being friends. Such a transitivity measure is also generally useful and can be defined in a graph for node i as the number of triangles where the node participates divided by the maximum possible number of such triangles given the node's degree:

$$c_i = \frac{t_i}{\frac{1}{2}k_i\,(k_i - 1)} = \frac{\sum_{j,k} A_{ij}A_{jk}A_{ki}}{k_i(k_i - 1)}. \tag{A.3}$$

If the clustering coefficient is defined this way, its value will depend heavily on the degrees of the neighboring nodes. To remove the effects of degree correlations, the clustering coefficient for node i can be defined as the number of triangles it forms divided by the maximum number of triangles it can form given its and its neighbors degrees [57].

The clustering coefficient can also be defined for weighted networks so that the weights affect its value [58–64]. This approach doesn't seem to be very fruitful as the value of the coefficient becomes very degenerate and hard to interpret [65].

Two nodes i and j have a *path* between them if there is a sequence of nodes $p_{ij} = \{v_k\}_{k=1,..,n}$ for which $v_1 = v_i$, $v_n = v_j$ and $A_{k,k+1} = 1$ for every $k = 1, .., n-1$. The length of a path is then $n - 1$ and there is a (possibly non-unique) *shortest path* $\hat{p}_{ij}$ between every pair of nodes. Two nodes are said to be in the same *component* if there is a path between them. The *diameter* D(G) of the network is defined as the maximum of the shortest path lengths between any two nodes in the network:

$$D(G) = max(\{|\hat{p}_{ij}| - 1 | i, j \in V\}). \tag{A.4}$$

A network G is said to be a *small-world* network [56], if the average path length $\langle p \rangle$ is small compared to the network size, but the average clustering coefficient $\langle c \rangle$ is large.

The *betweenness centrality* of node i is the number of shortest paths going through

the node. If a shortest path is not unique, its contribution to betweenness centrality is divided by the number of shortest paths between the same nodes:

$$B_i = \sum_{j \neq i \neq k} \frac{|\{\hat{p}_{jk}\} \cap \{\hat{p} | v_i \in \hat{p}\}|}{|\{\hat{p}_{jk}\}|}. \tag{A.5}$$

A *clique* of size k, is denoted here as C_k, is a set of nodes in which every pair of nodes has an edge between them. $C_{k'}$ is a *sub-clique* of C_k, if it is a clique and $C_{k'} \subset C_k$. *Maximal clique* is a clique in graph G which is only a sub-clique of itself.

A *subgraph* $G(V') = G'(V', E')$ of a graph $G(V, E)$ given a set of vertices V' is a graph, for which $V' \subset V$ and $E' = \{(v_1, v_2) | (v_1, v_2) \in E \wedge v_1, v_2 \in V'\}$. An *intensity* [59] can be defined for a weighted subgraph $G'(V', E', w)$:

$$I(G'(V', E', w)) = \prod_{e \in E'} w(e)^{|E'|^{-1}}. \tag{A.6}$$

A *tree* is a graph with no cycles. This means that there is an *unique* path between every node of the tree.

A *spanning tree* of a graph G is a tree that has the same set of nodes as the graph G. If the graph is not a tree, the spanning tree is not unique for that graph, and the set of spanning trees is called the *spanning tree forest* for the graph G.

A *minimum/maximum spanning tree* (MST) for a weighted graph G is the tree in the spanning tree forest of G, for which the sum of edge weights is minimal/maximal. Note that the MST might not be unique if there are multiple edges with similar weights in G.

Appendix B

A sequential thresholding algorithm for k-clique percolation

The algorithm given by Palla *et al.* in the paper introducing the clique percolation method [42] relies on finding maximal cliques in given networks, enumerating them and then constructing an overlap matrix with each element giving the number of shared nodes with two cliques. The matrix can be interpreted as a weight matrix, where the nodes are the maximal cliques in the network, and there is a link between two nodes if the corresponding cliques share a sub-clique. Removing the weights, which correspond to the subclique sizes, of size smaller than desired clique size $k-1$ yields a network whose components correspond to k-clique communities. Thus communities for all clique sizes can be found by adding the edges to the maximal clique network sequentially starting with the largest, and then observing merging of the components.

The maximal clique algorithm would correspond to a sweep in the vertical or topological direction in Figure 3.1, and as such has some critical limitations when used for dense weighted networks: First, it has to find the maximal cliques, which is an NP-complete problem and thus all known algorithms scale exponentially. Second, it has to be run again from the beginning for each weight threshold level, which can be a problem when the number of such levels is large. A solution to these problems would be to find an algorithm that would sweep the same space horizontally or in the weight threshold direction, as finding all cliques of given size is a polynomial problem and in most cases only a few smallest clique sizes are used [43]. This would mean that the algorithm would be required to run only

once for each desired clique size and the polynomial scaling exponents would not be very large. Such an algorithm was developed by the author and coworkers with the Posidonia oceanica data set in mind, and was used for all the k-clique percolation studies in this Thesis. The algorithm is described below.

B.1 Description of the algorithm

The common algorithmic solution for edge percolation analysis is to start from an empty network, reconstruct the original network by adding the edges one by one, and update the component structure after each addition. This way, the only updating needed to be done is the joining of two components corresponding to the nodes at the each end of the edge, which can be done by using disjoint-set forests [66]. When the nodes in the components are listed, that is the components are evaluated, we need to know the component where each node belongs to. For both of these operations the amortized time is related to the inverse Ackermann function [66, 67], which is in practice a constant factor. This makes the whole algorithm almost linear with respect to the number of added edges. Also the memory consumption is very low as the algorithm only keeps the disjoint-set tree in the memory, and there is no need to keep the entire network in memory. Thus, memory use scales linearly with respect to the number of nodes in the network.

In terms of cliques, edge percolation is equivalent to 2-clique percolation, where 2-communities correspond to components in the graph. Thus edge percolation algorithms are a good starting point for a threshold-wise k-clique percolation algorithm, as it should reduce to one of the fast edge percolation algorithms when $k = 2$.

B.1.1 K-clique percolation as edge percolation

The new sequential thresholding algorithm for k-clique percolation is based on edge percolation algorithms. This generalization requires a few observations to be made. First of all, a k-clique community can be interpreted as a component in a bipartite graph between k-cliques and $k-1$-cliques, where there is an undirected edge from each k-clique to each of its subcliques of size $k-1$. In this network, two adjacent k-cliques have a link to the same $k-1$-clique and thus a

path between them. Now, as k-clique communities are defined as maximal sets of k-cliques adjacent via $k-1$-cliques, they correspond to components of this bipartite graph. As the components of any bipartite network correspond to the components of any unipartite projection of that network, we can study k-clique percolation by tracking down components in the $k-1$-clique network, which is the unipartite projection of the bipartite network. Hence, one can, analogously to edge percolation, build the $k-1$-clique network sequentially and monitor its component structure in the process.

As k-cliques corresponds to k edges and $k-1$-cliques are represented by nodes in the unipartite $k-1$-clique network, unweighted clique percolation analysis is equivalent to detecting all k-cliques in the original network and adding them to the $k-1$-clique network in arbitrary order. The components of the $k-1$-clique network correspond to the k-clique components of the original network after all k-cliques are added. In order to use the same algorithm for weighted clique percolation [44], the only modification needed is to sort the list of k-cliques with respect to their intensities before adding them to the $k-1$-clique network. When the k-cliques are added in increasing order with respect to their intensities, *i.e.* weights, the weighted k-communities can be evaluated at any intensity threshold during the addition process analogous to weighted edge percolation.

In the hierarchical clique percolation method the weighted clique percolation method was not used as a staring point. Instead, clique communities were searched for each value of edge threshold in the original network. Thresholding the original network and applying the clique percolation method for each level of edge threshold is equivalent to adding the k-cliques to the $k-1$-clique network in the order they appear in the original network when the edge threshold level is raised, and evaluating the emerging k-communities in the $k-1$-clique network after each edge threshold level. There are two ways of finding the k-cliques in the order they are formed in the original network when the edge threshold is increased. The first way is to find all k-cliques, as is done in weighted clique percolation, and assign the smallest edge weight in each clique as a weight of that clique. Sorting the cliques with respect to these minimal edge weights will then result in the desired order for the cliques. The second way of building the list of sorted k-cliques is to follow the edge percolation procedure for the original network and to add newly formed k-cliques after each edge addition to end of the list.

Finding k-clique communities using the unipartite $k-1$-clique network when the sequence of k-cliques is found is known to scale almost linearly in time with respect to the number of k-cliques in the network and to scale linearly in memory consumption with respect to the number of $k-1$-cliques in the network. Finding and sorting the k-cliques in a general case scales log-linearly in time and linearly in memory with respect to the number of k-cliques in the network. Thus the sorting part of the algorithm has in the worst case a much poorer performance than the rest of the algorithm. This is discussed in the next subsection, and the algorithm for finding the cliques in their order of emergence in the edge thresholding process reduces the workload dramatically.

Algorithm 1 Pseudocode for finding k-clique communities when the sequence of k-cliques is known. This done by keeping track of the components of a $k-1$-clique network with the disjoint-sets forest.

```
for K in k-clique sequence do
  kr=a subclique of K
  for kr2 in subcliques of K (not kr) do
    join in disjoint-set tree: kr and kr2
  end for
end for
```

The alternative unipartite projection

Notice that the unipartite projection in the clique percolation algorithm could be defined by removing $k-1$-clique nodes instead of k-clique nodes without affecting the results of the algorithm. This was not done for two reasons: First of all, there are n times more k-cliques in the worst case than $k-1$ cliques when the network size n is a constant. This is also a valid point beyond the worst case, as for example there are more edges (2-cliques) in most networks than nodes (1-cliques). The second reason is that adding a k-clique to the unipartite $k-1$-clique network requires only combining all subcliques of the k-clique to the same component. As finding the each subclique takes a constant time, the required workload is $k-1$ times the effort required by the disjoint-set forest. On the other hand, adding a $k-1$-clique to the k-clique network would require finding all k-cliques having the $k-1$-clique as a subclique. Two straightforward solutions to this would be to either keep a lookup table of such cliques in hand, which is essentially equivalent of keeping the whole bipartite network in memory, or go through all the n possible k-cliques for each $k-1$-clique. Neither of these alternatives are good, but more complicated algorithms might exist somewhere between these two.

B.1.2 Finding the sequence of k-cliques

Finding all k-cliques and sorting them can become a bottle neck for the discussed weighted clique percolation algorithm. However, finding k-cliques their order of emergence when the weight threshold is increased can be done much faster. The procedure starts from an empty network and reconstructs the original network by adding the edges one by one, as is done in edge percolation algorithms. At each step when an edge $e = (i, j)$ is added to a network, the new k-cliques forming as a consequence of this addition can be found in the following way: First, find all the common neighbors N of i and j. After that, find all the $k-2$-cliques in the subgraph of those neighbors $G(N)$. When the nodes i and j are added to these $k-2$-cliques, they form all the new k-cliques born when e is added to the network.

This approach has three benefits over the brute-force way of finding and sorting all k-cliques: It does not require keeping all k-cliques in memory, but only the original network and possibly information related to the detecting $k-1$-cliques in each subnetwork depending on the algorithm used. It does not require sorting the k-cliques as only the edges need to be sorted. Lastly, the k-clique finding algorithm can be run simultaneously with the k-clique community finding algorithm updating the community structure immediately after each k-clique is found. This makes it possible to stop the k-clique search algorithm at any time in the community finding process. In some cases this is a huge advantage over exhaustive search of every k-clique: for example in ER random graphs [53] the number of cliques grows as $O(p^{k(k-1)/2})$ [68], where p is the probability that an edge exists. The k-clique percolation process can be stopped when all nodes are in the same community, or even before that, when some other criterion is fulfilled.

B.2 Scaling considerations

As the new algorithm for finding k-clique communities for each edge threshold level can be divided in two parts, the k-clique percolation and finding the k-clique sequence, worst case scaling is also studied separately for these parts. It turns out that the percolation part dominates the time and memory requirements in the worst case approximations. If the number of nodes in a network N and the clique size k are fixed, the worst case for this algorithm is a full network. This

Algorithm 2 Pseudocode for finding new k-cliques formed when an edge is added to the network. Notice that finding the $k-2$-cliques from a subnetwork can be done by calling this code recursively for each edge in the subnet, and by treating the cases of $k = 1$ and $k = 2$ as trivial separate cases.

```
i,j = nodes of the edge
for l in neighbors of i do
  if j is a neighbor of l then
    add l to list of common neighbors
  end if
end for
G_subnet = subnet of common neighbors
for all k-2-cliques in G_subnet do
  k-clique=nodes in k-2-clique, i and j
  add k-clique to the list of new k-cliques
end for
```

Algorithm 3 Pseudocode for sequential k-clique percolation with edge weight thresholding. Algorithms 1 and 2 are used as subroutines.

```
sort the list of edges in the network
G=an empty network
while stopping criterion for community structure is not fulfilled do
  pop edge from the sorted list
  get list of new k-cliques when the edge is added
  add edge to G
  update the community structure with the list of k-cliques
end while
```

analysis does not take into account that the algorithm can be stopped before all edges are processed when all nodes belong to the giant component.

B.2.1 Finding the k-clique sequence

For the k-clique sequence finding part of the algorithm, the worst case of a full network means that the number of edges the algorithm has to go through grows as $O(N^2)$, and for each for those edges the number of operations for finding all triangles they participate grows as $O(N)$. The number of triangles each edge participates in also grows as $O(N)$, and thus the time to find the $k-2$ cliques in a subnetwork of nodes at the corners of those triangles grows as $O(\binom{N}{k-2})$, which is the number of possible combinations of $k-2$ nodes in a set of size N. In all,

the time to find the k-cliques amounts to

$$O(N^2)(N + O(\binom{N}{k-2})) = O(N^2)O(\binom{N}{k-2}) = O(\binom{N}{k}). \tag{B.1}$$

This is also the number of the k-cliques in the network, which means that any algorithm listing all the k-cliques in any order must perform at least $\binom{N}{k}$ operations. Thus, in this sense the scaling of the algorithm is optimal. Note also that the scaling of the number of k-cliques can be written in the following way, when k is fixed, or N is fixed and large:

$$O(\binom{N}{k}) = O(\frac{N!}{k!(N-k)!}) = O(\frac{N!}{(N-k)!}) = O(N^k). \tag{B.2}$$

Thus, the number of k-cliques and the time needed to find the k-clique sequence in the worst case grows polynomially with respect to network size N when k is fixed, and exponentially with respect to k when the network size N is fixed.

B.2.2 Finding k-clique communities

For the percolation part of the algorithm, the same analysis is more straightforward, as we can use the results published for disjoint-sets forests [66, 67]. The algorithm requires k joining operations in the disjoint-sets tree for each k-clique, and if there are K of them in the network, the amortized amount of work has been proven to be $O(\alpha K)$, where α is the Ackermann function, which grows almost linearly. This means that the percolation part of the algorithm dominates the asymptotical required computation time for the described worst case scenario, and the algorithm as a whole is optimal for the task.

Real data can be considerably sparser than full networks, and for many dense networks, the algorithm can be stopped before adding all the links, so the real computation times often behave much better than the worst case scenarios. However, the real-world networks can locally resemble full networks, and those parts of the networks are often the ones taking most of the time for the k-clique percolation algorithm introduced here. Effects of dense subnetworks to the overall computation time can be approximated by using the above analysis.

Appendix C

Software toolbox for network analysis

C.1 Starting point and requirements

Most of the work done for this Thesis is related to using, implementing and developing methods from the field of network analysis. This is a rather new area of data analysis, and as such the choice of computational tools is limited. This is a problem especially when dealing with weighted dense networks, as is done in this Thesis. Published software usually offers solutions to specific problems only, and general purpose software packages were not considered suitable for use in this Thesis. Some such packages are listed below:

Pajek A toolbox for network analysis with graphical user interface. Not easy to extend and is designed for rather small networks. [69]

Boost graph library A C++ template library for graphs. Contains very few tools for statistical analysis. Also not very easy to use and extend as itself. [70]

Networkx A Python module designed for network science perspective using the Boost graph library. [71]

None of the above mentioned packages for network analysis seemed good enough for the purposes of the work described in this Thesis. Also the methods and code

developed for this project could be used and extended for other similar projects in the future. With this in mind, a list of requirements for a software toolbox was designed:

- The toolbox should have a suitable user interface for exploratory data analysis and rapid prototyping. It should be easy to implement scripts on it.
- There should be a possibility to write low level code for implementing computationally intensive methods.
- The underlying data structures should be efficient to allow usage of very large data sets even in the scripting mode.
- The above point should be true for both dense and sparse networks in such a way that the user interface remains transparent with respect to the type of underlying data structures. This means that algorithms implemented for one should work for both kinds of networks without any changes.
- The toolbox should be based on a framework which has already lots of fast code (possibly written in some low level language) for most common computationally expensive tasks, such as community detection and modeling networks.
- An automatic testing framework should be available.

C.2 Specifications

A software package for network analysis was developed as a part of this Thesis as other packages did not fulfill the above requirements. The closest one was the Networkx package, but among its other problems it was not mature enough, at least at the time. Despite this, it resembles the software package developed during this Thesis, as both of them are mostly written in Python and have a C++ library as a back-end.

The Python [72] scripting language was chosen as a front end for the toolbox for the following reasons: first of all, as a high level language it is easy to use and not as prone to programming errors as for example C is. Also as it is an interpreted language, it is easy to try out short pieces of code with the interpreter interface,

which is particularly important in data analysis where detailed specifications of the programs and scripts cannot be made beforehand, but the work consists of exploring different possibilities. Another reason for choosing Python is that there is an extensive set of libraries for most general-purpose tasks such as plotting [73], numerical analysis [74], interactive shell [75] and scientific analysis [76].

For high performance, the back-end library for sparse graphs made by Hyvönen [77] was chosen. It has been proven to be able to efficiently handle extremely large data sets, for example in analysis of mobile phone call networks [7]. It has also been proven to be suitable for network analysis in general, and has been used in the Complex networks group in Laboratory of Computational Science for several years for almost all data analysis. This use has also generated large amounts of code written for the library ranging from model generation to community detection.

The design of the software toolbox tries to follow the guidelines and requirements set in the previous subsection. The networking toolbox is organized in such a way that the network interface visible for the user is made with Python. Under that, the sparse network data structure is the same as in C++ library discussed in the previous paragraph [77,78], and dense networks are implemented with Numpy [74] matrices. This allows for writing C++ extensions for sparse networks by using the C++ library. The design is illustrated in Figure C.1.

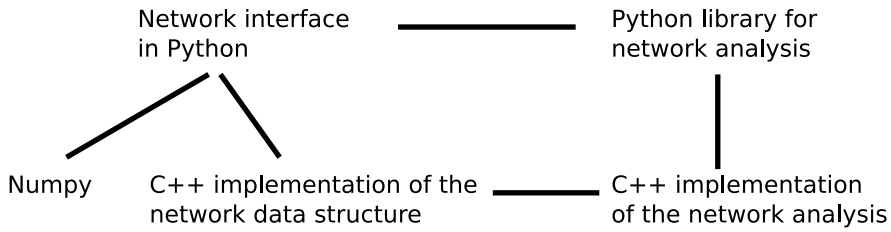


Figure C.1: Schematic picture of the Network Toolbox.

The design of the network Toolbox tries to encourage the user to follow a development cycle, which consist of the following steps:

1. Explore the problem/data in the Python interpreter using the existing library and modules.

2. Write an own module or extend an existing one with required functions in Python.

3. Write unit tests for the Python module/function.
4. Rewrite the speed/memory critical parts of the module/function with C++.
5. Now all analysis and unit tests can be done again with the C++ implementation with very small changes.

The goal of this cycle is to allow the researcher to mainly use high level scripting language such as Python, and minimize the risk of programming errors and loss of time related to writing large standalone C++ programs. Python also offers an easy interface to powerful network-related C++ libraries for people who could otherwise not use them.

Bibliography

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter: Molecular biology of the cell, Garland Science, NY (2002)

[2] W.J. Ewens: Mathematical Population Genetics, 2nd ed., Springer, NY (2004)

[3] A.F. Rozenfeld, S. Arnaud-Haond, E. Hernández-García, V.M. Equiluz, M.A. Matías, E. Serrão, C.M. Duarte: Spectrum of genetic diversity and networks of clonal organisms, Physica D 214, 166-173 (2006)

[4] D.H. Huson, D. Bryant: Application of Phylogenetic Networks in Evolutionary Studies, Mol. Biol. Evol. 23(2), 254-267 (2006)

[5] M. Nei, S. Kumar: Molecular Evolution and Phylogenetics, Oxford University Press, NY (2000)

[6] Z. Yang: Computational Molecular Evolution, Oxford University Press, Oxford, UK (2006)

[7] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabo, D. Lazer, K. Kaski, J. Kertész, A.-L. Barabási: Structure and tie strengths in mobile communication networks, PNAS 104, 7332-7336 (2007)

[8] R. Ferrer, R.V. Solé: The small world of human language, Proc. R. Soc. B 268, 2261-2265 (2001)

[9] R.N. Mantegna: Hierarchical structure in financial markets, Eur. Phys. J. B 11, 193 (1999)

[10] J.-P. Onnela, A. Chakraborti, K. Kaski, J. Kertész, and A. Kanto: Asset trees and asset graphs in financial markets, Physica Scripta 106, 48-54 (2003)

[11] R. Pastor-Satorras, A. Vespignani: Epidemic Spreading in Scale-Free Networks, Phys. Rev. Lett. 86, 3200-3203 (2001)

[12] M.E.J. Newman: The structure and function of complex networks, SIAM Review 45, 167-256 (2003)

[13] M.E.J. Newman, A. L. Barabási, D. J. Watts: The Structure and Dynamics of Networks, Princeton University Press (2006)

[14] S.N. Dorogovtsev, J.F.F. Mendes: Evolution of Networks: From Biological Nets to the Internet and WWW, Oxford University Press (2003)

[15] G. Caldarelli: Scale-Free Networks, Oxford University Press, Oxford (2007)

[16] R.J. Williams, E.L. Berlow, J.A. Dunne, A.-L. Barabási, N.D. Martinez: Two degrees of separation in complex food webs, PNAS vol. 99, no. 20, 12913-12916 (2002)

[17] H. Jeong, S.P. Mason, A.-L. Barabási, Z.N. Oltvai: Lethality and centrality in protein networks, Nature 411, 41-42 (2001)

[18] J.K. Pritchard, M. Stephens, P. Donnely: Inference of Population Structure Using Multilocus Genotype Data, Genetics 155, 945-959 (2000)

[19] R. Pastor-Satorras, A. Vázquez, A. Vespignani: Dynamical and Correlation Properties of the Internet, Phys. Rev. Lett. 87, 258701 (2001)

[20] G. Procaccini, L. Orsini, M.V. Ruggiero, M. Scardi: Spatial patterns of genetic diversity in Posidonia oceanica, and endemic Mediterranean seagrass, Molecular Ecology 10, 1413-1421 (2001)

[21] The photo is taken by Yorono and is distributed with the Creative Commons Attribution ShareAlike 3.0 licence http://creativecommons.org/licenses/by-sa/3.0/

[22] D.B. Goldstein, D.D. Pollock: Launching Microsatellites: A Review of Mutation Processes and Methods of Phylogenetic Inference, Journal of Heredity, 88:335-342 (1997)

[23] J.J. Doyle, J.L.I Doyle: A rapid DNA isolation procedure for small quantities of fresh leaf tissue, Phytochemistry Bulletin 11, 11-15 (1987)

[24] S. Arnaud-Haond, F. Alberto, S. Teixeira, G. Procaccini, E.A. Serrão, C.M Duarte: Assessing Genetic Diversity in Clonal Organisms: Low Diversity or Low Resolution? Combining Power and Cost Efficiency in Selecting Markers

[25] F. Alberto, L. Correia, S. Arnaud-Haond, C. Billot, C.M. Duarte, E. Serrão: New microsatellite markers for the endemic Mediterranean seagrass Posidonia oceanica, Molecular Ecology Notes 3, 253-255 (2003)

[26] G. Procaccini, L. Mazzella: Population genetic strcuture and gene flow in the seagrass Posidonia oceanica assessed using microsatellite analysis, Mar Ecol Prog Ser 169, 133-141 (1998)

[27] G. Procaccini, M.V. Ruggiero, L. Orsini: Genetic structure and distribution of microsatellite population genetic diversity in Posidonia oceanica in the Mediterranean basin, Bulleting of Marine Science 71, 1291-1297 (2002)

[28] P.H. Sneath, R.R. Sokal: Numerical taxonomy, W. H. Freeman, San Francisco (1973)

[29] E. Hernández-García, E.A. Herrada, A.F. Rozenfeld, C.J. Tessone, V.M. Eguíluz, C.M. Duarte, S. Arnaud-Haond, E. Serrão: Evolutionary and Ecological Trees and Networks, AIP Conf. Proc. 913, 78-83 (2007)

[30] C.E. Metz: Basic principles of ROC analysis, Semin Nucl Med. 8, 283-298 (1978)

[31] J.A. Hanley, B.J. McNeil: The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143, 29-36 (1982)

[32] K. Klemm: Entropies and mutual information in the Posidonia data set, Unpublished manuscript (2008)

[33] V. Kunin, L. Goldovsky, N. Darzentas and C.A. Ouzounis: The net of life: Reconstructing the microbial phylogenetic network, Genome Res. 15:954-959 (2005)

[34] E. Diaz-Almela, S. Arnaud-Haond, M.S. Vliet, E. Alvarez, N. Marba, C.M. Duarte, E.A. Serrão: Feed-backs between genetic structure and perturbation-driven decline in seagrass (Posidonia oceanica) meadows, Conservation Genetics 8:1377-1391 (2007)

[35] S. Arnaud-Haond, M. Migliaccio, E. Diaz-Almela, S. Teixeira, M.S. Vliet, F. Alberto, G. Procaccini, C.M. Duarte, E.A Serrão: Vicariance patterns in the Mediterranean Sea: east-west cleavage and low distpersal in the endemic seagrass Posidonia oceanica, Journal of Biogeography 34, pp. 963-976 (2007)

[36] A.F. Rozenfeld, S. Arnaud-Haond, E. Hernández-García, V.M. Eguíluz, E.A. Serrão, C.M. Duarte: Network analysis identifies weak and strong links in a metapopulation system, PNAS 105, 18824-18829 (2008)

[37] E. Hernández-García, A.F. Rozenfeld, V.M. Eguíluz, S. Arnaud-Haond, C.M. Duarte: Clone size distributions in networks of genetic similarity, Physica D 224, 166-173 (2006)

[38] M.E.J. Newman: Assortative Mixing in Networks, Phys. Rev. Lett. 89, 208701 (2002)

[39] Santo Fortunato, Claudio Castellano: Community Structure in Graphs, physics.soc-ph/0712.2716 (2007)

[40] M.E.J. Newman, M. Girvan: Finding and evaluating community structure in networks, Phys. Rev. E 69, 026113 (2004)

[41] D. Stauffer, A. Aharony: Introduction to Percolation Theory, 2nd Ed., CRC London (1994)

[42] G. Palla, I. Derényi, I. Farkas, T. Vicsek: Uncovering the overlapping community structure of complex networks in nature and society, Nature 435, 814-818 (2005)

[43] J.M. Kumpula, M. Kivelä, K. Kaski,and J. Saramäki: A sequential algorithm for fast clique percolation, Phys. Rev. E 79 (2008)

[44] I. Farkas, D. Abel, G. Palla, T. Vicsek: Weighted network modules, New J. Phys. 9 180 (2007)

[45] S. Fortunato, M. Barthélemy: Resolution limit in community detection, PNAS 104:36-41 (2007)

[46] J.M. Kumpula, J. Saramäki, K. Kaski, J. Kertész: Limited resolution in complex network community detection with Potts model approach, Eur. Phys. J. B 56, 41 (2007)

[47] M. Sales-Pardo, R. Guimerà, A.A. Moreira, L.A.N Amaral: Extracting the hierarchical organization of complex systems, PNAS vol. 104 no. 39 (2007)

[48] M.E.J. Newman: Analysis of weighted networks, Phys. Rev. E 70, 056131 (2004)

[49] N. Saitou, M. Nei: The neighbor-joining method: a new method for reconstructing phylogenetic trees, Molecular Biology and Evolution 4, 406-425 (1987)

[50] T.M. Cover, A. Joy: Elements of information theory, 2nd Ed., Wiley-Interscience NY (2006)

[51] A. Lancichinetti, S. Fortunato, J. Kertész: Detecting the overlapping and hierarchical community structure of complex networks, cond-mat/0808.1218 (2008)

[52] G. Schwarz: Estimating the dimension of a model, Annals of Statistics 6(2):461-464 (1978)

[53] P. Erdős, A. Rényi: On random graphs, Publ. Math. Debrecen 6, 290 (1959)

[54] T. Margush, F. R. McMorris: Consensus n-Trees, Bulletin of Mathematical Biology, Vol. 43(2), 239-244 (1981)

[55] J. Felsenstein: Confidence Limits on Phylogenies: An Approach Using the Bootstrap, Evolution 39(4), 783-791 (1985)

[56] D. Watts, S. Strogatz: Collective dynamics of small-world networks, Nature 393, 440-442 (1998)

[57] Sara Nadiv Soffer, Alexei Vázquez: Network clustering coefficient without degree-correlation biases, Phys. Rev. E 71, 057101 (2005)

[58] A. Barrat, M. Barthélemy, R. Pastor-Satorras, A. Vespignani: The architecture of complex weighed networks, PNAS 101, 3747 (2004)

[59] J.-P. Onnela, J. Saramäki, J. Kertész, K. Kaski: Intensity and coherence of motifs in weighted complex networks, Phys. Rev. E 71, 065103 (2005)

[60] J. Saramäki, J.-P. Onnela, J. Kertész, K. Kaski: Characterizing Motifs in Weighted Complex Networks, Science of Complex Networks - AIP Conference Proceedings 776, J.F.F. Mendes *et al.* (eds.) 108 (2005)

[61] P. Holme, S.M. Park, B.J. Kim, C.R. Edgling: Korean university life in a network perspective: Dynamics of a large affiliation network, Physica A 373, 821-830 (2007), cond-mat/04116334 (2004)

[62] P. Grindrod: Range-dependent random graphs and their application to modeling large small-world Proteome datasets, Phys. Rev. E 66, 066702 (2002)

[63] S.E. Ahnert, D. Garlaschelli, T.M. Fink, G. Caldarelli: An ensemble approach to the analysis of weighted networks, cond-mat/0604409 (2006)

[64] L. Lopez-Fernandez, G. Robles, J.M. Gonzalez-Barahona: Applying Social Network Analysis to the Information in CVS Repositories, doi:10.1049/ic:20040485

[65] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, J. Kertész: Generalizations of the clustering coefficient to weighted complex networks, Phys. Rev. E 75, 027105 (2007)

[66] B.A. Galler, M.J. Fischer: An improved equivalence algorithm, Communications of the ACM, Vol 7, Issue 5, pp.301-303 (1964)

[67] M. Fredman, M. Saks: The cell probe complexity of dynamic data structures. Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, pp. 345-354 (1989)

[68] I. Derényi, G. Palla, T. Vicsek: Clique Percolation in Random Networks, Phys.Rev. Lett. 94, 160202 (2005)

[69] V. Batagelj, A. Mrvar: Pajek - Analysis and Visualization of Large Networks, Graph Drawing Software, Springer, Berlin p. 77-103 (2003)

[70] J.G. Siek, L-Q. Lee, A. Lumsdaine: The Boost Graph Library: User Guide and Reference Manual, Addison-Wesley Professional (2001)

[71] https://networkx.lanl.gov

[72] G. van Rossum, F.L. Drake (editors): Python Reference Manual, PythonLabs, Virginia, USA, 2001. Available at http://www.python.org

[73] J. Hunter: matplotlib: plotting for Python, http://matplotlib.sf.net (2002)

[74] D. Ascher, P.F. Dubois, K. Hinsen, J. Hugunin, T. Oliphant, Numerical Python, Lawrence Livermore National Laboratory, Livermore, California, USA, 2001. Available at http://numpy.scipy.org/numpydoc/numdoc.htm

[75] F. Perez, B.E. Granger: IPython: A System for Interactive Scientific Computing, Computing in Science & Engineering Vol. 9, No. 3, 21-29 (2007)

[76] E. Jones, *et al.*: SciPy: Open Source Scientific tools for Python (2001)

[77] J. Hyvönen: An efficient library for simulating complex networks, Master's Thesis at Helsinki University of Technology (2006)

[78] D. Beazley: Simplified Wrapper and Interface Generator, http://www.swig.org/ (1995)