Imperial College London

Department of Mathematics

Local, multi-resolution detection of network communities by Markovian dynamics

Yun William Yu

March 2014

Supervised by Mauricio Barahona & Sophia Yaliraki

Submitted in part fulfilment of the requirements for the degree of
Master of Philosophy in Mathematics of Imperial College London

Declaration

I herewith certify that all material in this dissertation which is not my own work has been properly acknowledged.

Yun William Yu

Copyright

Acknowledgements

I would like to thank my supervisors, Sophia Yaliraki and Mauricio Barahona, for their immeasurable assistance through the entire process, and Jean-Charles Delvenne for insightful discussions and advice.

Also, I would like to thank all the members of both the Yaliraki and Barahona research groups, especially Antoine Delmotte and Michael Schaub, for useful comments and some of the example networks used.

The Imperial Marshall Scholarship, jointly funded by Her Majesty's Government and Imperial College, made this research possible.

Finally I would like to thank my parents for giving me the time and support to finish writing this thesis.

Abstract

Complex networks are used to represent systems from many disciplines, including biology, physics, medicine, engineering and the social sciences; Many real-world networks are organised into densely connected *communities*, whose composition gives some insight into the underlying network. Most approaches for finding such communities do so by partitioning the network into disjoint subsets, at the cost of requiring global information and that nodes belong to exactly one community. In recent years, some effort has been devoted towards the development of local methods, but these are either limited in resolution or ignore relevant network features such as directedness.

Here we show that introducing a dynamic process onto the network allows us to define a community quality function *severability* which is inherently multi-resolution, takes into account edge-weight and direction, can accommodate overlapping communities and orphan nodes and crucially does not require global knowledge. Both constructive and real-world examples—drawn from fields as diverse as image segmentation, metabolic networks and word association—are used to illustrate the characteristics of this approach. We envision this approach as a starting point for the future analysis of both evolving networks and networks too large to be readily analysed as a whole (e.g. the World Wide Web).

On more theoretical fronts, severability quantifies the relevance of a time scale separation of the dynamic process, allowing us to set apart the community at short times or aggregate it to a super node at large times. This also offers the potential for quantitatively exploring the underlying connections between network community detection, model reduction and diffusion-like processes.

To my family.

Contents

Acknowledgements **4**

1 Introduction **12**

1.1 Thesis structure 14

2 Literature review **16**

2.1 Mathematical notation 16

2.2 Hierarchical Clustering 17

2.2.1 Agglomerative methods 19

2.2.2 Divisive methods 20

2.3 Quality functions 24

2.3.1 Modularity 24

2.3.2 Spin glasses 25

2.3.3 Stability 26

2.3.4 Information theoretic 27

2.4 Local approaches 28

2.4.1 Local modularity variants 28

2.4.2 LFR fitness 30

2.4.3 k-clique percolation 31

2.5 Conclusion 32

3 Severability 33

3.1 Formal definition 35

3.2 Meaning 36

3.3 Relationship to existing methods in the literature 38

3.4 Optimisation procedure 41

3.5 Computational complexity 43

4 Constructed benchmarks 45

4.1 Hierarchical random graph 45

4.2 Variants of the LFR benchmark 47

4.2.1 Comparison methods 47

4.2.2 Unweighted, undirected, non-overlapping networks . . 50

4.2.3 Unweighted, undirected, overlapping networks 52

4.2.4 Weighted, directed, overlapping networks 53

4.3 Ring of rings 54

4.4 Square lattice 55

5 Real networks 57

5.1 Word association, overlap, & orphans 57

5.2 Metabolic networks 59

5.3 Image segmentation & commutativity 62

6 Future Work 65

6.1 Computational optimisation 65

6.1.1 Proof of concept 66

6.1.2 Numerical stability 68

6.1.3 Conclusion 69

6.2 Evolving and large networks 69

7 Conclusion **71**

List of Figures

2.1 Hierarchical clustering 18

3.1 Painting landscapes 34

3.2 Random walk history 35

3.3 Optimisation procedure 42

4.1 Hierarchical random graph 46

4.2 Unweighted, undirected, non-overlapping LFR benchmark . . 51

4.3 Unweighted, undirected, overlapping LFR benchmark 52

4.4 Weighted, directed, overlapping LFR benchmark 53

4.5 Ring of rings 54

4.6 Square lattice diffusion 56

5.1 Word association 58

5.2 Word association - a larger view 60

5.3 Citric acid cycle 61

5.4 Neuronal image segmentation 63

1 Introduction

Complex networks are used to represent systems from many disciplines, including biology, physics, medicine, engineering and the social sciences [65]. In the social sciences, one of the traditional bastions of network science, networks are used to represent people and the connections between them: friendships, romantic pairings, work colleagues, etc. [54, 72, 76]. In engineering, electrical wiring, on both macro and micro scales, from country-wide grids [3] to the components on individual chips [33], serves as another historically relevant example. More recently, this abstraction has also proven revealing in biochemistry, being used for kinetic transition networks [67], interaction networks [42] and structural networks [17]. By abstracting away the details of connections between members of a set into possibly weighted and/or directed edges, the same mathematical methods can be used to analyse networks generally.

It is widely accepted that one of the distinguishing characteristics of complex networks is community structure [54], where communities (also known as partitions, modules, or clusters) are sets of nodes with stronger internal than external links. Depending on the underlying network, these communities might map to groups of friends [76], genes and proteins involved in a particular function [75] or the pixels on an image representing an object [62]. Unfortunately, though intuitively appealing, there is no single mathematically precise definition of a community, and a whole host of detection methods have arisen, all with their own implicit assumptions about the nature of communities [12, 13, 18, 32, 36, 39, 43, 45, 48, 49, 52, 56, 57, 58, 62].

Traditional approaches partition networks into disjoint sets [18, 32, 36, 45, 48, 49, 56, 57, 58, 62], which works well when nodes must belong to exactly one community, such as in the partitioning of tasks to multiple nodes in a computing cluster for parallel processing [63]. However, in real networks, nodes will often belong to multiple or even no communities [39, 52]. For example, most people have both work and family social circles; hard partitioning excludes such overlap. Conversely, while many weblogs belong to "blog rings" sharing similar interests—e.g. cooking, mathematics—there are a significant number without such affiliations.

Furthermore, communities can not only overlap but be completely embedded within each other: all physicists are scientists, but not all scientists are physicists. To detect such embeddings, resolution parameters are needed [26]. Algorithms that are combinatorial in construction, based on counting and/or cutting links [12, 13, 28, 39, 43, 45, 48, 49, 52, 56, 62] do not naturally include such resolution parameters. These have often been added later, sometimes *ad-hoc* [39], sometimes through connections with other phenomena such as spin glasses [56] or Markov chains [36, 18].

Perhaps more importantly, methods that are global in scope, including spectral [32] and information theoretic [57, 58] algorithms, also result in an unintuitive artefact: whether a set of nodes is considered a good community is dependent upon the entire network in which it is embedded. This also results in practical problems where the entirety of a network is not known, as updating the network representation with more accurate information might completely change the results.

The role of community detection has increased greatly in prominence due to the massive data acquisition and proliferation caused by a wealth of recent technological advances. Its goal should be to give insight into such large but opaque and uninformative datasets which are not enlightening on their own. It is thus imperative to resolve the outstanding problems enumerated above. Here, we propose such a method that is local, multi-resolution and allows for both overlapping and orphan nodes, as well as directed and weighted links.

1.1 Thesis structure

This work has been organised around its main result, severability. The mathematical and historical context is first provided, the result is presented, and future implications are suggested.

Chapter 2 introduces the mathematical notation necessary and surveys the history and current status of community detection methods.

Chapter 3 introduces severability, a community quality function based on results from Markov chain quasi-stationarity. Furthermore, an optimisation procedure is presented that is well-suited for optimising local quality functions.

Chapter 4 assesses the performance of severability on appropriately chosen artifical benchmark graphs against existing popular methods.

Chapter 5 demonstrates some of the more pertinent properties of severability by application to several real-world networks, including word-association, image segmentation and a biochemical network.

Chapter 6 highlights some potential areas for future work, including the search for a fast method for determining the eigenvalues of an expanded matrix.

Chapter 7 concludes the thesis.

2 Literature review

There are far too many variations on community detection methods in the literature to describe in detail here [60]. In this chapter, we restrict ourselves to a subset of proposed methods, comprising those that have been key influences on the development of our method *severability* and/or popular in the field.

2.1 Mathematical notation

Formally, a mathematical graph G is defined as a set V of nodes (or vertices) together with another set E of links (or edges) between vertices. Let $n = |V|$ and $m = |E|$, the total number of nodes and vertices respectively. Depending on the context, links can be either directed (such as hyperlinks on the web) or undirected (such as collaboration networks). Additionally, weights can be assigned to links according to a connection strength.

The topology of a graph is encoded in the adjacency matrix A, where a_{ij} is the weight on the link from node i to node j. If there are multiple links, let a_{ij} be the sum of their weights. Define the out-degree d_i to be the sum of the weights of all links leaving node i. Then the out-degrees can be compiled in the vector $\boldsymbol{d} = A\mathbf{1}$, where $\mathbf{1}$ is the $n \times 1$ vector of ones. Define $D = \mathrm{diag}(\boldsymbol{d})$ to be the diagonal matrix of out-degrees.

The standard random walk on such a graph defines an associated Markov chain in which the probability of leaving a node is split amongst the outgoing links proportionally according to their weights, with a transition probability $\frac{a_{ij}}{d_i}$ for each link.

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t D^{-1} A \equiv \boldsymbol{x}_t P \tag{2.1}$$

where $\boldsymbol{x}_t$ is the $1 \times n$ normalised probability vector and P is the transition matrix.

Let $H \in M_{n \times c}(\mathbb{R})$, where c is the number of communities, be the partition indicator matrix such that

$$\begin{aligned} H_{ij} = 1 &\iff i \in C_j \\ H_{ij} = 0 &\iff i \notin C_j, \end{aligned}$$

where $C_j \subset V$ is a community. We use this notation only for hard partitions, so $C_i \cap C_j = \emptyset \iff i \neq j$. As there is an isomorphism between indicator matrices and partitions of a graph, it is possible without risk of confusion to refer to the *partition* H.

2.2 Hierarchical Clustering

One of the classical methods for finding groups in datasets is hierarchical clustering [70, 46]. A general method, hierarchical clustering requires only a dissimilarity measure between members of the set and thus is not restricted to network clustering. For networks, the dissimilarity measure is usually related to either the number of links or paths between clusters.

The method is based on building a dendrogram of clusters (figure 2.1). At the top is the entire graph, at the bottom the individual nodes, and in the intermediate slices, partitions consisting of clusters of varying sizes. As one

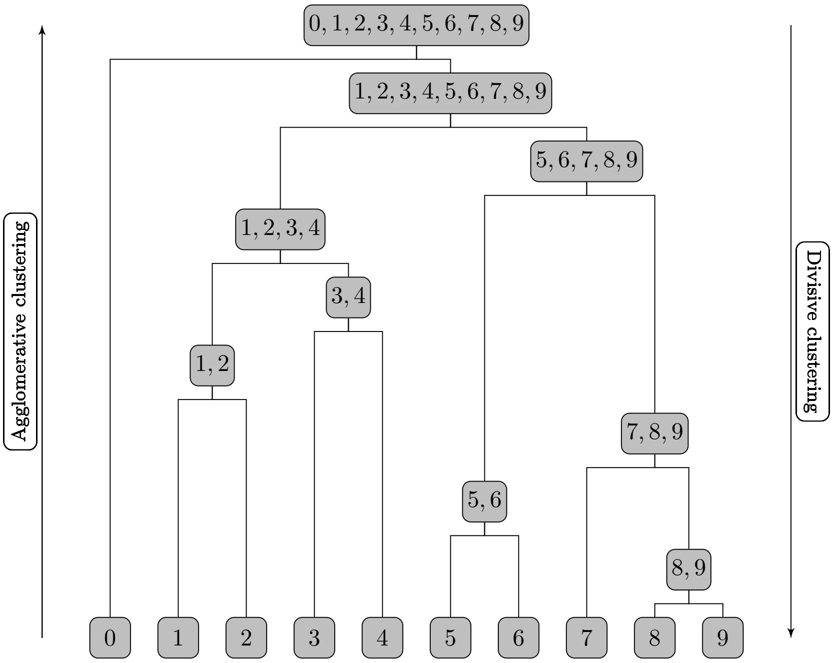


Figure 2.1: Hierarchical clustering can be depicted as a dendrogram. Agglomerative methods work by merging together smaller clusters (going up the dendrogram), while divisive methods split larger clusters (going down the dendrogram).

goes up the dendrogram, clusters merge together; as one goes down, clusters are divided into smaller ones. These two directions also correspond with the two types of hierarchical methods used, respectively the agglomerative and divisive methods.

2.2.1 Agglomerative methods

This is a family of methods that operates by iteratively forming clusters from nearest neighbours [46]. Using the original network, weights are assigned between every pair of nodes according to a measure of how dissimilar they are. Different measures of similarity between nodes include counting the total number of node-independent paths or total number of link-independent paths [72]. Choice of the measure is dependent both on computational considerations and on the nature of the network.

Initially, each node is assigned to its own cluster. At each step, the two "nearest" clusters are merged; the process stops when all nodes are assigned to a single cluster. Because the identity of the clusters change at every step, dissimilarity must be redefined; different algorithms use various approximations to minimise the computational cost [46]. Communities are defined as the clusters formed after t steps. Once nodes are grouped together in the same cluster, they remain so for the entirety of the method, resulting in a hierarchy of communities containing all the clusters that merged to form it. Similar techniques have been used in the chemical physics literature in the form of recursive regrouping to generate dendrograms of Markov states known as disconnectivity graphs [68].

One shortcoming of hierarchical clustering is its tendency to isolate nodes on the periphery. For instance, in variations that make use of counting paths, a node that is connected by only a single link to a cluster will remain isolated for a large number of agglomeration steps.

2.2.2 Divisive methods

Instead of iteratively merging smaller clusters together, divisive methods start with all nodes grouped together and iteratively separate them. Sometimes, this is done by removing links one-by-one according to some heuristic. For instance, the edge betweenness algorithm first lists all the shortest paths between all pairs of nodes, and then removes the links which are included in the most number of shortest paths [28]. As links between communities serve as the bridges between nodes in the respective communities, a higher number of shorter paths run along them; thus, they are cut away first.

More often though, divisions are made by optimising for some global partition penalty function. For hierarchical clustering, even when using partition penalty functions—which often permit k-way divisions for arbitrary k—each step generally consists of finding only the best 2-way partition. This has the advantage of being comparatively easy to optimise through spectral methods [61]. As penalty functions share many similarities with severability, our community quality function, we go into some detail on the construction of several graph partition penalty functions.

Minimum Cut

Possibly the simplest possible penalty function is one summing together the total weight of links cut [33]. The minimum cut

$$cut(H) = ||H^T AH||_1 - \text{trace}(H^T AH),$$

using the entry-wise matrix 1-norm. The number of clusters k must be specified *a priori*—otherwise, $k = 1$ trivially minimises the number of links cut. Note that this is equivalent to finding a permutation matrix P such that PAP^T is almost block diagonal.

The minimum cut criterion works well for the problem of partitioning graphs into k clusters of at most size c, such as appears in the division of tasks for parallel processing or the construction of circuit boards. However, in the problem of community detection, one often does not have prior knowledge of either the number of clusters or their respective sizes. Furthermore, this method can suffer from the same shortcoming as the agglomerative hierarchical clustering, in that it can isolate small groups nodes on the periphery. This is a ready consequence of the fact that assigning small groups of peripheral nodes to a community results in fewer links being cut [73].

For large graphs, minimising the penalty function through such methods as Kernighan-Lin switches [33] can be extremely computationally intensive. However, more efficient algorithms have been developed, including a recursive spectral method based on solving a generalised eigenvalue problem [69], which leads to hierarchical clustering.

Ratio Cut

To resolve the issue of isolating peripheral nodes, ratio cut was proposed:

$$RCut(H) = \frac{cut(H)}{|C_1|} + \frac{cut(H)}{|C_2|}.$$

where C_1, C_2 are the two clusters [71]. By normalising the cutting penalty against the size of each cluster, unbalanced partitions, wherein communities are of extremely different size, are selected against.

A spectral method for bisecting the network is offered through the laplacian matrix

$$L = D - A.$$

If the eigenvector corresponding to the 2nd smallest eigenvalue of L (also known as the Fiedler eigenvector [25]) is small, then bisecting the network according to the sign of each entry of the eigenvector will result in a small cut [1]. This provides the basis for a hierarchical partitioning using ratio cut through iterative 2-way partitions.

Normalised Cut

Another attempt to resolve the tendency of minimum cut to isolate peripheral nodes can be achieved by normalising the cutting penalty against the total link weight from each community, rather than the community size. This provides the basis of the normalised cut [62]. Let H be a bisection such that

$$H^T A H = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix},$$

so $cut(H) = L_{12} + L_{21}$. Then the normalised cut is

$$NCut(H) = \frac{cut(H)}{L_{11} + L_{12}} + \frac{cut(H)}{L_{21} + L_{22}}.$$

Although isolating small groups of peripheral nodes may cut fewer links, those links make up for a larger proportion of their total, and so the normalised cut penalty even more severely penalises unbalanced partitions than does ratio cut.

Normalised cut also has a spectral approximation. Each of the columns of the partition indicator matrix H can be approximated by the generalised eigenvector solutions q of

$$(D - A)q = \lambda Dq.$$

Note that the clusters identified are generally not exactly correct because of the relaxation necessary in transitioning to the continuous domain of real eigenvectors.

MinMax Cut

MinMax cut normalises the cutting penalty against the internal connectivity of each cluster [19]. Using the same notation as above,

$$MinMaxCut(H) = \frac{cut(H)}{L_{11}} + \frac{cut(H)}{L_{22}},$$

which can be rewritten

$$MinMaxCut(H) = \frac{x^T(D - A)x}{x^T Ax} + \frac{y^T(D - A)y}{y^T Ay}.$$

This turns out to provide even more balanced cluster sizes than ratio or normalised cut [50].

As with normalised cut, the usual method employs spectral relaxation and finding the generalised eigenvector solutions q where

$$(D - A)q = \lambda Dq.$$

Again, the solutions are not exact, and often, further refinements like k-means are needed in post-processing to find more optimal partitions; more recent modifications of MinMax cut provide better resolutions to this problem [50].

2.3 Quality functions

The distinction we draw between divisive hierarchical partition penalty functions and partition quality functions is highly artificial, lying only in the way they are applied. The penalty functions considered in the previous section are intimately tied to their respective optimisation methods, generally being used for recursive bisection to generate a dendrogram of communities. On the other hand, the quality functions considered in this section are in practice more separated from the heuristics used to optimise them.

Although both are functions from the space of possible partition matrices $\mathfrak{H}$ to $\mathbb{R}$, the separation of optimisation procedures from quality of partition marks an important paradigm shift. As a gross oversimplification, scientists no longer simply try to find partitions, but are more interested in evaluating how good those partitions are.

2.3.1 Modularity

One extremely popular alternative to purely cut-based approaches is modularity [48]. It's computed by taking the number of intra-community links and subtracting the expected number of intra-community links, given an appropriate model of a random graph.

More precisely, for a partition into c communities,

$$modularity = \frac{1}{2m} \sum_{l=1}^{c} \sum_{i,j \in C_l} \left(A_{ij} - \frac{k_i k_j}{2m} \right).$$

The A_{ij} term gives the actual number of intra-community links, while $\frac{k_i k_j}{2m}$

takes into account the expected number of links. That is to say, in this case, the expected number of links is proportional to the degree of the node, as would be the case in a random graph (under the configuration model of a random network).

Although modularity was originally employed through recursive bisection, as in the divisive hierarchical algorithms, it eventually grew to take on a life of its own as primarily a quality function for which many varied optimisation methods exist [15]. Unlike hierarchical clustering, using modularity to detect communities gives only a single partition, as opposed to an entire hierarchy of partitions—this is true even when using recursive bisections, as modularity gives reason to choose one particular slice of the hierarchy over all others. Although this is not necessarily a shortcoming of the method, it does result in modularity only being able to detect communities on certain scales [26]. When the most natural community size is below the resolution scale, modularity does not give the correct answer. Furthermore, in the case of naturally hierarchical graphs, wherein communities are obviously embedded within one another, modularity can only give a single slice.

2.3.2 Spin glasses

One class of attempts to resolve the resolution scale limitations of modularity comes from connections to statistical physics by viewing the clustering problem as the assignment of spins in the ground state of a spin glass [4, 56]. We focus here on the Pott's Hamiltonian of Reichardt and Bornholdt [56], given by

$$\mathcal{H}(\sigma) = -\sum_{i \neq j} (A_{ij} - \gamma p_{ij}) \delta(\sigma_i, \sigma_j),$$

where σ is a vector specifying the spin states of each node, p_{ij} is the expected probability of a link between nodes, and γ is a scaling factor determining the ratio of energy that can be contributed by links vs non-links.

Minimising the energy of the system takes the place of optimising for a quality function. Additionally, by varying γ, one is able to recover communities at different resolutions. Furthermore, when the expected probability of links between nodes $p_{ij} = k_i k_j / 2m$ and the scaling factor $\gamma = 1$, it is straight-forward to verify that the Hamiltonian becomes a constant multiple of modularity.

2.3.3 Stability

Stability is based on analysing the behaviour of a Markov process on the graph [18, 36]. Although theoretically, any choice of a Markov process could be used, the most natural choice is the standard random walk, with the nodes corresponding to Markov states and the probability of transition directly proportional to the weight of the out-links from nodes. Roughly, stability measures the probabilities for random walkers to remain in their starting communities.

Given the clustered auto-covariance matrix

$$R(t, H) = H^T(\Pi M^t - \boldsymbol{\pi}^T \boldsymbol{\pi})H, \tag{2.2}$$

$R(t, H)_{ij}$ is the probability that a random walker originating from community C_i will still be in C_j at time t, minus the contribution towards that probability arising solely from the respective vertex degrees.

However, that includes the probability that a random walker will leave and later return to the community. To account for that, stability is defined by

$$r(t; H) = \min_{0 \le s \le t} \mathrm{trace}[R(s, H)]. \tag{2.3}$$

Thus formulated, Markov time plays the role of the resolution parameter, allowing for discovery of communities at multiple scales.

Although it is not entirely obvious, stability turns out to have deep connections with several of the clustering methods presented above. At a Markov time $t = 0$, stability corresponds to normalised cut, at $t = 1$ to modularity under an appropriate null model, and at $t \to \infty$ to spectral clustering [18]. Furthermore, the linearisation of stability from $0 < t < 1$ happens to be equivalent to the Pott's Hamiltonian model [36].

2.3.4 Information theoretic

Another set of approaches proposed by Rosvall and Bergstrom utilises an information-theoretic framework for community detection [57, 58, 59]. We do not go into as much detail here because they are not as closely related to severability as many of the previous methods; however, this approach has been extremely successful at certain benchmarks [40].

Infomod views partition assignment through the lenses of signal transmission [57]. Say Alice has full information about a graph G and wants to inform Bob. However, there is limited transmission bandwidth, so Alice instead only transmits the partition and Bob tries to reconstruct G from the partition. Infomod's quality function measures how similar the reconstruction is to the original graph.

Infomap follows a similar scheme, but introduces random walk dynamics [58]. Alice is attempting to use a partition assignment to optimally compress the path-history of the infinite-length random walk, which is then transmitted to Bob. Instead of sending the path history as a list of nodes traversed, compression is achieved by readdressing the nodes according to community assignment using a prefix/postfix pattern. Then, if a random walker moves from one node within a community to another within the same community, only the postfix needs to be transmitted. Compression is thus optimal when the inter-community hops are rare.

Furthermore, to better understand hierarchical communities, infomap can be modified to produce multilevel partition trees [59]. Instead of using the 2-level prefix/postfix pattern, even more optimal compression of a random walker's infinite path history can often be achieved by subdividing communities.

2.4 Local approaches

One of the common features of all of the partitioning methods reviewed above is the lack of overlapping communities. Although it is possible to generalise some methods to permit "fuzzy" partitions [74], most recent attempts instead centre around local algorithms that do not depend on global knowledge. This approach has the advantage of naturally permitting both overlapping communities and communities of extremely varying size (including "orphan" nodes).

2.4.1 Local modularity variants

Given modularity's popularity as a community detection method, it is unsurprising that there have been many attempts to define a local variant. Let us define for sets $X \subset Y \subset V$

$$\mathcal{A}(X,Y) = \frac{1}{2}\left[\sum_{i\in X}\sum_{j\in Y}(A_{ij} + A_{ji}) - \sum_{i,j\in X} A_{ij}\right],$$

the sum over all edges in Y with at least one endpoint in X.

Now consider a community of nodes $C \subset V$ with boundary nodes $B \subset C$. Clauset's local modularity measure R [13] can be defined by

$$R(C) = \frac{\mathcal{A}(B,C)}{\mathcal{A}(B,V)},$$

the ratio of the sum over edges in C with at least one endpoint in B and the sum over all edges with at least one endpoint in B.

Similarly, Luo, et al, proposed a modularity M [43] defined by the ratio of the sum over edges in C to the sum over edges with exactly one endpoint in C. This can be expressed using the above notation as

$$M(C) = \frac{\mathcal{A}(C,C)}{\mathcal{A}(C,V) - \mathcal{A}(C,C)},$$

Unsurprisingly, it can be related to R: in the special case where $C = B$, the thresholds $M \geq 1$ and $R \geq 0.5$ are equivalent [12].

Chen, et al, proposed a similar local community metric L [12] defined by the ratio of the average number of neighbours each node in C has in C to the average number of neighbours each node in B has outside of C. Translated into a slightly different form, that is equivalent to

$$L(C) = \frac{2\mathcal{A}(C,C)/|C|}{(\mathcal{A}(B,V) - \mathcal{A}(B,C))/|B|}.$$

Unfortunately, all of these modularity variants suffer from a shared shortcoming that makes them unsuitable to use as community quality functions. Taking C to be the entire network is either always a global maximum (though use of the extended real line is needed to make the latter two functions well-defined when resulting in ∞). Much like the divisive hierarchical partitioning algorithms (whose penalty functions were minimised when all nodes belonged to a single cluster), making use of these functions requires specifying *a priori* the expected size of a community. The optimisation routine becomes as important as the quality function, so the latter cannot be an objective metric.

Furthermore, as with modularity, none of these local variants includes a resolution parameter of any sort. Thus, even when the expected community sizes are specified, these functions are unable to adequately probe at certain scales or reveal multilevel community information.

2.4.2 LFR fitness

One attempt to add a resolution parameter to these local modularity variants was proposed by Lancichinetti, et al [39] in the form of the following fitness function (using previous notation):

$$f_C = \frac{2\mathcal{A}(C,C)}{(\mathcal{A}(C,V) + \mathcal{A}(C,C))^{\alpha}}$$

The α factor allows tuning the method for different size communities. For $\alpha \leq 1$ the global maximum for the fitness function is always the entire network, but for $\alpha > 1$ that is not necessarily true (despite the original paper claiming it to be so [39]).

However, as presented, the fitness function is still intimately connected with the optimisation procedure and is not used as true fitness function.

In their agglomeration method for finding communities, instead of trying to find the global maximum (which is in many cases the entire network), Lancichinetti, et al, instead stop the process once the first local maximum is found [39].

Furthermore, the internal structure of the community is not taken into account by the fitness function beyond summing together all the edges. This causes the unintuitive result that even disconnected sets of nodes can score high on the fitness function, which is only prevented by their particular optimisation procedure.

Lastly, although α mostly works as a resolution parameter, it is a purely *ad hoc* construction, without any deeper theoretical justification like connections to Markov processes or spin glasses. For these reasons, although the motivations and ideas behind the construction of the LFR fitness function were laudable, there is still much room for a better local community quality function.

2.4.3 k-clique percolation

Of course, community quality functions are not entirely necessary for a local approach to work. k-clique percolation, as introduced by Palla and Vicsek, is one such example [52]. Given a network, k-cliques are defined as complete sub-graphs with k nodes. Further, define k-cliques to be adjacent if they share $k-1$ nodes. Then, a community can be defined as a set of nodes that can be fully reached by following adjacent k-cliques [52]. This approach does not suffer from the problem of the entire network always being the optimal solution, takes into account the internal structure of the community (as it must) and has an obvious, easily adjustable resolution parameter k.

Unfortunately, k-clique percolation does not readily lend itself to directed and/or weighted graphs, due to the very definition of a k-clique. In the case of weight, the naive approach is to use a cut-off to derive a related unweighted graph. More complex approaches define an intensity measure for k-cliques, using some of the weight information, but still requiring a hard cut-off, this time for intensity [51]. Similarly, a directed variant of k-clique percolation has also been proposed [53]. Although interesting for preserving a directionality of sorts even in the clique, it does not generalise well for weighted directed graphs.

2.5 Conclusion

As has been highlighted throughout this chapter, there are a plethora of different community detection algorithms, each with particular advantages and disadvantages. However, there exist no true community quality functions that are local in scope, permit discovery of communities at varying resolution scales and naturally handle link weight and directionality. It is to fill this gap that we introduce *severability* in the next chapter.

3 Severability

To introduce our method, we draw an analogy from energy landscapes which are often used to represent complex systems. A landscape can be imposed onto any network by introducing a dynamic process on it, for example, the standard random walk on a graph. Unlike the more familiar energy landscapes, there is no downward arrow pointing to a minimum energy state; however, the notions of barriers and roughness translate over to communities: barrier "heights" are inversely related to inter-community connection strength, while "roughness" is inversely related to intra-community connection strength. This is illustrated through a 3D representation of flat images, where barriers and roughness are based on differences in luminosity of adjacent pixels (figure 3.1).

We propose to define a community as *severable* if it has both high barriers and low roughness, as extracted by the behaviour of the random walkers on the underlying landscape. To do so, we borrow the concepts of mixing and retention from Markov chain quasi-stationarity [16]. Given random walkers restricted to a community C, C is weakly mixing if there is a strong correlation between the walkers' positions at Markov times 0 and t. As the exploration of C is also hindered by roughness, mixing is inversely related to the roughness (see figure 3.2). On the other hand, retention is directly related to the height of the barriers, so random walkers tend to stay within

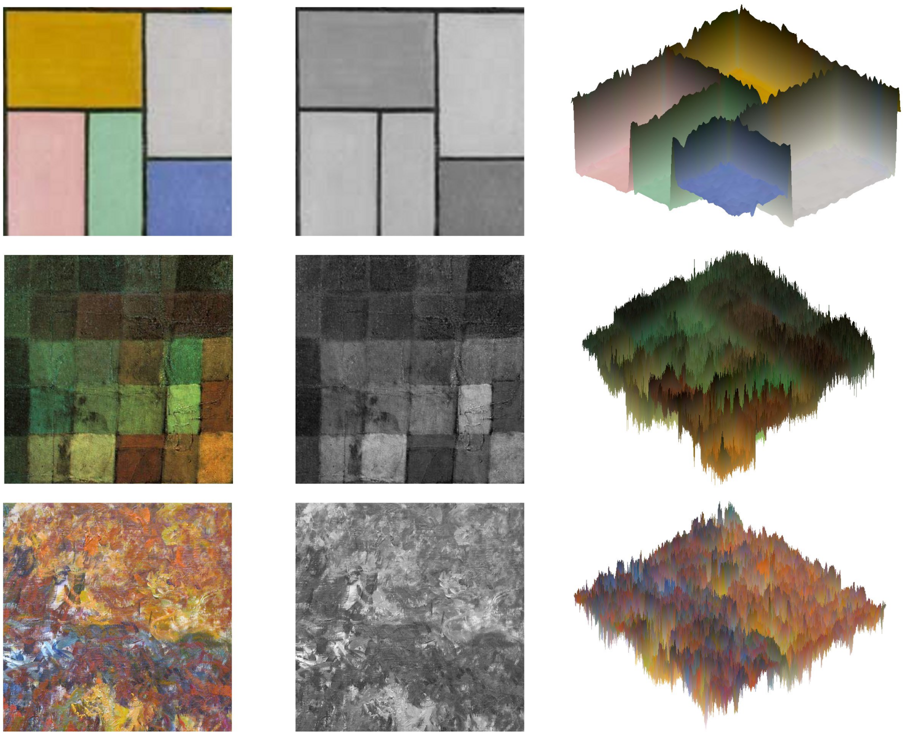


Figure 3.1: Small excerpts from Theo van Doesburg's *Composition in dissonances* (top), Paul Klee's *Ancient Sound* (middle) and Claude Monet's *the Japanese Footbridge* (bottom), their luminosity levels (grayscale) and the resultant landscapes. Top and bottom are extreme cases, respectively with either obvious or nonexistent communities. In middle, although there is significant internal roughness, slightly higher barriers exist.

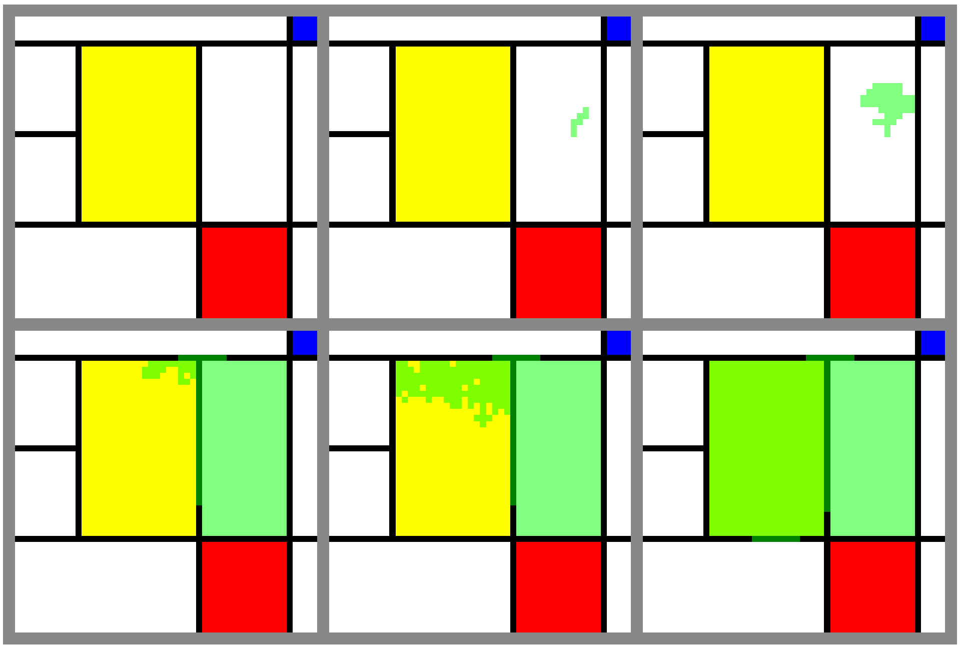


Figure 3.2: The path history of one particular random walker highlighted in green. Note that the regions of solid colour (low roughness) get explored before crossing the colour-shift barriers. Thus, as is natural, the solid colour regions (except for the black borders) correspond to communities, but not any other arbitrary areas.

C if it is hard for them to escape.

3.1 Formal definition

Given a connected subset $C \subset V$, we consider the behaviour of the random walk on C. Let Q be the sub-matrix of P corresponding to nodes in C, and let $k = |C|$, the size of the community. Then we define the retention $\rho(Q,t)$, the probability for a random walker starting with a uniform probability distribution in C not to have escaped by time t as

$$\rho(Q,t) = \frac{1}{k}\left(\mathbf{1}^{\mathrm{T}} Q^{t} \mathbf{1}\right). \tag{3.1}$$

To define mixing, first let $q_i^{(t)}$ be the ith row of the matrix Q^t. Because C is connected, $q_i^{(t)} \neq \mathbf{0}$. Thus $\frac{q_i^{(t)}}{q_i^{(t)}\mathbf{1}}$, a unit-normalised row of Q, is the probability distribution at time t for a random walker starting from node i, conditional upon the walker remaining in C. We can then define the internal mixing $\mu(Q,t)$ as

$$\mu(Q,t) = 1 - \frac{1}{k}\sum_{i=1}^{k} \|\bar{q} - \frac{q_i^{(t)}}{q_i^{(t)}\mathbf{1}}\|_{TV}, \tag{3.2}$$

where $\bar{q}$ is the arithmetic mean over the unit-normalised rows of Q^t. We have used the fact that the total variation distance norm is given by

$$\delta_{TV}(\boldsymbol{v}) = \frac{1}{2}\sum_i |v_i|. \tag{3.3}$$

Both ρ and μ are defined to range from 0 to 1, where the a value of 1 corresponds to perfect retention or mixing, respectively. We can now define the community quality function *severability* by

$$\sigma(Q,t) = \frac{\rho(Q,t) + \mu(Q,t)}{2}. \tag{3.4}$$

Severability has the intrinsic resolution parameter of Markov time; as t increases, the random walker will diffuse to larger parts of the graph. Additionally, since it depends only upon the out-links from nodes within C, it is a purely local function.

3.2 Meaning

For time t, the retention of a community is a measure of the error committed in assuming that the community is disconnected from the rest of the network, i.e. has a zero escape probability. The internal mixing on the other hand indicates how close the distribution within the community is from the quasi stationary distribution—for instance, in the context of energy landscapes and discrete path sampling, mixing is equivalent to the local equilibrium condition. Therefore, severability at time t is close to one if the dynamics of the random walker starting from a node in the community are equally unaffected by aggregation of all states in the community for times larger than t and disconnecting the community for times smaller than t. Moreover, for a given community, the severability will reach a maximum at a time which can be interpreted as the best separator between the short run and long run regimes.

This is related to the the existence of time scale separation in random walker dynamics. Given a partition of the states in a linear system, if inter-community transition probabilities are sufficiently low, then at short times, the Markov chain can be approximated by a description of its dynamics on completely disconnected communities. Conversely, because at high times the random walkers approach the quasi-stationary distribution within communities, the dynamics of the entire system can be accurately described as an aggregated Markov chain with every community collapsed into a single state [64, 2, 14].

Severability provides a way of quantifying the error committed in collapsing a set of states, without needing to know the entire system. Being a local approach allows in principle the detection of all overlapping or disjoint communities, each with their own time scale. On the other hand, the previous approaches cited require a strict partition into communities, all with the same characteristic time scale. Such a uniformity has little reason to emerge naturally in large complex, heterogeneous networks, and can only be obtained through the artificial grouping, splitting or trimming of natural communities.

3.3 Relationship to existing methods in the literature

In some ways, severability can be thought of as a local extension of stability [18] in that both objective functions rely on considering the dynamics of random walkers on a graph. However, it should be noted that the clustered autocovariance matrix for stability only has an analogue to the retention term and does not directly invoke the idea of mixing.

Indeed, the use of retention only for community detection has before been explored. The local modularity functions explored in the literature review can be thought of as measuring retention for a single Markov step under different initial distributions and renormalisations of random walkers. Further, in an approximate sense, LFR fitness' tuning factor also corresponds in some sense to retention at different (fractional) Markov times. As a more direct example, although not cast as a community detection method, the Markov Chain literature gives an example of a first passage time approach to create macrostates for more efficient MCMC [7], which also is equivalent to measuring retention.

However, as alluded to earlier, using only retention results in unintuitive pathological cases. First, retention-only approaches to creating a quality function will of necessity give a perfect score to considering the entire network as a community. Obviously, random walkers cannot escape from the network by definition. Thus, in all of the retention only local methods, a size limitation in the optimisation method is specified to prevent returning the entire network as an/the optimal community. For stability, this is not a problem because the stationary distribution is subtracted out. Second, retention-only approaches do not see internal structure. If a community is disconnected from the rest of the network, it will have perfect retention even if it is itself not connected. Although in the extreme, this pathological case tends to be avoided by the choice of optimisation procedure, this means that the objective function can not be relied upon without tying it to the optimisation. Furthermore, in almost pathological cases, perhaps a community consisting of two large cliques connected by a single link, retention-only approaches will not (as intuition suggests) favour the individual cliques over the combined double clique. Global approaches like stability avoid this because the entire network must be considered as a whole, and breaking up a two-clique community is favoured by the subtraction of the stationary distribution.

It is in order to avoid the problems of the previous paragraph that mixing must be introduced. Requiring the convergence to the quasi-stationary distribution has been implicitly used before, also in the Markov Chain literature in the form of lumping analysis [41]. Indeed, the distance from the quasi-stationary distribution is exactly a measure of how accurately a single macrostate can be used to replace a collection of other states in a Markov chain. However, the method of Lempesis, et al. method operates on the entire network and measures the accuracy of the lumping at a global level. We are able to resolve this issue locally by our variance-like mixing term, which uses both "local in time" and "local in space" information from the transition submatrix power.

Another connection was hinted at previously in the analogy drawn to energy landscapes. Previous work has shown that for protein folding, techniques such as transition path sampling can be used to generate a Markov chain of low-energy states connected by high-energy transition states [24, 9, 10, 11, 66]. Since in such systems, each state is associated with an actual energy level, and transitions that decrease energy level are favoured, one clever way of partitioning the state space into potential wells is to look at the disconnectivity graph. For any particular energy threshold, low-energy states are connected if the transition states between them (the barrier between them) have energy below that threshold, creating a rapidly mixing community of states. Indeed, here, energy plays the same role as time in severability, whereby the communities found by the disconnectivity graph are exactly those that rapidly interconvert with low probability of jumping to disconnected states for which the barriers are too high. In this way, both retention and mixing are captured by energy.

However, although we use the language of energy landscapes to motivate the definition of severability, it is crucial to clarify that there is no requirement for networks to correspond to actual energy landscapes. It is for this reason that instead of looking at energies, we instead look directly to the dynamic behaviour of random walkers on the graph. By directly measuring mixing and retention of random walkers, we are able to probe the same sorts of phenomenon in the more general case, at a cost of some additional complexity. Even when actual energy landscapes are involved, severability can still be applied since all it requires is the transition submatrix. Indeed, should the transition matrix be given for states on an energy landscape, severability of a community can be computed without severability having to "know" that the system is indeed an energy landscape.

A more subtle difference has to do with the nature of severability's locality. Although the disconnectivity graph can be almost completely charactised by the energy levels of the states and transition states between them, which allows for local determinations of energy wells, it still assumes the global structure of an energy landscape. This shows itself in the disconnectivity graph being a proper partition at any level and exhibiting the structure of a dendrogram. As we will later see, severability permits overlapping communities, whereas disconnectivity graphs do not. In the application of protein transition pathways, we probably do not want overlapping communities, so severability is slightly worse in that setting; however, in many other settings, it is natural to think of communities overlapping, making methods like severability more applicable.

3.4 Optimisation procedure

We apply a semi-greedy search algorithm to find the optimal community C for a starting node n_0, at a chosen Markov time t and search size S (see figure 3.3). Without loss of generality, define $\sigma(C) = \sigma(Q, t)$. Initially, only $n_0 \in C$. Aggregate nodes greedily, except let every third step be a Kernighan-Lin switch of a single node on the boundary of C to maximise $\sigma(C)$ [33]. After the initial semi-greedy optimisation, the intermediate community $\mathfrak{C}$ that has maximal severability is fine-tuned using Kernighan-Lin switches to find a local maximum. If n_0 is in the resulting community, then done; otherwise, start over, but choose a neighbour that was not chosen previously for the first step. If all neighbours of n_0 have been attempted without success, declare n_0 an orphan (i.e. a community of size one).

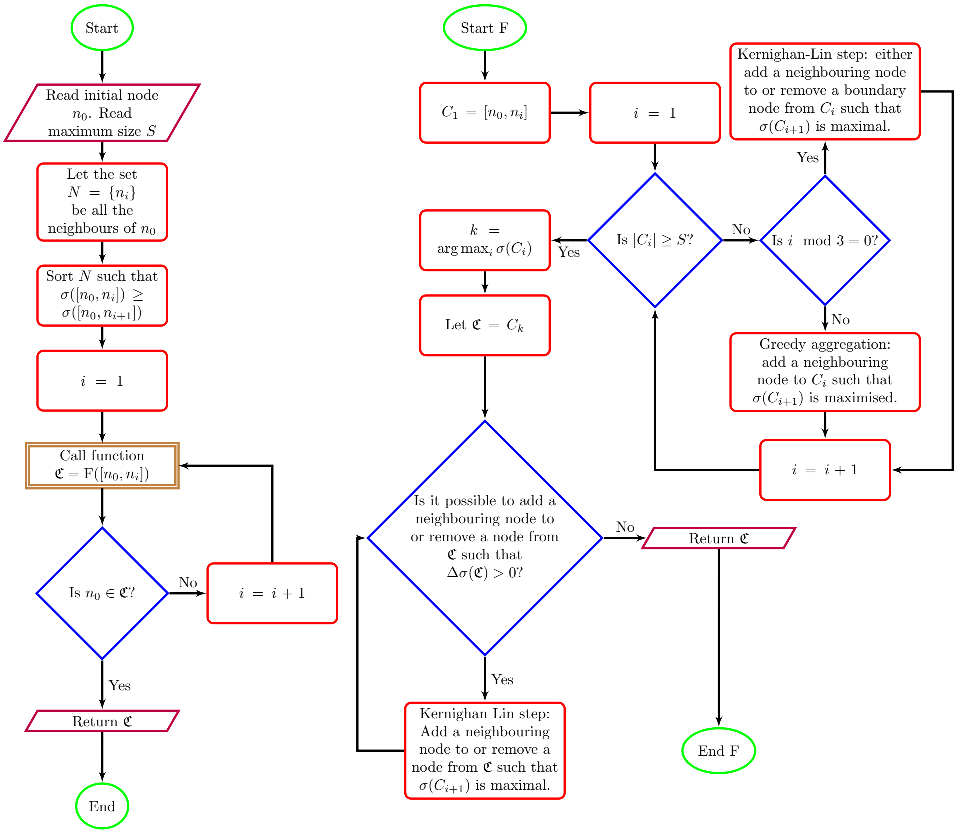

Start
Read initial node n_0. Read maximum size S
Let the set $N = \{n_i\}$ be all the neighbours of n_0
Sort N such that $\sigma([n_0, n_i]) \geq \sigma([n_0, n_{i+1}])$
$i = 1$
Call function $\mathfrak{C} = \mathrm{F}([n_0, n_i])$
Is $n_0 \in \mathfrak{C}$?
No
$i = i + 1$
Yes
Return $\mathfrak{C}$
End
Start F
$C_1 = [n_0, n_i]$
$i = 1$
Kernighan-Lin step: either add a neighbouring node to or remove a boundary node from C_i such that $\sigma(C_{i+1})$ is maximal.
Yes
$k = \arg\max_i \sigma(C_i)$
Yes
Is $|C_i| \geq S$?
No
Is $i \mod 3 = 0$?
No
Greedy aggregation: add a neighbouring node to C_i such that $\sigma(C_{i+1})$ is maximised.
Let $\mathfrak{C} = C_k$
$i = i + 1$
Is it possible to add a neighbouring node to or remove a node from $\mathfrak{C}$ such that $\Delta\sigma(\mathfrak{C}) > 0$?
No
Return $\mathfrak{C}$
Yes
Kernighan Lin step: Add a neighbouring node to or remove a node from $\mathfrak{C}$ such that $\sigma(C_{i+1})$ is maximal.
End F

Figure 3.3: Flowchart of the optimisation procedure for finding the most severable community to which a node belongs. Note that in the diagram, the Markov time t is assumed to be constant.

3.5 Computational complexity

Let n be the number of nodes in a graph. The severability of a community C of size k for a Markov time t can be computed in $P(k,t) = O(k^3 \log_2 t)$ time, where the cubic term comes from schoolbook matrix multiplication. Computation of mixing and retention given Q^t are both $\mathrm{O}(k^2)$ operations, so the total cost is dominated by matrix exponentiation.

The cost can be reduced using fast matrix multiplications techniques; e.g., using Strassen's method, the total cost would only be $O(k^{2.807} \log_2 t)$. Alternately, for large t, matrix diagonalisation can be first employed, which makes the t term negligible, giving a $O(k^3)$ solution, but with additional lower-order cost. As this paper was largely a proof of principle, we have not fully explored computational optimisation.

However, finding good communities is more involved than simply computing the severability of a single set of nodes. The community optimisation algorithm described in figure 3.3 is also costly, and more difficult to characterise, as it depends strongly on the number of nodes neighbouring the putative community throughout the procedure. In pathological cases, cost is $O(nM \cdot P(M,t))$, where M is the maximum number of nodes permitted in the community. Luckily, this upper bound only occurs in complete graphs and is largely irrelevant as most real networks are far sparser. However, by specifying M, one can choose the maximal computational resources one wants to spend trying to find a community.

As before, this procedure has not been optimised for time, and there are several obvious ways of reducing cost. Besides the computation of severability itself, the primary contributor to the cost is that all "neighbouring" communities' severabilities are checked as part of the aggregation process and subsequent Kernighan-Lin switches. A smarter algorithm could perhaps use a random walk to highlight likely candidate neighbours; for instance, by choosing only the l nodes that a random walker uniformly distributed in C would most likely walk to in the next step, or for removal of nodes, the l nodes in C that have the least density of probability. Such an algorithm would only cost $O(M \cdot P(M,t))$, a significant improvement.

More subtly, the computational cost of the matrix powers might also be reduced, by taking advantage of the fact that $Q(C^*)$ for each of the neighbouring communities is effectively a rank-2 perturbation of $Q(C)$. Furthermore, as briefly mentioned in the discussion, severability is only one way of quantifying the mixing and retention of random walkers. Other alternate methods may be found that are quicker, perhaps using Monte Carlo simulations and the like.

4 Constructed benchmarks

To illustrate this mathematical formulation, we first apply severability to constructed benchmarks with known, pre-seeded community structure. Naturally, it is unavoidable that every benchmark comes with its own set of implicit definitions about the definition of a community. A community detection method whose design follows the same definition, whether intentionally or not, will generally perform best on that particular benchmark.

Thus, it is important to evaluate not only how well a community detection method performs on a benchmark, but whether that benchmark matches our intuition of what a community ought to be. We have attempted to use a range of very different benchmarks, but it is important to note that because the implicit definition of a community can vary, it is highly unlikely that one single method will be optimal for every problem.

4.1 Hierarchical random graph

We optimise for severability on a hierarchical random graph with 256 nodes. Level 1 (L1) communities are composed of 16 nodes each, level 2 (L2) communities contain 64 nodes (four L1 communities) while the entire graph (L3) is made up of four L2 communities. The probabilities that pairs of nodes are connected when they are in the same level 1, 2, or 3 community, but not in the same lower-level community are $p_1 = 0.8452$, $p_2 = 0.0549$ and

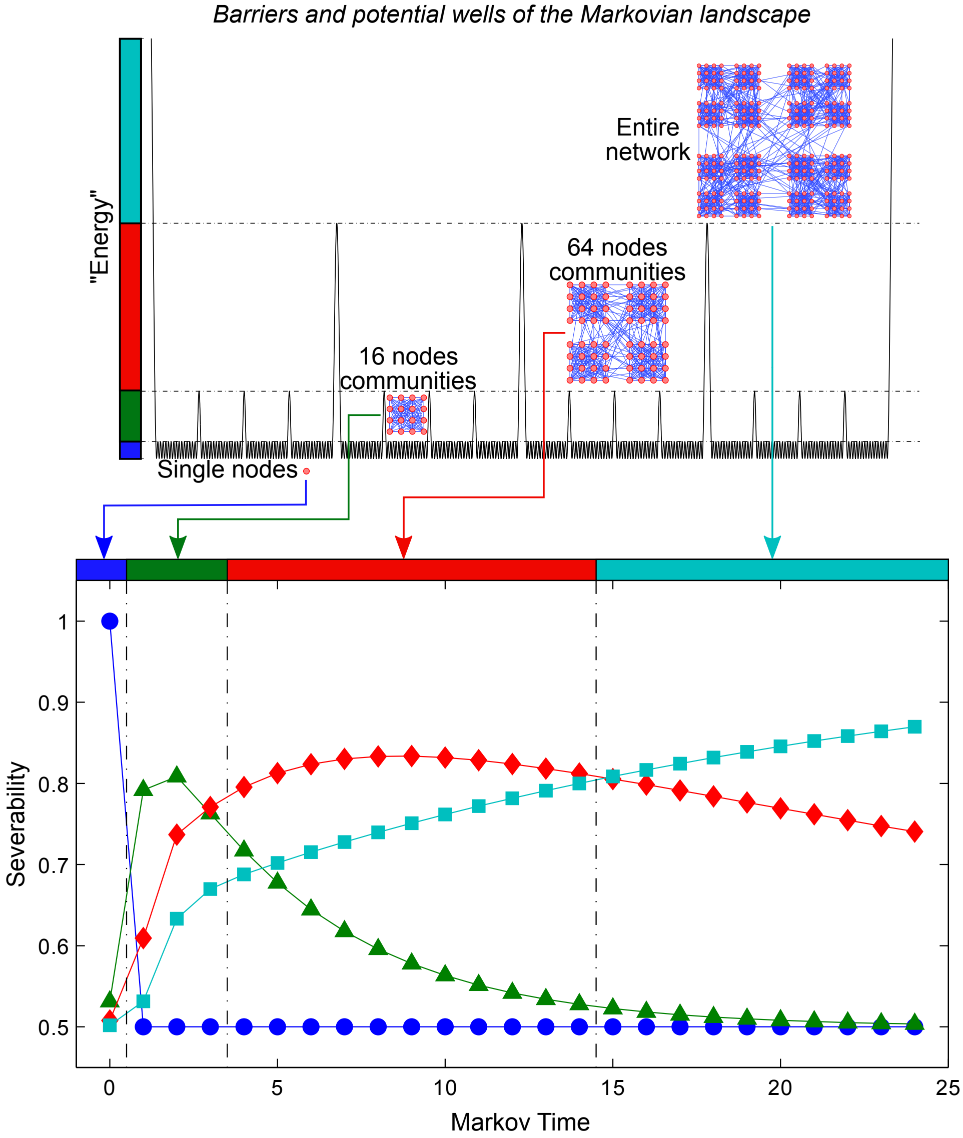


Figure 4.1: The "Markovian" energy landscape of a hierarchical random graph with multi-level communities of sizes 16, 64 and 256. Severability gives optimal community sizes as a function of time. Depicted are the severabilities of a single node (blue circles), 16-node community (green triangles), 64-node community (red diamond) and the entire network (cyan square).

$p_3 = 0.036$ respectively, resulting in an average degree $< k >= 16$. Variations on the exact probabilities and number of levels are of course possible, but we chose here a single representative network for illustration purposes only.

This benchmark serves to illustrate the use of Markov time as a resolution parameter for networks with multi-level structure. As Markov time increases, the random walkers gain sufficient probability to overcome "energy barriers" and hence diffuse to larger portions of the network so that the optimal community grows from single nodes to the entire network, passing through each of the intermediate levels (figure 3.1). For smaller times, it makes sense that the lower-level structure (the L1 communities) would be recovered, whereas later on, larger super-structures dominate.

4.2 Variants of the LFR benchmark

However, many real-world networks do not have such a simple distribution of community sizes. In order to provide an alternative to the Girvan-Newman model of identically sized communities [28], Lancichinetti, Fortunato and Radicchi recently proposed several new classes of benchmarks, in which the community size and node degree follow power-law distributions [40]. Additionally, these benchmarks also allow for overlapping, directed and weighted networks [37].

4.2.1 Comparison methods

Optimal community cover

To compare against benchmarks which permit overlapping communities, it is necessary to generate a list of communities to cover the network. Simply taking the optimal communities of each node is suboptimal, because then there are many duplicate communities in the list. Instead, we chose the following naive method: cover the network by communities, starting community optimisations only from unassigned nodes that are on average more connected to other unassigned nodes than to previously found communities.

More elaborate schemes for generating covers of a network by communities are of course possible and may indeed give better results. However, this method has the advantage of reducing the computational time required, as well as being simple to implement.

Optimal partition

To compare severability with other partitioning methods, it is necessary to turn the optimal community cover into a (hard) partition. To do so, first order the communities of the cover arbitrarily; we used the order in which the optimal community cover found the communities. Assign each node to the first community it appears in. This procedure is obviously dependent upon the ordering of the communities. In other settings, it may behoove the practitioner to develop a smarter way of assigning nodes to communities. However, in networks with well-defined partition structure, this method works sufficiently well, as will be demonstrated in analysis of the LFR benchmark graphs.

Choice of Markov Time

For hierarchical networks, Markov time serves as a useful resolution parameter, allowing for severability to pick out optimal community structure at different levels. However, existing metrics [15, 39] require the selection of a single time t. It is not always clear *a priori* what t should be chosen for finding optimal community covers, which is a problem common to many algorithms with resolution parameters. However, for optimal partitioning, this can be done by choosing a Markov time to minimise the total number of orphan and overlapping nodes.

Quantifying similarity

For quantifying similarity, we have employed normalised mutual information [27, 15, 35]. Coming from information theory, this metric takes the form

$$I_{norm}(X:Y) = \frac{H(X) + H(Y) - H(X,Y)}{(H(X) + H(Y))/2},$$

where $H(X)$ is the entropy of a random variable X and $H(X,Y)$ represents the joint entropy. Thus, normalised mutual information is a measure of the shared information between the two random variables X, Y, normalised against the sum of their respective entropies.

In comparing partitions, the partitions take the place of the random variables, and the frequency of shared nodes is used as an approximation of probability. Let X, Y be partitions, and let c_X, c_Y be the number of communities in each partition respectively. Define a $c_X \times c_Y$ *confusion matrix* N, where N_{ij} is the number of nodes in the ith community of X that are also in the jth community of Y. Let $N_{i*} = \sum_{j=1}^{c_Y} N_{ij}$ and $N_{*j} = \sum_{n=1}^{c_X} N_{ij}$. Then the normalised mutual information is given by

$$I_{norm}(X:Y) = \frac{-2\sum_{i=1}^{c_X}\sum_{j=1}^{c_Y} N_{ij} log(N_{ij}N/N_{i*}N_{*j})}{\sum_{i=1}^{c_X} N_{i*} log(N_{i*}/N) + \sum_{j=1}^{c_Y} N_{*j} log(N_{*j}/N)}.$$

For comparing community covers, a generalisation of normalised mutual information that allows for overlapping nodes has been used [39]. This is of course more complicated because each node can belong to multiple communities within a partition.

We refer to the generalised variant as simply "normalised mutual information" without loss of precision as only the generalised variant can be used in the benchmarks with overlapping communities.

4.2.2 Unweighted, undirected, non-overlapping networks

Here is highlighted the analysis of a class of networks in which communities are extremely unevenly sized, a situation in which many popular partition quality functions perform sub-optimally. These multi-scale graphs are randomly constructed such that both degree and community size distributions follow power laws, with exponents γ and β, respectively. Additional parameters include the total number of nodes N, the average degree $< deg >$, the maximum degree deg_{max}, and the mixing parameter μ - not to be confused with the mixing component of severability. The fraction of links from a node to other nodes within the same community is given by $1 - \mu$ [40]. Graph generation parameters were chosen at values matching those in paper of Lancichinetti et al. [40]: $\gamma = 2$, $\beta = 2$, $N = 1000$, $< deg >= 15$ and $deg_{max} = 50$. Severability optimisation was performed with a maximum search size $S = 50$, and partitions were generated from the community cover.

As can be seen from the figure, severability performs well, always finding the natural community structure up to until around $\mu = 0.5$, when communities are no longer defined in a strong sense [55]. That severability begins failing at $\mu = 0.5$ is expected, since at that point random walkers are as likely to escape during each step as to remain within any of pre-seeded communities. Recalling the definition, a community is defined as severable precisely when random walkers tend to stay and mix within it. Even so,

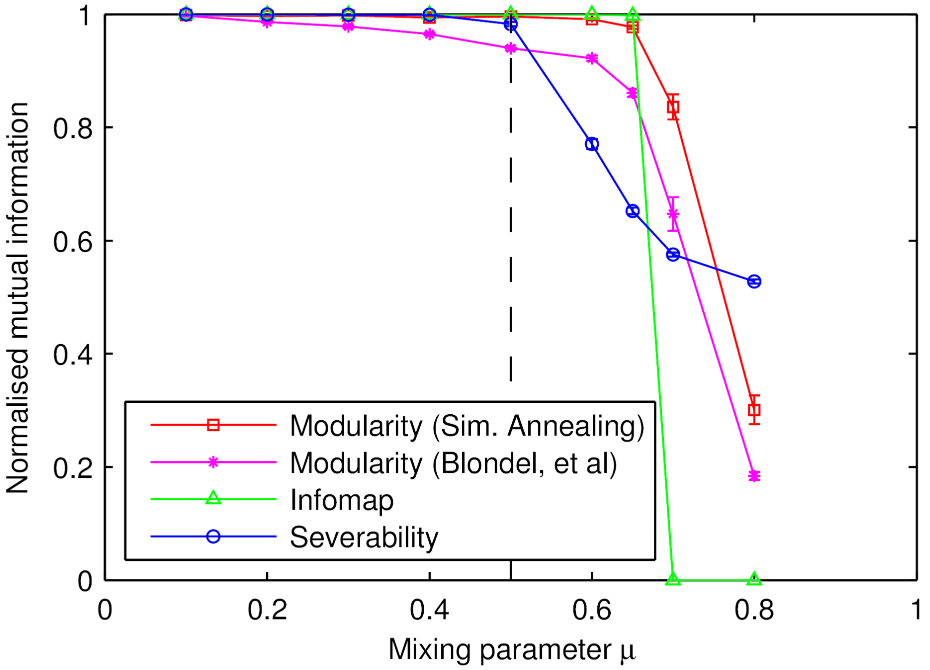


Figure 4.2: Comparison of severability with modularity and infomap the LFR benchmarks with exponents $\gamma = 2$, $\beta = 2$, average degree $< deg >= 20$ and maximum community size of 50. Severability optimisation was performed with maximum search size of 50 and Markov time $t = 3$ (a value determined as a result of minimising the number of orphan nodes and overlapping nodes). Modularity was optimised for using both simulated annealing [29], which is extremely slow but gives good results, and a faster heuristic by Blondel, et al [5]. Each point is an average over ten random realisations.

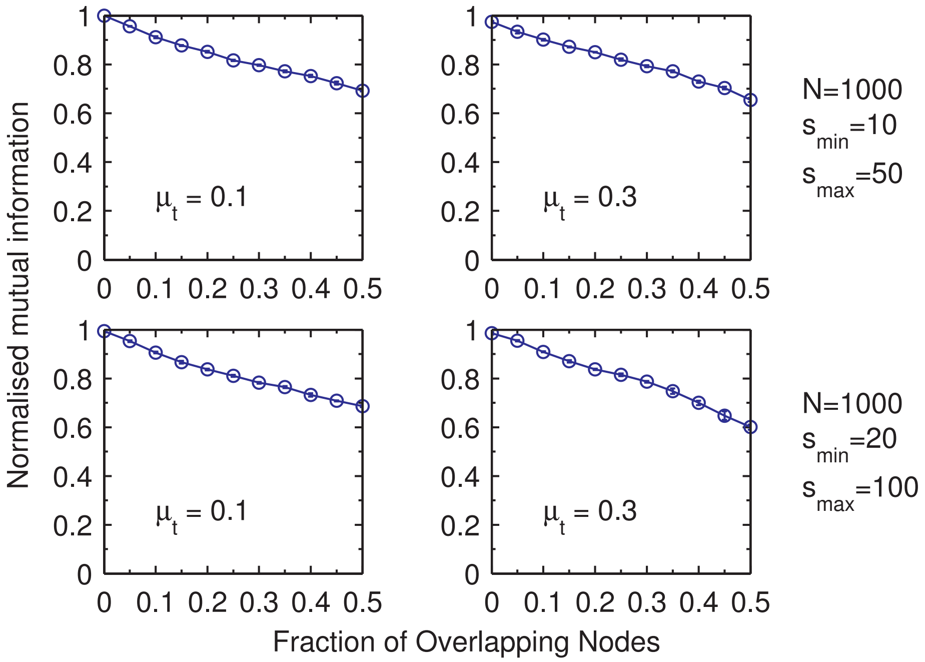


Figure 4.3: Severability at Markov time $t = 4$, with an unweighted, undirected, overlapping variant of the LF Benchmark [37]. The networks have 1000 nodes; the other parameters are $\tau_1 = 2$, $\tau_2 = 1$, $< deg >= 20$, $deg_{max} = 50$. Each point is an average over five random realisations.

the results are comparable to that of Infomap and modularity optimisation using simulated annealing, which have been found to be amongst the most successful methods for this benchmark [38].

4.2.3 Unweighted, undirected, overlapping networks

Further extensions to the LFR benchmark were made to allow for communities to overlap [37]. In figure 4.3, we compare the community covers from severability to the pre-seeded communities. For the optimisation, the maximum search size $S = 50, 100$ was used for the upper and lower panels, respectively.

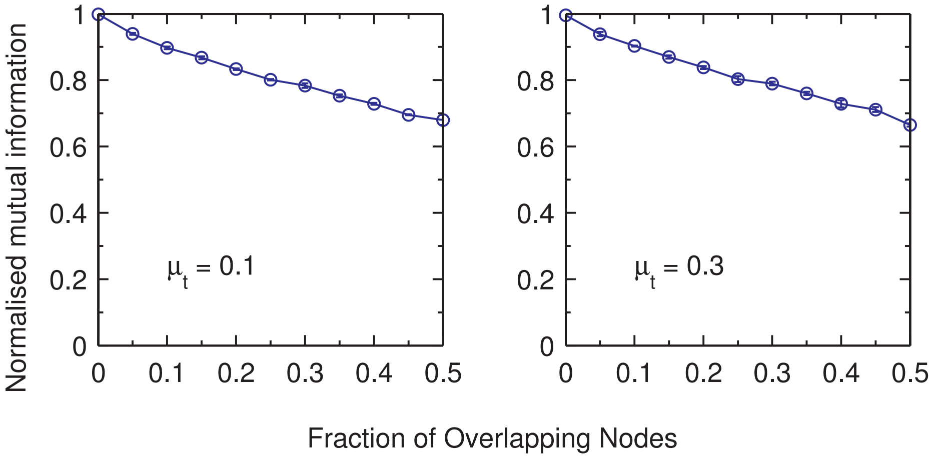


Figure 4.4: Severability at Markov time $t = 4$, with a weighted, directed, overlapping variant of the LF Benchmark [37]. The networks have 1000 nodes; the other parameters are $\tau_1 = 2$, $\tau_2 = 1$, $\mu_w = 0.2$, $< deg >= 20$, $deg_{max} = 50$, $s_{min} = 20$, $s_{max} = 100$. Each point is an average over five random realisations.

The parameters chosen were identical to those used for the evaluation of k-clique percolation [52] in figure 6 of Lancichinetti and Fortunato, 2009 [38]. Comparison with those results will show that severability performs comparably to slightly worse for the smaller community sizes, but significantly better for larger communities.

4.2.4 Weighted, directed, overlapping networks

Furthermore, severability also loses no accuracy when direction and weight are added to the benchmark [37] (figure 4.4). This is expected, since the Markov chain formulation naturally includes both without any modifications to either the optimisation procedure or the quality function. For the optimisation shown, the maximum search size $S = 100$.

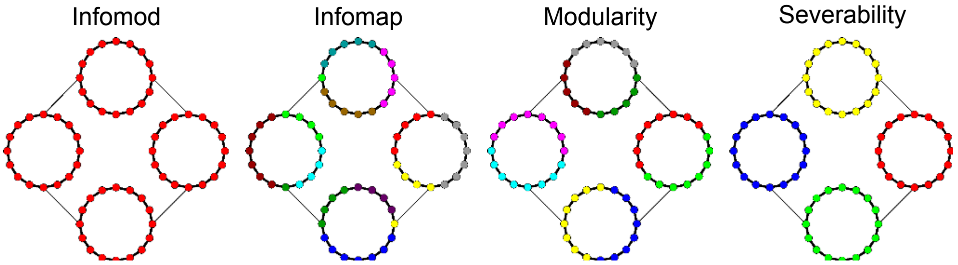


Figure 4.5: Ring of rings. Heavy lines (within rings) correspond to undirected links with weight 2, while light lines between rings to links with weight 1. Only severability is able to recover the seeded ring structure (at Markov times $3 \leq t < 10$).

4.3 Ring of rings

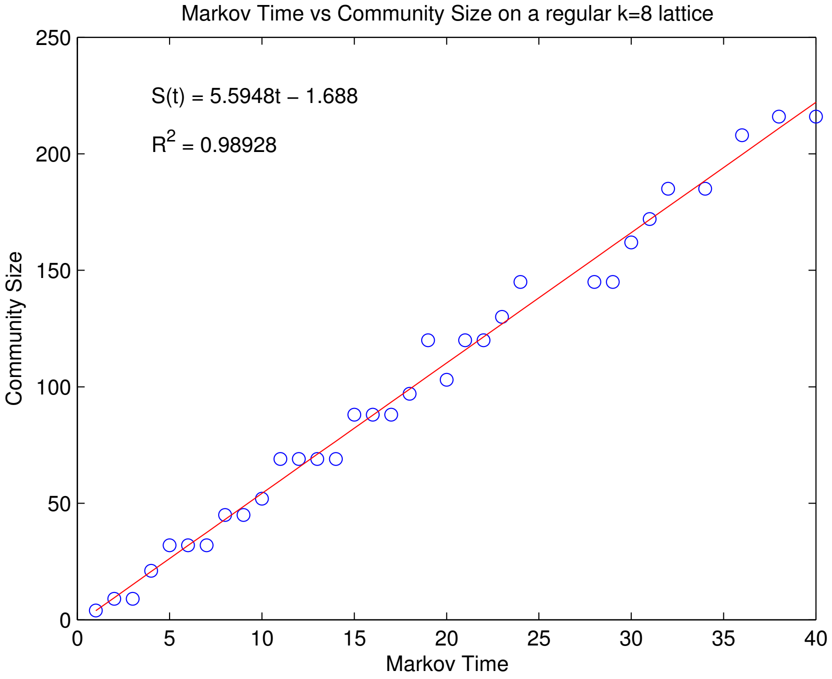
Most constructed benchmarks [28, 40], including the ones we analysed above, base their communities on a very high density of links, modelled by cliques or random high density edge selection from a clique. Similarly, most community detection methods also adhere to this assumption; some, like k-clique percolation, make this an explicit requirement [52], whereas others like modularity have null models that implicitly lead to favouring cliques.

Cliques are of course a key feature of many real world networks that severability is able to handle, as demonstrated by performance on the LFR benchmarks. However, it is instructive to examine other possibilities for community structure, since they may turn up to be of importance in man-made networks, which often have very non-random placement of links. As an illustration, consider here a collection of rings, in which a network of 64 nodes is divided into 16-node rings, with strong intra-ring links of weight 2 and weak links of weight 1 between rings (figure 4.5). Modularity (using simulated annealing [29]) and infomap [58] fail to find the most natural communities, while infomod [57] finds only the trivial community of the entire graph. Because severability is based on the retention and mixing of random walkers, it performs considerably better; at a Markov time $3 \leq t < 10$, severability recovers the rings as the optimal community cover.

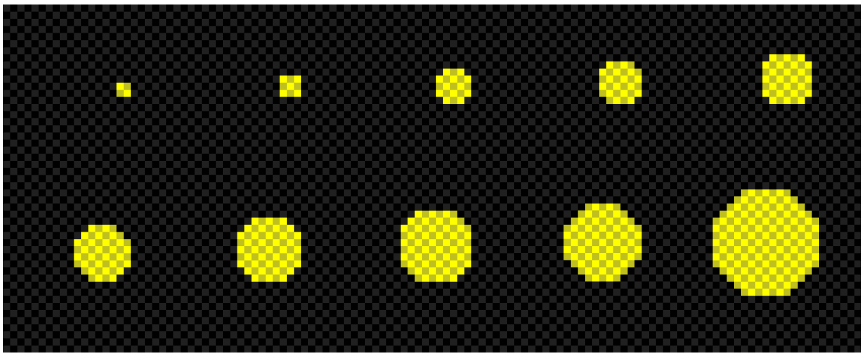
4.4 Square lattice

As a negative control, It is instructive to consider a network in which there is clearly no community structure. For that, we chose a regular 2-D square lattice with each node connected to all 8 neighbours (including diagonal links). We visualise this using a uniformly coloured discrete image, in which each pixel is connected to all of the adjacent pixels with links of equal strength. As can be seen in figure 4.6, after accounting for symmetry considerations, all communities found are transients, which is the expected result.

Additionally, these images strongly suggest a relationship between severability optimisation and diffusion. This is of course quite closely related both to the dependance of severability on random walk dynamics and to the optimisation procedure outlined in 3.3. Along these lines, the optimisation procedure we outlined can be thought of as a modified random walk in which previously explored states are immediately accessible to the random walker, but probability barriers in the “energy landscape” are magnified.



(a)



(b)

Figure 4.6: (a) Correlation of Markov time with size on an square lattice. (b) The transient communities found by severability on a regular lattice at Markov times $t = \{1, 2, 3, 5, 8, 10, 11, 15, 18, 32\}$. Each block is connected to all eight of its neighbouring blocks by a single undirected edge of weight 1.

5 Real networks

It is (tautologically) always possible to construct artificial benchmarks arbitrarily, but only when they approximate real-world networks do they become relevant. In this chapter, we give examples of real-world networks that exhibit overlapping communities, orphan nodes, ring-like structure and lattices. Additionally, we highlight one of the key advantages of local methods, that they can be effectively used on partial graphs.

5.1 Word association, overlap, & orphans

One important aspect of community detection is the possibility of overlapping communities or orphans nodes. To illustrate this, we turn to a word association network that was among the examples used to highlight overlapping communities in the introduction of k-clique clustering [52], namely the University of South Florida Free Association Norms dataset [47]. Researchers at the University of South Florida presented vocabulary words to study participants, who were then asked for the first word that came to mind. In constructing this network, each node corresponds to a single word, and directed links between nodes are weighted according to the proportion of responses linking those two words. For example, when cued with "science", 21.4% of participants wrote "biology". Thus, there exists a directed link from "science" to "biology" with weight 0.214.

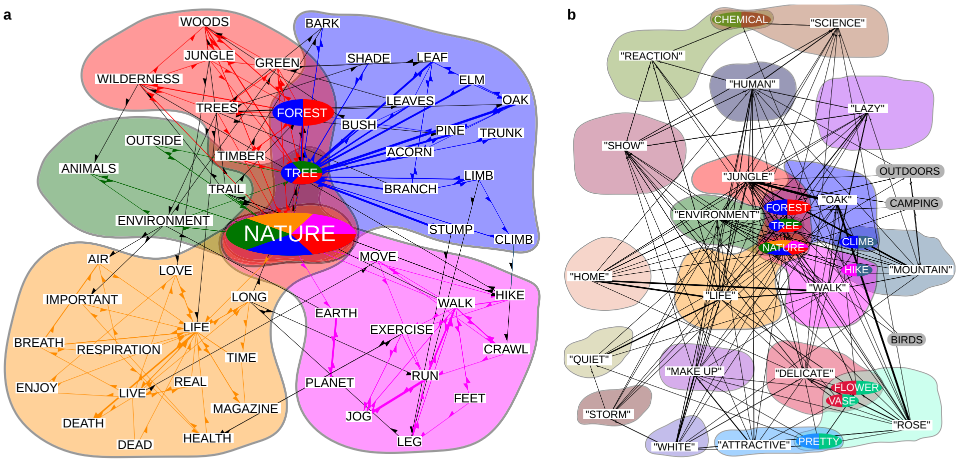


Figure 5.1: (a) The five communities that the word “nature” belongs to. For the word association network, every neighbour of “nature” was attempted for the first step, giving the overlapping communities. Nodes and links are coloured by community identification; coloured ovals represent multiple community membership. (b) A broader view of the community landscape surrounding “nature”, depicting also communities connected to, but not containing, “nature”, including three orphan nodes. Nodes belonging to just one of the communities are combined into a single block labelled by the most central word of the community, while nodes belonging to more than one community are separately mentioned in the grey ovals. Note that in many cases, the words used to “tag” the community blocks themselves have multiple “tags”. Communities were found by optimising severability for Markov time $t = 2$ and maximum search size $s = 50$.

Because cliques are only well-defined on unweighted, undirected networks, using them directly requires thresholding by weight and ignoring direction, which already discards quite a bit of information. Markov chains do not require such false dichotomies, so severability naturally admits both link direction and weight. Additionally, as a local method, it is not necessary to analyse the entire graph to find communities. Rather, by analysing more and more of the network, an ever-expanding view presents itself; figure 5.1 shows in part (a) "nature" and the communities it belongs to, while part (b) depicts the communities and orphan nodes (which are simply communities of size 1) "nature" is directly linked to (see figure 5.2).

For this analysis, we limited the maximum sizes of the communities considered. However, another advantage of the method is that this in itself does not limit its applicability. This is made possible by inherent checks in the method. For example, when the maximum community size (i.e. the number of nodes) is chosen to be too small, nearly all the resulting communities will be truncated at the limit, while too large of a choice is not an issue because the Markov time itself limits the community size. Indeed, in some cases where prior knowledge of the network structure is given, artificially limiting the maximum community size can be beneficial.

5.2 Metabolic networks

A real-world example of ring structure can be found in the basic biochemistry of the citric acid cycle. To construct the network from the citrate pathway schematic (map00020) in the KEGG database [31], each compound was used as a node, and if there exists a reaction with A as a substrate and B a product, a directed, unweighted link from A to B was assigned.

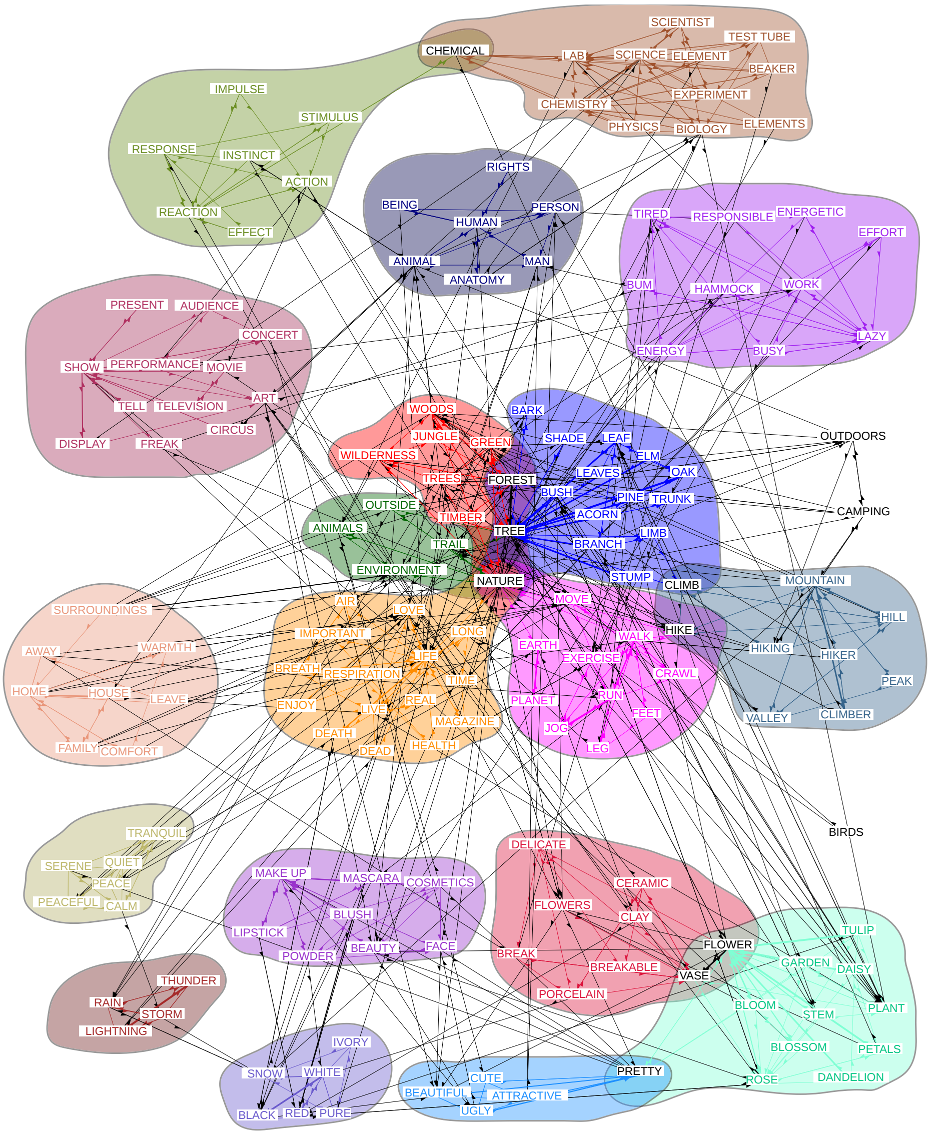


Figure 5.2: Figure 5.1 only depicted the communities including “nature” and the communities directly connected to that word. Here, we have displayed the full membership of all the other communities that have at least one link to “nature” but do not include the word itself.

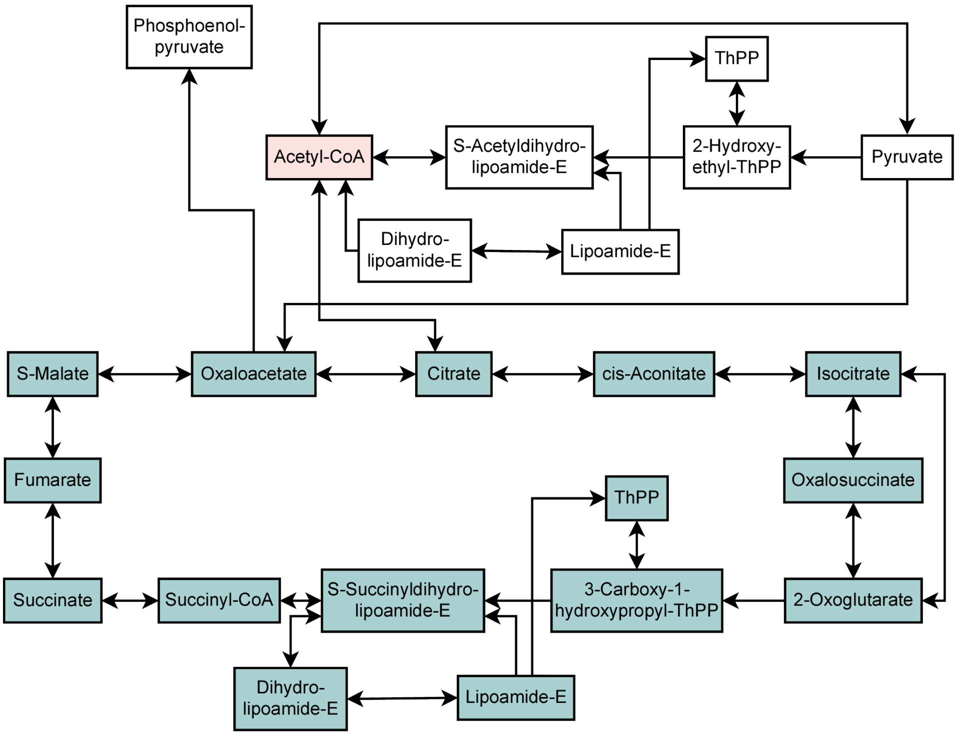


Figure 5.3: The citric acid cycle [31]. The blue region is a stable community from Markov time $17 < t \leq 21$ and adds acetyl-CoA from $21 < t \leq 31$.

Given the similarity to the ring of rings construct, it is not surprising that the severability finds the central cycle (figure 4.5). However, it not only serves to highlight the fact that some real systems are based around structures other than cliques, but also that the random walk in this network has an obvious physical interpretation, that of biochemical transformations. Such physical interpretations are not necessary for the detection of interesting communities, but when existent can provide additional insight into the meaning of network structure.

5.3 Image segmentation & commutativity

In this final example, we apply severability optimisation to segment a cell-fluorescence image, automating the identification of stained neurons (figure 5.4) to illustrate another powerful aspect of the method, namely that it does not rely on global information in order to detect communities faithfully. Indeed, we show below that the results are similar whether the algorithm was run on only some part of the image or on the entirety.

As suggested by figure 3.1 and the square lattice network (which is equivalent to a constant image under the following), one can apply standard graph partitioning techniques to segmenting images. One of the classical means of generating networks from image data is to assign a link weight between pixels by using differences in luminosity and distance (up to a cut-off) [8]. Connecting only adjacent pixels (using the maximum metric), the link weight $w = \exp\left(-(\Delta I)^2/\sigma_I^2\right)$, where ΔI is the difference in luminosity and σ_I is an adjustable parameter controlling the exponential weight decay. Pre-processing consisted of reducing the image resolution to a manageable size and converting to a network.

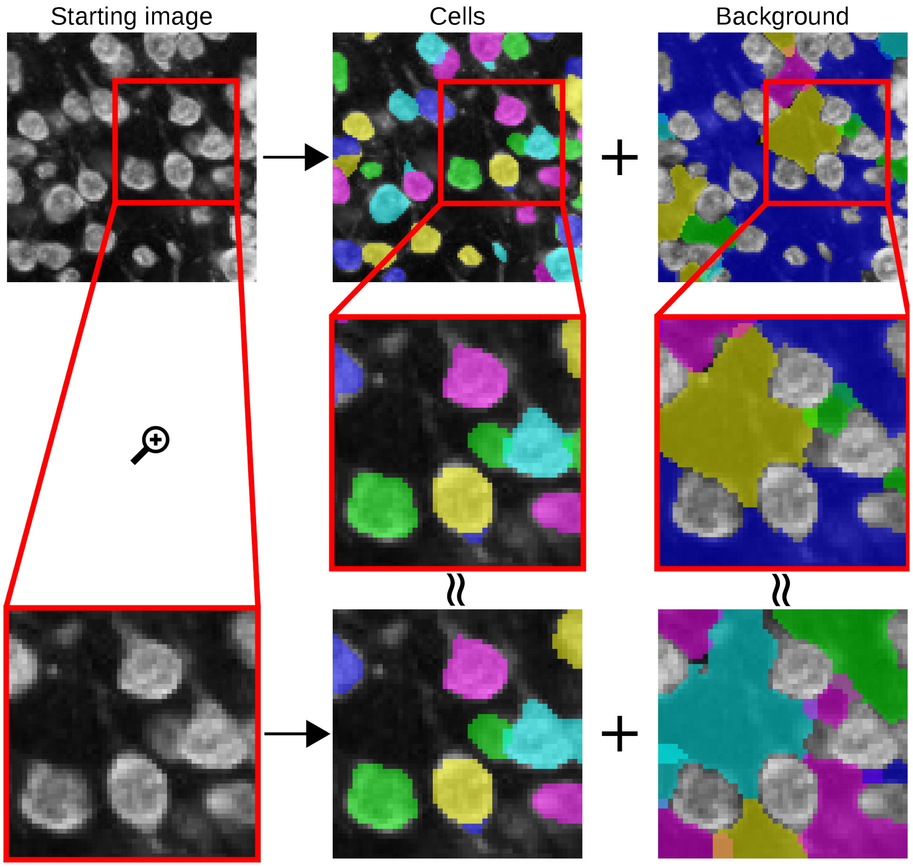


Figure 5.4: Neocortical pyramidal neurons, stained with a fluorescent dye, with resolution reduced to 102 × 102 and converted to grayscale by luminosity. (Cyan) Using $\sigma_I = 20$, Markov time $t = 32$, and maximum size $s = 200$, communities largely corresponding to cells were found, as well as background patches, which were merged together if more than 20-pixels overlapped to form super-communities. (Yellow) Furthermore, repeating the procedure with a cropped subregion of the image gives largely the same results, with some minor variations along the borders. This near-commutativity is a key feature of local methods. Original image courtesy of Simon Schultz and Marie-Therese Vasilache.

Similarly, little post-processing was necessary, limited to removing bad communities, merging overlapping ones and then ordering by luminosity. Communities with mixing $\mu > 0.9$ and retention $\rho < 0.1$ were considered bad, because at high Markov times they correspond to nearly disconnected communities (the aberrant values of μ and ρ result partly from numerical errors when $Q^t \to 0$).

Merging is sometimes possible when a feature of the network is much larger than the maximum search size; in this case the optimisation method gives overlapping patches of the background, which can then be pieced together. If a community C_1 was completed embedded in C_2 ($C_1 \subset C_2$), we kept only the one with higher severability. Communities were then inductively merged if they overlapped by more than 20 pixels until no more merges were possible. The communities were ordered by average luminosity, and the darker patches were assigned to the background. The ordering by luminosity is not as easily generalised, but it works here because the luminosity corresponds to the cell fluorescence. These steps are sufficient for good results, despite the fact that severability was not designed for the analysis of images.

Of particular note is that because severability does not depend on global information, the results are similar whether the algorithm was run on only some part of the image or on the entirety. Communities that are not on the boundary of the subset will not differ at all; although those that are on the boundary will differ, they still tend to be similar as the part of the community not on the boundary will largely retain its severability. This is of potential application to evolving networks, as communities are stable against perturbations outside of a local neighbourhood.

6 Future Work

Although we have already successfully demonstrated the utility of severability to a number of problems, there is still much work to be done. In particular, we suggest that the two obvious outstanding directions include (1) finding better numerical methods and (2) applying severability to evolving and/or large networks.

6.1 Computational optimisation

The method presented in chapter 3 is not very computationally efficient. Much of this is due to having to recompute the matrix power from scratch every time a node is added to the community during the optimisation process. It is reasonable to hypothesise that the shared entries of a matrix and its principal sub-matrix could provide a more efficient algorithm for computing their respective powers.

To justify this conjecture, we first restrict ourselves to exact arithmetic, undirected graphs and adding a single node to a community. Recall that the Markovian transition matrix $P = D^{-1}A$, where A is the adjacency matrix. Then

$$P^t = (D^{-1}A)^t = D^{-\frac{1}{2}}(D^{-\frac{1}{2}}AD^{-\frac{1}{2}})^t D^{\frac{1}{2}} \equiv D^{-\frac{1}{2}}\mathring{A}^t D^{\frac{1}{2}}.$$

Since A is symmetric for undirected graphs, $\mathring{A}$ is also symmetric. Thus, adding a node is equivalent to adjoining a single row and column to the symmetric matrix $\mathring{A}$.

Unfortunately, a direct expansion of $(X+Y)^t$ for arbitrary square matrices X and Y results in an unwieldy expression that requires more than t matrix multiplications. However, prior results on eigenvalue interlacing [6] and the spectral changes resulting from special low rank perturbations of [21, 20, 77, 44, 34] suggest an alternate line of inquiry. Using the close relationship between matrix powers and eigendecomposition, we choose to instead focus on the corresponding eigendecomposition problem.

6.1.1 Proof of concept

Conjecture 6.1.1. *Let $\mathring{A}$ be a $k \times k$ real symmetric matrix, α a $k \times 1$ vector, $d \in \Re$, and*

$$B = \begin{bmatrix} \mathring{A} & \alpha \\ \alpha^T & d \end{bmatrix}.$$

Then given the orthogonal eigendecomposition $\mathring{A} = R \Lambda R^T$, where Λ is diagonal and R is orthogonal,

1. *the eigenvalues of B*

2. *and an orthogonal transform of the eigenvectors of B*

are computable in $O(k^2)$ steps ***in exact arithmetic****.*

Proof. Let

$$S = \begin{bmatrix} R & 0_{k\times 1} \\ 0_{1\times k} & 1 \end{bmatrix}.$$

Then

$$S^T B S = \begin{bmatrix} R^T & 0_{k\times 1} \\ 0_{1\times k} & 1 \end{bmatrix} \begin{bmatrix} \mathring{A} & \alpha \\ \alpha^T & d \end{bmatrix} \begin{bmatrix} R & 0_{k\times 1} \\ 0_{1\times k} & 1 \end{bmatrix} = \begin{bmatrix} \Lambda & R^T\alpha \\ \alpha^T R & d \end{bmatrix}.$$

As Λ is known, $S^T BS$ only requires one matrix-vector multiplication, $P^T\alpha$, to compute at a cost of k^2 operations.

$Q^T BQ$ has no more than $3k+1$ non-zero entries, so by using sparse matrix multiplication and the Lanczos algorithm, it can be orthogonally transformed in $O(k^2)$ time into a tridiagonal matrix (recall that we are working in exact arithmetic). Then, the QR algorithm can give the eigenvalues in $O(k^2)$ operations.

Let λ be an eigenvalue of B. Then the corresponding eigenspace is the kernel of $B - \lambda I$, which is difficult to solve for. However, a simpler, related problem is to find the kernel of $S^T BS - \lambda I$.

Suppose that $S^T BS - \lambda I$ has no zero entries along the diagonal. Then, because the kernel is not empty, the $(k+1)$ row $\begin{bmatrix} \alpha^T R & d-\lambda \end{bmatrix}$ lies in the span of the first k rows. Since the $(k+1)$ row can be ignored, solving for $(S^T BS - \lambda I)X = 0$ requires only k multiplications. If $S^T BS - \lambda I$ has at least one zero entry along the diagonal, then back-substitution must be used to reduce the matrix to Gauss-Jordan form, but this still requires only $O(k)$ multiplications.

Since the eigenspaces need to be computed for each of $k+1$ eigenvalues, computing the eigenvectors for $S^T BS$ requires $O(k^2)$ steps, giving an orthogonal transform of the eigenvectors of B. □

Unfortunately, recovering the eigenvectors of the original matrix B requires conjugation by Q, which using naive matrix multiplication takes

$O(n^3)$ operations. However, in the case of computing severability, this step may not be necessary. The retention ρ is clearly preserved under orthogonal transforms, and the mixing μ is at the very least approximated by an orthogonal transform. Thus, for finding the neighbouring node that best optimises for severability, this can potentially reduce the complexity by a factor of k.

6.1.2 Numerical stability

As highlighted several times, conjecture 6.1.1 only holds in exact arithmetic. However, most of the involved steps take place using orthogonal transformations, so they should be numerical stable. There are two major exceptions, the Lanczos iteration and the Gaussian elimination, which is why this procedure has not actually been attempted.

In the case of the Lanczos iteration, orthogonality is lost after sufficient steps due to rounding errors. Re-orthogonalisation is costly, but not generally necessary because Lanczos is normally only used to find the extremal eigenvalues. However, iterating many ($> k$) times to find the full spectrum, while possible[23], is generally very error-prone due to round-off. The interlacing property does provides an easy way to determine whether or not the full spectrum has been found [6], so there is some hope that a resolution might be possible.

For the Gaussian elimination, note that in the case with no zero diagonal entries, no back-substitution is required, so even without pivoting, it should be fairly numerically stable. In the other case, only forward substitution is only required for the last row, so again, it seems plausible that a numerically stable algorithm is possible.

6.1.3 Conclusion

Although we have here proposed in theory a quicker method of computing the severability change given by adding a node to a community, it cannot be used. Until the issues of numerical stability can be resolved, reducing the time complexity by a factor of k is probably still a pipe-dream. Additionally, further work would need to be done to compute the change in severability from removing a node, as well as generalising the process to non-symmetric matrices (i.e. directed graphs). Noting that non-Hermitian matrices do not generally have a full spectrum of real eigenvectors/values, the problem seems much less tractable.

6.2 Evolving and large networks

As hinted at in figure 5.4, one of the key advantages of using a purely local quality function is that it is more robust to the addition of new nodes. This key advantage of severability over global methods suggests a potential avenue for future inquiry, namely that of evolving networks.

For constantly changing systems (e.g. the World Wide Web), it is desirable to be able to cheaply update a network analysis. When using global methods for partitioning, it is impossible to tell whether a change to the underlying network will significantly alter the results. On the other hand, severability, as a trivial consequence of its locality, provides explicit guidelines on when a previously detected community will be altered by the addition of a new node.

Another related problem is that of extremely large static networks. These sorts of problems may not easily fit into computer memory and are often broken up into smaller problems, whose results are then integrated together.

As a consequence of its locality, severability is particularly amenable to being used for this sort of analysis. Stitching together the results from two different portions of the graph requires only properly accounting for those communities that are partially found in both sections.

7 Conclusion

Much of this thesis has been devoted to the particular quirks and characteristics of severability. It performs comparably to existing partition quality functions on traditional benchmarks, allows for communities that are not clique-based and includes an inherent resolution parameter for better understanding of multilevel networks. Additionally, as a local community quality function based on Markov chains, it not only naturally admits several key features of real networks—including overlapping communities, orphan nodes and link direction and weight—but also is stable against distant graph perturbations. As a huge wealth of data is being collected in all areas of science, technology and social life, severability fulfils the requirements to be able to harvest information and insight even with partial knowledge.

However, just as importantly in our opinion, it presents a different notion of what it means for a set of nodes to be a community. The appeal to the coexistence of time scales in Markov processes as a means to unfold communities from the inside is not limited to severability; there are many different ways mixing and retention can be quantified and combined, some of which might lead to better or more computationally efficient approaches. Indeed, one could imagine even Markov processes themselves being eschewed in favour of other means of quantifying the barriers and roughness of the graph landscape.

Furthermore, because the mathematical construction used is remarkably similar to Markov chain lumping [22, 30], there are strong hints of potential connections to model reduction. Similarly, connections to diffusion processes are likely, given both the behaviour of severability on square lattices and the random walk nature of the method. Thus, while future explorations of these themes may not involve matrix powers or vector averages, the importance of dynamics as a key to characterising networks will undoubtedly persist.

Bibliography

[1] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[2] A. Ando and F. M. Fisher. Near-Decomposability, partition and aggregation, and the relevance of stability discussions. *International Economic Review*, 4(1):53–67, Jan. 1963.

[3] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509 –512, Oct. 1999.

[4] M. Blatt, S. Wiseman, and E. Domany. Clustering data through an analogy to the Potts model. *Advances in Neural Information Processing Systems*, page 416–422, 1996.

[5] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008, 2008.

[6] B. Bollobás and V. Nikiforov. Graphs and Hermitian matrices: eigenvalue interlacing. *Discrete Mathematics*, 289(1-3):119–127, Dec. 2004.

[7] G. C. Boulougouris and D. N. Theodorou. Dynamical integration of a Markovian web: A first passage time approach. *The Journal of chemical physics*, 127(8):084903, 2007.

[8] A. Browet, P. Absil, and P. Van Dooren. Community detection for hierarchical image segmentation. *Combinatorial Image Analysis*, page 358–371, 2011.

[9] J. M. Carr and D. J. Wales. Global optimization and folding pathways of selected α-helical proteins. *The Journal of chemical physics*, 123(23):234901, 2005.

[10] J. M. Carr and D. J. Wales. Folding pathways and rates for the three-stranded β-sheet peptide beta3s using discrete path sampling. *The Journal of Physical Chemistry B*, 112(29):8760–8769, 2008.

[11] J. M. Carr and D. J. Wales. Refined kinetic transition networks for the GB1 hairpin peptide. *Physical Chemistry Chemical Physics*, 11(18):3341–3354, 2009.

[12] J. Chen, O. Zaiane, and R. Goebel. Local community identification in social networks. In *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, page 237–242, 2009.

[13] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72(2):026132, 2005.

[14] P. Courtois. On time and space decomposition of complex structures. *Communications of the ACM*, 28(6):590–603, 1985.

[15] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008–P09008, Sept. 2005.

[16] J. N. Darroch and E. Seneta. On Quasi-Stationary distributions in absorbing Discrete-Time finite markov chains. *Journal of Applied Probability*, 2(1):88–100, June 1965.

[17] A. Delmotte, E. W. Tate, S. N. Yaliraki, and M. Barahona. Protein multi-scale organization through graph partitioning and robustness analysis: application to the myosin–myosin light chain interaction. *Physical Biology*, 8(5):055010, Oct. 2011.

[18] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona. Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences*, June 2010.

[19] C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *ICDM 2001, Proceedings IEEE International Conference on Data Mining, 2001*, pages 107–114. IEEE, 2001.

[20] J. Ding and G. Yao. The eigenvalue problem of a specially updated matrix. *Applied Mathematics and Computation*, 185(1):415–420, Feb. 2007.

[21] J. Ding and A. Zhou. Eigenvalues of rank-one updated matrices with some applications. *Applied Mathematics Letters*, 20(12):1223–1226, Dec. 2007.

[22] W. E, T. Li, and E. Vanden-Eijnden. Optimal partition and effective dynamics of complex networks. *Proceedings of the National Academy of Sciences*, 105(23):7907–7912, June 2008.

[23] J. T. Edwards, D. C. Licciardello, and D. J. Thouless. Use of the Lanczos method for finding complete sets of eigenvalues of large sparse

symmetric matrices. *IMA Journal of Applied Mathematics*, 23(3):277 –283, Apr. 1979.

[24] D. A. Evans and D. J. Wales. Free energy landscapes of model peptides and proteins. *Journal of Chemical Physics*, 118(8):3891–3897, 2003.

[25] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.

[26] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36 –41, Jan. 2007.

[27] A. Fred and A. Jain. Robust data clustering. In *IEEE computer society conference on computer vision and pattern recognition*, volume 2, 2003.

[28] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821 –7826, June 2002.

[29] B. H. Good, Y. de Montjoye, and A. Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, Apr. 2010.

[30] M. N. Jacobi. A robust spectral method for finding lumpings and meta stable states of non-reversible markov chains. *0810.1127*, Oct. 2008.

[31] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27 –30, Jan. 2000.

[32] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.

[33] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2):291–307, 1970.

[34] M. Krupnik. Changing the spectrum of an operator by perturbation. *Linear Algebra and its Applications*, 167(0):113–118, Apr. 1992.

[35] L. I. Kuncheva and S. T. Hadjitodorov. Using diversity in cluster ensembles. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1214– 1219 vol.2. IEEE, Oct. 2004.

[36] R. Lambiotte, J. C. Delvenne, and M. Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv*, 0812.1770, Dec. 2008.

[37] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, July 2009.

[38] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, Nov. 2009.

[39] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, Mar. 2009.

[40] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, Oct. 2008.

[41] N. Lempesis, D. G. Tsalikis, G. C. Boulougouris, and D. N. Theodorou. Lumping analysis for the prediction of long-time dynamics: From monomolecular reaction systems to inherent structure dynamics of

glassy materials. *The Journal of chemical physics*, 135(20):204507, 2011.

[42] E. D. Levy and J. B. Pereira-Leal. Evolution and dynamics of protein interactions and networks. *Current Opinion in Structural Biology*, 18(3):349–357, June 2008.

[43] F. Luo, J. Wang, and E. Promislow. Exploring local community structures in large networks. In *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, pages 233–239, 2006.

[44] J. Moro and F. Dopico. Low rank perturbation of Jordan structure. *SIAM journal on matrix analysis and applications*, 25(2):495–506, 2003.

[45] S. Muff, F. Rao, and A. Caflisch. Local modularity measure for network clusterizations. *Physical Review E*, 72(5):56107, 2005.

[46] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354 –359, Nov. 1983.

[47] D. L. Nelson, C. L. McEvoy, and T. A. Schreiber. The University of South Florida word association, rhyme, and word fragment norms, 1998.

[48] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577 –8582, June 2006.

[49] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, Feb. 2004.

[50] F. Nie, C. Ding, D. Luo, and H. Huang. Improved minmax cut graph clustering with nonnegative relaxation. *Machine Learning and Knowledge Discovery in Databases*, page 451–466, 2010.

[51] J. Onnela, J. Saramäki, J. Kertész, and K. Kaski. Intensity and coherence of motifs in weighted complex networks. *Physical Review E*, 71(6):065103, June 2005.

[52] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.

[53] G. Palla, I. J. Farkas, P. Pollner, I. Derényi, and T. Vicsek. Directed network modules. *New Journal of Physics*, 9(6):186–186, June 2007.

[54] M. A. Porter, J. P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

[55] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658 –2663, Mar. 2004.

[56] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1):16110, 2006.

[57] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327–7331, May 2007.

[58] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118 –1123, Jan. 2008.

[59] M. Rosvall and C. T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PLoS ONE*, 6(4):e18209, Apr. 2011.

[60] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, Aug. 2007.

[61] A. J. Seary and W. D. Richards. Partitioning networks by eigenvectors. *Proceedings of the International Conference on Social Networks*, 1:47–58, 1995.

[62] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[63] H. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2(2–3):135–148, 1991.

[64] H. A. Simon and A. Ando. Aggregation of variables in dynamic systems. *Econometrica*, 29(2):111–138, Apr. 1961.

[65] S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, Mar. 2001.

[66] D. J. Wales. Calculating rate constants and committor probabilities for transition networks by graph transformation. *The Journal of chemical physics*, 130(20):204111, 2009.

[67] D. J. Wales. Energy landscapes: some new horizons. *Current Opinion in Structural Biology*, 20(1):3–10, Feb. 2010.

[68] D. J. Wales and P. Salamon. Observation time scale, free-energy landscapes, and molecular symmetry. *Proceedings of the National Academy of Sciences*, 111(2):617–622, 2014.

[69] C. Walshaw and M. Berzins. Dynamic load-balancing for PDE solvers on adaptive unstructured meshes. *Concurrency: Practice and Experience*, 7(1):17–28, Feb. 1995.

[70] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, Mar. 1963.

[71] Y. C. Wei and C. K. Cheng. Ratio cut partitioning for hierarchical designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(7):911–921, July 1991.

[72] D. R. White and F. Harary. The cohesiveness of blocks in social networks: Node connectivity and conditional density. *Sociological Methodology*, 31(1):305–359, Jan. 2001.

[73] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, Nov. 1993.

[74] J. Yan and P. Hsiao. A fuzzy clustering algorithm for graph bisection. *Information Processing Letters*, 52(5):259–263, Dec. 1994.

[75] E. Yeger-Lotem, S. Sattath, N. Kashtan, S. Itzkovitz, R. Milo, R. Y. Pinter, U. Alon, and H. Margalit. Network motifs in integrated cellular networks of transcription–regulation and protein–protein interaction.

Proceedings of the National Academy of Sciences, 101(16):5934 –5939, Apr. 2004.

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

[77] Y. Zhou. On the eigenvalues of specially low-rank perturbed matrices. *Applied Mathematics and Computation*, 217(24):10267–10270, Aug. 2011.