Word Count: 7657

TP01: Detecting False Negatives in Networks

Candidate Number: 49845
Supervisors: Dr. N Jones and Dr. M Porter

The study of networks is of increasing importance across a large range of scientific fields, providing insight into the functional properties of many systems of complex interactions. Inaccuracies in network data are, however, an ever-present concern and it is important that we investigate means of minimising them. While good progress has been made in detecting random errors, much less work has focused on the effects of bias, when certain types of interactions are more likely to be misrepresented than others. In this work I address this problem by examining how well an existing error detection method deals with this kind of systematic error. I found that introducing a bias in the errors detrimentally affects the performance and that the details of this performance can provide insight into network structure. Inroads were also made in determining specific network properties that affect the efficacy of the method.

I. INTRODUCTION

Networks are an extremely concise and powerful way of representing systems of complex interactions and they have been applied in many diverse disciplines: network methods have been used in the study of neural structure, food webs, online social groups, the World Wide Web (WWW), airline transportation systems, and protein interactions, to name just a handful[1–6].

A network is a graph which corresponds to a real-world system. In mathematics, a graph is a pair, $G = (V, E)$, where V is a set of vertices (nodes), and E is a set of edges (links), two-element ordered subsets of V. In a network, vertices and edges represent, respectively, individual components of the system and their interactions (see fig. 1). The structure of a network is represented by an adjacency matrix, A, in which element A_{ij} indicates the presence or absence of an edge between vertices i and j. The number of adjacencies (links) for a given node is referred to as its 'degree' and is a widely used property in characterising networks. There are many different types of network, each appropriate in a different context[7]. However, in this work only simple networks are considered: those which are connected, undirected and unweighted.

The wide applicability of network methods stems from the fact that a network is essentially a representation of an interaction matrix. To understand the dynamics of the system, therefore, it is important to understand the network structure. One of the most powerful aspects of this approach is that it strips away field-specific complications and allows useful parallels to be drawn between disparate areas. In recent years the field has developed considerably with the application of a physical approach; current network science has much of its foundation in methods from statistical physics[8,9].

Of major concern in the study of networks is the reliability of the empirical data used to construct them. In many contexts there are large uncertainties in the means of data gathering. For example, modern methods used in constructing protein interaction networks have been shown to be highly error-prone: in 2002 von Mering et al. reported that "more than half of all current high-throughput [interaction] data are spurious"[4,13]. In general, sources of error can arise due to different effects, such as incomplete network sampling (inherent in web-crawling methods for acquiring data on the WWW[14]), and biased survey responses in social network studies (for example, two people may have different ideas of what constitutes a 'friend')[15]. Questionable data means that any conclusions are uncertain at best, and at worst positively misleading.

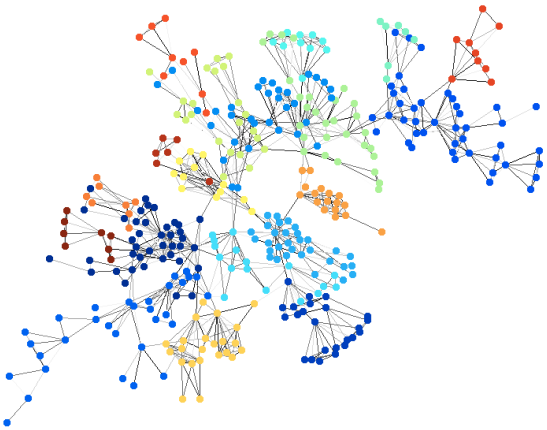

Figure 1: The largest connected component of the network of citations amongst network scientists[10]. Each node (vertex) represents an individual working in network science, and each link (edge) represents at least one instance of co-authorship between individuals. I used the Kamada-Kawai graph visualisation method[11], and colours represent structural community membership. These were determined by maximising the quality function 'modularity' (see section III A) using a greedy method after Blondel et al. 2008[12].

A natural way to combat these inaccuracies in the data is to endeavour to improve the experimental technique used in their acquisition. This can be very difficult however, often requiring labour-intensive methods of manually verifying each interaction[16]. Even when possible, experimental improvements can only ever apply to a small number of network types. It is important, therefore, to develop ways of using currently available data to predict which interactions are incorrectly represented: by identifying the most likely errors in the data, ex-

perimentalists can focus more on individually re-testing just those links, and concentrate less on those which are likely to be correct. This could dramatically reduce the time it would take to obtain accurate network structures, and improve the quality of inferences made based on them. It is also possible that prediction methods could shed light on the evolution of time-varying networks, in which links between nodes are formed and broken over time, as they might predict the changes which are likely to occur. This could have applications ranging from the prediction of the spread of viral epidemics to friend suggestion in online social networks such as Facebook[3,17].

There are many approaches to error prediction that have been discussed in the literature, most of which are targeted at a specific domain, such as protein-protein interaction networks[18] or social networks[15,19]. There have, however, been recent advances in domain-independent methods which rely on factors such as the overall network topology (for a review, see ref. 20). Simple examples of topological prediction methods are the 'common neighbours', 'degree product' and 'shortest path' methods[21], which predict links based on joint properties of the nodes. These methods can make useful predictions; however, they make strong assumptions about the structure of the network, and their efficacy varies accordingly.

Two more versatile methods are those proposed recently by Guimerà et al.[22], and Clauset et al.[23] Both of these methods make assumptions as to a general model which may be used to represent the network (stochastic block models[24] in the case of the former, and hierarchical random graphs, HRGs, in the latter), and predict links based on the likelihood of a link's existence given that model. The two differ in their details, but both methods have shown impressive results.

Previously, work using such model-based topological methods has concentrated mainly on prediction performance with regards to random errors[22,23]. This, however, is not representative of the mechanisms that give rise to the errors which can often have systematic origins which depend on the underlying structure of the data. For example, web pages that are less frequently linked to are more likely to be omitted from a web-crawling search[14], and responses to social surveys can be systematically biased[15]. Other data collection methods are likely, to a greater or lesser extent, to be skewed in unforeseen ways.

It is not clear how this kind of bias in the errors would affect a given method's ability to detect them. In this work I simulate both random and systematic errors over a range of real-world and artificially generated networks, and investigate the performance of Clauset et al.'s HRG method in predicting these errors. This gives insight into the limitations of the prediction algorithm, and also allows us to probe the structure of the networks themselves.

I will proceed in section II by first explaining the way that errors were simulated, describing some of the limitations encountered in the process. I will then introduce the HRG prediction algorithm in detail in section III in order to better understand the results presented in section IV. Here I will look in turn at each of the network types used, explaining the choice of each. Finally, I will conclude in section V with a discussion of the main results and suggestions for future directions of study.

II. NETWORK ERROR SIMULATION

In order to be able to evaluate the accuracy of error detection, it is necessary to compare the predictions with a known set of errors. This can be achieved by selecting an initial network which is taken to be the 'true' structure (i.e. we assume that there are no errors in this original data), and then introducing errors to produce an 'observed' network which is passed to the prediction algorithm. The accuracy of prediction is then calculated by scrutinising how well these known errors are identified. Both the choice of initial network and the method of error introduction are important aspects to control. In this section I will discuss the latter, explaining the choices made in this study.

There are many conceivable ways in which errors can be introduced into network data, however here I focus exclusively on creating missing links. The reasons for this are twofold: firstly, due to the topological nature of the prediction algorithm, it makes sense to consider edge-based errors, taking the observed nodes to be the true ones; secondly, limitations in computer time made addressing spurious links impractical. This computational intensity is because in order to add a link in a biased way, metrics for all potential links would need to be calculated: for a sparse network (number of edges, $m \ll$ number of possible edges, $\frac{1}{2}n(n-1)$, where n is the number of nodes) the computer time is approximately $\mathcal{O}(n^2)$. Having decided to concentrate on missing links, a method of selecting links to remove from the original network is needed.

The most obvious way to remove links is to do so uniformly at random, with all edges having an equal probability of removal. This method can provide insight into the effects of stochasticity in data collection, and the ease of detecting random network damage. This is the method that has been used in previous studies[22,23], and I also implemented it here. Purely random errors are of limited use, however, as they ignore the possibility of bias in the data: the possibility that errors occur systematically in the data collection process. In this work, in order to address this I also removed links according to a set of specific metrics, each of which are discussed in section II A below.

Once each link is labelled with a value according to the chosen metric, we can choose to then remove them in two ways: stochastically, with probability (inversely) proportional to the link's value; or in an ordered fashion, starting with the largest (smallest) value and working down (up). Although a stochastic method is likely to be a better reflection of real-world processes, the speed of the prediction algorithm made it impractical to carry out the multiple runs which are necessary to obtain a reasonable average when using a probabilistic method. Instead

an ordered method was used on the basis that these results would give a 'worst case scenario' of the response of the algorithm to different systematic sources of error. If any links shared the same value, their ordering was determined at random.

In real-world link-prediction applications it is improbable that we would use a topology-based approach to predict links to nodes which are entirely unconnected to the network, since there is no topological information available for these nodes. In the light of this, I worked only with connected network components, discarding nodes (and their former links) from the system when they became isolated by link removal. I also ceased to remove links when the network dropped to fewer than 10% of its original number of nodes.

Discarding isolated nodes has important implications for the prediction accuracy: since there is no topological information available for an isolated node, any topology-based prediction algorithm is bound to estimate all possible links to it as having a very low probability of existing. Since the truth is the reverse (there is in fact a certainty that at least one link was originally connected to the node), failure to discard these nodes results in an accuracy which is lower than it should be.

While discarding isolated nodes makes little difference for random removal, it is of great importance when targeted removal is applied. For example if links to nodes with high degree are targeted, the network fragments much faster than if links had been removed at random[25]. This would result in an extremely low prediction accuracy if isolated nodes were not ignored.

A. Targeted Removal

To simulate systematic errors in network data, links were removed according to their value with respect to one of four different metrics. The metrics used were based on ones common in the literature[26]: *betweenness*, *closeness*, *clustering coefficient*, and *degree assortativity*. Where relevant, I used slightly modified versions of these so that they apply to individual edges, rather than to vertices or the network as a whole. Several of these metrics have been used in the study of community structure[27–30](see section III A), and since the HRG method relies to a large extent on such structure, they offer a direct way of probing the response of the algorithm.

The 'betweenness' of an edge is a natural extension of vertex betweenness introduced by Freeman[31]. It is calculated as a weighted sum of the number of geodesic (shortest) paths between nodes which run through the given edge. Edge-betweenness can be viewed as a measure of how important a link is in connecting different sections of a network, and is well known from Newman and Girvan's use of it in identifying community structure[28].

Also based on the shortest paths through a network, closeness has not, to the best of my knowledge, previously been applied as an edge-metric. Conventionally, the closeness is the inverse of the harmonic mean distance of any given node from all of the other nodes in the network, and is given by[7]

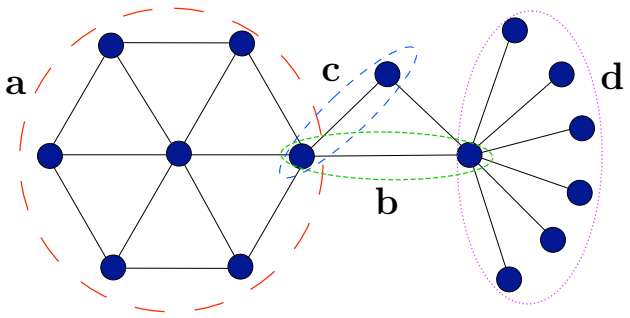

Figure 2: A simple network to demonstrate the metrics being used. **a.** All the edges in this group have high clustering coefficients: they all take part in the maximum number of 'triangles' possible. **b.** This edge has both high betweenness and closeness. It also has low asymmetry. **c.** Similarly to **b.**, the closeness for this edge is high, however its betweenness is very low since almost all paths are shorter if they do not take this detour. **d.** Edges in this group have high asymmetries: they run between nodes with very dissimilar degrees.

$$L_i = \frac{1}{n-1}\sum_{j\neq i} d_{ij}^{-1}, \tag{1}$$

where n is the number of nodes in the network, d_{ij} is the geodesic distance from node i to node j, and L_i is the closeness of node i. In adapting this metric to apply to links instead of nodes, I treat an edge (i, j) as if it were simultaneously at both node i and node j, so that the edge-closeness is given by

$$l_{ij} = \frac{1}{n-2}\sum_{k\neq i,j} \frac{1}{\min(d_{ik}, d_{jk})}. \tag{2}$$

This is equivalent to using the definition in eqn. 1, if in that equation d_{ij} is redefined as counting the minimum number of *nodes* passed through between *edges* i and j, instead of the minimum number of *edges* between *nodes* i and j. The closeness measures the ease of communication between an edge and the rest of the network.

The clustering coefficient, as one might expect, gives a measure of how tightly grouped a network is. In an extension to a commonly used vertex-based definition[32], Radicchi et al.[30] define the edge clustering coefficient as

$$C_{ij} = \frac{z_{ij}+1}{\min(k_i-1, k_j-1)} \tag{3}$$

in which k_i is the degree of node i, and z_{ij} is the number of 'triangles' (fig. 2) to which edge (i, j) belongs. The coefficient gives the fraction of triangles to which an edge contributes out of the total number possible, given the degree of the nodes it connects. The addition of 1 in the numerator is to remove the degeneracy that occurs when $z_{ij} = 0$, which would give zero clustering, irrespective of k_i and k_j.

The last of the metrics used in this work is based on the degree assortativity, a widely used measure that

quantifies how similar the degrees are likely to be on either end of a given edge. As a global network property, it is given by the Pearson correlation coefficient between the degrees of nodes at either end of a link[33]. This has been found to characterise the the resilience of certain types of networks, with assortative networks being resistant to damage, and disassortative networks being vulnerable[33]. In adapting this measure to apply to individual edges, I used a modification of a metric introduced by Saavedra et al.[34], so that the edge-asymmetry is given by

$$a_{ij} = \frac{|k_i - k_j|}{k_i + k_j}. \tag{4}$$

It should be noted that high average asymmetry corresponds to low degree assortativity, and thus to a disassortative network.

III. THE HRG PREDICTION METHOD

Having summarised how errors were introduced to network data, I will now explain the method used to predict them. In this study, I used the algorithm of Clauset et al.[23], which is based on a model of Hierarchical Random Graphs (HRGs). This method fits an HRG to the observed network, and uses this to predict the missing structure. This is done using a Markov Chain Monte Carlo (MCMC) method[35] to converge on the set of HRGs with the highest likelihood of generating the observed network. I will first explain the term 'community structure' before describing the HRG model. An understanding of both will be essential in the analysis of my results. I will then discuss how predictions are made, and how the accuracy is calculated.

A. Community Structure and Hierarchical Random Graphs

The phrase 'community structure' refers to any partition of the set of nodes into subsets called 'communities', usually taken to be non-overlapping. This partition is normally chosen in such a way that each community has a higher density of internal than external links. Community determination is important in linking network structure with function. However, deciding exactly how best to partition a network is a non-trivial problem and a more detailed definition of community structure is necessarily algorithmic.

There are many algorithms for determining the best partition[27,29], a widely used class of which works by maximising the quality function 'modularity'[28]. The modularity provides a measure of the extent to which connections are within, rather than between, communities, compared to what we would expect if the communities were randomly allocated. As such it can be a useful measure of how clear the communities are in any given partition.

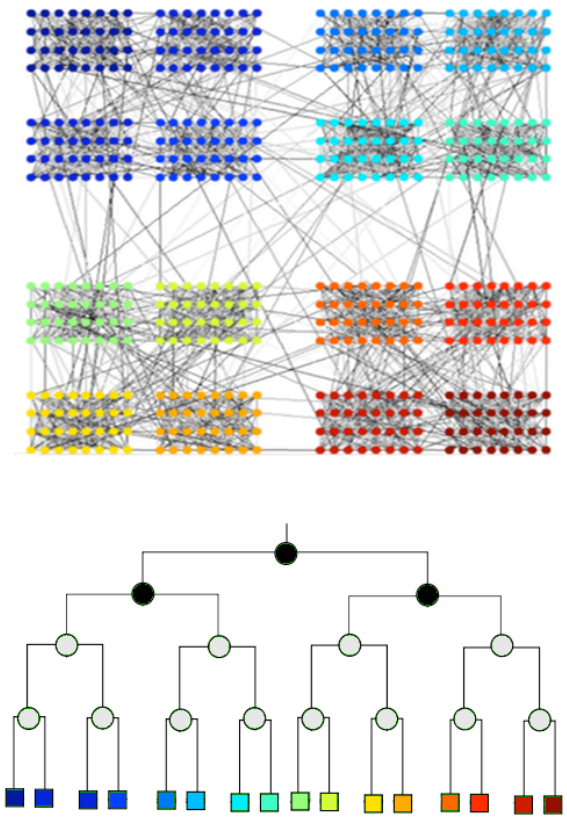

Figure 3: A network displaying hierarchical structure: each of the coloured 32-node communities is also part of a larger 128-node one. The binary dendrogram is one of the possible HRGs representing this hierarchy. The shading of internal junctions represents the probability of links between nodes from either sub-branch, with black low and white high. Here I have truncated the dendrogram so that leaves depict the smallest communities rather than individual nodes. I based this figure on that of Clauset et al.[23], using a network structure after Lancichinetti et al. 2009[36].

Real network structure, however, can be more complicated than the single partition found by conventional community detection methods. For example, structures may be overlapping, with a node best described as a member of several communities, or hierarchical, where communities are nested within one another on different scales. Such features can provide important information about the functional organisation of a network, and as such they are important to consider in any model of network structure. The HRG model deals explicitly with the latter: it is a way of generating random networks with hierarchical structure built-in.

An example of a network with hierarchy can be seen in fig. 3: clusters of nodes are coloured by their communities (as determined by modularity maximisation); however, we can also see larger scale groupings, corresponding to higher levels of hierarchical structure. In general, this structure can be either assortative or disassortative at each level, meaning that a set of nodes may be grouped because they are more (assortative) or less (disassortative) likely to have links between them. An HRG builds in this structure using a binary dendrogram (a tree diagram, as in fig. 3) in which each leaf corre-

sponds to a node in the network. Each internal junction, r, is assigned a value p_r corresponding to the probability that nodes on either side of it share a link. In this way, random networks can be generated with a predetermined hierarchy, mixing both assortative and disassortative structure.

B. Link Prediction

Using MCMC, the prediction algorithm is able to sample from the space of HRGs with probability proportional to the likelihood of the HRG generating the observed network. For each sampled HRG, every possible link, (i, j), has a probability p_r of existing, where r is the lowest common ancestor of leaves i and j in the dendrogram. By tallying up these probabilities for each link over a certain number of samples, a list of possible missing links is assembled, ranked in order of their probability. In sampling across multiple dendrogram structures, this algorithm explicitly takes account of the fact that a given network can have more than one plausible hierarchy.

From this ranked list of possible missing links, the prediction accuracy was calculated as the probability that any randomly chosen *actual* missing link (a false negative in the network data) is ranked higher than a randomly chosen *possible* missing link (a true negative). The accuracy is thus a value in the range $[0, 1]$, where prediction at random would have an accuracy of 0.5: with possible missing links randomly ordered, there is an equal probability that any false negative is ranked either above or below a randomly chosen true one. This measure of accuracy is equivalent to the area under the ROC curve (the AUC statistic) and is the same as that used by both Clauset et al. and Guimerà et al.[22,23].

I found that a reasonable number of HRG samples to take was 10000, since increasing the number above this point did not significantly increase the consistency of the accuracy. The standard deviation in the accuracy at this number of samples was 0.006.

We can see that the link prediction algorithm is based on the fundamental assumption that the HRG model is a good way of describing the network structure. This then is the main limitation of the method, as there are several features of network structure which cannot easily be captured by an HRG, such as overlapping communities, or indeed structure which is non-hierarchical. On top of this, with links missing, the observed network topology may be misleading in the recovery of the original structure, though this is a limitation for any topology-based method. As my results will show, these limitations can account for much of the variation in the performance of the algorithm on different types of networks.

IV. RESULTS

In presenting the results, I will introduce in turn each class of network that was used, and will discuss the performance of the link prediction on it. I will then discuss some general results which apply to all network types. It is worth noting that due to restrictions in computer time, and the slow speed of the algorithm, the largest network I was able to consider contained less than 400 nodes, while most of them contained 100 or less. This heavily influenced my choice of both real and generated networks.

Specifically, for a network of 100 nodes it took $\sim$40s to take 10000 samples on a machine with an Intel Pentium 4 HT 3 GHz processor and 2 GB RAM. The number of possible missing links is $\mathcal{O}(n^2)$ for a sparse network, where n is the number of nodes, so we can take this as an estimate of the computational complexity of the HRG prediction algorithm. With the multiple runs required ($\sim$180 for each network), we can see that times quickly become impractical for networks much bigger than those I considered.

In the interests of greater clarity in the following sections, I will here explain the way that results were taken, and introduce a measure that I use in summarising much of the data. For each network being studied, nine link removal methods were applied: random removal, followed by removal according to each of the four metrics discussed above in both ascending and descending order. Thus 'clustering ascending' refers to removal of links in order, starting with those with the lowest clustering coefficients. For each removal method, a specified fraction of links was removed (where the fraction is calculated ignoring links to isolated nodes, as discussed in section II), and the prediction algorithm was run. This enabled the construction of a profile of the accuracy at different levels of damage according to different removal methods across a range of networks. These profiles form the basis of my results.

In order to simplify the data for ease of comparison and analysis, I used a *characteristic accuracy* (CA) to reduce a series of accuracies at different levels of error to a single number. This was calculated as the normalised area under the curve of an accuracy profile. Not all accuracy curves were of the same length: with isolated nodes being discarded, the remaining connected networks became too small at different levels of link removal, depending on the specific removal method. For this reason, areas were normalised by $\frac{1}{N-1}$, where N is the number of data points in the curve. The baseline is shifted so that $\mathrm{CA} = 0$ corresponds to random prediction.

Unless otherwise noted, all uncertainties are given by the sample standard deviation about the mean. Correlations were calculated using the Pearson correlation coefficient, and p-values are computed using a Student's t distribution for a transformation of the correlation[46].

A. Random Networks

The Erdős-Rényi random graph is one of the simplest and best understood of all artificial networks structures[37]. It is generated, given a network size, by linking any two nodes at random, with fixed probability, p.

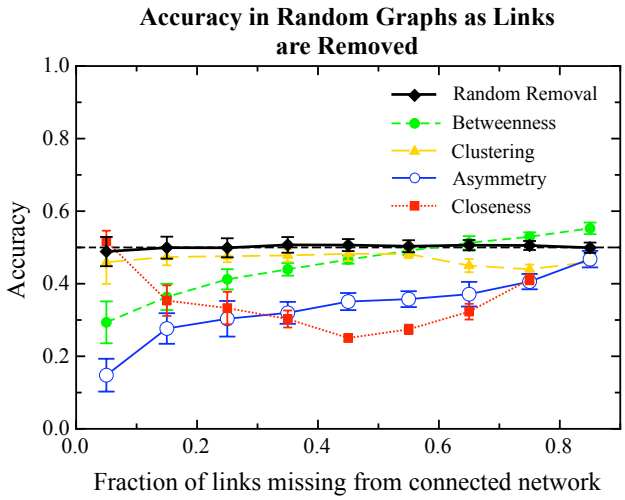

Figure 4: The accuracy of link prediction as links are progressively removed in Erdős-Rényi random graphs (100 nodes, $p = 0.2$). All targeted removals shown are carried out in descending order with regards to the given metric. Note that almost all targeted link removal produces worse-than-random accuracy.

This algorithm results in a network without any natural partitioning of the nodes into communities. Since the HRG model relies on detectable community structure when fitting the observed network, random graphs provide a useful baseline against which all other results can be compared.

Here I consider a set of random graphs with 100 nodes, generated with $p = 0.2$. This value was chosen in order to create networks which are almost certainly connected (this is the case when $p > \ln(n)/n$, where n is the number of nodes in the network)[37], and have a reasonable quantity of links available to remove. The value of p was kept relatively low in order to avoid saturation. Exactly how the prediction accuracy would depend on p is unknown, and would be worth further study. All results are averaged over three independent instances of networks with these parameters, allowing errors to be calculated even when links were removed in an ordered fashion for the targeted link removal. Computational costs limited the number of network instances that could be considered.

As expected, we find that when links are removed uniformly at random from these networks, the accuracy is consistent with random prediction (fig. 4). Since there is no clear community structure in the 'true' network, there is no structure to which the HRG model can be fitted. This results in the prediction of links at random, half of which are correct by chance.

In contrast, almost all targeted link removal methods result in an accuracy which is considerably worse than random. This is likely because, by preferentially removing links with specific properties, the remaining network develops a spurious structure. For example, by removing links with high betweenness, we leave behind groups of nodes which are more separate from the rest of the network than from each other, thus creating clear communities where there were none before. The HRGs are then fitted to this structure and are consistently misled by it in making predictions. The varying extent to which the accuracy is less than 0.5, depending on the metric, would then reflect the extent to which they create this spurious structure; removing links with high clustering coefficients clearly does this much less than removing those with high asymmetry, for example.

The aggregate positive trend in the asymmetry and betweenness curves can be explained if only those links which are removed first, ones with the highest values of these metrics, leave distinct structure behind. All links which are removed from that point on act to degrade this structure, thereby reducing the extent to which the algorithm is misled.

We also notice that although starting out as for random removal, accuracy in detecting links with high closeness decreases until roughly 45% of links are removed before then increasing again. This implies that the links with the very highest closeness are more or less randomly distributed throughout the network, while those with slightly lower values are more instrumental in leaving spurious structure behind. This continues to be the case until the point at which all further removals only act to degrade whatever false structure has been created.

Entirely different qualitative behaviour in the accuracy of prediction for systematic errors, even in a network with no well-defined initial structure, goes a long way to help justify the motivation behind this study. It helps corroborate the premise that systematic errors will provoke a notably different response from link prediction algorithms. The finding that structure is quick to appear after damage to as little as 5% of the links, suggests perhaps that random structures are particularly fragile; for many kinds of systematic damage, they are quick to change their fundamental properties and develop a non-random structure.

B. Community-Based Networks

Given the importance of community structure for the HRG method, it is useful to study its performance when applied to a suite of networks generated with this structure explicit. In 2008, Lancichinetti et al. proposed a method for constructing networks with communities, designed as a benchmark for testing community detection algorithms[38]. I made extensive use of these networks in testing the HRG prediction algorithm.

In Lancichinetti et al.'s network model, both the degree distribution and the community size distribution follow power laws ($p(x) = x^a$, where $p(x)$ is the distribution of x and a is a constant). Nodes are placed inside communities and then form links either without or within that community at a ratio μ, the mixing parameter. Thus community structure disappears at $\mu = 1$, and becomes indistinct for $\mu \gtrsim 0.5$. Parameters which can be adjusted are: number of nodes n, average degree $\langle k \rangle$, maximum degree $k_{\max}$, exponents for degree and community size distributions α, β, and mixing parameter, μ.

In the networks used, parameters held constant were $n = 100$, $k_{\max} = 20$, $\alpha = 3$ and $\beta = 2$. I chose this

Removal Method	Corr. between μ and CA ($\langle k\rangle$ const.)		Corr. between $\langle k\rangle$ and CA (μ const.)	
	r	p	r	p
Random	-1.00	0.0003	0.49	0.32
Betweenness Descending	-0.87	0.0542	**-0.78**	**0.068**
Betweenness Ascending	-0.99	0.0007	0.55	0.26
Closeness Descending	-0.99	0.0010	**-0.97**	**0.0018**
Closeness Ascending	-0.96	0.0089	-0.58	0.23
Clustering Descending	-0.96	0.0110	0.10	0.84
Clustering Ascending	-0.87	0.0537	**0.85**	**0.030**
Asymmetry Descending	-0.99	0.0008	-0.10	0.84
Asymmetry Ascending	-0.99	0.0006	**-0.95**	**0.0043**

Table I: Correlations of mixing parameter and average degree with the CA for each removal method. As μ is varied, $\langle k\rangle$ is held at 7; as $\langle k\rangle$ is varied, μ is held at 0.3. Statistically significant correlations are ones with low p-values. Mixing Parameter is strongly correlated with all removal methods, while average degree only significantly affects accuracy for four of them, highlighted here in bold.

value for the power law exponent of the degree distribution, α, in order to reflect real network properties; many networks are known to have power law degree distributions with exponents between 2 and 4[39]. I tested link prediction across different instances of these networks, varying first the mixing parameter, keeping $\langle k\rangle = 7$, and then the average degree, keeping $\mu = 0.3$. By changing the mixing parameter, I was able to probe the effect of varying how distinct the communities were; in changing the average degree I looked at the effect of different levels of connectivity.

First of all, it should be noted that, unlike with the Erdős-Rényi networks, the algorithm performs consistently well for most methods of link removal, at least for $\mu \lesssim 0.5$, giving accuracies significantly better than random (average CAs are above 0.1 for all removal methods apart from closeness ascending; see section IV D for a discussion of why this method does poorly). This tells us that the presence of even damaged community structure in the network enables much better prediction accuracy. Similarly to the random networks, random errors are the easiest to predict.

There was found to be a significant correlation (r-value $\cong -0.98$, p-value $\cong 0.0023$) between the mixing parameter and the accuracy when all other variables were held constant (see fig. 5). This is expected given the reliance of the HRG method on a well-defined community structure. Accuracy was much higher for low values of μ, when there were clearly defined communities, showing that the HRGs were able to provide a good fit to the observed network. As the structure becomes more washed out with increasing μ, the accuracy drops accordingly.

When the mixing parameter was held constant at $\mu = 0.3$, it was found that for most link removal metrics, varying the average degree made very little difference (p-value > 0.2; see table I, columns 3–4 for details of correlations for individual metrics). This shows that community structure is of greater importance than

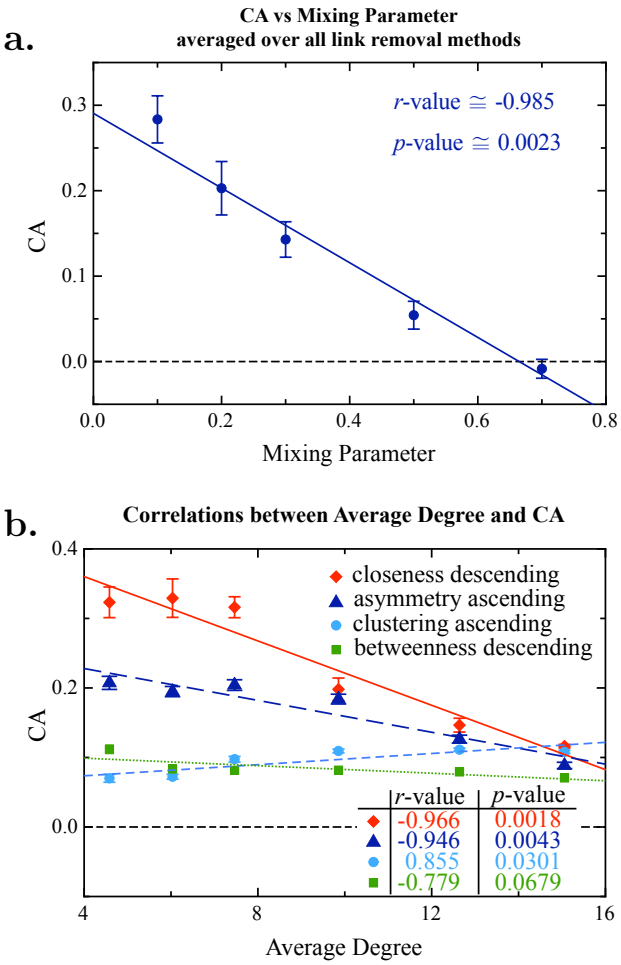

Figure 5: **a.** The negative correlation between mixing parameter and CA, demonstrating that the HRG algorithm does better when community structure is more defined. Here the CAs are averaged over all link removal methods. **b.** Correlations between average degree, $\langle k\rangle$, and the CA for link removal according to betweenness and closeness in descending order, and asymmetry and clustering in ascending order.

the connectivity in determining the performance of prediction. There were, however, found to be four removal methods which showed strong correlations with $\langle k\rangle$: closeness and betweenness descending, and clustering and asymmetry ascending. This tells us that with increasing connectivity, errors in either highly 'close' links (those with short paths to the rest of the network), highly symmetric ones (connecting nodes of similar degree), or links with high betweenness (many paths running through them) are increasingly damaging to the overall structure of the network, while the absence of low clustering links becomes easier to detect.

While the reasons behind these patterns are not entirely clear from my analysis, it is clear that they give important information about the network structure. It is possible that they are due to the fact that at low $\langle k\rangle$, the networks are simply too homogeneous for the effects of the targeted removal by these metrics to become apparent. By carrying out more tests on a wider range of networks, we would hope to be able to probe these trends more closely.

Network	Average CA
Net. Science	0.31(2)
Les Miserables	0.25(3)
Dolphins	0.20(1)
Karate Club	0.16(2)
Cat Brain	0.12(1)
Adj. Noun	0.06(1)

Table II: The CA in real-world networks, averaged over all removal methods. Numbers in parenthesis give the error in the last digit. We can clearly see that link prediction performs best on the citation network, and worst on the semantic network. See main text for discussion.

C. Real World Networks

In addition to looking at the response of the HRG prediction to artificial network structures, it is of interest to apply it to real-world networks. Real networks can vary in size from just a few nodes in small social networks, up to tens of millions or more in networks such as the WWW. Being restricted to networks of no more than 400 due to computational time, therefore, was a major limitation in the kinds of networks that could be studied — for instance, there are no reported protein-protein interaction networks of this size. The six networks that were used consisted of a karate club social network[40], a dolphin social network[41], a semantic network of adjacencies between adjectives and nouns[42], a character adjacency network from the novel *Les Miserables*[43], a coarse grained description of a cat brain, and the citation network of network scientists[10]. All of these networks have been well studied in the literature, and are considered to be of high quality, in the sense that they are as error-free as we could hope for. This makes them good candidates for use as 'true' network structures in this study.

Due to the complicated nature of real systems, the small number of sample networks studied, and their disparate properties, it is difficult to uncover many underlying trends in the link prediction accuracy for these networks. However two patterns can be noticed. Firstly we find that, as for the random and community-based networks, removal of links at random allows the algorithm to predict links with the highest accuracy: for several of the networks (net. science, Les Miserables and cat brain), this remained higher than 80% until more than 50% of the original links had been removed from the network. Secondly, as can be seen from table II, there was a distinct ordering to how well links could be predicted, on average, in each of the networks. It was found that there is a strong positive correlation (r-value $\cong$ 0.75) between the average CA and the average path length between nodes in these networks. Although this correlation would not usually be considered statistically significant with a p-value of 0.083, this relationship would make sense intuitively: with nodes widely separated, the local neighbourhood of each one is more clearly identifiable, and it is easier for the algorithm to pick out the likely candidates for missing links. It is plausible that this correlation would become significant if a larger number of networks were considered.

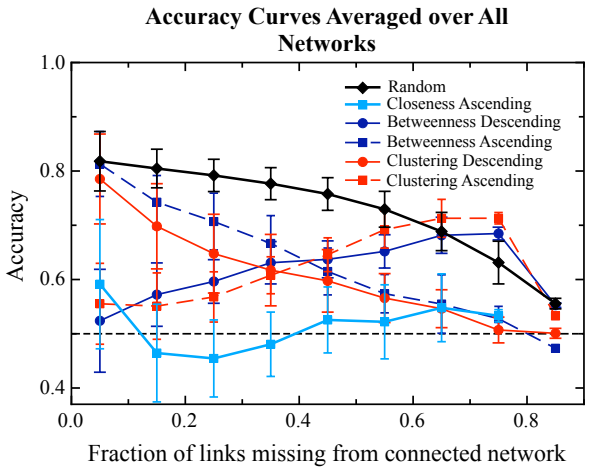

Figure 6: The overall prediction accuracy for several removal methods, averaged over all networks. Random errors give the best accuracy, while those according to closeness ascending give the worst. Also there is a clear inverse relationship between the clustering and betweenness curves, with high betweenness corresponding to low clustering coefficients. See main text for a discussion of these points. Error bars are the standard error of the mean.

D. Results Over All Networks

Having described the response of the HRG method on individual networks, I will now present results averaged over all networks considered here to see if there are any features which apply more generally. These features would be useful in providing indications of trends we might look for when examining new networks.

The first point to extract from the results as a whole is that targeted link removal almost always produces worse accuracy than random removal. This is a non-trivial result, as it is plausible that the structure could be accentuated by systematic removal rather than degraded, which would have resulted in higher accuracy than for random removal. This tells us that the type of error present in the network data is very important in allowing us to reconstruct the 'true' network. In particular, errors which give rise to misleading topologies can seriously hinder our ability to do so.

Finding that random errors are least damaging to link prediction is important since it tells us that this accuracy can provide an indication of the upper limit to the possible performance of the algorithm. We can therefore expect that in most real applications, where there is likely to be at least some systematic bias in any errors, the link prediction will not be as reliable as it has appeared to be in the previous studies by Guimerà et al. and Clauset et al. It would be useful to test if this result holds in general for other link prediction methods — if not, it would be an important disadvantage of the HRG method.

In contrast to the result that the highest accuracy is achieved in predicting random errors, we find that the worst accuracy (among the quantities that I considered) is obtained when links are removed in ascending order based on their closeness. The average CA for this re-

moval method was only 0.02 ± 0.02, which is only just better than prediction at random. The reason that the HRG method performs so badly in detecting these errors must lie in the nature of the links removed: here they are those which are furthest, on average, from the rest of the network.

In light of this, it seems likely that this behaviour is an artefact of the same effect which contributed to the decision to disregard isolated nodes: that with no topological information about the node, all its possible links are given a disproportionately low probability of existing. Although those nodes were ignored, any group of nodes with connections remaining were retained, regardless of whether they were disconnected from the main component of the network or not. This was done because it is not always clear what the 'main' component is, and because a network could otherwise become fragmented after only a few links were removed. Since ascending closeness removal preferentially deletes links on the periphery of the network, it is likely to leave small detached groups of two or three nodes which, as with completely isolated nodes, lead to poor prediction accuracies. This would explain the poor response to this type of link removal.

One final overall trend can be clearly seen in fig. 6: the inverse relationship between accuracies for betweenness and clustering based removal. We find that the 'betweenness descending' and 'clustering ascending' curves are strongly correlated (r-value $\cong 0.94$, p-value $\cong 2.1 \times 10^{-4}$), and there is a similar correlation between the same two metrics' curves when removals are oppositely ordered. This result implies that edge-betweenness and edge-clustering coefficient are similar metrics, picking out links which have the same qualitative effects on the structure of the network, though doing so in an inverse order. Indeed, it makes intuitive sense for this to be the case: links which take part in a small fraction of the possible 'triangles' are also the ones which have a high number of paths running through them, since there are few alternative paths to take in the local vicinity.

The clarity of this relationship in the response of the HRG algorithm shows that link prediction can provide insight into network structure over and above simply recovering missing data. Although indirect, this could be a useful tool for probing the interplay between the many interdependent metrics that can be applied to networks.

V. DISCUSSION

In this work I addressed the problem of detecting systematic errors in network data. Using a leading detection method, which had previously shown promising results[23], I investigated the accuracy in detecting errors introduced artificially according to five informative metrics.

In doing this, I confirmed that the presence of systematically biased errors in the data results in a qualitatively different accuracy in their detection, with consistently worse performance for systematic errors than for ones which were randomly introduced. In probing how network properties affect link prediction, I found that community structure is a key factor in determining how well the HRG method will work. For different metrics, error depended in a correlated way with the clarity of structure in the network. Finally, I found a strong inverse relationship between the betweenness and clustering coefficient metrics with regard to the ease of predicting links with these properties. This highlights their similar relationship to the network structure.

Previous work investigating the prediction of errors in networks has found that the HRG method performs extremely well in this task when detecting random defects. My results agree with this, often showing very high prediction accuracy. In looking at errors which are systematically biased according to the underlying structure of the true network however, it was found that entirely different behaviour was displayed (fig. 4).

The hypothesis underlying this work was that qualitatively different types of network errors will vary in their ease of detection. This was indeed borne out by my results, which showed that across all network types, randomly missing links were easier to predict than biased ones, even in random networks in which there is initially no well-defined community structure on which to base predictions (see fig. 6). Given an understanding of the HRG prediction method which was used, this result implies that the presence of any systematic errors in network data will result in significantly misleading structure in the observed network — often prediction accuracy was worse by as much as 50% for biased errors relative to random ones (for an example, see closeness (red) in fig. 4). With an increasing reliance on network methods in numerous areas of science[44], my results serve to highlight the importance of eliminating any systematic errors from data acquisition.

With this in mind, my finding that random networks with no community structure were quick to develop spurious structure with the introduction of systematic errors has important consequences. The implication is that the observed data for an unstructured network with structured errors would appear to have structure. This could lead to entirely inaccurate conclusions, and so it is important that we develop ways of distinguishing misleading from true structure. It is possible that further investigation into the prediction of biased errors could uncover ways of achieving this.

Perhaps the most important avenue of research building on this work would be to determine whether this inability to predict systematic errors is shared by other link prediction methods, such as those mentioned in the 2005 review article by Getoor et al[20]. If it were found that any performed significantly better in this regard, this would constitute a major advantage over other more affected methods.

In order to simplify the computations, it has been assumed here that the most important errors are caused by the under-representation of interactions actually present (false negatives). However, this is not necessarily the case: it has been suggested that in protein-protein inter-

action networks, as much as one in every ten reported interactions is spurious[18]. It has also been found that the HRG method is particularly bad at predicting such spurious links, due to over-fitting of the observed networks[22]. An important line of future study would be to investigate this further to see how this changes if links are introduced systematically rather than at random. It is possible that this would produce even worse accuracies, which would render the method effectively useless for application in spurious link detection.

One of the restrictions in the results found here is the limited number of types of systematic error which were dealt with. I only considered the effect of bias based on four properties of edges, while there are many possible methods that we can imagine using. These include, but are not limited to, using node-, rather than edge-metrics or introducing errors depending explicitly on a pre-determined community structure. There are also other edge-metrics which could have been used. The specific effects of these other methods would be worth further study, and it would be interesting to see whether the underlying finding, that systematic errors are harder to predict, would remain true.

By using artificial networks with in-built community structure as a controlled environment, I was able to test the ease of error detection over a range of specific parameters. By varying the mixing parameter, which controls how clearly defined the communities are, I was able to test how the presence of distinct communities affected the accuracy. Since the HRG method relies on community structure to make its predictions, we expect that this would have a clear effect. This was indeed the case: a strong correlation between the mixing parameter and the accuracy was found, with consistently accurate error detection observed when the communities were most distinct (fig. 5). Since there is a direct relationship between the mixing parameter and the modularity, I expected to see a similar correlation with this property across all networks. Interestingly, this was not the case. Indications as to why can be found in the results of varying average degree: despite a constant modularity, the accuracy itself did not remain unchanged. Instead, effects caused by varying the connectivity complicated the results. Although it is only part of the story, then, it is nevertheless important to note that with the HRG method we expect to be able to predict links more effectively in networks with clearly defined communities: those with high modularity.

In building upon these results, an important factor will be in the study of networks with a wider range of properties, and a larger number of nodes, so that statistical methods can be more accurate. There are two requirements in order to do this: the ability to tune the properties of a network manually, and a means of overcoming the computational restraints which lead to the network size restriction. Much work has been carried out with regards to the first of these points, with models such as '$p*$' able to produce networks with many desired properties[45]. The easiest way around the second requirement, without simply using another prediction method, would be to parallelise the procedure. Since it is the multiple, independent runs of the algorithm which is time consuming when run sequentially, a parallel computation would speed up the process to a great extent. The limitation to this approach, naturally, is the number of processors available. As was noticed with regards to networks constructed from real data, it is likely that examining larger and more diverse networks would also reveal new patterns, and help strengthen those already observed.

One final conclusion can be drawn from the results of this work: by studying the accuracy of link prediction in response to targeted network errors we can do more than simply evaluate the effectiveness of an error detection method, but can in fact probe the relationships between different metrics. The correlations found here between accuracies for clustering and betweenness-based removal when applied in opposite directions imply that there is an inverse relationship between these two measures, at least for the networks I studied. This finding could have implications in other applications of these measures and it is possible that with further investigation, more relationships such as this could be discovered.

While analysis generally focusses on structure in network data, I have demonstrated the importance of studying the structure in the errors, showing that this can have major effects on features that are seen. By investigating the effects of different sources of bias in the data we can probe the interplay between them. This process could have the potential to deepen our understanding of how these different sources of error affect the structure of networks, and ultimately provide insight into the function of the systems they represent.

VI. ACKNOWLEDGEMENTS

I would like to thank Sumeet Agarwal, Anna Lewis and Phillip Staniczenko for their indispensable advice and guidance throughout this project. The code used for the HRG prediction algorithm can be found on Aaron Clauset's website at `www.santafe.edu/~aaronc/hierarchy/`, and that used in creating the Lancichinetti community-based networks is on Santo Fortunato's website at `sites.google.com/site/santofortunato/inthepress2/`. Also used were the MatlabBGL package and various network tools available from NetWiki: `netwiki.amath.unc.edu/`.

[1] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., et al. *Computer Networks* **33**, 309–320 (2000).

[2] Guimerà, R., Mossa, S., Turtschi, A., and Amaral, L. A. N. *PNAS* **102**(22), 7794–7799 (2005).

[3] Traud, A. L., Kelsic, E. D., Mucha, P. J., and Porter, M. A. *arXiv* (2008), physics.soc-ph/0809.0690v2.

[4] von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., et al. *Nature* **417**, 399–403 (2002).

[5] White, J., Southgate, E., Thomson, J., and Brenner, S. *Phil. Trans. R. Soc. B* **314**(1165), 1–340 (1986).

[6] Williams, R. J. and Martinez, N. D. *Nature* **404**, 180–183 (2000).
[7] Newman, M. E. J. *SIAM Review* **45**(2), 167–256 (2003).
[8] Newman, M. E. J. *Physics Today* **61**(11), 33–38 (2008).
[9] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. *Physics Reports* **424**, 175–308 (2006).
[10] Newman, M. E. J. *Phys. Rev. E* **74**, 036104 (2006).
[11] Kamada, T. and Kawai, S. *Inf. Proc. Lett.* **31**, 7–15 (1989).
[12] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. *arXiv* (2008), physics.soc-ph/0803.0476v2.
[13] Sprinzak, E., Sattath, S., and Margalit, H. *Journal of molecular biology* **327**, 919–923 (2003).
[14] Lawrence, S. and Giles, C. L. *Nature* **400**, 107–109 (1999).
[15] Kossinets, G. *Social Networks* **28**, 247–268 (2006).
[16] Martinez, N. D., Hawkins, B. A., Dawah, H. A., and Feifarek, B. P. *Ecology* **80**(3), 1044—1055 (1999).
[17] Pastor-Satorras, R. and Vespignani, A. *Phys. Rev. Lett.* **86**(14), 3200–3203 (2001).
[18] Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., et al. *Science* **302**(5644), 449 (2003).
[19] Robins, G., Pattison, P., and Woolcock, J. *Social Networks* **26**, 257–283 (2004).
[20] Getoor, L. and Diehl, C. P. *ACM SIGKDD Explorations Newsletter* **7**(2), 12 (2005).
[21] Liben-Nowell, D. and Kleinberg, J. *J. Am. Soc. Inf. Sci. Tec.* **58**(7), 1019–1031 (2007).
[22] Guimerà, R. and Sales-Pardo, M. *PNAS* **106**(52), 22073–22078 (2009).
[23] Clauset, A., Moore, C., and Newman, M. E. J. *Nature* **453**, 98–101 (2008).
[24] Blackmond, H., Paul, W., and Leinhardt, S. *Social Networks* **5**(2), 109–137 (1983).
[25] Albert, R., Jeong, H., and Barabási, A.-L. *Nature* **406**, 378–382 (2000).
[26] da F Costa, L., Rodrigues, F. A., Travieso, G., and Boas, P. R. V. *Advances in Physics* **56**(1), 167–242 (2007).
[27] Fortunato, S. *Phys. Rep.* **486**, 75–184 (2009).
[28] Newman, M. E. J. and Girvan, M. *Phys. Rev. E* **69**, 026113 (2004).
[29] Porter, M. A., Onnela, J.-P., and Mucha, P. J. *Notices of the AMS* **56**(9), 1082–1166 (2009).
[30] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. *PNAS* **101**(9), 2658–2663 (2004).
[31] Freeman, L. C. *Sociometry* **40**(1), 35–41 (1977).
[32] Watts, D. and Strogatz, S. H. *Nature* **393**, 440–442 (1998).
[33] Newman, M. E. J. *Phys. Rev. Lett.* **89**(20), 208701 (2002).
[34] Saavedra, S., Reed-Tsochas, F., and Uzzi, B. *PNAS* **105**(43), 16466–16471 (2008).
[35] Geyer, C. J. *Statistical Science* **7**(4), 473–483 (1992).
[36] Lancichinetti, A., Fortunato, S., and Kertész, J. *New Journal of Physics* **11**, 033015 (2009).
[37] Bollobas, B. *Modern Graph Theory*. Graduate Texts in Mathematics. Springer, New York, (1998).
[38] Lancichinetti, A., Fortunato, S., and Radicchi, F. *Phys. Rev. E* **78**, 046110 (2008).
[39] Barabási, A.-L. and Albert, R. *Science* **286**(5439), 509–512 (1999).
[40] Zachary, W. *Journal of Anthropological Research* **33**(4), 452–473 (1977).
[41] Lusseau, D. *Proc. R. Soc. Lond. B* **270**, 186–188 (2003).
[42] Knuth, D. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. Addison-Wesley, Reading, MA, (1993).
[43] Scannell, J. W., Burns, G. A. P. C., Hilgetag, C. C., O'Neil, M. A., and Young, M. P. *Cereb. Cortex* **9**(3), 277–299 (1999).
[44] Strogatz, S. H. *Nature* **410**, 268–276 (2001).
[45] Anderson, C. J., Wasserman, S., and Crouch, B. *Social Networks* **21**, 37–66 (1999).
[46] Matlab documentation, function `corr()`: `www.mathworks.com/access/helpdesk/help/toolbox/stats/corr.html`

Appendix A: LIST OF CODE

Here I give a brief list of the code I wrote for this project, in alphabetical order. All code was written as functions for Matlab.

- `accuracy.m`: Given the ranked list of links output by the HRG algorithm, and the list of actual missing links, calculates the accuracy of link prediction.
- `accuracyStats.m`: Calculates the CA, average accuracy across multiple instances of a network, and their errors.
- `assortativity.m`: Calculates the degree assortativity of a network.
- `averagePathLength.m`: Calculates the average geodesic path length or the average vertex closeness of a network. Can also give the network diameter.
- `connectedNetworkAverage.m`: Calculates average accuracies across different runs of the HRG algorithm in which isolated nodes have been discarded.
- `edgeAsymmetry.m`: Calculates the asymmetries for all edges in a network.
- `edgeCloseness.m`: Calculates the edge-closeness for all edges in a network.
- `edgeClustering.m`: Calculates the edge-clustering coefficient for all edges in a network.
- `edgeProb.m`: Converts edge-metric values into removal probabilities.
- `globalProperties.m`: Calculates several global properties for a network.
- `makeErrors.m`: Given link removal/addition probabilities, removes or adds links to a network in either an ordered ascending/descending or stochastic fashion. If no link probabilities are provided, addition/removal is carried out uniformly at random.
- `runExperiment.m`: Brings together the other functions and runs the whole process, including the HRG prediction algorithm's C++ code, from Matlab.
- `structMean.m`: Takes average of a field across a multi-dimensional data structure.