Bayesian inference of network structure from unreliable data

JEAN-GABRIEL YOUNG[†]

*Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109, USA,
Department of Computer Science, University of Vermont, Burlington, VT 05404, USA and Vermont
Complex Systems Center, University of Vermont, Burlington, VT 05404, USA*
[†]*Corresponding author. Email: jean-gabriel.young@uvm.edu*

GEORGE T. CANTWELL

*Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA and Santa Fe Institute,
1399 Hyde Park Road, Santa Fe, NM 87501, USA*

AND

M. E. J. NEWMAN

*Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA and Center for the Study of
Complex Systems, University of Michigan, Ann Arbor, MI 48109, USA*

Edited by: Tiago P. Peixoto

Most empirical studies of complex networks do not return direct, error-free measurements of network structure. Instead, they typically rely on indirect measurements that are often error prone and unreliable. A fundamental problem in empirical network science is how to make the best possible estimates of network structure given such unreliable data. In this article, we describe a fully Bayesian method for reconstructing networks from observational data in any format, even when the data contain substantial measurement error and when the nature and magnitude of that error is unknown. The method is introduced through pedagogical case studies using real-world example networks, and specifically tailored to allow straightforward, computationally efficient implementation with a minimum of technical input. Computer code implementing the method is publicly available.

Keywords: network structure, Bayesian inference, mixture models.

1. Introduction

Networks are widely used as a convenient mathematical representation of the relationships between elements of a complex system [1]. Network methods have been fruitfully applied to aid our understanding of systems in physics, biology, computer and information sciences, the social sciences and many other areas. A typical empirical network study will first determine the structure of a network of interest, using experiments, field observations or archival data, then analyse that structure to reveal features of interest, using any of the many quantitative analysis techniques developed for this purpose [2].

In this article, we focus on the first part of this process, on how we determine the structure of a network from appropriate empirical observations. At first sight, this appears to be a straightforward task. Most studies aim to measure the presence or absence of edges in a network, either singly or in some collective

fashion, and then assemble a picture of the complete network by aggregating many such measurements. Upon further consideration, however, it is apparent that the situation is not so simple, because most network measurements do not tell us unambiguously about the presence or absence of edges. At best, they give us a noisy indication, and in many cases they merely hint obliquely at the network structure [3]. As an example, consider a protein–protein interaction network representing the pattern of physical interactions between proteins in the cell. Such networks can be measured in a relatively direct manner. Techniques such as affinity purification and two-hybrid screens can be used to determine whether a given protein interacts with others [4, 5]. These techniques are notoriously unreliable, however, returning high rates of both false positives and false negatives, so that individual measurements do not themselves tell us the structure of the network [6]. Measurements may have to be repeated many times in order to separate the signal from the noise [7].

A more complex example is the measurement of a social network such as a network of who is friends with whom. Such networks can be measured by conducting surveys in which people are asked to identify their friends [8]. Here, however, things can get complicated. For instance, it happens often that person A identifies person B as a friend, but person B does not identify person A. In some communities, fully a half of all friendships are 'one-way' friendships of this kind [9, 10]. Should two such people be considered friends? Why do they disagree? Is one of the individuals mistaken, or forgetful, or not telling the truth? Perhaps the two are using different definitions of friendship? Things become harder still when we consider other data that are commonly gathered in such surveys, such as age, gender, race, occupation, income or educational level of participants. Friendships are well known to depend strongly on such personal characteristics and in cases where we are uncertain about a hypothesized friendship between two people we may be able to use their traits to make a better informed decision about whether they are really friends [11]. The best estimate of whether two people are friends may thus be the result of a complex calculation that combines many inputs.

Most empirical studies of network structure, however, do not take such an approach. Even though network data are known to be noisy and imperfect [6, 12–17], many experimenters nonetheless rely on simple direct measurements of the presence of edges and effectively make the assumption that the resulting data *are* the network. In a protein–protein interaction network, they assume an interaction to be present if, for example, it is observed to exist enough times when measured. In a friendship network, two people are assumed to be friends if one identifies the other as a friend. And yet, it is clear from the discussion above that the true situation is more complicated than this. In realistic circumstances, we have a heterogeneous body of data, potentially of many different types, potentially unreliable and varying in its relationship to the actual network structure. In such circumstances, traditional methods for inferring network structure from data may be inadequate and in some cases outright misleading.

In this article, we present a general framework for inferring network structure from measurements using methods of Bayesian inference, along with a complete software pipeline implementing that framework. The methods we describe take empirical measurements of a system and return a posterior distribution over possible network structures, thereby telling us not only what the likely structure is but also giving us an estimate of our certainty about that structure. Our approach requires a minimum of input from the experimenter, the only information they need supply (other than the data) being a description of how the measurements depend on the underlying network structure, which is specified at a high level in the form of probability distributions—we give a number of examples in this article. The rest of the calculation, from data to posterior distribution and final network structure, is performed automatically. Application of Bayesian inference methods often requires experimentation to find the best approach for a particular application. The level of automation in our framework makes this a straightforward task, allowing one to easily experiment with different strategies and quickly see the results [18].

The problem we address falls in the general area of *network reconstruction*, the challenge of determining the structure of a network from the available data, though one might also think of it as simply network measurement, akin to the experimental or observational determination of any quantity in the natural and social sciences. There has been a considerable amount of previous work on network reconstruction, much of it in the subject-specific literature and directed at the reconstruction of particular types of networks or using particular types of data. Methods have been proposed, for example, for geographic co-location data [19–22], social networks [23], ecological networks [24], brain scans [25–27] and biochemical networks [5, 28–30]. Perhaps the closest precursor to our work is that of Butts [23], who developed Bayesian methods and Gibbs sampling techniques for estimating social network structure from unreliable social surveys. Priebe *et al.* [31] have developed similar ideas, incorporating structured priors over networks to improve inference, an approach that has been echoed in statistics [25–27] and network science [32]. A key difference between these approaches and our own, however, is that they were developed with particular measurement processes in mind, whereas our methods can accommodate almost arbitrary ones. The approach we propose also differs from our own related previous work [3, 33, 34] in putting forward an estimation algorithm that works essentially automatically with arbitrary models, using ideas borrowed from the literature on finite mixture models [35, 36].

Looking more broadly, there is also a considerable volume of work that tackles other problems concerning the reliability of network data, many of which could also be addressed using variants or extensions of the methods proposed here. The problem of *link prediction*, for instance, involves predicting missing edges in partially observed networks and a range of algorithms have been proposed that achieve good performance [37–41]. The work of Guimerà and Sales-Pardo [38], for instance, uses a model sampled from a marginal posterior distribution reminscent of our approach, but does not explicitly include any mechanism for measurement error or attempt to reconstruct the network, focusing instead on the link prediction problem. Also much studied is the problem of inferring network structure from non-network data [42], such as time-series [43], gene expression data [44], the spread of information or disease [45, 46] and various local network features and node properties [47, 48]. A separate literature deals with problems of *network sampling*, addressing how choices such as which nodes or edges we measure can affect our estimates of network properties and how best to make those choices [2, 49–51]. *Disambiguation* or *entity resolution* is the process of accurately identifying the nodes in a network in the presence of potential sources of error such as duplicate nodes or nodes that have been inadvertently combined [15, 52, 53]. Some methods have also been proposed that aim to perform more than one of these tasks at once, such as simultaneous link prediction and disambiguation [54]. Finally, there is also a growing literature on making best estimates of derived network metrics starting from probabilistic representations of network structure [55–59]. These methods use the kinds of structural estimates we develop in this article as input to calculations of further quantities of interest, such as centrality measures, path lengths, correlations, community structure or core decompositions [55, 57].

This article is organized as follows. In Section 2, we first give an overview of the framework we propose for inference of network structure from unreliable data. Then in Section 3, we describe in depth two example applications, illustrating the construction of appropriate measurement models and the practical implementation of the method. In Section 4, we give our conclusions and suggest some directions for future work. An Appendix gives technical details of the workings of our method.

2. Description of the method

Suppose we have made some measurements of a networked system. They may include direct measurements of network structure, such as measurements of the presence or absence or edges, but they may also

include other quantities that could have indirect bearing on network structure, such as measurements of node properties or traits. Our objective is to make the best possible estimate of the structure of the network from these measurements and to do so efficiently and almost automatically, requiring a minimum of technical effort, so that practitioners can focus on making the measurements and interpreting the results. In this section, we give a broad overview of the concepts and essential equations underlying our method. A complete derivation and accompanying technical discussion can be found in Appendix A.

In a nutshell, we posit that there exists some underlying network whose structure affects measurements made on the system of interest. This network is represented by its adjacency matrix A, which is initially unknown. Our goal is to estimate this matrix. We assume that the observational data depend on the adjacency matrix but in a potentially noisy way: even exact repetition of the same experiment could produce different observations. This means that, even though the network has a well-defined and unambiguous structure, it will not in general be possible to tell exactly what that structure is from the measurements. To accommodate this uncertainty, we adopt a Bayesian point of view. Instead of inferring the exact network structure itself from measurements, we instead infer a probability distribution over possible structures compatible with the observations.

Apart from the data themselves, the only input our method requires of the user is the specification of a model that represents, in general terms, how the data depend on the network structure. This model can take a variety of different forms: networks and the experiments used to measure them vary widely, so no single model applies in all cases. Our method allows the user to specify the model that is most appropriate to their situation. The model may contain parameters (sometimes many of them), but it is not necessary to know the values of these parameters: our method automatically infers the best values from the data.

For the sake of concreteness, we concentrate in this article primarily on the most common situation encountered in network studies, in which the network is simple, undirected, and unweighted, and the data consist of individual measurements of the presence or absence of edges. The methods we describe are applicable to other situations as well, but this one covers many cases of interest and will allow us to use a more transparent and explicit notation. The measurements themselves can be as simple as observed interactions between pairs of nodes but can also take more complex forms, such as paths across the network, time-series, tags associated with relationships or any of a variety of other possibilities. We encode the measurements in an array X, whose entry X_{ij} contains all the information we have about the interaction of nodes i and j. We also make a further crucial assumption about the model, namely that observations of different node pairs are *conditionally independent*, which in this case means that the observations X_{ij} of the interaction between i and j depend only on the adjacency matrix element A_{ij} and not on any other elements. This assumption is not strictly necessary for the method work but, as shown in Appendix A, it improves the computational efficiency substantially. And, since it is true of most commonly used models anyway, it is in practice not a significant restriction. There are exceptions: in certain ('fixed choice') social network studies, for instance, participants are limited to naming a maximum number of friends or contacts, such as 10. This means that every time a participant names a contact, contacts with other individuals become less likely because the participant has fewer 'slots' left to fill, and hence contacts are (weakly) negatively correlated. In this article, we neglect such dependencies and assume that contacts are independent. (However, see Refs. [3, 33] for a discussion of methods that can handle dependencies.)

With these assumptions, selecting a model boils down to making three basic choices. The first and most crucial of these is specifying how the pairwise measurements X_{ij} depend on the underlying network, as represented by the adjacency matrix A. Specifically, for a given pair of nodes i, j we need to specify the expected distribution of values X_{ij} for the case where i and j are connected by an edge and for the

case where they are not. We will write these two distributions respectively as

$$\mu_{ij}(1,\theta) = P(X_{ij}|A_{ij} = 1, \theta) \tag{2.1}$$

and

$$\mu_{ij}(0,\theta) = P(X_{ij}|A_{ij} = 0, \theta), \tag{2.2}$$

where θ denotes any parameters of the distribution. (We will in some cases drop θ from the notation where it is clearly understood.) We will refer to Eqs. (2.1) and (2.2) as the *data model*.

The second modelling choice is the specification of the prior probability ascribed to the edge between i and j. What is the probability that the edge exists *before* we make any measurements of it? Do all edges have an equal chance of existing *a priori*, or are some more likely than others? Again we assume that different edges are independent, although again this assumption, while computationally convenient, is not strictly necessary. Mimicking the notation introduced for the data model, we write the prior probability of an edge between i and j as

$$\nu_{ij}(1,\theta) = P(A_{ij} = 1|\theta) \tag{2.3}$$

and the probability of no edge is $\nu_{ij}(0,\theta) = 1 - \nu_{ij}(1,\theta)$. Again θ collectively denotes the set of parameters (if any). We call this second set of probabilities the *network model*.

The third and last modelling choice is the specification of the prior distribution on the parameters θ. Our framework does not place any restriction on the possible form of the prior on θ. In many cases, a simple uniform prior works well, but the prior can also be chosen for example to encode specific knowledge about the system or to rule out unphysical values of the parameters.

Once these three choices are made, the rest of the procedure is essentially automatic. Given the model choices and a set of measurements, our method will generate a string of pairs (A_r, θ_r) of networks and parameters compatible with the measurements. By inspecting these networks and parameters, we can determine any other network properties we might care about. For example, if we want to determine whether i and j are connected by an edge, we can inspect the matrix element $A_{ij}^{(r)}$ for all r and compute the fraction of the time that $A_{ij}^{(r)} = 1$, which gives us (a Monte Carlo estimate of) the posterior probability of the edge's existence.

More precisely, the sample networks and parameter sets returned by our algorithm are drawn from the joint posterior distribution $P(A, \theta|X)$, which allows us to compute an estimate of the expected value of any function $f(A, \theta)$ of the network and parameters thus:

$$\begin{aligned}\langle f(A,\theta)\rangle &= \sum_A \int f(A,\theta)P(\theta,A|X)\,d\theta \\ &\simeq \frac{1}{N}\sum_{r=1}^N f(A_r,\theta_r),\end{aligned} \tag{2.4}$$

where N is the number of samples generated.

Our computer code implementing these calculations is freely available online with accompanying tutorials explaining its use.[1]

---

[1] The code can be downloaded from https://github.com/jg-you/noisy-networks-measurements.

3. Examples

To demonstrate how our method operates in practice, we present two detailed case studies. The first involves a network of animal interactions.

3.1 *Network of dolphin companionship*

Many animals form lasting social networks, including monkeys, deer, horses, cattle, dolphins and kangaroos [16]. Here, we analyse data from Connor *et al.* [60] of interactions among a small ($n = 13$) group of male bottlenose dolphins as they swim in a shallow lagoon. This is a typical example of an animal observational study that aims to determine social ties indirectly by observing behaviour. Standard techniques of social network analysis would typically be used to transform the observations into association indices that quantify the level of interaction between pairs of individuals [61]. These indices, however, can be hard to interpret and their definition relies on somewhat ad hoc assumptions. Our methods give us a more principled way to infer connections by interpreting the data as noisy measurements of an underlying social network.

3.1.1 *Basic model*   In this particular study the recorded data X_{ij}, shown in Fig. 1, represent the number of times each pair of dolphins is observed swimming in close proximity. We can use these data to reconstruct the underlying network as follows. First, it is reasonable to assume that the number of interactions between two dolphins will depend on whether they have a network connection or not. But also we expect there to be some randomness in the numbers, both because of circumstances and because of observational error. We thus model the number of interactions as a Poisson random variable with mean λ_1 or λ_0 depending on whether there is or is not a network connection, respectively. That is,

$$\mu_{ij}(0, \lambda_0) = \frac{\lambda_0^{X_{ij}}}{X_{ij}!}e^{-\lambda_0}, \tag{3.1a}$$

$$\mu_{ij}(1, \lambda_1) = \frac{\lambda_1^{X_{ij}}}{X_{ij}!}e^{-\lambda_1}. \tag{3.1b}$$

We assume that $\lambda_1 > \lambda_0$, that is, that individuals interact more often if they have a network connection than if they do not.
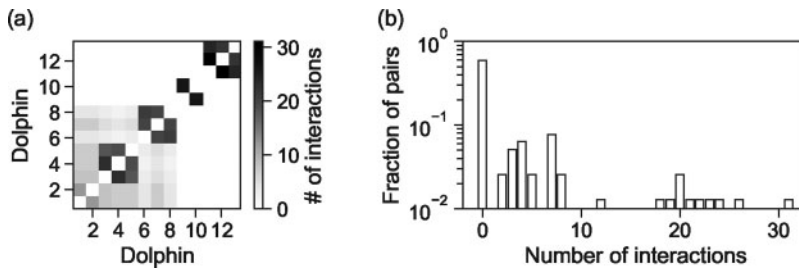

FIG. 1. Data on interactions among a group of dolphins, from Connor *et al.* [60]. Thirteen male dolphins were observed as they swam in a shallow lagoon and tabulations were made of pairs that swam together. (a) Observed frequency of interaction for each pair. (b) Histogram of frequencies.
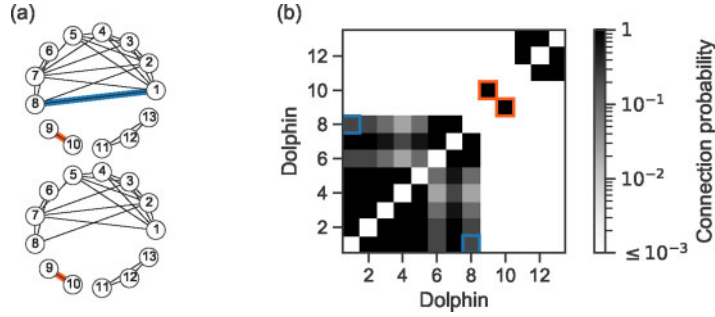
FIG. 2. Reconstruction of the network of dolphins from the data shown in Fig. 1. (a) Two examples of sampled network structures. (b) Matrix of posterior edge probabilities, obtained by averaging over 4000 network samples. Entries in the matrix corresponding to the edges highlighted in panel (a) are shown in matching colours.

We also need to choose our network model and prior on the model parameters. Having no prior information about the probabilities of individual network edges, we assume that all edges are *a priori* equally likely, which implies

$$\nu_{ij}(0, \rho) = 1 - \rho, \tag{3.2a}$$

$$\nu_{ij}(1, \rho) = \rho. \tag{3.2b}$$

where ρ is the probability of an edge.

For the priors on the parameters, we make minimal assumptions. For ρ, which is a probability, we assume a uniform prior on the interval $[0, 1]$. For λ_0 and λ_1, which have semi-infinite support, we cannot use a uniform prior or the posterior distribution would become improper. So, instead, we assume a slowly varying probability with a wide range of plausible values, specifically a semi-normal distribution (i.e. the right half of a normal distribution centred on zero) with large variance:

$$P(\lambda_k) \propto e^{-\lambda_k^2/2\sigma^2}, \tag{3.3}$$

where $\sigma \gg 1$ is a fixed hyperparameter and $P(\lambda_k) = 0$ if $\lambda_k < 0$. In our calculations, we use $\sigma = 100$.

With the model specified, we now run the algorithm described in Appendix A and obtain a series of samples (A_r, θ_r) from the joint distribution $P(A, \theta|X)$, where θ in this case collectively refers to the parameters λ_0, λ_1 and ρ, and θ_r is one realization of these parameters. Two examples of sampled network structures are shown in Fig. 2a, out of thousands generated. As the figure shows, the two structures are similar but not identical. This is expected: we should see variability from sample to sample. When looking over the entire sample set, for example, the edge between nodes 9 and 10, highlighted in orange, appears almost always, whereas the edge between 1 and 8, in blue, appears only rarely. To capture this variability, we average over samples and compute the posterior probability of every edge as the fraction of samples in which the edge occurs. The result is shown in matrix form in Fig. 2b. Comparing with Fig. 1, we see that these probabilities are quite different from the distribution of number of interactions between dolphin pairs. Moreover, while the numbers of interactions span a wide range of values, the probabilities of edges are clustered around zero and one. This is good news. Edge probabilities near zero and one indicate edges about which we are relatively certain, either that they exist or that they do not. Thus, we have turned a
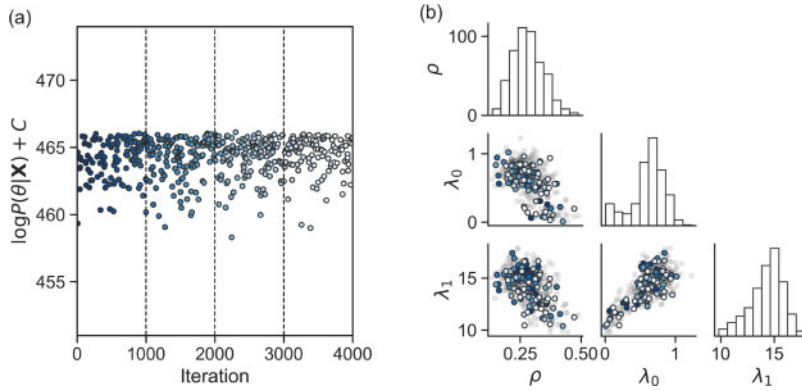
FIG. 3. Statistics of samples drawn from the posterior distribution $P(\lambda_0, \lambda_1, \rho|\boldsymbol{X})$ for the data shown in Fig. 1 and the measurement model given in Eqs. (3.1)–(3.2) over four runs with randomly chosen initial conditions. Only 500 of the 4000 total samples taken are shown for the sake of visual clarity. (a) Logarithm of the posterior probability, with different runs separated by vertical dotted lines. The colour of the points varies from left to right for easier comparison with panel (b). (b) Pair plot showing the relation between sampled values of parameters, as well as the distribution of individual parameters. We highlight a subset of samples with colours matching panel (a).

dataset with considerable variability into a network structure about which we have high confidence. We discuss this point further below.

In conjunction with these estimates of the network's structure, we also compute estimates of the parameters θ, by averaging the parameter samples, just as we did with the network samples. For instance, we find that $\lambda_0 = 0.62$ and $\lambda_1 = 14.3$ (with 95% credible interval (CI) of [0.06.0.98] and [10.8, 16.8], respectively), meaning that dolphin pairs interacted an average of 14.3 times during the experiment if they shared a network connection but only 0.62 times if they did not. In other words, the effect of network connections is very pronounced and highly statistically significant. We also find that $\rho = 0.28$ (95% CI [0.18, 0.43]), indicating that the network is quite dense. (This would be an unusually high value in human social networks, although the network in this example is small, which tends to increase density.)

3.1.2 *Sampling quality and goodness of fit*   While these results are promising, there is further work to be done if we are to be confident in them. In particular, as is standard with Bayesian calculations, there are two specific things we need to check [62]. First, we need to be sure that the Monte Carlo algorithm is sampling correctly from the posterior distribution and, second, we need to test whether our proposed model is in fact a good fit to the data.

In Monte Carlo calculations of this kind the posterior probability distribution is typically *rugged*, meaning it has multiple local optima, and the sampling algorithm can as a result get stuck for periods of time in suboptimal portions of the sampling space. To test for this issue, we plot in Fig. 3a the log probability of our samples as a function of time over four different runs of the algorithm. The plot shows that the distribution of values appears roughly consistent within each run and across different runs, which we take as a sign that the samples have not failed in obvious ways. In Fig. 3b, we compare how the sampled values for the parameters λ_0, λ_1 and ρ relate to one another. These plots, colloquially known as pair plots, are conventional in Bayesian analysis. In addition to showing the distribution of values for the individual parameters on the diagonal (which are more informative than the simple averages reported in the text above), these plots can help verify the quality of the samples and provide some sanity checks.
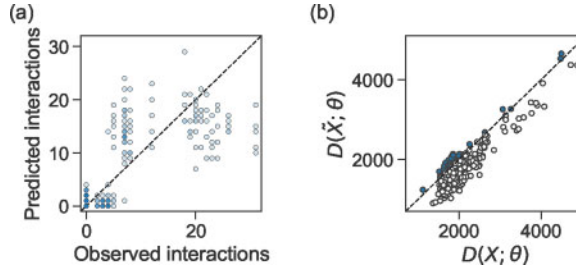
FIG. 4. Posterior-predictive tests of the model used for the dolphin network. (a) Average predicted number of interactions between dolphin pairs as a function of the actual observed number of interactions, for five random data sets generated from the model ($R^2 = 0.50$). Colours indicate the numbers of pairs of dolphins with each number of predicted/observed interactions across five random replications of the data. (b) Scatter plot of discrepancy values. Each point in this plot corresponds to one of 500 sets of parameter values, selected at random from a total of 4000 such sets drawn from the posterior distribution $P(\lambda_0, \lambda_1, \rho | X)$ during the calculation. Sets having higher model–model discrepancy $D(\tilde{X}, \lambda_0, \lambda_1, \rho)$ than data–model discrepancy $D(X, \lambda_0, \lambda_1, \rho)$ are highlighted in blue, above the diagonal. The fraction of such sets gives us the p-value, which in this case is $p = 0.136$.

In our case, the plots reveal that all the samples come from approximately the same region of parameter space regardless of the different initialization of the four runs, which tallies with the uniform sample quality found in Fig. 3a and gives further evidence that the algorithm is sampling consistently.

Another function of pair plots is to diagnose problems with the model specification. Models of the kind we consider here can for instance possess symmetries, such as invariance under the interchange of edges and non-edges, which can cause problems when averaging chains that break the symmetry differently [35]. Such symmetries are easy to visualize in a pair plot, though we see none in Fig. 3b. (We have imposed $\lambda_1 > \lambda_0$ to avoid them.) A model can also suffer from parameters that are not identifiable, for example when two or more parameters can be combined into one. The pair plot alerts us to such issues because the parameter values will be perfectly correlated, but again we see no such issues in Fig. 3b.

Having confirmed that the samples are plausibly drawn from a correctly specified posterior distribution, the other thing we need to check is whether our model is actually a good fit to the data. If the model is a poor one, then even the best fit it provides may not actually be a *good* fit. To test the goodness of fit, we use two Bayesian tests of the type known as *posterior-predictive assessments*. (See Appendix A and Refs. [18, 63] for discussions.) In these tests, we use the data model $P(X | \hat{A}, \hat{\theta})$ with parameter values $\hat{A}$ and $\hat{\theta}$ drawn from our Monte Carlo sample to generate new data $\tilde{X}$, as if we were making measurements on a system that obeyed the fitted model exactly. Then, we compare these new data to the original inputs. If the model is a good fit, the two should look similar.

An example of such a comparison is shown in Fig. 4. Figure 4a shows the number of times a pair of dolphins interacts in the simulated data as a function of the number of times they are observed to interact in the original data, in five artificial data sets $\tilde{X}$ generated as above. If data and model agreed well, most of the points in this scatter plot would concentrate along the diagonal line, but in this case they do not. This is our first indication that the fit we have found may not be as good as we would like. We will see shortly how to make the fit much better, but let us proceed for the moment with what we have as an illustration of our goodness-of-fit analysis.

To more accurately quantify the performance of the model, we can calculate a *discrepancy* [63]:

$$D(X, \theta) = \sum_{ij} X_{ij} \log \frac{X_{ij}}{\langle \tilde{X}_{ij}(\theta) \rangle},$$ (3.4)

where $\langle \tilde{X}_{ij}(\theta)\rangle$ is the average of the synthetic data generated from the model, with parameter values θ. The discrepancy is essentially a Kullback–Leibler divergence between the observed and synthetic data for one sampled value of the parameters. It functions in this situation as a goodness-of-fit measure: the smaller the discrepancy, the closer the synthetic data are to the input.

The discrepancy is not very informative by itself, however, since it is not clear what kind of values we should expect to see. For example, even if the model is an excellent fit, we should not expect the discrepancy to be zero, since the randomness of both the data and the model mean that they are unlikely to agree exactly. To obtain a point of comparison, therefore, we compute discrepancies between a large number of pairs of simulated datasets drawn from realizations of the model with the same parameter values used for the observed data. If the model were correct, so that simulated and observed data have the same distribution, then these values would tell us the typical magnitude we should expect the discrepancy to have. If the values are mostly smaller than the observed discrepancy then it indicates the model is unlikely to be correct; if they are larger then the model is not ruled out. The fraction of generated discrepancy values that are larger than the observed discrepancy gives us the *p*-value for the model, that is, the probability that the observed discrepancy would have been generated if the model were correct.

Note that the use of the *p*-value in this situation is different from the way it is used in traditional frequentist statistics, where it represents the probability of getting a particular observed value if a null hypothesis were true. In the traditional scenario, a small *p*-value leads us to reject the null hypothesis, and, since this is often the goal of an experiment, small *p*-values are 'good'. In the present case, the *p*-value is applied to the model we are fitting (there is no null/alternative hypothesis) and it just counts the fraction of artificial datasets for which we find discrepancies more extreme than the value found when fitting the model to the true data. So in this situation small *p*-values are 'bad'.

Figure 4b shows values of the discrepancy for both artificial and real data, for 500 sampled values of the model parameters. Instances where the artificial discrepancy is greater than the observed one appear above the diagonal in the plot and the fraction of such points tells us the *p*-value. In this case, we find $p = 0.136$. While it is not appropriate to apply arbitrary cutoffs to this (or any) *p*-value [63], the value is not as high as we would like it to be, and though we cannot completely rule out the model the evidence suggests that the fit is not ideal.

3.1.3 *Improved model*   What can we do to improve the fit? The standard approach is to adopt a more elaborate model that is capable of representing a wider range of data distributions. In the model we have used so far connections are binary: dolphin pairs either have a connection or they do not. We can create a more nuanced model by allowing three levels of connection, corresponding to no tie, a weak tie or a strong tie. Denoting the three levels by adjacency matrix elements $A_{ij} = 0$, 1 and 2, we introduce a new distribution for X_{ij} when $A_{ij} = 2$ which is Poisson as before but with mean λ_2:

$$\mu_{ij}(2, \lambda_2) = \frac{\lambda_2^{X_{ij}}}{X_{ij}!} e^{-\lambda_2} \tag{3.5}$$

where $\lambda_2 \geq \lambda_1 \geq \lambda_0$, and the prior of Eq. (3.2) becomes

$$\nu_{ij}(0, \rho) = \rho_0 = 1 - \rho_1 - \rho_2, \tag{3.6a}$$

$$\nu_{ij}(1, \rho) = \rho_1, \tag{3.6b}$$

$$\nu_{ij}(2, \rho) = \rho_2. \tag{3.6c}$$

The fitting and model verification procedures follow the same lines as previously.
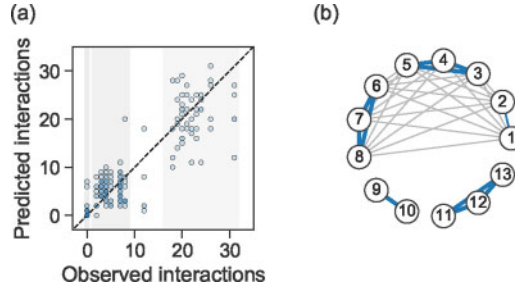
FIG. 5. Network inference with multiple edge types for the data shown in Fig. 1 and the measurement model given in Eqs. (3.1)–(3.6). The estimated mean numbers of interactions are $\lambda_0 \simeq 0.02$ when there is no edge between a pair of dolphins, $\lambda_1 = 5.13$ when the pair shares a weak tie, and $\lambda_2 = 21.97$ when they share a strong one. The corresponding prior probabilities of edge types are $\rho_0 = 0.58$, $\rho_1 = 0.28$ and $\rho_2 = 0.14$. (a) Posterior predicted number of interactions between dolphins as a function of observed number for five random datasets generated from the fitted model ($R^2 = 0.82$). The highlighted regions correspond, from left to right, to dolphins having no tie, a weak tie and a strong tie. (b) The inferred structure of the network, with weak ties represented by thin grey edges and strong ties by thicker blue edges. (Nodes 1 and 2 are connected by a thin blue line to reflect the fact that the calculation is ambiguous about the type of this tie.)

This modified model now fits the data significantly better, as shown in Fig. 5a. It divides the observed numbers of pair interactions into three clear groups centred around values of about 0, 5 and 25, and the *p*-value is now a very respectable 0.722, indicating that there is no statistical basis to reject this model at all: the model truly captures the structure of the observed behaviour. Indeed a *p*-value significantly larger than this could be a sign of problems, indicating overfitting. Unless the distribution of the discrepancy is strongly skewed, the expected *p*-value will normally be around 0.5 when the model is a perfect fit.

Having found a model that fits the data well, we examine the inferred network structure, which is shown in Fig. 5b. The network has three disconnected subgroups of dolphins, two comprised of strong connections only and one, the largest of the three, having a mix of strong and weak connections. The posterior probabilities on all of the interaction types are close to one, indicating high confidence in the structure of the network. For instance, the model predicts that nodes 8 and 9 are not connected with probability 0.99(6), that nodes 7 and 8 share a strong connection with probability 0.99(9), and that nodes 1 and 8 share a weak connection with probability 0.99(9). There is just one pair of nodes (1 and 2) whose connection is hard to classify. The model indicates that the tie between these nodes is either weak or strong, with probabilities 0.51(4) and 0.48(5), respectively. This pair of dolphins was observed swimming together 12 times, a number that falls between the weak and strong domains in the fitted model (see Fig. 5a).

3.2 *Friendship network of school students*

For our second example, we revisit a network analysed previously using different methods in Refs. [3, 33], a network of friendships between high-school students taken from the US National Longitudinal Study of Adolescent to Adult Health (the 'AddHealth' study). Although the method proposed in this article is mathematically more complex than that of [3, 33], it is arguably easier to apply since our analysis pipeline performs most of the calculation automatically. The most demanding step is the formulation of the model, but once we have a plausible model the rest of the process is straightforward and mechanical.

The AddHealth study is a large study of networks of social contact among students in schools across the USA. Students in participating schools were asked to identify their friends and the basic unit of resulting

data is a friendship nomination: one student says they are friends with another. Our data matrix X in this case is thus a non-symmetric one: $X_{ij} = 1$ if i names j as a friend and 0 otherwise. There is no guarantee that j will also name i and in fact there are many instances in which reported friendships only run in one direction. If we assume that friendship is fundamentally a bidirectional interaction, then this lack of symmetry indicates that the data are necessarily unreliable.

As is done in Refs. [3, 23, 33], we fit the data using a model in which each student i has an individual true-positive rate α_i and false-positive rate β_i. The true-positive rate is the probability that i names as a friend another student who is in fact a friend, as determined by the adjacency matrix. The false-positive rate is the probability of naming someone who is not actually a friend. We explicitly allow for different true- and false-positive rates for different individuals, since it is widely accepted that survey respondents vary in the accuracy of their responses.

In the notation of this article, the equations for the model are:

$$\mu_{ij}(1, \alpha_i, \alpha_j) = \alpha_i^{X_{ij}}(1 - \alpha_i)^{1-X_{ij}} \alpha_j^{X_{ji}}(1 - \alpha_j)^{1-X_{ji}}, \tag{3.7a}$$

$$\mu_{ij}(0, \beta_i, \beta_j) = \beta_i^{X_{ij}}(1 - \beta_i)^{1-X_{ij}} \beta_j^{X_{ji}}(1 - \beta_j)^{1-X_{ji}}. \tag{3.7b}$$

For instance, supposing that i and j truly are friends, the probability of i saying that they are ($X_{ij} = 1$) while j says they are not ($X_{ji} = 0$) is $\mu_{ij}(1) = \alpha_i(1 - \alpha_j)$. Conversely, if they are not in fact friends then we instead get $\mu_{ij}(0) = \beta_i(1 - \beta_j)$.

For the priors we again make the assumption of Eq. (3.2) that all edges are a priori equally likely, and assume a uniform prior on the edge probability ρ and a uniform distribution over all values of α_i and β_i that satisfy $\beta_i < \frac{1}{2} < \alpha_i$. (One could simply assume a uniform prior on both α_i and β_i in the range $[0, 1]$ but this leaves some ambiguity in the model because of the inherent symmetry between edges and non-edges: if we exchange the values of all α_i and all β_i and set ρ to $1 - \rho$ the model remains the same. By making the reasonable assumption that $\alpha_i > \beta_i$, we break this symmetry. The assumption that $\beta_i < \frac{1}{2} < \alpha_i$ is not strictly necessary, but turns out to be helpful for narrowing down the parameter space and hence improving the speed and convergence of the calculation [23].)

Figures 6 and 7 show the results of fitting this model to the data for a single school from the AddHealth data set. We use one of the smaller schools as our example, with 521 students who completed a survey and 2182 declared ties, primarily in order to make visualization of the results easier. We find that the Monte Carlo algorithm converges well and gives samples that appear to accurately characterize the posterior
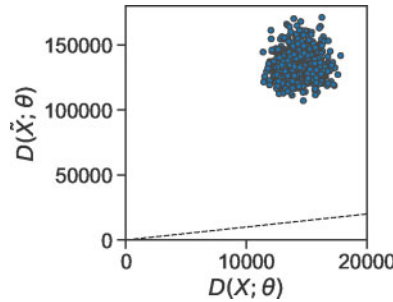

FIG. 6. Goodness of fit testing for the AddHealth model. Each point in this plot corresponds to a random parameter set drawn from the posterior distribution $P(\alpha, \beta, \rho | X)$. Samples associated with a higher model–model discrepancy $D(\tilde{X}, \theta)$ than data–model discrepancy $D(X, \theta)$ appear above the dotted line, indicating a good fit between data and model.
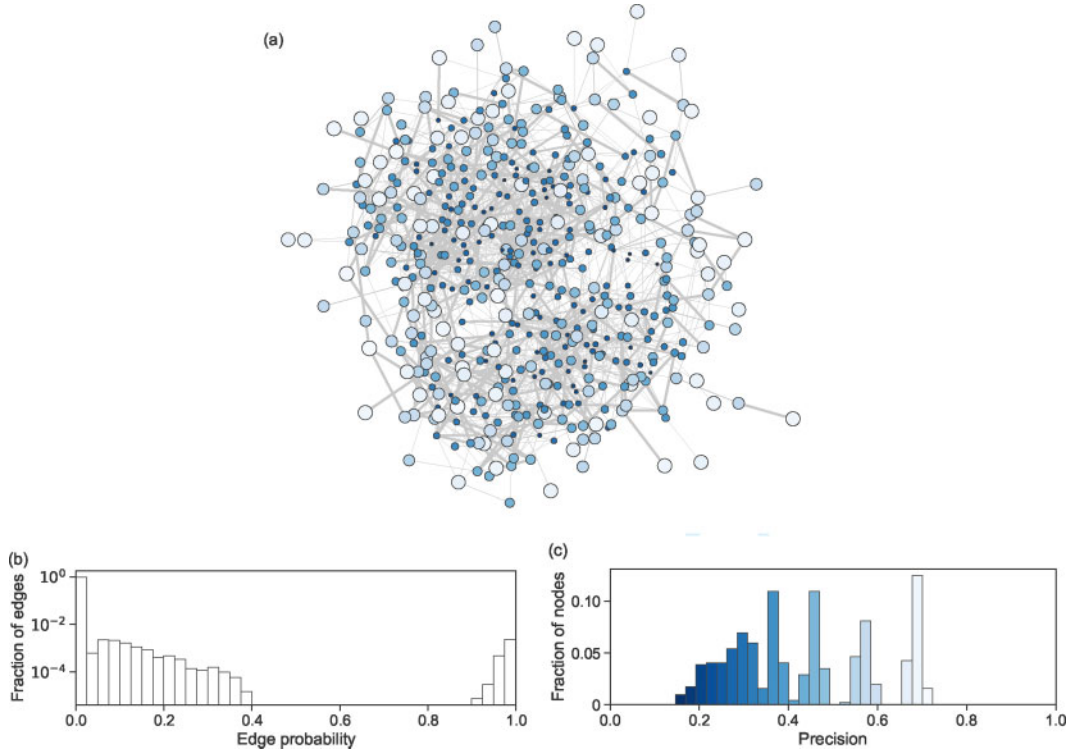
FIG. 7. Inference of a school friendship network from noisy data. (a) The 521 nodes in this figure represent the students at a single school in the AddHealth study and inferred friendships are shown as edges whose thickness indicates the estimated probability that they exist. The size and colour of the nodes indicates the estimated precision of friendship reports by the corresponding individual, that is, the fraction of their reported friendships that are inferred to actually exist. Darker shades indicated less precise individuals and correspond to the shades in the histogram in panel (c). The average values of the parameters of the model are $\langle \alpha \rangle \simeq 0.7605$, $\langle \beta \rangle \simeq 0.0065$ and $\langle \rho \rangle \simeq 0.004$. (b) The distribution of the probability of existence of edges. Many values are close to zero or one, indicating confidence that the corresponding edge does or does not exist, although a significant number fall at intermediate values. (c) The distribution of estimated precision values for participants.

distribution. Figure 6 shows discrepancy values in a manner analogous to Fig. 3b for the dolphin network and all values are well above the diagonal, indicating a good fit to the data.

The inferred network structure is shown in Fig. 7a. By contrast with the dolphin network example, the posterior probabilities of edges now vary more widely, as represented by the thickness of the edges in the figure. Figure 7b shows the distribution of edge probabilities as a histogram and many probabilities are again close to either 1 or 0, indicating a high degree of certainty that these edges either exist or do not, but there are also a significant number of edges with intermediate probabilities, edges about which we are less certain.

The fit also returns values of the true- and false-positive rates α_i and β_i for each node, which allow us to make quantitative statements about how accurately each individual reports his or her friendships. The average value of α_i over all individuals and all samples is 0.76, meaning that an estimated 24% of friendships are going unreported. The average false-positive rate is 0.0065, which sounds small but this result is somewhat misleading. The network is very sparse, meaning that almost all edges that

could exist do not. It takes only a small fraction of false positives among these many non-edges to generate a significant number of errors. Arguably a more informative measure of false positives is the *precision*, which is the fraction of reported friendships that are actually present and is given in this case by $\rho\alpha_i/[\rho\alpha_i + (1-\rho)\beta_i]$ [33]. The distribution of values of the precision is shown in Fig. 7c and ranges from a little under 0.2 to a little over 0.75, indicating that in fact a significant fraction of reported friendships—anywhere from 25% to 80%—are false positives.

These results are largely in agreement with previous work [33], although there are modest differences in estimated parameter values and network structure, due to the different methodology. We would argue that the fully Bayesian methodology employed here is more correct in that it accounts for intrinsic uncertainty in the parameter values. The methodology of [33], which makes use of an expectation-maximization (EM) algorithm, might be described as 'semi-Bayesian', computing a full posterior distribution over the network structure but relying on point estimates of the parameters. Because the model used here is a large one, having $O(n)$ parameters, we expect there to be significant uncertainty in the parameter values, which is captured by our Bayesian sampling method. That said, in practice the two methods do lead to qualitatively consistent conclusions. The key benefits of the current approach in this case are that it is simpler to implement using standard software, is formally more correct, and incorporates a natural means for checking the goodness of fit.

One potential issue with the results is the fact that the discrepancy values in Fig. 6 are all well above the dotted line, indicating close fits of the model to the data and a *p*-value near 1. A *p*-value this large can be a warning sign for overfitting, which is a possibility given the large number of parameters in the model. Such an issue could not be diagnosed with the methods previously used in Refs. [3, 33], but our approach makes this possible. One could address the problem by changing the model, say by using a more complex model in which instead of fitting the true- and false-positive parameters we instead draw them from a hyperprior distribution, such as a beta distribution, with an associated (small) set of hyperparameters that are fit using Monte Carlo. This approach can reduce the chances of overfitting and would be a good direction for future work.

4. Conclusions

In this article, we have introduced a general Bayesian framework for reconstructing networks from observational data in the case where the data are error prone, even when the magnitude of the errors is unknown. Our methods work by fitting a suitable model of the measurement process to the data and there is a large class of models that is both expressive enough to represent real data sets accurately and yet simple enough to allow for easy and automatic statistical inference. The output of the fitting process is a complete Bayesian posterior distribution over possible network structures and possible values of model parameters. We have demonstrated our methods with two case studies showing how to formulate suitable models, fit them, assess goodness of fit and infer reliable estimates of network structure.

With this work, we hope to promote the adoption of more rigorous methods for handling measurement error in network data in a principled manner. The methods we propose not only achieve this but do so in a manner that is straightforward and requires a minimum of technical expertise on the part of the user. Practitioners can use the framework we propose to apply appropriate, application-specific models to their data and obtain estimates of network structure in a matter of minutes.

Funding

A. Methods

In this appendix, we describe the mathematical and statistical foundations of our method in detail.

A.1 *Generative models of measurement*

Consider an experimental setting in which we have measurements X of a network's structure. The measurements could be as simple as a number of observed interactions between pairs of nodes, but could also incorporate time-series, vector measurements, etc. In general these measurements do not tell us the exact structure of the network, but instead give us indirect and potentially noisy information. Our goal is to make the best estimate we can of the true network structure given the measurements.

In the general framework we consider here, two nodes i and j can share connections of various types. In the simplest case there are just two types: nodes can be either connected by an edge (type 1) or not (type 0). In a more complex three-type case, the connection could be absent (type 0), weak (type 1), or strong (type 2) and so on. For a network of n nodes we encode these connections by an $n \times n$ adjacency matrix A where the matrix element A_{ij} records the type of connection between nodes i and j. We can also represent directed networks using an asymmetric adjacency matrix with A_{ij} being the type of the directed connection from j to i and A_{ji} being the type from i to j.

Our approach rests on the hypothesis that the matrix X of pairwise measurements is dependent, in a probabilistic fashion, on the adjacency matrix A. Both A and X can be either symmetric (for undirected networks) or asymmetric (for directed ones) and they need not be of the same type. In friendship networks, for example, the symmetric relationship of being friends is commonly probed using asymmetric measurements (person i says they are friends with person j).

It is this dependence between network and measurement that we exploit to estimate A from X. We formalize the relation using a generative model that specifies the probability $P(X|A, \theta)$ of making the measurements given the network, plus optionally some additional parameters represented collectively by θ. Then, applying Bayes' rule, we can write the probability of the unknown quantities A and θ given the measurements as

$$P(A, \theta|X) = \frac{P(X|A, \theta)P(A|\theta)P(\theta)}{P(X)}. \tag{A.1}$$

Our goal is to use this equation to infer the network structure A from the measurements X and to quantify the errors we might make in doing so.

A.2 *A flexible class of models*

To further simplify the discussion and improve the efficiency of the numerical calculations, we make some additional assumptions about the model, while keeping the approach as broad as possible to allow users to easily adapt it to various types of data and experimental settings.

Of the four probabilities that appear on the right-hand side of Eq. (A.1) one of them $P(X)$ is a constant (since it depends only on X which is fixed by the experiment) and hence will play no part in our calculations. The others must be specified to define our model. We refer to these three probabilities as the data model $P(X|A, \theta)$, the network model $P(A|\theta)$ and the prior on the parameters $P(\theta)$. Let us consider each of these in turn.

A.2.1 *Data model*

The data model $P(X|A, \theta)$ specifies the probability of making a particular set of measurements X given the network and the model parameters. In specifying this probability, we will make two key assumptions. First, we assume that the measurement X_{ij} is only influenced by the corresponding element A_{ij} of the adjacency matrix and not by any other elements. Second, we assume that, conditioned on the network structure A and parameters θ, the measurements X_{ij} for different node pairs are independent. Thus, for instance,

$$P(X_{ij}, X_{kl}|A, \theta) = P(X_{ij}|A_{ij}, \theta)P(X_{kl}|A_{kl}, \theta).$$

The notation here is a bit unwieldy, so for clarity we introduce the notation $\mu_{ij}(A_{ij}, \theta)$ to denote the probability $P(X_{ij}|A_{ij}, \theta)$ of making the measurement X_{ij} given the type A_{ij} of the connection between nodes i and j and given the parameter values θ. (Where the meaning is clear we may drop the explicit dependence on θ to simplify our expressions.) With this notation and our assumption of conditional independence, the probability $P(X|A, \theta)$ for the data model is simply

$$P(X|A, \theta) = \prod_{(i,j)} \mu_{ij}(A_{ij}, \theta). \tag{A.2}$$

The product $\prod_{(i,j)}$ is taken over all unordered pairs of nodes when the network is undirected and over all ordered pairs when it is directed.

Table A1 gives a selection of possible forms for the data model for networks with only a single edge type. Generalization to multiple edge types is straightforward. (See also Ref. [33] for a discussion of a range of models.)

A.2.2 *Network model*

The network model $P(A|\theta)$ can be thought of as our prior expectation of what the network should look like, before we make the measurements. By analogy with the factorized form of the data model in Eq. (A.2), we consider network models with the factorized form

$$P(A|\theta) = \prod_{(i,j)} \nu_{ij}(A_{ij}, \theta), \tag{A.3}$$

where we define $\nu_{ij}(A_{ij}, \theta)$ in a similar manner to $\mu_{ij}(A_{ij}, \theta)$, as the prior probability $P(A_{ij}|\theta)$ that nodes i and j share a connection of type A_{ij}, given the parameters θ. Many standard network models can be written in this form, including the Erdős–Rényi random graph, the configuration model, and the stochastic block model. Some examples of network models are given in Table A2 and Ref. [33].

A.2.3 *Prior on the parameters* The third component of our generative model, the prior $P(\theta)$ on the parameters, is the simplest. Our method does not place any significant constraints on the form of this probability, so one is free to choose almost any form appropriate to problem at hand, ranging from simple flat priors or factorized forms to ones that incorporate complex correlations between parameters. The only stipulation we make is that the parameters should be continuous-valued variables (not discrete-valued), which allows for more efficient sampling procedures (see Section A.4).

A.3 *Inference in theory*

Gathering the elements defined above and substituting them into Eq. (A.1) we obtain the complete joint posterior distribution for the model:

$$P(A, \theta | X) = \frac{P(X | \theta, A) P(A | \theta) P(\theta)}{P(X)},$$
$$\propto P(\theta) \prod_{(i,j)} \mu_{ij}(A_{ij}) \nu_{ij}(A_{ij}). \tag{A.4}$$

This distribution tells us the probability of a network structure and a set of parameter values given the observed measurements. From it, we can derive a variety of further useful quantities, such as the probability of the network structure independent of the parameters, which is given by

$$P(A | X) = \int P(A, \theta | X) \, d\theta. \tag{A.5}$$

Even more useful, perhaps, is the probability of having an edge of a given type between two specific nodes i, j:

TABLE A1 *Example data models for undirected networks with one edge type. Here, N_{ij} represents the number of times the node pair i, j was measured and X_{ij} represents how many of those times an edge was observed to exist.*

Model	Parameters	Data probability
Binomial with uniform errors	True-positive rate $\alpha \in [0, 1]$ False-positive rate $\beta \in [0, 1]$	$\mu_{ij}(1) = \alpha^{X_{ij}} (1 - \alpha)^{N_{ij} - X_{ij}}$ $\mu_{ij}(0) = \beta^{X_{ij}} (1 - \beta)^{N_{ij} - X_{ij}}$
Binomial with node-dependent errors	True-positive rate $\alpha_i \in [0, 1]$ for node i False-positive rate $\beta_i \in [0, 1]$ for node i	$\mu_{ij}(1) = \alpha_i^{X_{ij}} (1 - \alpha_i)^{N_{ij} - X_{ij}}$ $\mu_{ij}(0) = \beta_i^{X_{ij}} (1 - \beta_i)^{N_{ij} - X_{ij}}$
Poisson with uniform errors	Means λ_1, λ_0 for edges and non-edges	$\mu_{ij}(1) = \lambda_1^{X_{ij}} e^{-\lambda_1} / X_{ij}!$ $\mu_{ij}(0) = \lambda_0^{X_{ij}} e^{-\lambda_0} / X_{ij}!$
Poisson with node propensity	Normalized node propensity $0 < \eta_i < 1$ ($\sum \eta_i = 1$) and base rates λ_1, λ_0	$\mu_{ij}(1) = (\lambda_1 \eta_i \eta_j)^{X_{ij}} e^{-\lambda_1 \eta_i \eta_j} / X_{ij}!$ $\mu_{ij}(0) = (\lambda_0 \eta_i \eta_j)^{X_{ij}} e^{-\lambda_0 \eta_i \eta_j} / X_{ij}!$

TABLE A2 *Network models for the prior probability v_{ij} of an edge between nodes i and j.*

Model	Parameters	Edge probability
Random graph	Edge probability ρ	$v_{ij}(1) = \rho$
'Soft' configuration model	Node pseudo-degree λ_i	$v_{ij}(1) = 1/(1 + e^{-\lambda_i \lambda_j})$
Stochastic block model	Node i belongs to group g_i and edge probability between groups r and s is ω_{rs}	$v_{ij}(1) = \omega_{g_i g_j}$
Random graph with multiple edge types	Probability of type-k edge ρ_k	$v_{ij}(k) = \rho_k$
Poisson multigraph	Mean edge number ω	$v_{ij}(k) = \omega^k e^{-\omega}/k!$

$$P(A_{ij} = k|X) = \int P(A_{ij} = k, \theta|X)\, d\theta$$

$$\propto \int \mu_{ij}(k,\theta) v_{ij}(k,\theta) P(\theta)\, d\theta, \tag{A.6}$$

where we have used Eq. (A.4).

If we instead want to learn something about a parameter $\phi \in \theta$ then we can compute its distribution as

$$P(\phi|X) = \sum_A \int P(\theta', \phi, A|X)\, d\theta', \tag{A.7}$$

where θ' is the parameter set with ϕ excluded.

Each of these quantities can be considered a special case of the posterior average of a general function $f(A, \theta)$ of network structure and parameters, thus:

$$\langle f(A,\theta) \rangle = \sum_A \int f(A,\theta) P(\theta, A|X)\, d\theta. \tag{A.8}$$

There are a number of approaches we could take to computing expectations of this form [35]. One possibility is to use an expectation–maximization (EM) algorithm to compute the distribution over potential networks $P(A|\theta, X)$ as well as a point estimate of θ [3, 33]. Alternatively, following [32, 64], we can integrate out the parameters θ analytically to derive the marginal distribution $P(A|X)$ over the networks alone. However, neither of these approaches is in line with our goal of providing near-automatic inference for arbitrary models, the EM approach because it calls for the solution of (often non-linear) equations specific to the model and the marginal-based approach because it works only for models amenable to closed-form integration. The EM approach moreover gives only point estimates of θ and therefore provides no estimate of parameter uncertainty.

Instead, therefore, we employ a generalization of a method introduced in [34], which harnesses standard mixture-modelling techniques, adapting them to the network context. The method can be viewed as a general sampler for models in the family of Refs. [3, 23, 32, 33, 64].

A.4 *Inference in practice*

The general idea behind our method is to compute expectations of the form (A.8) in two manageable steps by factorizing the joint posterior as

$$P(A, \theta|X) = P(A|\theta, X)P(\theta|X). \tag{A.9}$$

This factorization tells us that we can draw samples from the joint posterior by first sampling sets of parameter values θ from the marginal distribution $P(\theta|X)$ and then sampling networks A from $P(A|\theta, X)$ with these parameter values. If we sample m different parameter sets and then n networks for each set, we end up with mn network/parameter pairs, which we number $r = 1 \ldots mn$. Then we can estimate the average in Eq. (A.8) as

$$\langle f(A, \theta) \rangle = \sum_A \int f(A, \theta)P(A|\theta, X)P(\theta|X) \, d\theta$$

$$\simeq \frac{1}{mn} \sum_{r=1}^{mn} f(A_r, \theta_r). \tag{A.10}$$

This expression is completely general and holds for any posterior, but for the class of models we consider here there are, as we now show, particularly efficient methods that can help us quickly generate the samples we need.

A.4.1 *Generating parameter samples*

The first step of the sampling algorithm draws values of the parameters θ from the marginal distribution

$$P(\theta|X) = \sum_A P(\theta, A|X), \tag{A.11}$$

where the sum runs over all the possible matrices A. For models with the factorized form (A.4), we have

$$P(\theta|X) \propto P(\theta) \sum_A \prod_{(i,j)} \mu_{ij}(A_{ij}, \theta) \nu_{ij}(A_{ij}, \theta)$$

$$\propto P(\theta) \prod_{(i,j)} \sum_k \mu_{ij}(k, \theta) \nu_{ij}(k, \theta). \tag{A.12}$$

Modern probabilistic programming languages make it easy to generate random samples from factorized marginals of this kind. Our code is written in the probabilistic language Stan, which implements the technique known as Hamiltonian Monte Carlo to generate samples automatically and efficiently—see Refs. [65, 66] for an introduction. Evaluating $P(\theta|X)$ involves a product over pairs (i, j) of nodes, of which there are $O(n^2)$, meaning that in general generating a sample takes $O(n^2)$ time. In many cases, however, the time complexity can be reduced to $O(n)$ by pooling terms in the product, as discussed in Section A.6.2.

A.4.2 *Generating network samples*   Given sampled values $\theta_1, \ldots, \theta_m$ of the parameters, the next step is to generate samples of the network A from the distribution $P(A|\theta, X)$ for these parameter values. This is straightforward for the factorized model assumed here, since node pairs are independent and we can sample each one separately. Specifically, using Eqs. (A.4) and (A.12), we have

$$P(A|\theta, X) = \frac{P(\theta, A|X)}{P(\theta|X)} = \frac{\prod_{(i,j)} \mu_{ij}(A_{ij}) \nu_{ij}(A_{ij})}{\prod_{(i,j)} \sum_k \mu_{ij}(k) \nu_{ij}(k)}$$
$$= \prod_{(i,j)} Q_{ij}(A_{ij}, \theta), \tag{A.13}$$

where

$$Q_{ij}(k, \theta) = \frac{\mu_{ij}(k) \nu_{ij}(k)}{\sum_{k'} \mu_{ij}(k') \nu_{ij}(k')} \tag{A.14}$$

is the posterior probability that nodes i and j are joined by an edge of type k. Generating networks is simply a matter of drawing a value $A_{ij} = k$ for each node pair independently from the distribution over k implied by $Q_{ij}(k)$. Again, naively this takes time $O(n^2)$ for all node pairs, but on a sparse network the speed can be improved, see Sec. A.6.1.

To estimate the average $\langle f(A, \theta) \rangle$, we generate a series of parameter sets θ using Eq. (A.12) and for each of these a series of networks using Eq. (A.13), then evaluate the average with Eq. (A.10).

A.5 *Assessing goodness of fit*

The method described above is simple, efficient and often gives good results. As described in the main text, however, the method can fail if the model itself is faulty—if the model is a poor representation of the system, failing to fit the data for any parameter values. It is important therefore to verify that the fit between model and data is good, which can be done with the standard technique of *posterior-predictive assessment*. As described in the main text, this involves generating synthetic data $\tilde{X}$ from the distribution implied by the fitted model:

$$P(\tilde{X}|X) = \int \sum_A P(\tilde{X}|\theta, A) P(\theta, A|X) \, d\theta. \tag{A.15}$$

This distribution weights all the possible parameters θ and networks A with their appropriate posterior probabilities and tells us the probability that a new data set $\tilde{X}$ would have if it were truly generated by the model with these inputs. The idea of the posterior-predictive assessment is to compare these synthetic data with the original input X. If the two look alike then the model has captured the data well; otherwise, it has not.

There are a number of ways to quantify the similarity of $\tilde{X}$ and X. For instance, one can compute the average

$$\langle \tilde{X}_{ij} \rangle = \sum_{\tilde{X}} P(\tilde{X}|X) \tilde{X}_{ij}, \tag{A.16}$$

and compare the result with X_{ij}. Visualizing the matrix of residues $\langle \tilde{X} \rangle - X$, the distribution of these residues, or how they depend on X_{ij} allows one to easily spot systematic issues with the model [18]. Such calculations are not costly in practice: the distribution (A.15) is just an average of a known function of A, θ over the posterior distribution and has the same general form as Eq. (A.10), so it can be evaluated numerically by the same methods. In this particular case, however, we can do even better, skipping the network sampling step altogether and making an estimate directly from the parameter samples. To do this, we write the distribution of Eq. (A.15) in the form

$$
\begin{aligned}
P(\tilde{X}|X) &= \int \sum_A P(\tilde{X}|\theta, A) P(A|\theta, X) P(\theta|X) \, d\theta \\
&= \int P(\theta|X) \sum_A \prod_{(i,j)} \tilde{\mu}_{ij}(A_{ij}, \theta) \, Q_{ij}(A_{ij}, \theta) \, d\theta \\
&= \int P(\theta|X) \prod_{(i,j)} \sum_k \tilde{\mu}_{ij}(k, \theta) \, Q_{ij}(k, \theta) \, d\theta,
\end{aligned}
\tag{A.17}
$$

where have used Eqs. (A.2) and (A.13) in the second line, and $\tilde{\mu}_{ij}(k, \theta)$ is the probability of generating a synthetic measurement $\tilde{X}_{ij}$ given that (i, j) is an edge of type k. This expression is now independent of A and only requires an average over θ to evaluate.

Using this expression for $P(\tilde{X}|X)$, we can write the average $\langle \tilde{X}_{ij} \rangle$ in Eq. (A.16) as

$$
\langle \tilde{X}_{ij} \rangle = \int P(\theta|X) \sum_k \langle \tilde{\mu}_{ij}(k, \theta) \rangle Q_{ij}(k, \theta) \, d\theta,
\tag{A.18}
$$

which we evaluate numerically as

$$
\langle \tilde{X}_{ij} \rangle \simeq \frac{1}{m} \sum_{r=1}^m \sum_k \langle \tilde{\mu}_{ij}(k, \theta_r) \rangle Q_{ij}(k, \theta_r).
\tag{A.19}
$$

Note that $\langle \tilde{\mu}_{ij}(k, \theta_r) \rangle$ usually has a simple closed form, since it is just the mean of $\tilde{X}_{ij}$ within the data model with parameters θ_r.

A visual inspection of the residues between $\tilde{X}$ and X is often enough to reveal issues with goodness of fit, but one can carry out a more formal model assessment using any of a variety of *discrepancy* measures that quantify the distance between the synthetic data $\tilde{X}$ and the original X [63]. The average value of such a discrepancy will always be greater than zero, since one does not expect the synthetic and original data to agree perfectly even with a perfect model. To obtain a baseline against which discrepancy values can be compared, we therefore compute the discrepancy between synthetic measurements $\tilde{X}$ and *their* associated predictions, calculating a model-versus-model discrepancy distribution.

In the calculations presented here, we make use of the log-likelihood ratio discrepancy:

$$
D(X, \theta_r) = \sum_{(i,j)} X_{ij} \log \frac{X_{ij}}{\langle \tilde{X}_{ij}(\theta_r) \rangle},
\tag{A.20}
$$

where $\langle \tilde{X}_{ij}(\theta_r) \rangle$ is evaluated using a single term of the sum appearing in equation Eq. (A.19), with the sampled parameter values θ_r. This discrepancy is reminiscent of a Kullback–Leibler divergence, with

the primary difference being that it compares unnormalized quantities rather than normalized probability distributions. That said, the norm of the two sets of measurements should be similar, since the whole purpose of the calculation is to reproduce the original observations. Hence, one can usually interpret the discrepancy in more or less the same way as a divergence: the smaller the divergence the better the fit (although values slightly less than zero can occur, which is not true of a true divergence).

We compute the distribution of the discrepancy and the reference distribution $\tilde{X}$ simultaneously using the method introduced in Ref. [63]. We go through each network/parameter sample A_r, θ_r and generate a single realization $\tilde{X}$ of the synthetic data from the data model, then compute the two discrepancies $D(\tilde{X}, \theta_r)$ and $D(X, \theta_r)$ using Eq. (A.19). From the resulting sets of discrepancy values one can then compute the p-value $p = P[D(X, \theta_r) > D(\tilde{X}, \theta_r)]$, which is the fraction of artificial datasets with discrepancy at least as large as the observed value. If the p-value is too small the model is rejected. If the p-value is too large then there is a danger that it is overfitting the data, which can be treated by regularizing the model using hierarchical priors or by changing the model entirely. This calculation does not cost much computation time since we are merely reusing the samples already generated for estimation purposes.

A.6 *Implementation*

In this section, we discuss details of implementation, including a number of techniques for optimizing the speed and numerical accuracy of the algorithm which can be useful with large data sets. Even without such optimizations the algorithm should run reasonably quickly on typical hardware for networks with up to a few hundred nodes. But with these optimizations—and with a suitable choice of models—the method can scale to hundred of thousands of nodes or more.

A.6.1 *Sampling networks*   One of the more computationally costly steps in the algorithm is the generation of sample networks from the conditional posterior distribution $P(A|\theta, X)$. Naively generating the network by flipping a biased coin for every node pair i, j takes time $O(n^2)$ on a network of n nodes.

For some models on sparse networks, this time can be reduced by explicitly sampling only the edges that exist. That is, all edges are assumed not to exist, except for a sparse sample that are generated in accordance with the fitted model. For instance, with the simple 'uniform error' model of Table A1, the posterior probabilities Q_{ij} of edges are a unique function $Q(X)$ of the number of observations X_{ij} of the edge in question. With this in mind we define $\Sigma = \sum_{(i,j)} Q_{ij} = \sum_X n(X)Q(X)$ where $n(X) = \sum_{(i,j)} \delta(X, X_{ij})$ is the number of node pairs with X observations and $\delta(x, y)$ is the Kronecker delta.

The value of Σ can be calculated rapidly once $n(X)$ is known, then we can generate a sampled network by first drawing an integer $M \sim \text{Poisson}(\Sigma)$ to represent the number of edges in the network, and then generating M random edges with probabilities Q_{ij} with standard 'roulette wheel' proportional sampling using binary search. The complete process takes time $O(M \log n)$, which on a sparse network will be much faster than the $O(n^2)$ of the naive algorithm.

In other cases, we may be able to skip the process of network sampling altogether, although at the price of still having to perform $O(n^2)$ operations. Specifically, when we want to calculate the average of a function f that factorizes over node pairs thus

$$f(A, \theta) = \prod_{(i,j)} g_{ij}(A_{ij}, \theta), \tag{A.21}$$

we can write the average as

$$\langle f(\mathbf{A}, \theta) \rangle = \sum_{\mathbf{A}} \int f(\mathbf{A}, \theta) P(\mathbf{A}|\theta, \mathbf{X}) P(\theta|\mathbf{X}) \, d\theta$$

$$= \int P(\theta|\mathbf{X}) \prod_{(i,j)} \sum_{k} \left[g_{ij}(k, \theta) Q_{ij}(k, \theta) \right] d\theta. \tag{A.22}$$

Now, we sample m sets of parameter values θ_r as usual, but generate no networks $\mathbf{A}$, and the average we want is given by

$$\langle f(\mathbf{A}, \theta) \rangle \simeq \frac{1}{m} \sum_{r=1}^{m} \prod_{(i,j)} \sum_{k} g_{ij}(k, \theta_r) Q_{ij}(k, \theta_r). \tag{A.23}$$

A.6.2 *Sampling parameters*   Generating sample values of the parameters also takes time $O(n^2)$ in general, because the right-hand side of Eq. (A.12) involves a product over pairs of nodes. For some models, however, we may be able to evaluate this product more rapidly by methods similar to those described for sampling networks above. Taking again the example of the 'uniform error' model from Table A1, the probability $\mu_{ij}(k)$ is a function $\mu(X, k, \theta)$ only of the number of observations X_{ij} of the corresponding edge (and k and θ) and $\nu_{ij}(k)$ is a function of k and θ only. This means we can group terms in the product and write

$$\prod_{(i,j)} \sum_{k} \mu_{ij}(k, \theta) \nu_{ij}(k, \theta) = \prod_{X} \left[\sum_{k} \mu(X, k, \theta) \nu(k, \theta) \right]^{n(X)}, \tag{A.24}$$

which saves considerable time.

REFERENCES

1. NEWMAN, M. (2018) *Networks*, 2nd edn. Oxford: Oxford University Press.
2. KOLACZYK, E. D. (2009) *Statistical Analysis of Network Data*. New York, NY: Springer.
3. NEWMAN, M. E. J. (2018) Network structure from rich but noisy data. *Nat. Phys.*, **14**, 542–545.
4. ITO, T. et al. (2000) Toward a protein–protein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc. Natl. Acad. Sci. USA*, **97**, 1143–1147.
5. KROGAN, N. J. et al. (2006) Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature*, **440**, 637–643.
6. SPRINZAK, E., SATTATH, S. & MARGALIT, H. (2003) How reliable are experimental protein-protein interaction data? *J. Mol. Biol.*, **327**, 919–923.
7. ROLLAND, T. et al. (2014) A proteome-scale map of the human interactome network. *Cell*, **159**, 1212–1226.
8. WASSERMAN, S. & FAUST, K. (1994) *Social Network Analysis*. Cambridge: Cambridge University Press.
9. BALL, B. & NEWMAN, M. E. J. (2013) Friendship networks and social status. *Netw. Sci.*, **1**, 16–30.
10. VAQUERA, E. & KAO, G. (2008) Do you like me as much as I like you? Friendship reciprocity and its effects on school outcomes among adolescents. *Soc. Sci. Res.*, **37**, 55–72.
11. MCPHERSON, M., SMITH-LOVIN, L. & COOK, J. M. (2001) Birds of a feather: homophily in social networks. *Annu. Rev. Sociol.*, **27**, 415–444.

12. AMINI, L., SHAIKH, A. & SCHULZRINNE, H. (2004) Issues with inferring Internet topological attributes. *Comput. Commun.*, **27**, 557–567.

13. HOLLAND, P. W. & LEINHARDT, S. (1973) The structural implications of measurement error in sociometry. *J. Math. Sociol.*, **3**, 85–111.

14. SPORNS, O. (2010) *Networks of the Brain*. Cambridge, MA: MIT Press.

15. WANG, D. J., SHI, X., MCFARLAND, D. A. & LESKOVEC, J. (2012) Measurement error in network data: a re-classification. *Soc. Netw.*, **34**, 396–409.

16. WHITEHEAD, H. (2008) *Analyzing Animal Societies: Quantitative Methods for Vertebrate Social Analysis*. Chicago, IL: University of Chicago Press.

17. WIESE, J., MIN, J.-K., HONG, J. I. & ZIMMERMAN, J. (2015) You never call, you never write: call and SMS logs do not always indicate tie strength. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. New York, NY: Association for Computing Machinery, pp. 765–774.

18. GELMAN, A. & SHALIZI, C. R. (2013) Philosophy and the practice of Bayesian statistics. *Br. J. Math. Stat. Psychol.*, **66**, 8–38.

19. CRANDALL, D. J., BACKSTROM, L., COSLEY, D., SURI, S., HUTTENLOCHER, D. & KLEINBERG, J. (2010) Inferring social ties from geographic coincidences. *Proc. Natl. Acad. Sci. USA*, **107**, 22436–22441.

20. CRANSHAW, J., TOCH, E., HONG, J., KITTUR, A. & SADEH, N. (2010) Bridging the gap between physical location and online social networks. *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*. New York, NY: Association for Computing Machinery, pp. 119–128.

21. EAGLE, N. & PENTLAND, A. (2006) Reality mining: sensing complex social systems. *J. Pers. Ubiquitous Comput.*, **10**, 255–268.

22. EAGLE, N., PENTLAND, A. S. & LAZER, D. (2009) Inferring friendship network structure by using mobile phone data. *Proc. Natl. Acad. Sci. USA*, **106**, 15274–15278.

23. BUTTS, C. T. (2003) Network inference, error, and informant (in)accuracy: a Bayesian approach. *Soc. Netw.*, **25**, 103–140.

24. FARINE, D. R. & STRANDBURG-PESHKIN, A. (2015) Estimating uncertainty and reliability of social network data using Bayesian inference. *R. Soc. Open Sci.*, **2**, 150367.

25. LE, C. M., LEVIN, K. & LEVINA, E. (2018) Estimating a network from multiple noisy realizations. *Electron. J. Stat.*, **12**, 4697–4740.

26. TANG, R. et al. (2018) Connectome smoothing via low-rank approximations. *IEEE Trans. Med. Imaging.*, **38**, 1446–1456.

27. WANG, L., ZHANG, Z. & DUNSON, D. (2019) Common and individual structure of brain networks. *Ann. Appl. Stat.*, **13**, 85–112.

28. BIRLUTIU, A., D'ALCHÉ BUC, F. & HESKES, T. (2014) A Bayesian framework for combining protein and network topology information for predicting protein–protein interactions. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **12**, 538–550.

29. JANSEN, R. et al. (2003) A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, **302**, 449–453.

30. JIANG, X. & KOLACZYK, E. D. (2012) A latent eigenprobit model with link uncertainty for prediction of protein-protein interactions. *Stat. Biosci.*, **4**, 84–104.

31. PRIEBE, C. E., SUSSMAN, D. L., TANG, M. & VOGELSTEIN, J. T. (2015) Statistical inference on errorfully observed graphs. *J. Comput. Graph. Stat.*, **24**, 930–953.

32. PEIXOTO, T. P. (2018) Reconstructing networks with unknown and heterogeneous errors. *Phys. Rev. X*, **8**, 041011.

33. NEWMAN, M. E. J. (2018) Estimating network structure from unreliable measurements. *Phys. Rev. E*, **98**, 062321.

34. YOUNG, J.-G., VALDOVINOS, F. S. & NEWMAN, M. E. J. (2019) Reconstruction of plant-pollinator networks from observational data. Preprint bioRxiv:754077.

35. MCLACHLAN, G. J. & PEEL, D. (2004) *Finite Mixture Models*. New York, NY: Wiley.

36. TITTERINGTON, D. M., SMITH, A. & MAKOV, U. E. (1985) *Statistical Analysis of Finite Mixture Distributions*. New York, NY: Wiley.

37. CLAUSET, A., MOORE, C. & NEWMAN, M. E. J. (2008) Hierarchical structure and the prediction of missing links in networks. *Nature*, **453**, 98–101.

38. GUIMERÀ, R. & SALES-PARDO, M. (2009) Missing and spurious interactions and the reconstruction of complex networks. *Proc. Natl. Acad. Sci. USA*, **106**, 22073–22078.

39. HUISMAN, M. (2009) Imputation of missing network data: some simple procedures. *J. Soc. Struct.*, **10**, 1–29.

40. KIM, M. & LESKOVEC, J. (2011) The network completion problem: inferring missing nodes and edges in networks. *Proceedings of the 2011 SIAM International Conference on Data Mining* (Liu, B., Liu, H., Clifton, C., Washio, T. & Kamath, C., eds). Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. 47–58.

41. LIBEN-NOWELL, D. & KLEINBERG, J. (2007) The link-prediction problem for social networks. *J. Assoc. Inf. Sci. Technol.*, **58**, 1019–1031.

42. BRUGERE, I., GALLAGHER, B. & BERGER-WOLF, T. Y. (2018) Network structure inference, a survey: motivations, methods, and applications. *ACM Comput. Surv.*, **51**, 24:1–24:39.

43. LI, Q., ZHENG, Y., XIE, X., CHEN, Y., LIU, W. & MA, W.-Y. (2008) Mining user similarity based on location history. *Proceedings of the 16th ACM Sigspatial International Conference on Advances in Geographic Information Systems*. New York, NY: Association for Computing Machinery.

44. BANSAL, M., BELCASTRO, V., AMBESI-IMPIOMBATO, A. & DI BERNARDO, D. (2007) How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, **3**, 78.

45. GOMEZ-RODRIGUEZ, M., LESKOVEC, J. & KRAUSE, A. (2012) Inferring networks of diffusion and influence. *ACM Trans. Knowl. Discov. Data*, **5**, 21.

46. NETRAPALLI, P. & SANGHAVI, S. (2012) Learning the graph of epidemic cascades. *Proceedings of the 12th ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*. New York: Association of Computing Machinery, pp. 211–222.

47. SQUARTINI, T. & GARLASCHELLI, D. (2017) *Maximum-Entropy Networks: Pattern Detection, Network Reconstruction and Graph Combinatorics*. Berlin:Springer.

48. YUAN, M. & LIN, Y. (2007) Model selection and estimation in the Gaussian graphical model. *Biometrika*, **94**, 19–35.

49. LEE, S. H., KIM, P.-J. & JEONG, H. (2006) Statistical properties of sampled networks. *Phys. Rev. E*, **73**, 016102.

50. ORBANZ, P. (2017) Subsampling large graphs and invariance in networks. Preprint arXiv:1710.04217.

51. STUMPF, M. P. H., WIUF, C. & MAY, R. M. (2005) Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proc. Natl. Acad. Sci. USA*, **102**, 4221–4224.

52. BUTTS, C. T. (2009) Revisiting the foundations of network analysis. *Science*, **325**, 414–416.

53. FERREIRA, A. A., GONCALVES, M. A. & LAENDER, A. H. F. (2012) A brief survey of automatic methods for author name disambiguation. *SIGMOD Record*, **41**, 15–26.

54. NAMATA, G. M., LONDON, B. & GETOOR, L. (2016) Collective graph identification. *ACM Trans. Knowl. Discov. Data*, **10**, 25.

55. BONCHI, F., GULLO, F., KALTENBRUNNER, A. & VOLKOVICH, Y. (2014) Core decomposition of uncertain graphs. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY: Association for Computing Machinery, pp. 1316–1325.

56. KHAN, A., YE, Y. & CHEN, L. (2018) *On Uncertain Graphs*. San Rafael, CA:Morgan & Claypool.

57. MARTIN, T., BALL, B. & NEWMAN, M. E. J. (2016) Structural inference for uncertain networks. *Phys. Rev. E*, **93**, 012306.

58. PFEIFFER, J. J. & NEVILLE, J. (2011) Methods to determine node centrality and clustering in graphs with uncertain structure. *Fifth International AAAI Conference on Weblogs and Social Media*. Menlo Park, CA: The AAAI Press.

59. POISOT, T., CIRTWILL, A. R., CAZELLES, K., GRAVEL, D., FORTIN, M.-J. & STOUFFER, D. B. (2016) The structure of probabilistic networks. *Methods Ecol. Evol.*, **7**, 303–312.

60. CONNOR, R. C., SMOLKER, R. A. & RICHARDS, A. F. (1992) Dolphin alliances and coalitions. *Coalitions and Alliances in Humans and Other Animals*. (A. H. Harcourt & F. B. M. de Waal eds). New York: Oxford University Press, p. 443.

61. BRASK, J. B., ELLIS, S. & CROFT, D. P. (2020) Animal social networks–an introduction for complex systems scientists. Preprint arXiv:2005.09598.

62. GELMAN, A., STERN, H. S., CARLIN, J. B., DUNSON, D. B., VEHTARI, A. & RUBIN, D. B. (2013) *Bayesian Data Analysis*, 3rd edn. New York, NY: Chapman and Hall/CRC.
63. GELMAN, A., MENG, X.-L. & STERN, H. (1996) Posterior predictive assessment of model fitness via realized discrepancies. *Stat. Sin.*, **6**, 733–760.
64. PEIXOTO, T. P. (2019) Network reconstruction and community detection from dynamics. *Phys. Rev. Lett.*, **123**, 128301.
65. BETANCOURT, M. (2017) A conceptual introduction to Hamiltonian Monte Carlo. Preprint arXiv:1701.02434.
66. CARPENTER, B. et al. (2017) Stan: a probabilistic programming language. *J. Stat. Softw.*, **76**, 1.