1 Introduction

Community detection is the task of dividing a network — typically one which is large — into many smaller groups of nodes that have a similar contribution to the overall network structure. With such a division, we can better summarize the large-scale structure of a network by describing how these groups are connected, rather than describing each individual node. This simplified description can be used to digest an otherwise intractable representation of a large system, providing insight into its most important patterns, how these patterns relate to its function, and the underlying mechanisms responsible for its formation.

Because of its important role in network science, community detection has attracted substantial attention from researchers, specially in the last 20 years, culminating in an abundant literature (see Refs. [1, 2] for a review). This field has developed significantly from its early days, specially over the last 10 years, during which the focus has been shifting towards methods that are based on statistical inference (see e.g. Refs. [3–5]).

Despite this shift in the state-of-the-art, there remains a significant gap between the best practices and the adopted practices in the use of community detection for the analysis of network data. It is still the case that some of the earliest methods proposed remain in widespread use, despite their many serious shortcomings that have been uncovered over the years. Most of these problems have been addressed with more recent methods, that also contributed to a much deeper theoretical understanding of the problem of community detection [3, 4, 6, 7].

Nevertheless, some misconceptions remain and are still promoted. Here we address some of the more salient ones, in an effort to dispel them. These misconceptions are not uniformly shared; and those that pay close attention to the literature will likely find few surprises here. However, it is possible that many researchers employing community detection are simply unaware of the issues with the methods being used. Perhaps even more commonly, there are those that are in fact aware of them, but not of their actual solutions, or the fact that some supposed countermeasures are ineffective.

Throughout the following we will avoid providing "black box" recipes to be followed uncritically, and instead try as much as possible to frame the issues within a theoretical framework, such that the criticisms and solutions can be justified in a principled manner.

We will set the stage by making a fundamental distinction between "descriptive" and "inferential" community detection approaches. As others have emphasized before [8], community detection can be performed with many goals in mind, and this will dictate which methods are most appropriate. We will

provide a simple "litmus test" that can be used to determine which overall approach is more adequate, based on whether our goal is to seek inferential interpretations. We will then move to a more focused critique of the method that is arguably the most widely employed — modularity maximization. This method has an exemplary character, since it contains all possible pitfalls of using descriptive methods for inferential aims. We will then follow with a discussion of myths, pitfalls, and half-truths that obstruct a more effective analysis of community structure in networks.

(We will not give a throughout technical introduction to inferential community detection methods, which can be obtained instead in Ref. [5]. For a practical guide on how to use various inferential methods, readers are referred to the detailed HOWTO[1] available as part of the `graph-tool` Python library [9].)

2 Descriptive vs. inferential community detection

At a very fundamental level, community detection methods can be divided into two main categories: "descriptive" and "inferential."

Descriptive methods attempt to find communities according to some context-dependent notion of a good division of the network into groups. These notions are based on the patterns that can be identified in the network via an exhaustive algorithm, but without taking into consideration the possible rules that were used to create the patterns uncovered. These patterns are used only to *describe* the network, not to explain it. Usually, these approaches do not articulate precisely what constitutes community structure to begin with, and focus instead only on how to detect such patterns. For this kind of method, concepts of statistical significance, parsimony, and generalizability are usually not evoked.

Inferential methods, on the other hand, start with an explicit definition of what constitutes community structure, via a generative model for the network. This model describes how a *latent* (i.e. not observed) partition of the nodes would affect the placement of the edges. The inference consists on reversing this procedure to determine which node partitions are more likely to have been responsible for the observed network. The result of this is a "fit" of a model to data, that can be used as a tentative explanation of how the network came to be. The concepts of statistical significance, parsimony, and generalizability arise naturally and can be quantitatively assessed in this context.

[1] Available at https://graph-tool.skewed.de/static/doc/demos/inference/inference.html.

Descriptive community detection methods are by far the most numerous, and those that are in most widespread use. However, this contrasts with the current state-of-the-art, which is composed in large part of inferential approaches. Here we point out the major differences between them and discuss how to decide which is more appropriate, and also why one should in general favor the inferential varieties whenever the objective is to derive generative interpretations from data.

2.1 Describing vs. explaining

We begin by observing that descriptive clustering approaches are the methods of choice in certain contexts. For instance, such approaches arise naturally when the objective is to divide a network into two or more parts as a means to solve a variety of optimization problems. Arguably, the most classic example of this is the design of very large scale integrated (VLSI) circuits [10]. The task is to combine from up to billions of transistors into a single physical microprocessor chip. Transistors that connect to each other must be placed together to take less space, consume less power, reduce latency, and reduce the risk of crosstalk with other nearby connections. To achieve this, the initial stage of a VLSI process involves the partitioning of the circuit into many smaller modules with few connections between them, in a manner that enables their efficient spatial placement, i.e. by positioning the transistors in each module close together and those in different modules farther apart.

Another notable example is parallel task scheduling, a problem that appears in computer science and operations research. The objective is to distribute processes (i.e. programs, or tasks in general) between different processors, so they can run at the same time. Since processes depend on the partial results of other processes, this forms a dependency network, which then needs to be divided such that the number of dependencies across processors is minimized. The optimal division is the one where all tasks are able to finish in the shortest time possible.

Both examples above, and others, have motivated a large literature on "graph partitioning" dating back to the 70s [11–13], which covers a family of problems that play an important role in computer science and algorithmic complexity theory.

Although reminiscent of graph partitioning, and sharing with it many algorithmic similarities, community detection is used more broadly with a different goal [1, 2]. Namely, the objective is to perform *data analysis*, where one wants to extract scientific understanding from empirical observations. The communities identified are usually directly used for representation and/or interpretation of the data, rather than as a mere device to solve a particular optimization

problem. In this context, a merely descriptive approach will fail at giving us a meaningful insight into the data, and can be misleading, as we will discuss in the following.

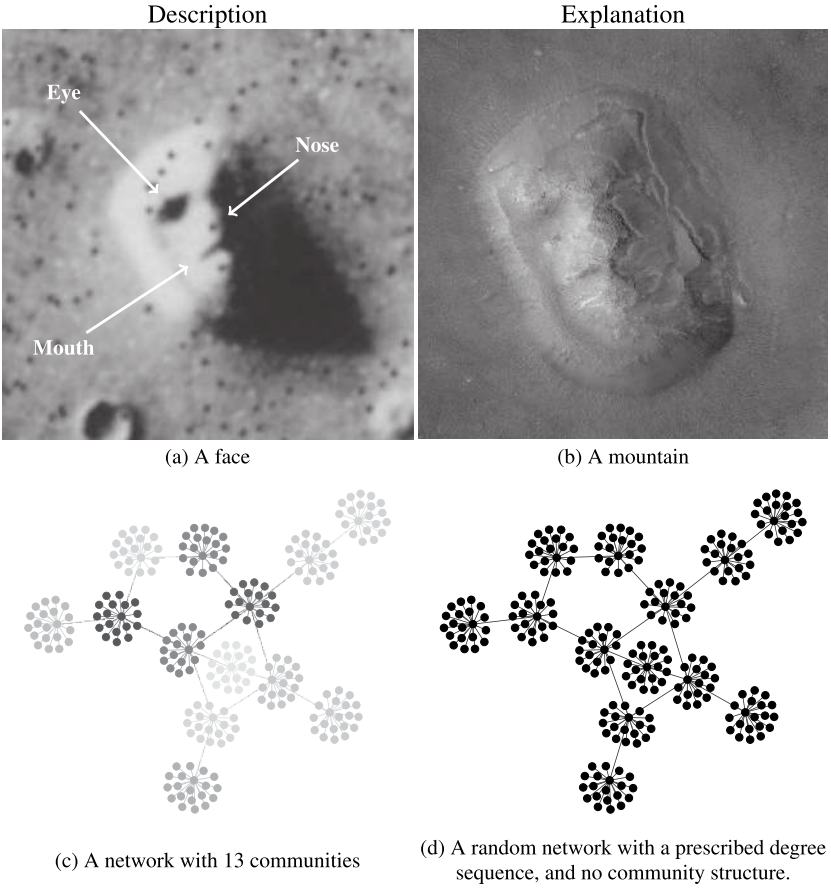
We illustrate the difference between descriptive and inferential approaches in Fig. 1. We first make an analogy with the famous "face" seen on images of the *Cydonia Mensae* region of the planet Mars. A merely descriptive account of the image can be made by identifying the facial features seen, which most people immediately recognize. However, an inferential description of the same image would seek instead to *explain* what is being seen. The process of explanation must invariably involve at its core an application of the law of parsimony, or **Occam's razor**. This principle predicates that when considering two hypotheses compatible with an observation, the simplest one must prevail. Employing this logic results in the conclusion that what we are seeing is in fact a regular mountain, without denying that it looks like a face in that picture and instead acknowledging that it does so accidentally. In other words, the "facial" description is not useful as an explanation, as it emerges out of random features rather than exposing any underlying mechanism.

Going out of the analogy and back to the problem of community detection, in Fig. 1(c) and (d) we see a descriptive and an inferential account of an example network, respectively. The descriptive one is a division of the nodes into 13 assortative communities, which would be identified with many descriptive community detection methods available in the literature. Indeed, we can inspect visually that these groups form assortative communities,[2] and most people would agree that these communities are really there, according to most definitions in use: these are groups of nodes with many more internal edges than external ones. However, an inferential account of the same network would reveal something else altogether. Specifically, it would explain this network as the outcome of a process where the edges are placed at random, without the existence of any communities. The communities that we see in Fig. 1(c) are just a byproduct of this random process, and therefore carry no explanatory power. In fact, this is exactly how the network in this example was generated, i.e. by choosing a specific degree sequence and connecting the edges uniformly at random.

In Fig. 2(a) we illustrate in more detail how the network in Fig. 1 was generated: The degrees of the nodes are fixed, forming "stubs" or "half-edges," which are then paired uniformly at random forming the edges of the network.[3]

[2] See Sec. 4.6 for possible pitfalls with relying on visual inspections.

[3] This uniform pairing will typically also result in the occurrence of pairs of nodes of degree one connected together in their own connected component. We consider instances of the process



(a) A face (b) A mountain

(c) A network with 13 communities

(d) A random network with a prescribed degree sequence, and no community structure.

Figure 1 Difference between descriptive and inferential approaches to data analysis. As an analogy, in panels (a) and (b) we see two representations of the *Cydonia Mensae* region on Mars. Panel (a) is a descriptive account of what we see in the picture, namely a face. Panel (b) is an inferential representation of what lies behind it, namely a mountain (this is a more recent image of the same region with a higher resolution to represent an inferential interpretation of the figure in panel (a)). More concretely for the problem of community detection, in panels (c) and (d) we see two representations of the same network. Panel (c) shows a descriptive division into 13 assortative communities. In panel (d) we see an inferential representation as a degree-constrained random network, with no communities, since this is a more likely model of how this network was formed (see Fig. 2).

In Fig. 2(b), like in Fig. 1, the node colors show the partition found with descriptive community detection methods. However, this network division carries no

where this does not happen for visual clarity in Fig. 2(c) and (d), but without sacrificing its main message.

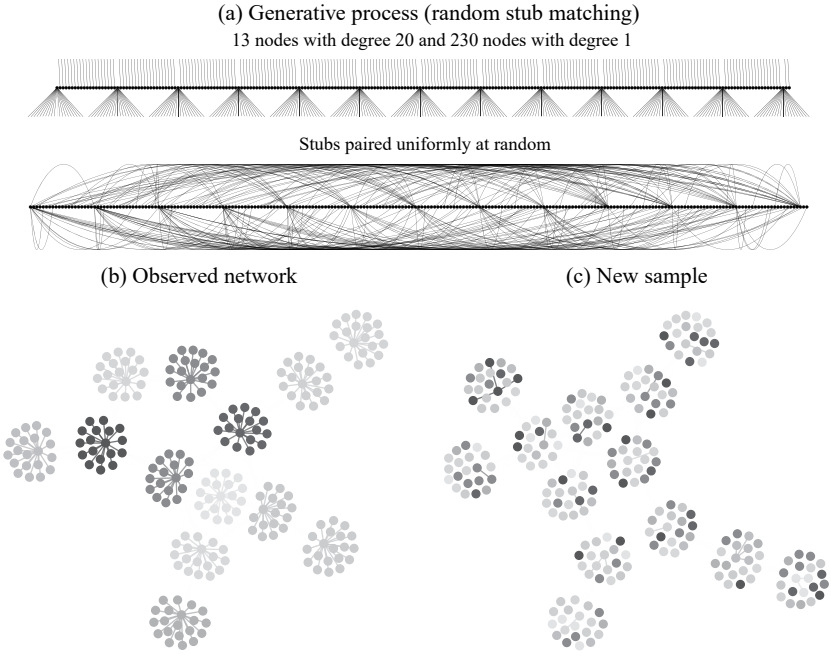


Figure 2 Descriptive community detection finds a partition of the network according to an arbitrary criterion that bears in general no relation to the rules that were used to generate it. In (a) is shown the generative model we consider, where first a degree sequence is given to the nodes (forming "stubs", or "half-edges") which then are paired uniformly at random, forming a graph. In (b) is shown a realization of this model. The node colors show the partition found with virtually any descriptive community detection method. In (c) is shown another network sampled from the same model, together with the same partition found in (b), which is completely uncorrelated with the new apparent communities seen, since they are the mere byproduct of the random placement of the edges. An inferential approach would find only a single community in both (b) and (c), since no partition of the nodes is relevant for the underlying generative model.

explanatory power beyond what is contained in the degree sequence of the network, since it is generated otherwise uniformly at random. This becomes evident in Fig. 2(c), where we show another network sampled from the same generative process, i.e. another random pairing, but partitioned according to the same division as in Fig. 2(b). Since the nodes are paired uniformly at random, constrained only by their degree, this will create new apparent "communities" that are always uncorrelated with one another. Like the "face" on Mars, they can be seen and described, but they cannot (plausibly) explain how the network came to be.

We emphasize that the communities found in Fig. 2(b) are indeed really there from a descriptive point of view, and they can in fact be useful for a variety of

tasks. For example, the *cut* given by the partition, i.e. the number of edges that go between different groups, is only 13, which means that we need only to remove this number of edges to break the network into (in this case) 13 smaller components. Depending on context, this kind of information can be used to prevent a widespread epidemic, hinder undesired communication, or, as we have already discussed, distribute tasks among processors and design a microchip. However, what these communities *cannot* be used for is to *explain* the data. In particular, a conclusion that would be completely incorrect is that the nodes that belong to the same group would have a larger probability of being connected between themselves. As shown in Fig. 2(a), this is clearly not the case, as the observed "communities" arise by pure chance, without any preference between the nodes.

2.2 To infer or to describe? A litmus test

Given the above differences, and the fact that both inferential and descriptive approaches have their uses depending on context, we are left with the question: Which approach is more appropriate for a given task at hand? In order to help answering this question, for any given context, it is useful to consider the following "litmus test":

Litmus test: to infer or to describe?

Q: "Would the usefulness of our conclusions change if we learn, after obtaining the communities, that the network being analyzed is maximally random?"

If the answer is "yes," then an inferential approach is needed.

If the answer is "no," then an inferential approach is not required.

If the answer to the above question is "yes," then an inferential approach is warranted, since the conclusions depend on an interpretation of how the data were generated. Otherwise, a purely descriptive approach may be appropriate since considerations about generative processes are not relevant.

It is important to understand that the relevant question in this context is not whether the network being analyzed is *actually* maximally random,[4] since this

[4] "Maximally random" here means that, conditioned on some global or local constraints, like the number of edges or the node degrees, the placement of the edges is done in uniformly at random. In other words, the network is sampled from a maximum-entropy model constrained in a manner unrelated to community structure, such that whatever communities we may ascribe to the nodes could have played no role in the placement of the edges.

is rarely the case for empirical networks. Instead, considering this hypothetical scenario serves as a test to evaluate if our task requires us to separate between actual latent community structures (i.e. those that are responsible for the network formation), from those that arise completely out of random fluctuations, and hence carry no explanatory power. Furthermore, most empirical networks, even if not maximally random, like most interesting data, are better explained by a mixture of structure and randomness, and a method that cannot tell those apart cannot be used for inferential purposes.

Returning to the VLSI and task scheduling examples we considered in the previous section, it is clear that the answer to the litmus test above would be "no," since it hardly matters how the network was generated and how we should interpret the partition found, as long as the integrated circuit can be manufactured and function efficiently, or the tasks finish in the minimal time. Interpretation and explanations are simply not the primary goals in these cases.[5]

However, it is safe to say that in network data analyses very often the answer to the question above question would be "yes." Typically, community detection methods are used to try to understand the overall large-scale network structure, determine the prevalent mixing patterns, make simplifications and generalizations, all in a manner that relies on statements about what lies behind the data, e.g. whether nodes were more or less likely to be connected to begin with. A majority of conclusions reached would be severely undermined if one would discover that the underlying network is in fact completely random. This means that these analyses suffer the substantial risk of yielding misleading answers when using purely descriptive methods, since they are likely to be *overfitting* the data — i.e. confusing randomness with underlying generative structure.[6]

2.3 Inferring, explaining, and compressing

Inferential approaches to community detection (see Ref. [5] for a detailed introduction) are designed to provide explanations for network data in a principled manner. They are based on the formulation of generative models that include

[5] Although this is certainly true at a first instance, we can also argue that properly understanding *why* a certain partition was possible in the first place would be useful for reproducibility and to aid the design of future instances of the problem. For these purposes, an inferential approach would be more appropriate.

[6] We emphasize that the concept of overfitting is intrinsically tied with an inferential goal, i.e. one that involves interpretations about an underlying distribution of probability relating to the network structure. The partitioning of a graph with the objective of producing an efficient chip design cannot overfit, because it remove does not elicit an inferential interpretation. Therefore, whenever we mention that a method overfits, we refer only to the situation where it is being employed with an inferential goal, and that it incorporates a level of detail that cannot be justified by the statistical evidence available in the data.

the notion of community structure in the rules of how the edges are placed. More formally, they are based on the definition of a likelihood $P(\boldsymbol{A}|\boldsymbol{b})$ for the network $\boldsymbol{A}$ conditioned on a partition $\boldsymbol{b}$, which describes how the network could have been generated, and the inference is obtained via the posterior distribution, according to Bayes' rule, i.e.

$$P(\boldsymbol{b}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b})}{P(\boldsymbol{A})}, \tag{1}$$

where $P(\boldsymbol{b})$ is the prior probability for a partition $\boldsymbol{b}$. The inference procedure consists in sampling from or maximizing this distribution, which yields the most likely division(s) of the network into groups, according to the statistical evidence available in the data (see Fig. 3).

Overwhelmingly, the models used to infer communities are variations of the stochastic block model (SBM) [14], where in addition to the node partition, it takes the probability of edges being placed between the different groups as an additional set of parameters. A particularly expressive variation is the degree-corrected SBM (DC-SBM) [15], with a marginal likelihood given by [16]

$$P(\boldsymbol{A}|\boldsymbol{b}) = \sum_{\boldsymbol{e},\boldsymbol{k}} P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b}), \tag{2}$$

where $\boldsymbol{e} = \{e_{rs}\}$ is a matrix with elements e_{rs} specifying how many edges go between groups r and s, and $\boldsymbol{k} = \{k_i\}$ are the degrees of the nodes. Therefore, this model specifies that, conditioned on a partition $\boldsymbol{b}$, first the edge counts $\boldsymbol{e}$ are sampled from a prior distribution $P(\boldsymbol{e}|\boldsymbol{b})$, followed by the degrees from the prior $P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b})$, and finally the network is wired together according to the probability $P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})$, which respects the constraints given by $\boldsymbol{k}$, $\boldsymbol{e}$, and $\boldsymbol{b}$. See Fig. 3(a) for a illustration of this process.

This model formulation includes maximally random networks as special cases — indeed the model we considered in Fig. 2 corresponds exactly to the DC-SBM with a single group. Together with the Bayesian approach, the use of this model will inherently favor a more parsimonious account of the data, whenever it does not warrant a more complex description — amounting to a formal implementation of Occam's razor. This is best seen by making a formal connection with information theory, and noticing that we can write the numerator of Eq. 1 as

$$P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b}) = 2^{-\Sigma(\boldsymbol{A},\boldsymbol{b})}, \tag{3}$$

where the quantity $\Sigma(\boldsymbol{A},\boldsymbol{b})$ is known as the *description length* [17–19] of the network. It is computed as[7]

[7] Note that the sum in Eq. 2 vanishes because only one term is non-zero given a fixed network $\boldsymbol{A}$.

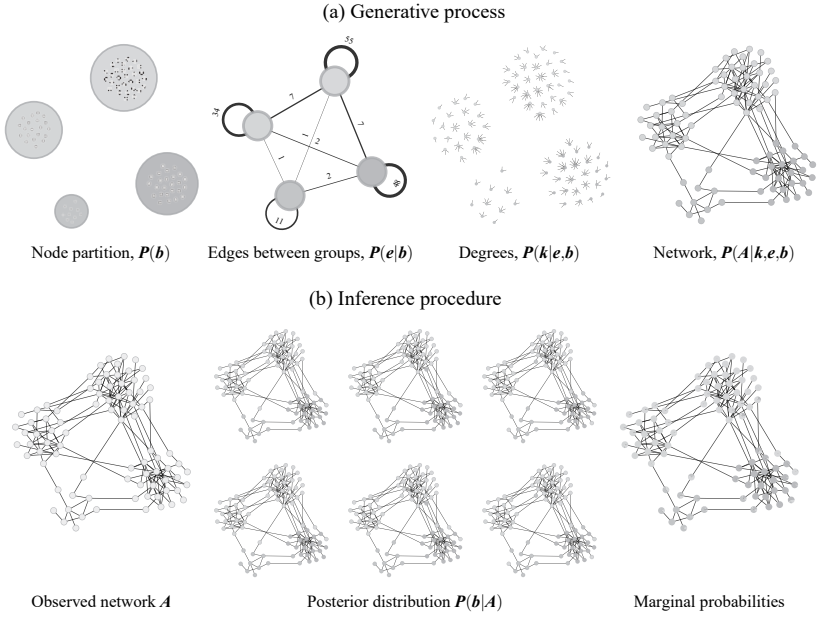


Figure 3 Inferential community detection considers a generative process (a), where the unobserved model parameters are sampled from prior distributions. In the case of the DC-SBM, these are the priors for the partition $P(\boldsymbol{b})$, the number of edges between groups $P(\boldsymbol{e}|\boldsymbol{b})$, and the node degrees, $P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b})$. Finally, the network itself is sampled from its model, $P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})$. The inference procedure (b) consists on inverting the generative process given an observed network $\boldsymbol{A}$, corresponding to a posterior distribution $P(\boldsymbol{b}|\boldsymbol{A})$, which then can be summarized by a marginal probability that a node belongs to a given group (represented as pie charts on the nodes).

$$\Sigma(\boldsymbol{A},\boldsymbol{b}) = \underbrace{-\log_2 P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})}_{\mathscr{D}(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})} \underbrace{-\log_2 P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b}) - \log_2 P(\boldsymbol{e}|\boldsymbol{b}) - \log_2 P(\boldsymbol{b})}_{\mathscr{M}(\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})}. \quad (4)$$

The second set of terms $\mathscr{M}(\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})$ in the above equation quantifies the amount of information in bits necessary to encode the parameters of the model.[8] The first term $\mathscr{D}(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})$ determines how many bits are necessary to encode the network itself, once the model parameters are known. This means that if Bob wants to communicate to Alice the structure of a network $\boldsymbol{A}$, he first needs to

[8] If a value x occurs with probability $P(x)$, this means that in order to transmit it in a communication channel we need to answer at least $-\log_2 P(x)$ yes-or-no questions to decode its value exactly. Therefore we need to answer one yes-or-no question for a value with $P(x) = 1/2$, zero questions for $P(x) = 1$, and $\log_2 N$ questions for uniformly distributed values with $P(x) = 1/N$. This value is called "information content," and essentially measures the degree of "surprise" when encountering a value sampled from a distribution. See Ref. [20] for a thorough but accessible introduction to information theory and its relation to inference.