MODULARITY-BASED APPROACHES TO COMMUNITY DETECTION IN MULTILAYER NETWORKS WITH APPLICATIONS TOWARD PRECISION MEDICINE

William Harrington Weir

A dissertation presented to the faculty at University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Curriculum of Bioinformatics and Computational Biology in the School of Medicine.

Chapel Hill
2020

Approved by:

Peter Mucha

William Kim

Timothy Elston

Mark Niethammer

Jeremy Purvis

ABSTRACT

William Harrington Weir: Modularity-based approaches to community detection in multilayer networks with applications toward precision medicine
(Under the direction of Peter J. Mucha and William Y. Kim)

Networks have become an important tool for the analysis of complex systems across many different disciplines including computer science, biology, chemistry, social sciences, and importantly, cancer medicine. Networks in the real world typically exhibit many forms of higher order organization. The subfield of networks analysis known as community detection aims to provide tools for discovering and interpreting the global structure of a networks-based on the connectivity patterns of its edges. In this thesis, we provide an overview of the methods for community detection in networks with an emphasis on modularity-based approaches. We discuss several caveats and drawbacks of currently available methods. We also review the success that network analyses have had in interpreting large scale *omics* data in the context of cancer biology. In the second and third chapters, we present CHAMP and *multimodbp*, two useful community detection tools that seek to overcome several of the deficiencies in modularity-based community detection. In the final chapter, we develop a networks-based significance test for addressing an important question in the field of oncology: are mutations in DNA damage repair genes associated with elevated levels of tumor mutational burden. We apply the tools of network analysis to this question and showcase how this approach yields new insight into the structure of the problem, revealing what we call the *TMB Paradox*. We close by demonstrating the clinical utility of our findings in predicting patient response to novel immunotherapies.

To my loving and devoted wife, Elizabeth: thank you for your constant encouragement and support.

And to my amazing parents: thanks for always stoking my curiosity.

ACKNOWLEDGMENTS

It has truly taken the entirety of the village to make this work and the other components of my graduate and medical training possible. I am but a small node in a much larger network of love and support that has enabled the work contained here.

Thanks to Peter for being a wonderful mentor academically, professionally, and personally. I have always enjoyed our lively conversations: those about math and science; but also the epic, drawn out golfing stories (that frequently end with a tragic three or four putt). I have very much enjoyed the flexibility you have given me over the years to pursue a wide range of projects and have benefitted tremendously from your encouragement and insight. You have also done a splendid job of assembling fantastic people in the group, which has created an amazing milieu to work in. From Natalie, Sarai, Dane, and Peter Diao, all of whom helped shepherd me along during the early years of my training; to Andrew, Eun, and Zach who have helped sharpen my work in the last half of my time in the group. Thanks also to the undergraduates in the group for keeping us all honest, and in particular to Ryan and Scott for their tremendous work on the CHAMP project and many other contributions.

Thanks to Billy for being an awesome co-mentor and keeping me rooted in medicine. For all of the help soldiering through the F30 and for taking me under your wing for my longitudinal clinical rotation. I have learned a great deal about the genetics of bladder cancer, but also more broadly about how to conduct rigorous science – to think critically about biological questions in whatever context they arise. I have enjoyed becoming a part of the lab and appreciate all that you have invested in my training.

Thanks to my family. To Mom and Dad for their encouragement, and all they have invested in my education over the years. I owe far more to you than I could ever repay, and I am truly grateful for your love and sacrifice. To Forest and Annelise, for

being superb siblings and lifelong sounding boards and compatriots.

Finally, thanks to my wife, Elizabeth for the daily love, support, and encouragement. You are truly a wife of noble character and I am blessed to have you as my anchor.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS AND NOTATION

$\mathcal{G}(\mathcal{V}, \mathcal{E})$	Network containing set of nodes $\mathcal{V}$ with the set of edges $\mathcal{E}$
$\mathbf{A}$	Network adjacency matrix
$\mathbf{D}$	Diagonal degree matrix: $diag(k_i)$
$\mathbf{L}$	The graph Laplacian matrix: $\mathbf{D} - \mathbf{A}$
N	Number of nodes in a network
M	Number of edges in a network
k_i	The degree (or strength) of node i.
$\mathbf{c}$	Vector of node-to-community assignments
c_i	The community assignment of node i
p_k	The degree distribution of the network.
$\delta_{x,y}$	Indicator function on $x = y$. Assumes value of 1 if $x = y$ otherwise 0.
ER(N,M)	Erdős-Renyi network model with N nodes and M edges.
SBM	Stochastic Block Model
p_{in}	Within-community edge probability under planted partition SBM
p_{out}	Between-community edge probability under planted partition SBM
CHAMP	Convex Hull of Admissible Modularity Partitions (see Chapter 2)
AMI	Adjusted Mutual Information

CHAPTER 1: INTRODUCTION TO NETWORKS, COMMUNITY DETECTION, AND APPLICATIONS TO ONCOLOGY

We begin this thesis with an overview of the basic concepts underpinning the main ideas presented in this work, including networks and community detection with a focus on personalized medicine. Advances in networks science have arisen from a plethora of fields: sociology, physics, statistics, pure and applied mathematics, computer science, and many others. This convergence of problems arising in radically disparate domains with a common set of solutions is one of the most exciting parts of working in the field. However, it can make it difficult for the casual reader to find an appropriate entry point into the discipline without getting drawn into other minutia. In this introductory chapter, we hope to provide enough of a flavor of this fascinating field to keep the reader engaged, and to provide a foothold for accessing this deep and diverse body of work. For the interested reader who wishes to dive deeper, we recommend Newman's *Networks* [167] or Kolaczyk's *Statistical Analysis of Network Data* [117], both of which provide a hearty, self encapsulated introduction to networks. We begin by giving an account of what networks are and why they arise in so many different contexts. We discuss community detection in networks, detailing the different approaches and the philosophies underpinning them. Finally, we conclude the chapter by giving an overview of how network approaches, including (but not limited to) community detection have been used in the field of cancer genomics. We segue into the next chapter by giving an overview of the rest of the thesis and summarizing the main contributions of this work.

1.1 Networks

We conceptualize a network as an abstract collection of objects and relationships between those objects. We refer to the "objects" as nodes or vertices and the relationships as edges. In math terms a network, also called a graph, is a set of vertices and edges: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Edges

typically represent pairwise interactions, involving two nodes only[1]. We assign each node an arbitrary index, $i \in 1, \ldots, N$ and denote a specific edge by the pair of nodes it involves: $(i, j) \in \mathcal{E}$. Our definition of a network is quite simple and quite broad; hence there are many systems in the real world that can be conceptualized as networks:

$$\begin{aligned}\mathcal{G} &= (\{\text{genes}\}, \{\text{physically interact with each other}\}) \\ &= (\{\text{cities}\}, \{\text{connected via a road}\}) \\ &= (\{\text{researchers}\}, \{\text{have published a paper together}\}) \\ &\ldots\end{aligned}$$

All of these have a notion of individual entities (nodes) who come together through interactions (edges) to form a more complex system. We typically visualize a network by drawing circles for the nodes and lines connecting the circles for the edges as demonstrated in Figure 1.1, where we have assigned each node a coordinate in 2D space based on an algorithm to reveal clusters, although the layout is just one of infinitely many we could have chosen to show this network. It is merely a representation of the underlying structure and not itself intrinsic to that structure. Such visualizations can make us think of each network as living in 2D or 3D space; however, they are much higher dimensional objects themselves as each network can have up to $\binom{N}{2} = N(N-1)/2$ possible edges and thus can have the same number of degrees of freedom. In most networks of interest, there are many correlations between edges that can vary across the network, greatly reducing the complexity of the information content within the network. Networks can range in scale from tens to hundreds of millions of nodes[2], and the appropriate visualization will depend on the size of the network as well as the density of edges.

Another way networks are commonly represented, especially to perform computations, is

[1]Higher order networks (*e.g.* involving triplets or quartets) are called hypergraphs

[2]There is much interest in the dynamics of consumer preference on Amazon, which currently has over 600 million products listed. Facebook had over 2.4 billion users as of 2019.

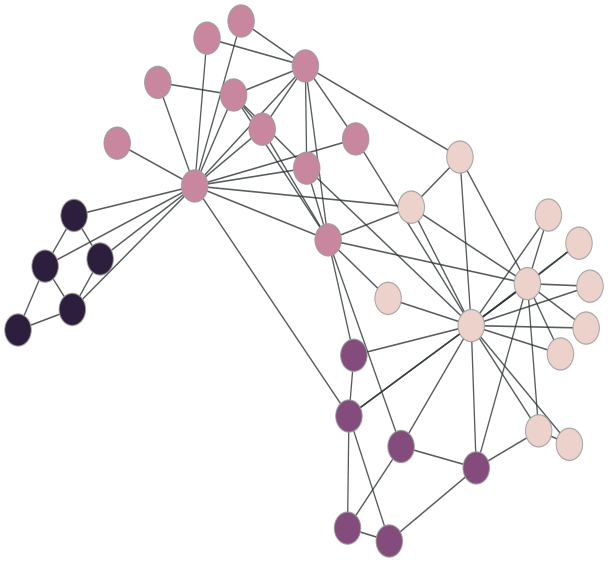

Figure 1.1: The famous Zachary karate club network. [255]. Each of the 34 nodes represents a member of the karate dojo Zachary was studying over several years, while edges represent whether or not Zachary observed consistent social interactions between the members outside of the official functions of the club. During the course of the the study, a disagreement between the leaders caused a split into two different clubs.

by an adjacency matrix, denoted $\mathbf{A}$ where:

$$A_{ij} = \begin{cases} 1 & (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} . \tag{1.1}$$

For a network with N nodes, $\mathbf{A}$ will be an $N \times N$ binary matrix[3] (which we also can denote $\mathbf{A} \in \{0,1\}^{N \times N}$). As any reindexing of the nodes will result in a permutation of the rows and columns of $\mathbf{A}$, there are many adjacency matrices that can correspond to the same underlying network. The adjacency matrix is computationally useful because it allows us to use both the theoretical and computational tools of linear algebra to address questions concerning networks.

Networks theory has found application in nearly all applied sciences. Much of the early work in the study of networks arose in the context of sociology, focusing on the empirical distribution of node-level metrics of observed networks as well as developing simple models of network formation to understand how those distributions could have arisen. For example, one of the major network quantities of interest is the distribution of the degrees of a network. The degree

[3]In the case where the edges of the network are weighted, we let elements of A take on arbitrary values (*i.e.* $\mathbf{A} \in \mathbb{R}^{+N \times N}$).

of a node, k, is the number of other nodes attached to it: $k_i = \sum_j A_{ij}$. The degree distribution of a network, denoted p_k, is the probability that a randomly chosen node has degree, k (i.e. the number of nodes with degree, k divided by the total number of nodes in the network). In most social networks, it was found that often there are a few high degree nodes, referred to as *hubs*, that play important roles in the network. This simple fact concerning the distribution of the degrees explains the surprising *small-world* effect described by Milgram [150]: despite the fact that there are almost 7 billion people in the world, any two randomly selected people are separated (with high likelihood) by a path of no less than six hops along edges within the networks. This same phenomenon accounts for the fact that even though there are tens of thousands of airports in the world, in most cases you can fly commercially between any two of them with only 1 or 2 layovers. These structural facts have implications for a number of other fields as well. Both sociologists as well as epidemiologists are interested in how phenomenon propagate themselves over a network: mathematical models of how news, fashion trends, or voting preferences spread on a network will have similar considerations to understanding an epidemic of a deadly disease.

Many domains in the biological sciences have also been impacted by developments in network theory, providing researchers with the ability to take a systems-level approach. From understanding the organization and functioning of the human brain at multiple structural levels; to the modeling of chromatin folding and disruption [169]; to assembling and interpreting the human genome [42]; All of these and many more systems can be tackled with the tools of networks. Recently, as we shall overview in Section 1.3, the field of oncology has been transformed by the exciting merger of large scale genomics data with the development of networks-based approaches.

There are many possible extensions of the basic network described above that allow one to capture more of the complexity of systems in the real world. One could allow for different strengths of interactions, placing weights on the edges (weighted networks), or allow for the interactions to have a directionality (directed network). We could also allow for multiple edges to exist between pairs of nodes (multigraph) or allow for different kinds of edges (multilayer network). For most of these, the operational representations of the network can be extended in a natural way to make use of the available tools. For instance, in the case of the weighted networks, allowing $\mathbf{A}$ to take on values other than 1 or 0 is usually sufficient to capture additional

complexity added by the weights. In other cases, a more careful handling is required; throughout the work we attempt to specify the kind of network we are using in each case when special considerations are required. We provide a greater explanation in the next few subsections on several needed concepts including the treatment of networks with multiple layers (multilayer), as these are the focus of much of this thesis.

1.1.1 The Graph Laplacian

Earlier, we saw how the adjacency matrix, $\mathbf{A}$, encodes the structural connectivity of the network in a relatively straightforward way. We introduce here the concept of the graph Laplacian, another matrix encoding the same structural information that arises in a number of contexts, especially in considering the dynamics of processes occurring on the graph. The graph Laplacian is a discrete version of the Laplacian operator ∇^2 that provides a notion of smoothness for a function over a graph.

If we have a function, $\mathbf{f}$, defined on each node by f_i, we can derive the graph Laplacian by considering how different the values of $\mathbf{f}$ are across neighboring nodes of the network:

$$||\mathbf{f}||_{\mathcal{G}}^2 = \sum_{ij}(f_i - f_j)^2 A_{ij} \; . \tag{1.2}$$

This equation then provides us with a simple notion of what it means for a function to be smooth across a network. If differences in the value of $\mathbf{f}$ for neighboring nodes is usually small, then $||\mathbf{f}||_{\mathcal{G}}^2$ will also be small. If we rewrite this equation as follows:

$$\begin{aligned} ||\mathbf{f}||_{\mathcal{G}}^2 &= \sum_{ij}(f_i - f_j)^2 A_{ij} \\ &= \sum_{ij} f_i^2 A_{ij} + f_j^2 A_{ij} - 2 f_i f_j A_{ij} \\ &= 2(\sum_{ij} f_i^2 D_{ij} - \sum_{ij} f_i f_j A_{ij}) \\ &= 2(\mathbf{f}^T \mathbf{D} \mathbf{f} - \mathbf{f}^T \mathbf{A} \mathbf{f}) \\ &= 2(\mathbf{f}^T \mathbf{L} \mathbf{f}) \end{aligned} \tag{1.3}$$

where we have introduced the degree diagonal matrix $\mathbf{D}$:

$$D_{ij} = \begin{cases} k_i & \text{If } i = j \\ 0 & \text{otherwise} \end{cases}, \tag{1.4}$$

and have defined the graph Laplacian, $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Thus the graph Laplacian can be thought of as a discrete difference operator on the graph. The graph Laplacian appears in many places in graph theory, including the identification of the minimum spanning tree and in regularization for machine learning approaches to graph structured data. The eigenvalues of the graph Laplacian also contain information about the community structure of the graph as can be seen in Figure 1.2.

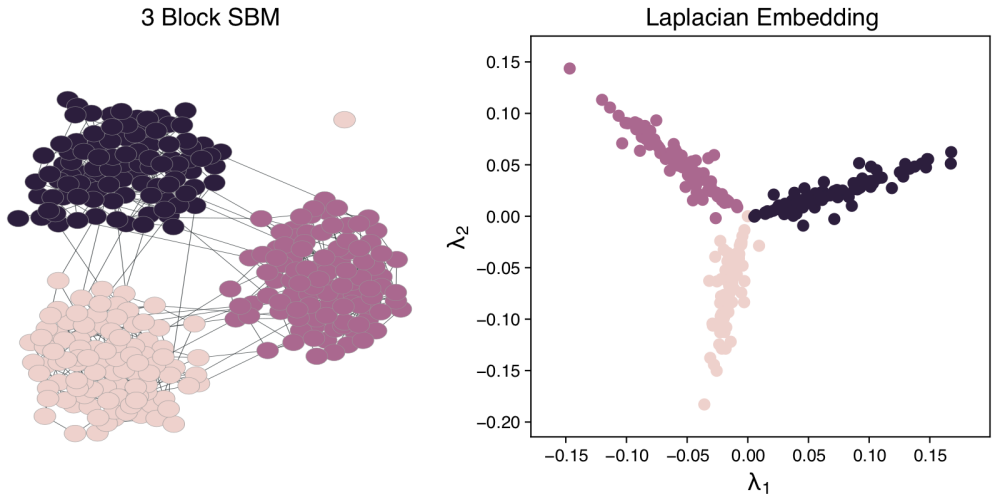

Figure 1.2: Embedding of 3 block SBM using the graph Laplacian. On the left we show the layout for a graph drawn from a 3 block non-degree corrected stochastic block model (planted partition model) with $p_{in} = .05$ and $p_{out} = .001$. To the right we plot the first two eigen values of the graph Laplacian of the network, with each node colored according to the ground truth community assignment. We see that the three communities of the model are well separated by the first two eigen values of the Laplacian.

1.1.2 Models of Networks

Much of network science has centered around the development of models of networks that capture aspects of real world networks. For instance, the Watts-Strogatz's "small world" model attempts to generate networks with relatively short paths between possible pairs of nodes [241]; the Barabási-Albert model of preferential attachment model produces networks with a

fat-tailed, power law degree distribution,[4] which is common in many real world networks [4] [5]. Here we briefly introduce the concept of a network model and describe the Erdős-Rényi model and the configuration model, both of which will be useful throughout this thesis.

In broad terms, a random network model defines a distribution over the space of possible networks one could observe. Often we use a model to look at how certain statistics behave over an ensemble of networks drawn randomly from our model. The Erdős-Rényi (ER) model is one of the simplest random network models: we allot equal probability weight to every network that has a given number of nodes and a specified number of edges. We denote this distribution of networks by : $ER(N, M)$. Instead of specifying a fixed number of edges, it is often more useful (and analytically tractable) to specify the independent and identically distributed (IID) probability each edge has of occurring, p, written as $ER(N, p)$. In this case, the expected number of edges is $M = p \times \binom{N}{2}$. For a large enough N the two models converge to each other. As each edge has an equal probability of being selected, the degree distribution for each node is simply the binomial distribution: $p_k \propto \binom{n-1}{k} p^k (1-p)^{n-1-k}$.

Another random graph model that will appear several times in this work is the configuration model. The configuration model is interesting in that the specified parameter is the degree distribution itself, p_k. The configuration model assigns equal probability to every network with the specified degree distribution (and zero to all others). Typically, we can use the configuration model to interrogate whether or not the properties we have observed in a real world distribution are independent from its degree distribution. Given a network, one way to construct the configuration model is to cut each edge in two, leaving a corresponding stub on each node. The configuration model is formed by placing equal weight on all of the possible ways of reconnecting the stubs, as illustrated in figure 1.3. We can calculate the probability of there being an edge between any two nodes, i and j, under the configuration model. Each stub has an equal probability of connecting with any other stub in the network. For a given stub on node i, there are k_j stubs on node j for it to connect to; therefore the probability of it being connected to a stub on

[4] A power law distribution follows the following functional form: $P(X = x) \propto x^{-\alpha}$. Typically $2 < \alpha < 3$

[5] There is some debate as to whether true power law distributions are actually commonly seen or whether other fat tailed distributions provide a better fit for most networks. See [29, 41] for assessing whether an observed distribution is fit by a power law, as well as [92] for additional discussion.

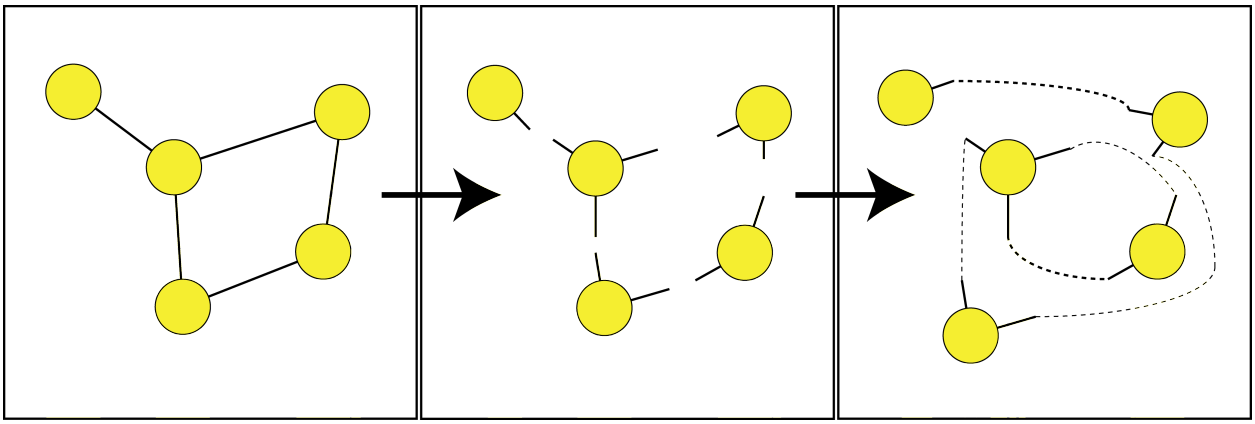

Figure 1.3: Conceptualization of the configuration model. Given a network (left panel), we can construct the configuration model by cutting every edge in the network into two (middle panel), and then randomly re-connecting the stubs (right panel). Depending on the context, networks with multi-edges or with self loops are not allowed. See [65] for a discussion of sampling from the configuration model and how to avoid having self loops or multi-edges.

node j is $\frac{k_j}{2m-1}$. And since there are k_i independent chances of a stub on node i connecting to a stub on node j, then the total probability of an edge between the two is given by $\frac{k_i k_j}{2m-1}$ ($\approx \frac{k_i k_j}{2m}$ for large enough m). There are many other network models that attempt to capture various aspects of observed networks. In section 1.2, we discuss a few more complicated models that attempt to capture higher order structure in the network by defining a notion of "communities".

1.1.3 Introduction to Multilayer Networks

In recent years, there has been great interest in extending the traditional network models to encompass more complexity that can better reflect a variety of systems. Specifically, in many systems, interactions between objects can vary over time, be of multiple types, or other complexities. For example, in a network of social actors, there could be multiple types of relationships that are captured: are friends, have worked on a project together, are family members, etc. We can incorporate this kind of structure by stratifying the nodes of the network across multiple "layers" or dimensions. A multilayer network allows us to represent multiple types of relationships between nodes in a unified way.[6] Throughout this work, we have adapted the multilayer notation and terminology from Ref. [114].

In the multilayer formulation, all edges representing a certain kind of relationship are

[6]The general multilayer network structure developed in [114] includes multidimensional networks, multiplex networks, networks of networks, hypernetworks, and others.

present within a layer. We show a depiction of such a network in Figure 1.4. In addition, we can allow for relationships for nodes in different layers encoded by the interlayer edges (shown by dashed lines in Figure 1.4). Typically, the interlayer edges encode the persistence of identity of

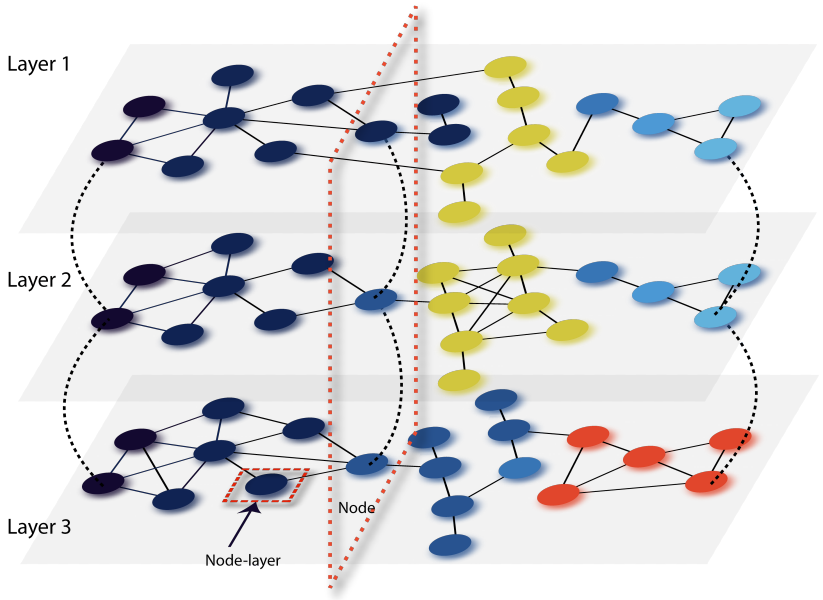

Figure 1.4: Depiction of a multilayer network with three different layers. Solid lines represent edges within each layer of the network while dashed lines represent edges across layers (interlayer edges). Node colors represent communities within the network. Note that if this were to represent a temporal network, we have deliberately only included a few of the interlayer edges for visibility. We have highlighted a node in the network that extends across all three layers as well as an example of a node-layer.

nodes across the various layers. For example, consider a multilayer where each layer represents observed email correspondence between the executives of a firm within a given time frame. Each executive would be represented in each layer by a specific node, with nodes in adjacent layers (representing neighboring time frames) connected by an interlayer edge. To distinguish between the single node or object of analysis across all layers, and its representation in any given layer, we refer to a particular node as it exists within a layer as a "node-layer".

There are several ways to encode a multilayer network. In the single-layer (also referred to as monoplex) network, we represented the edges between all pairs of nodes through a single, $A \in \mathbb{R}^{N \times N}$ matrix, with each element A_{ij} indicating the presence of an edge between nodes i and j (and the weight of the edge in the weighted case). In the multilayer case, we can represent each layer as A^l, with $l \in [1, L]$, indexing the layer, and the whole of the network as a single tensor

$\mathbf{M} \in \mathbb{R}^{N \times N \times L}$, where L is the number of layers in the multilayer network.[7] It is common to flatten $\mathbf{M}$ into an $\mathbf{A} \in \mathbb{R}^{NL \times NL}$, "supra-adjacency" matrix, where each node $k \in [1, N]$ is identified with node-layers indexed $\{i, i+N, \ldots, i+N(L-1)\}$. We use the notation $i \cong j$ to denote that two node-layers are identified with the same node. Typically we index the node-layers such that $i \cong i+N \cong ... \cong i+N*(L-1)$. We use the "supra-adjacency" notation in the rest of the paper and i, j, k to refer to node-layers in the network unless otherwise specified. We also use a single vector to keep track of which layer each node-layer resides in: $\vec{l} = [l_1, ..., l_{NL}]$ where $l_i \in [1, L]$ specifies the layer that node-layer i is in. We use $\mathcal{V}_{l_i}$ to denote the set of node-layers in layer l_i (*i.e.* nodes in the same layer as i including i itself). In addition to the edges within each of the layers, we also have to specify the overall topology across the layers. We encode the interlayer topology through the matrix $\mathbf{C} \in \mathbb{R}^{N*L \times N*L}$. We only allow elements of $\mathbf{C}$ to be non-zero if the corresponding node-layers are in different layers : $C_{ij} = 0$ if $l_i = l_j$. Typically each node-layer will be connected to a subset of the other node-layers corresponding to the same node in different layers, but C can represent any number of interlayer topologies.

There are many different kinds of multilayer topologies that could be encoded by the supra-adjacency format. Two of the most common types are the (discrete layer) temporal topology and the multiplex topology. In the temporal topology, we have an inherent ordering of the layers, usually representing observations of our system over time. Node-layers are typically connected by interlayer edges only for adjacent layers. Examples of temporal networks include observed functional connectivity patterns across regions in the brain at different times while performing a cognitive task [19] or observed patient referral patterns for a group of physicians across different years [228]. Multilayer networks can also encode a multiplex topology. Each layer represents a different kind of relationship; however, there is no inherent ordering on the layers. Identified node-layers are connected across all possible pairs of layers. Examples of multiplex multilayer networks could include an observed social network where edges are constructed separately from phone, text, and face-to-face contact information [94], or perhaps a transportation network of the London underground where different lines connecting various

[7] To make the collection of layer-adjacencies into a single tensor, we need each layer to have the same set of node-layers. In the case where a node is not present in a given layer (*i.e.* does not have a node-layer in that layer), we add in a place holder node-layer that remains unconnected to everything else in that layer.

stops are represented by different layers[196]. It is also possible to have a multilayer network with both multiplex and temporal topology. In these, we can imagine several "dimensions" of layers, with each layer being described by multiple aspect labels. For example we could observe a multiplexed social network over many time frames.

The earliest approaches to multilayer network analysis attempted to characterize them on the basis of well established single layer metrics. For example, to detect community structure, researchers attempted to apply single layer tools on each layer of the network individually, and then perform post-hoc alignment of the different layers [15]. There have also been attempts to map multilayer networks into a monoplex network and apply existing single layer methods. [25]. There are several different ways to collapse a multilayer network into a single layer network (for instance an using an OR operation on the existence of any edges or requiring an edge to be present in every layer [218]). While it is possible that the loss of information from collapsing does not significantly affect the downstream analysis, there are many cases when the multilayer structure is vital to understanding how a system behaves. For instance, in modeling the spread of a disease, the temporal arrangement of the edges can make a huge difference as to whether contagion occurs [131]. There has been a push to extend the definitions and metrics used to characterize single layer networks onto multilayer networks including various measures of centrality, random walks, clustering coefficients, as well as notions of communities. In addition a number of new metrics have arisen that have no single layer analog. These include metrics such as the *global overlap*-the number of common edges present by any two layers [26]. Likewise, the *degree of multiplexity* is the number of nodes with multiple edge types between them divided by the total number of adjacent nodes [151]. These measures are inherently multilayer and provide additional tools for researchers to characterize the statistical properties of these networks.

Several of the results that we present in this thesis concern the identification and characterization of the structure of multilayer networks through the development of community detection tools. In the next section we provide an introduction to the challenge of community detection in networks, beginning with single layer and then discussing several approaches for multilayer networks.

1.2 Community Detection in Networks

Network analyses are useful in the context of analyzing real world data because in most networks of interest, placement of the edges is not totally random. In fact, networks tend to exhibit numerous types of higher order structure as a result of the underlying processes that formed the network. Community detection is the development of models and algorithms to identify and interpret such higher order structure in networks. In essence it is the attempt to divide the network into "communities" in order to provide greater interpretability to the network data. In its most basic form, community detection falls under the category of "unsupervised learning": the attempt to discover broader patterns within a dataset without any known labels of the data.[8] Although there is no single definition of what defines network community structure, many approaches seek to identify groups of nodes which are more densely connected to each other than to the rest of the network [62, 167, 186]. The task of community detection is therefore to assign a label to each node such that this definition is optimally satisfied. Although this seems like a relatively straightforward notion, there are many ways in which it has been more precisely formulated, and there are many, many tools available to compute the communities of a network.

Some of the earliest work in the vein of community detection was on the problems of graph partitioning that arise in several contexts in computer science. In graph partitioning problems, the number and size of clusters is known in advance. For example the min-cut/max-flow problem depicted in Figure 1.5 seeks to identify a cut that minimizes the number of severed edges.

This min-cut problem is well defined and has an answer for any graph that is identifiable in polynomial time [72]. However, identification of the solution might not provide much insight into the structure of the network itself. The goal of community detection is much broader in that it seeks to identify partitions of the network *and* provide interpretability to those results. Thus community detection deals with a broad range of questions: How well grouped or distinguished are the communities? How certain am I that communities exist in the first place? Are certain nodes more classifiable than others? Can I find communities within communities (*i.e.* at different scales)?

[8] More recently, a few approaches have attempted to take known, node metadata into account, which could be consider a "semi-supervised" approach. For examples see [166],[213] and [56]

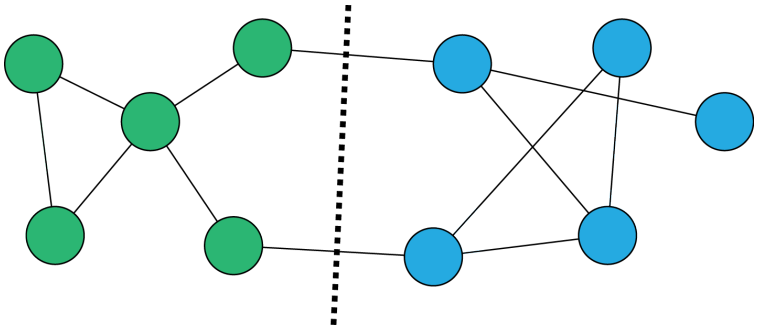

Figure 1.5: The min-cut, graph partitioning problem. The goal of the min-cut problem is to partition a graph into k-parts with a minimum sized cut. A cut of a graph is the removal of edges such that subset of nodes defined by a partition are disjoint. The size of the cut is the number of edges removed (or some function thereof). In this figure, there is a cut of size 2 that separates the green nodes from the blue nodes. Note that there are other cuts of this size in this particular example network.

There is a huge (and ever growing) body of research devoted to the identification of communities in networks. Here, we provide a brief overview of two widely used community detection methods: modularity and stochastic block models. We comment on how they relate to each other and on the particular benefits of each in addressing the challenges enumerated above. We provide an overview of other challenges that fall within the domain of community detection. We conclude this section with a discussion of how results of community detection are interpreted and applied to real world data.

1.2.1 Modularity

There are a number of community detection algorithms that define a score of how well a partition "divides" a network, then, try to find partitions which optimize this score. Broadly speaking, if we write down a score function, $f(\mathcal{G}, \mathbf{c}_\mathcal{G})$, with the network, $\mathcal{G}$, and $\mathbf{c}_\mathcal{G} = [c_1, \ldots, c_N]$, a partition of the network as inputs, then we can attempt to identify the partition that achieves the maximum score:

$$\mathbf{c}^* = \arg\max_{\mathbf{c}_\mathcal{G}} f(\mathcal{G}, \mathbf{c}_\mathcal{G}) \; .$$

One such quality function, developed by Newman and Girvan that has become a popular mainstay of community detection is called modularity [168]. Modularity has the form:

$$Q = \frac{1}{2m} \sum_{i,j} (\overbrace{A_{ij}}^{\text{internal edges}} - \underbrace{\gamma \frac{k_i k_j}{2m}}_{\text{expected edges}}) \delta(c_i, c_j) \; , \tag{1.5}$$

where $\delta(c_i, c_j)$ is one if nodes i and j belonging to the same community and zero otherwise, and γ is a user specified "resolution" parameter that was first added by Reichardt and Bornholdt [191] in showing the connection between modularity and the Potts spin glass model from statistical mechanics. We discuss the resolution parameter in more detail later on in this section. Originally, Newman and Girvan developed modularity as a metric to select an appropriate threshold in a different hierarchical clustering approach based on the removal of edges in order of decreasing betweenness. However, they and others soon developed algorithms to optimize modularity directly [27, 163].

To explain the form of modularity, in Equation 1.5, we have labeled the two terms that contribute to a partition's modularity score on a given network. The first term sums over the elements of the adjacency matrix that are internal to any of the communities. That is, how many of the observed edges fall between nodes that are in the same community? This aligns well with the idea that a good partition of the network should be dense within each community. However, if this were the only term in our function (or we set $\gamma = 0$), one could trivially maximize modularity by putting every node in a connected component into the same community. Thus, the second term in Equation 1.5 gives the number of internal edges one would expect to see under a random model of networks called the configuration model (see section 1.1.2 for details). We sum the probability of each edge occurring, $\frac{k_i k_j}{2m}$, over all possible pairs of nodes within each community. The modularity score tells us how the number of edges we have observed within each community compares to what we would expect if we sampled the network under the configuration model with the partition fixed. This is then normalized by the $\frac{1}{2m}$ prefactor such that the maximum value of modularity is 1. Since modularity was introduced, numerous other null models for different network topologies have been developed including those for directed networks [8, 132], bipartite networks [14], signed networks [74, 222], and multilayer networks [156], which is discussed in more detail in Section 1.2.1.

While modularity was developed based on the structural properties of a network (*i.e.* the degree distribution), Lambiotte et al. showed that it can also be derived by considering the dynamics of a random walk taking place on the network [122]. They define *stability* as the likelihood of a random walker remaining within the same community after the passage of a certain amount of time. They show that the formulation for stability, under the assumption of a

continuous-time random walk via a Poisson process, is approximated to first order by modularity, with the time scale of the random walk related to the resolution parameter. This provides an interesting connection between the strength of a community and the diffusion of a random walker on a network. A good partitioning of the network is one in which a random walker on the network is more likely to remain within a community than leave it over a small enough time scale. We note there is yet another connection between modularity and the field of statistical mechanics. As we shall further explore in Chapter 3, modularity also arises in the context of a spin-glass system of interacting particles.

Optimization of modularity

Although modularity was initially constructed as a quality metric for a different community detection method, it wasn't long before approaches were developed to optimize modularity directly. By writing Equation 1.5 in terms of matrix multiplication, Newman developed a spectral based approach to identifying a high modularity partition [164]. Another commonly used approach is known as the "Louvain" algorithm, after the corresponding institute in France where it was developed by Blondel *et al.* [27]. The Louvain method shown in Figure 1.6, begins by starting each node in its own community, then greedily attempting to swap nodes between communities to increase the modularity score. Once no community swaps are found, all nodes that are within the same community are then merged into a single, collapsed node in a condensed graph. Multi-edges and self loops are placed in the condensed graph so the value of modularity is maintained. Node-swaps between communities in the condensed graph correspond to the same changes in modularity for the original graph. Once no possible moves are found in a condensed graph, each node is assigned to the community corresponding to all the nodes it was merged with at the highest level condensed graph. The advantage of collapsing the graph after no moves can be found is that the algorithm can identify communities whose combination yields an overall increase in modularity, even though there is no series of individual moves where each move gives an increase. In other words it helps the algorithm overcome the problem of getting trapped in local optima.

The Louvain algorithm is fast and usually finds a partition with fairly high modularity. It has also been extended to multiple different modularity-like functions with very fast and efficient

code in the *Genlouvain* package [104]. More recently, Traag *et al.* released an improvement to the Louvain approach which they called the Leiden algorithm [225]. Their approach combats the tendency of Louvain to produce badly (or completely disconnected) communities by locally splitting nodes within each community of the partition before aggregating nodes into a condensed graph. We present results using the Leiden algorithm on several datasets in Chapter 2.

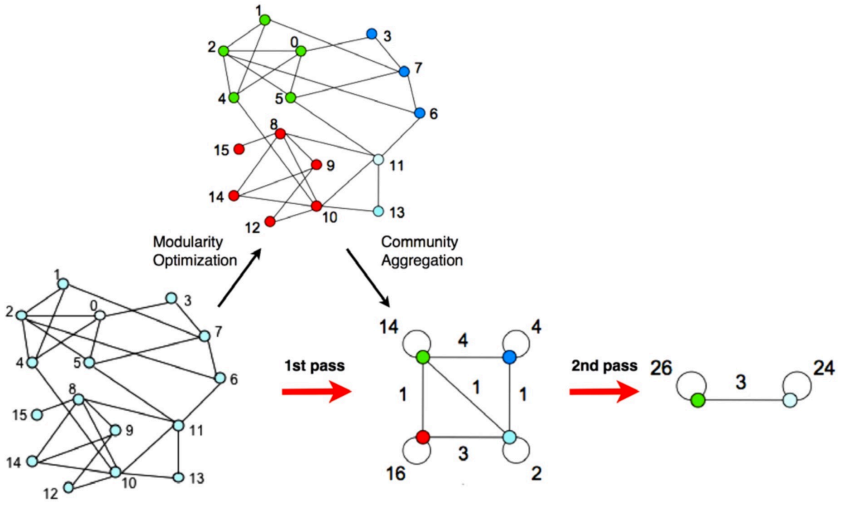

Figure 1.6: The *Louvain* algorithm developed by Blondel *et al.* [27]. Each node starts in its own community, then each node is moved (in random order) into the community that gives the largest increase in modularity. Once no more moves can be identified the graph is condensed (with self loops and multiedges) on the basis of the partition and the algorithm is repeated. This figure was taken directly from [27].

However, as all of these are greedy algorithms, they are subject to entrapment in local optima. In fact all known algorithms with a feasible run time (on anything beyond small networks) will only produce a locally optimal partition, because the problem of optimizing modularity has been shown to be NP-hard [28][9]. This is not surprising given the number of possible partitions of a network grows exponentially with the size of the network. To counter this, one typically runs an algorithm many times stochastically, randomly choosing the order of moves, with the goal of exploring as much of the space as possible. However, incorporating information from the many stochastic runs is made more challenging by the introduction of one or more free parameters into the formulation for modularity. An approach to combining the results from many different algorithmic runs is one of the contributions of this thesis developed in Chapter 2.

[9]There is a linear programming approach to modularity that find a guaranteed optimum, however this runs in exponential time. See Ref. [28] for details.

The resolution limit of detection

One of the major issues with modularity, initially pointed out by [63], is that under certain conditions it will fail to detect smaller clusters within a larger network. They show that the condition for a community contributing positively to the modularity score depends on the total number of edges within the network rather than a local comparison of relative edge densities inside and outside of the community. If the number of edges within two distinct modules connected to each other weakly is less than some fraction of the total edges within the network, modularity will be optimized by merging the communities. This is not merely a failure of algorithms attempting to maximize modularity; it is a fault in the metric itself and indeed any such metric that uses a global null model. The resolution limit occurs because if the network grows larger while keeping the size of individual modules fixed, the number of expected edges between any two modules vanishes. Thus even a single link between modules can be enough to increase modularity by merging them if the surrounding network is large enough.

Fortunato and Barthélemy provide several examples of real world networks where they were able to identify additional sub-communities by re-applying modularity optimization to the subgraphs induced by the individual communities identified by the first pass of modularity [63]. Kumpula *et al.* applied a similar approach to identify the conditions when modularity-based methods would correctly identify subcommunities with the incorporation of the resolution parameter, γ [119]. The resolution parameter was originally introduced by Reichardt and Bornholdt [191] when they showed that modularity, as originally formulated, was a specific version of a more general spin glass problem. We discuss the connection between modularity and the spin glass problem in greater detail in Chapter 3. Although the parameter was not specifically meant to address the resolution limit, they show how the communities identified under various values of γ yield a hierarchy of communities at different scales. Kumpula *et al.* show that by identifying communities across a range of the resolution parameter, one can overcome the resolution limit. They caution, though, that one should question the validity of the hierarchical structure observed by this approach due to the tendency of modularity to artificial merge communities. Several other recursive approaches have been proposed that attempt to refine identified partitions as different scales using spectral methods and local graph searches

[194, 200].

Arenas *et al.* contend that the resolution limit arises from the existence of community structure at multiple scales in most networks rather than the intrinsic defect in modularity [7]. Organization at multiple scales is a natural feature of complex systems, and the ability to tune modularity to select for communities at different scales can be seen as a feature rather than a bug. They develop an approach that adds a self loop to each node and then varies the weight of the self loop while using the original version of modularity ($\gamma = 1$) to identify communities at different scales [7]. Traag *et al.* suggest that for a community detection approach to be truly "resolution-limit free", that method should not split the induced subgraph of the individual communities identified for some level of the resolution parameter [223]. To create an approach that is truly resolution free, the objective function cannot rely on a null model that is dependent on the global properties of the graph (as $p_{ij} = \frac{k_i k_j}{2m}$ clearly does). Despite issues posed by the resolution limit of detection, modularity-based approaches remain a popular and widely used approach to community detection, largely in part due to the existence of easy to use, fast implementations. While there is no *a priori* correct value of γ for arbitrary networks, there are clearly networks for which a particular choice or range of γ's produces well clustered results. Exploring and characterizing how the identified structure changes as γ is varied remains an important and practical question in the field. Chapter 2 in this thesis contributes the CHAMP method to further address this question.

There are a number of other deficiencies to modularity-based approaches that have been pointed out. Good *et al.* show that the modularity function exhibits a high degree of degeneracy on most networks [76]. That is to say for most networks there exists a large number of structurally *dissimilar*, high modularity partitions that make identifying a unique, optimal solution quite challenging. Nevertheless, modularity-based community detection continues to be widely employed, and as such, we offer several approaches in 2 and 3 of this thesis to help surmount these well known deficiencies.

Multilayer Modularity

Another reason why modularity remains popular is that it is one of only a few principled approaches for the detection of communities within multilayered networks. Additionally, there

are very fast and efficient implementations of multilayer modularity that is well maintained [101]. Mucha *et al.* developed multilayer modularity by extending the Laplacian dynamics approach by Lambiotte *et al.* in [122] to include the conditional probabilities of traveling along various edge types [156]. The equation they derived can be written in the supra-adjacency form (see Section 1.1.3 for notation):

$$Q(\gamma,\omega) = \sum_{i,j} \left(A_{ij} - \gamma P_{ij} + \omega C_{ij}\right) \delta(c_i, c_j) \tag{1.6}$$

where i and j each index the distinct node-layers, A_{ij} is the supra-adjacency encoding the intralayer edges, P_{ij} describes the expected number of intralayer edges based on the selected random model(s), C_{ij} encodes the interlayer connections, and ω is the inter-coupling parameter that sets the strength of the interlayer edges relative to the intra-layer connections. Written in this form, one can see that like its single layer counter part, multilayer modularity is a trade off between edges (both interlayer and intralayer) within communities and those expected under the null model. One can of course choose a variety of null models, even employing different models across the various layers of the same network. In the original formulation derived by Mucha *et al.*, the Laplacian dynamics on temporal networks give rise to a within layer restricted configuration model:

$$P_{ij} = \begin{cases} \frac{d_i d_j}{2m_{l_i}} & l_i = l_j \\ 0 & l_i \neq l_j \end{cases} \tag{1.7}$$

where l_i is the layer containing node-layer i, (*i.e.* $i \in \mathcal{V}_{l_i}$), $d_i = \sum_j A_{ij}$, and $m_{l_i} = \sum_{i,j \in \mathcal{V}_{l_i}} A_{ij}$ is the total weight of edges in layer l_i. We see that the null-model in Equation 1.7 is very similar to that introduced in the original form of modularity in Equation 1.5; the major difference being that the multilayer null model has a layer specific denominator ($2m_l$, the total edge weight in layer, l), and is zero for all pairs of node-layers that are not in the same layer. The multilayer nature of these networks allows for a much larger number of possible null models than in the single-layer setting and the choice of null model can greatly influence the detected community structure. See supplement of [156] for null models for bipartite, signed, and directed networks as well as [17] for examples of other possible null models in a multilayer context.

Most multilayer approaches to community detection attempt to identify community structure that persists throughout the various layers (*i.e.* sets of node-layers in different layers that in the same community). The multilayer modularity approach encourages identified node-layers to remain in the same community across the layers of the network through the interlayer coupling term, ω. This represents another free parameter that must be tuned, which is one of the major benefits of using the CHAMP approach detailed in Chapter 2. Other approaches to selecting ω have been tailored specifically for the system being studied (for example see[176]). Multilayer modularity has been employed successfully in understanding the organization of a number of systems including long-term changes in voting patterns in the US senate [155, 156], the evolution of functional modules in the human brain during learning [16, 18], as well as the prediction of conflict between countries over time based on trading networks [43]. As it continues to be widely used for the detection of communities across a wide range of multilayer networks, we hope the novel extensions we provide in Chapters 2 and 3 of this thesis will be found useful by the general community.

Other score based approaches to community detection

One of the main other score-based community detection method is the *Infomap* method developed by Rosvall and Bergstrom [198]. Like the Lambiotte treatement of modularity previously discussed, *Infomap* approaches community detection from the perspective of the dynamics of a random walk on the network. Rather than compare the stability of a random walk with respect to the community assignments, *Infomap* attempts to minimize the information cost of encoding such a random walk. Each community is given its own coding system for its nodes and the brevity of the overall encoding trades off between the complexity induced by having multiple code books, and the decreased size of codes within each community. If a random walk is likely to stay within a given community, having a shorter code is worth the cost of designating switches between each community's set of code assignments. The *Infomap* approach makes explicit that a well structure network (*i.e.* with strong communities) allows for a compressed encoding of the network. Infomap's emphasis on efficiently encoding random walks within a network represents a shift from focusing on the "topological properties of its links" (in the modularity approach) to "patterns of flow that its structure induces" [198]. Often times the

communities identified between these two approaches are similar. However, in [198] they illustrate a couple of cases where the optimal solutions diverge wildly between Infomap and modularity. The authors suggest that Infomap is more appropriate when the links of the network "represent patterns of movement among nodes" [198]. Infomap is also one of the few community detection approaches that has been adapted for multilayer networks [48], as well as several other extensions (e.g. [56, 199]).

1.2.2 Statistical Models of Network Communities

While the previously discussed approaches to community detection were based on finding partitions that optimize some *a priori* notion of what a good community is, there is also a class of methods that employ statistical models of communities in network. Like the networks models discussed in Section 1.1.2, these approaches define a probability distribution over the space of possible networks, with parameters that reflect the strength and structure of the identified communities. In this sense, they are usually generative models, meaning that once one has inferred the parameters of the model from the observed data, one can sample from the model to create new networks from the distribution. Generally, one attempts to infer the parameters of the model based on an observed network via a maximum likelihood estimation approach.

There are a number of benefits to fitting a generative model of community structure to one's data. Fitting a generative model to the observed data allows one to test whether other statistical properties of the observed data (degree distribution, clustering coefficient, other centrality measures, etc.) are significantly different than networks from the generative model. The generative model can also be useful in assessing the performance of an algorithm over many different random realizations of a model fit to the real data of interest. The inferred parameters can also allow for comparing different size networks or for a way to summarize the networks in a compressed from. For a more detailed discussion of the advantages of generative models of network structure, as well as another example of a flexible, generative model that can capture communities, see [135].

The most basic model of community structure, first developed by [91], is known as the Stochastic Block Model (SBM). In the vanilla SBM, we divide all of the nodes into K classes, with each node's class given by its corresponding element in vector $\mathbf{c}^* = [c_1, \ldots c_N]$. The probability of

any two nodes being connected by an edge is conditional only on the classes of the two nodes. We encode the probabilities of connection between each possible pair of the K communities with the $K \times K$ matrix, $\Theta \in \mathbb{R}^{K \times K}$ as demonstrated by the example in Figure 1.7.A.

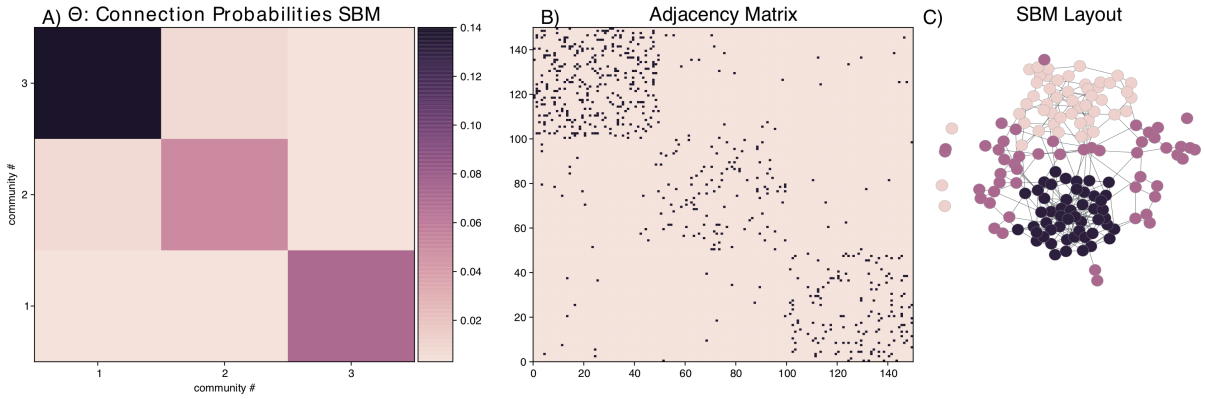

Figure 1.7: Example of a 3 community stochastic block model (without degree correction). A) The matrix, Θ giving the probabilities of connections within and between the various communities. B) Adjacency matrix for a network sampled from this model and C) layout of the corresponding network, colored by the block each node is assigned to.

The probability of an edge between nodes i and node j is given by Θ_{c_i,c_j}, the element of Θ corresponding to the communities of the two nodes. Thus each node is statistically indistinguishable from any other node within the same block. The probability of an observed adjacency, given our model parameters is:

$$P(A|\Theta,\eta) = \prod_{i<j} \left(\Theta_{c_i,c_j}^{A_{ij}} (1 - \Theta_{c_i,c_j})^{1-A_{ij}} \right) . \tag{1.8}$$

We see from Equation 1.8 that the probability of observing an adjacency is given by the product of the independent probabilities for each edge. This implies that the edges, restricted to each individual block, are distributed according to an Erdős-Rényi null model (see Section 1.1.2). This produces generally unrealistic results when applied to real world networks (that have heavy tailed degree distributions) and tends to disproportionally place high degree nodes together in the same community. We discuss towards the end of this section a correction to the model that accounts for more realistic degree distributions.

The goal of fitting the SBM to an observed network is to infer the parameters, Θ and $\mathbf{c}$ that

maximizes the log of our likelihood, $\log P(A|\Theta, \mathbf{c})$:

$$\hat{\Theta}, \hat{\mathbf{c}} = \arg\max_{\theta,\eta} \left(\log P(A|\Theta, \mathbf{c})\right) \ . \tag{1.9}$$

In other words, we want the maximum likelihood estimate (MLE) of our model. While there is no closed form solution to this problem, we can use variational techniques such as the Expectation-Maximization algorithm to identify approximate solutions. In essence, these methods work by treating $\mathbf{c}$ as our "latent variable"; that is we let $\mathbf{c}$ becomes a probability distribution for each node over all the classes,ψ. If we fix ψ, we can analytically find the values for Θ that maximize our likelihood. On the other hand, for a fixed Θ, there are mean-field approaches to identify ψ that approaches the lower bound of our likelihood [46], as well as a belief propagation approach [261].

To account for the heterogenous degree distributions observed in real networks, Karrer and Newman introduced the degree corrected stochastic block model (dcSBM) [108]. They modify the probability for an edge between two nodes by incorporating an extra parameter for each node that dictates that nodes inherent likelihood of participating in an edge. To make their approach more tractable, rather than treat each edge as a Bernoulli variable, they allow for multi-edges under their model, with the number of edges being drawn from a Poisson distribution.[10] In the limit where the size of the network continues to increase while the edge density remains sparse, these two formulations converge in expectation as the contribution from multi-edges and self loops becomes negligible.

The likelihood under their model becomes:

$$\begin{aligned} P(A|\Theta, \eta) = &\prod_{i<j} \frac{(\theta_i \theta_j \Theta_{c_i,c_j})^{A_{ij}}}{A_{ij}!} \exp(-\theta_i \theta_j \Theta_{c_i,c_j}) \times \\ &\prod_i \frac{(\frac{1}{2}\theta_i^2 \Theta_{c_i,c_i})^{A_{ij}/2}}{(A_{ij}/2)!} \exp(-\frac{1}{2}\theta_i^2 \Theta_{c_i,c_i}) \ . \end{aligned} \tag{1.10}$$

where there is now an additional parameter for each node, θ_i that can be interpreted as the

[10] Recall that the Poisson distribution is a discrete distribution over counts with the following probability mass function: $P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$.

probability that an edge connected to community c_i is incident to node i. The maximum likelihood estimate of θ_i ends up being proportional to the node's degree, k_i; hence the model appropriately accounts for increased connections between high degree nodes. This model can also be approximately solved using a variational approach.

There are numerous other extensions of the stochastic block model that have been proposed. For example, Peixoto introduced a hierarchical, full Bayesian approach that incorporates priors and hyperpriors for the number and size of communities, the degree distributions, and the hierarchical nesting of communities at different scales [184]. He proposes an approach based on a Markov Chain Monte Carlo (MCMC) importance sampling of the posterior distribution of community memberships given the observed network, which he implemented in the GRAPH_TOOL python package [181]. Several variants of the stochastic block model have also been developed for multilayer networks including fitting individual SBM to each layer with fixed node classes throughout the various layers [83], allowing for classes switching within subsets of different layers for multiplex networks [212], as well as a belief propagation approach for a temporal version known as the Dynamic Stochastic Block Model [70] (which we describe in more detail in Section 3.3.2). There have also been several tools that attempt to determine whether or not an observed single layer network is derived from an aggregated multilayer stochastic block model structure including [229] and [183].

One interesting feature of the stochastic block model is that it can fit to a number of different types of community architecture. For instance, it could be the case that edges are less likely to occur within a block than outside of a block (*i.e.* the off diagonal elements of Θ are larger than the diagonal elements), which represents a kind of disassortative or bipartite like community structure. This could represent a network such as a food web where similar nodes are not those that interact with each other but have similar neighbors (*e.g.* predators that share the same prey)[6]. One could also have a core-periphery structure where one, *core* block is highly connected internally and to other blocks, but everything else is weekly connected [102]. The stochastic block model has the flexibility to represent multiple notions of community based on the requirements of the specific domain.

Equivalence of the stochastic block model with modularity

There is an equivalence between maximum likelihood estimation of the degree corrected stochastic block model under certain conditions and the optimization of modularity [165]. With some basic algebra, and dropping the constant terms, we can write the log-likelihood for the dcSBM as follows :

$$logP(A|\Theta,\eta) \propto \frac{1}{2}\sum_{ij}\left(A_{ij}log\Theta_{c_i,c_j} - \frac{k_i k_j}{2m}\Theta_{c_i,c_j}\right) \tag{1.11}$$

where we have substituted in the degrees k_i for what was originally θ_i based on the maximum likelihood estimate for a fixed η. The equivalence between the SBM and modularity can be seen when we restrict ourselves to a particular form of the SBM known as the planted partition model. In the planted partition model we can have any number of communities, however the elements of Θ are restricted to one of two values: $\Theta_{r,s} \in \{p_{in}, p_{out}\}$. We let p_{in} denote the probability of an edge connection within each of the communities (*i.e.* the diagonal elements of Θ) and p_{out} represent the probability of an edge connection between nodes of different communities (the off-diagonal elements of Θ). Therefore in the planted partition model, all communities are assumed to have the same in-group and between group connection rates. We can write out the elements of Θ as:

$$\Theta_{c_i,c_j} = (p_{in} - p_{out})\delta_{c_i,c_j} + p_{out} \tag{1.12}$$

$$\log\Theta_{c_i,c_j} = (\log p_{in} - \log p_{out})\delta_{c_i,c_j} + \log p_{out}\ . \tag{1.13}$$

Plugging this into Equation 1.11, we obtain

$$\begin{aligned} logP(A|\Theta,\eta) &= \frac{1}{2}\sum_{ij}\left(A_{ij}(\log p_{in} - \log p_{out})\delta_{c_i,c_j} + \log p_{out} - \frac{k_i k_j}{2m}(p_{in} - p_{out})\delta_{c_i,c_j} + p_{out}\right) \\ &= \frac{1}{2}\sum_{ij}\left(A_{ij}\log\frac{p_{in}}{p_{out}}\delta_{c_i,c_j} + \log p_{out} - \frac{k_i k_j}{2m}((p_{in} - p_{out})\delta_{c_i,c_j} + p_{out})\right) \\ &= \frac{1}{2}\log\frac{p_{in}}{p_{out}}\sum_{ij}\left(A_{ij} - \frac{p_{in} - p_{out}}{\log\frac{p_{in}}{p_{out}}}\frac{k_i k_j}{2m}\right)\delta_{c_i,c_j} + m(\log p_{out} + p_{out})\ . \end{aligned} \tag{1.14}$$

The community assignments that maximize Equation 1.14 will also maximize the modularity function for the resolution parameter

$$\gamma = \frac{p_{in} - p_{out}}{\log \frac{p_{in}}{p_{out}}} \,. \tag{1.15}$$

Thus we see that if for fixed set of parameters of our planted partition model, p_{in}, p_{out}, K, finding the community assignments that maximize the likelihood of our SBM is equivalent to optimizing modularity for a particular choice of γ. Newman suggests that this equivalence can be used in an iterative manner to identify the correct value of γ for a network as follows: Select an arbitrary value for γ and run modularity optimization; One can then calculate the values of p_{in}, p_{out} given the identified communities, as well as the value of γ implied by Equation 1.15; Run modularity optimization at the updated value of γ and repeat the process until the identified value of γ converges. This process is not guaranteed to converge, especially in the equivalent multilayer process which we discuss next. It is also possible that there could be multiple basins of attraction for networks with community structure at multiple scales. Note that this procedure requires the number of communities to remain fixed through each iteration in order for the equivalence to hold, which is problematic as usually the appropriate number of communities is not known *a priori* (and, moreover, most modularity optimization codes are not constrained to a specified number of communities).

There is also an equivalence between certain versions of the multilayer stochastic block model and multilayer modularity as shown by Pamfil *et al.* [174]. In the case where the network has temporal topology and each node-layer in layer l has probability, p of copying its corresponding node-layer's community assignment in layer[11], $l-1$, one can show that the model implies a similar equivalence for the interlayer coupling parameter, ω:

$$\omega = \frac{\ln(1 + \frac{p}{1-p}K)}{T\langle \ln p^l_{in} - \ln p^l_{out} \rangle_l} \tag{1.16}$$

where T is the total number of layers, and $\langle , \rangle_t$ denotes the average over the layers. Pamfil *et al.* explore the dynamics of an iterative procedure in the (γ, ω) domain analogous to that suggested

[11] This is the dynamic stochastic block model (DSBM) detailed in Section 3.3.2 with the exception that it is degree corrected within each layer

by Newman [165], showing how it can be used to identify an optimal $(\hat{\gamma}, \hat{\omega})$. They also show an equivalence in the case where there is non-uniform coupling between layers (*i.e* p is allowed to vary across layers) and they identify an approximate formula for ω in the case when the topology of the network is multiplex rather than temporal. The connection between modularity-based approaches and the variants of the stochastic block model sheds light on why modularity approaches have been successful as well as on some of the underlying assumption it makes. This equivalence also leads to a more principled way to selecting appropriate values for (γ, ω).

1.2.3 Real world applications of community detection

While community detection has been proven to provide valuable insight into network problems across a wide range of scientific domains, it is, in general, not an ultimate end in and of itself. Rather, community detection can be thought of a useful approach towards generating and, in some cases, testing hypotheses. For example, Weng *et al.* examine how the community structure within the network of Twitter users influences the spread of viral memes [247]. They showed that the extent to which memes were "trapped" by the community structure influenced whether it would take off in the broader community, and they developed a metric to compare the observed phenomenon with several models of social contagion. Importantly, they were able to show that incorporating knowledge of the network's community structure greatly enhances the ability to predict which memes would go viral. This provides a useful example of how the results of community detection can provide interpretable features for use in other supervised analyses further downstream.

While we tend to consider community detection among the "unsupervised" analyses that extract patterns from the data, the best approach is to develop questions that can be interrogated from the data before application of the community detection methods. While the details will be domain specific, examples of the kinds of downstream questions one can ask using community detection include:

Are there any node attributes that tend to localize across the communities?

Are the community labels themselves useful in predicting a downstream task?

Does the structure I observe in my data suggest a process by which the network arose?

Knowledge of the particulars of a given dataset as well as of the downstream task will highly influence the definition of "community" that a researcher will look to use. We have discussed several conceptions of communities that are assumed by the different community detection methods detailed in the prior sections. There are also many other approaches to clustering data that don't assume the existence of a network at all. The details of the questions researchers aim to answer, as well as the known laws governing the nature of the network itself should dictate the types of community detection tools that are applied [149]. For a case study in several successful applications of community detection to real world problems, see [207]. In this section we discuss the approaches that have been used to validate and interpret various community detection algorithms as well as some caveats for these approaches.

Assessing performance of community detection algorithms

Like other forms of unsupervised learning, assessing the performance of community detection algorithms yields its own particular set of challenges. In general, community detection tools have been assessed on the basis of three criteria: 1) the ability to resolve known community structure in networks generated from synthetic model 2) alignment of identified communities with metadata for real world networks for which there is justification of it contributing to community structure and 3) demonstrated usefulness of detected community structure in the downstream analysis of real world data. We provide an overview of each of these approaches below. Understanding the first two approaches requires an understanding of how we compare identified communities.

Metrics for comparing partitions

In this section we discuss several metrics that are available for comparing partitions. In general, we need a function that quantifies how similar two partitions, $\mathbf{c}$ and $\mathbf{c}^*$ are to each other. Ideally, this function should be high when a similar set of nodes is assigned to each group between the two partitions and low when there is little concordance. It is also important for such metric to be permutation invariant so that similarity between the two partitions is not dependent on the arbitrary ordering of labels assigned to identified groups within each partition. It is for this reason that simply counting the elements of $\mathbf{c}$ and $\mathbf{c}^*$ that agree (also known as the accuracy) is

not a very useful metric.

ADJUSTED MUTUAL INFORMATION Throughout the paper, we use an information theoretic metric to assess how much the partitions are varying across the dominant domains. Mutual Information (MI) quantifies the decrease in entropy for one random variable that comes from knowing the value of a second random variable. Here the two random variables are discrete, community labels on the nodes. If two partitions are highly similar, knowledge of the community label of node i in the first partition drastically reduces the uncertainty of the label of node i in the second partition. We let s and t index the unique labels for $\mathbf{c}$ and $\mathbf{c}^*$ respectively, and let $p_s = \sum_i \delta_{\mathbf{c_i},s}/|\mathbf{c}|$ be the proportion of the nodes of $\mathbf{c}$ that belong to community s. Likewise, let p_t^* be the proportion of nodes of $\mathbf{c}^*$ that belong to community t. Finally, let $p_{s,t} = \frac{\sum_i \delta_{c_i,s}\delta_{c_i^*,t}}{|\mathbf{c}||\mathbf{c}^*|}$, be the joint proportion for each pair of community labels s and t. Then we can define the mutual information for the random variable $\mathbf{c}$ and $\mathbf{c}^*$ as :

$$MI(\mathbf{c},\mathbf{c}^*) = \sum_{s,t} p_{s,t} \log \frac{p_{s,t}}{p_s p_t^*} \tag{1.17}$$

This score can be normalized by the average entropy of the two variables individually to yield a value between 0 and 1, known as the normalized mutual information [68]. In this thesis, we use a more stringent, normalized version of the metric introduced by Vinh *et al.* [235] called Adjusted Mutual Information (AMI),

$$AMI(\mathbf{c},\mathbf{c}^*) = \frac{MI(\mathbf{c},\mathbf{c}^*) - E(MI(\mathbf{c},\mathbf{c}^*))}{\max\left(H(\mathbf{c}),H(\mathbf{c}^*)\right) - E(MI(\mathbf{c},\mathbf{c}^*))}\,, \tag{1.18}$$

where $H(\mathbf{c}) = -\sum_s p_s \log p_s$ is the entropy of the random variable $\mathbf{c}$. The expected value, $E(MI(\mathbf{c},\mathbf{c}^*))$, is calculated over random partitions sampled from a hypergeometric null distribution (see [235] for details). The AMI between two partitions equals 1 to indicate perfect concordance, with the value 0 representing alignment no better than random. AMI tends to be a more conservative measure of alignment because it is less biased than normalized mutual information or the Rand Index towards partitions with a larger number of communities [234]. With our multilayer examples, we have applied AMI in two ways to assess different aspects of the alignment of the discovered partitions. We calculated the AMI between all node-layers and the

entire partition, each taken as a single vector. We refer to this simply as AMI, and it is the main metric we use throughout the paper. We also use a layer-averaged version of AMI where we compute the AMI of the partition induced within each layer with the ground truth, weighted by the size of the layer:

$$\langle \text{AMI} \rangle = \sum_{l=1}^{L} \text{AMI}(\mathbf{c}_l, \mathbf{c}_l^*) \frac{|l|}{N} \tag{1.19}$$

where $\mathbf{c}_l$ is the identified partitioning of the node-layers restricted to layer l, $\mathbf{c}_l^*$ is the ground-truth communities of the node-layers within layer l, $|l|$ is the number of node-layers in layer l, and N is the total number of node-layers. $\langle \text{AMI} \rangle$ is useful in assessing how well multilayer community detection methods are leveraging information across layers to detect communities within each layer (see discussion in [21] for advantages of a layer averaged metric).

VARIATION OF INFORMATION Variation of information (VI) is an information theoretic measure that assess the degree of information lost switching from from one variable to another [146]. Unlike AMI, VI is actually a measure of dissimilarity between two partitions and is a true metric. This is useful in that one can use VI to compare sets of partitions and it will obey the triangle inequality. If we index the unique values of $\mathbf{c}$ by s as above (*i.e.* the possible community assignments) and likewise for $\mathbf{c}^*$ by t, and use the same definitions for p_s, p_t^*, p_{st} as above, then we can compute the variation of information as follows:

$$VI(\mathbf{c}, \mathbf{c}^*) = \sum_{st} -p_{st} \left[\log \frac{p_{st}}{p_s} + \log \frac{p_{st}}{p_t^*} \right] . \tag{1.20}$$

In the event that the two community assignments overlap perfectly, then $p_{st} = p_s = p_t^*$, and both terms inside each element of the sum will be zero. Variation of information can also be written in terms of the mutual information between the two clusters:

$$VI(\mathbf{c}, \mathbf{c}^*) = H(\mathbf{c}) + H(\mathbf{c}^*) - 2MI(\mathbf{c}, \mathbf{c}^*) = H(\mathbf{c}|\mathbf{c}^*) + H(\mathbf{c}^*|\mathbf{c}) \tag{1.21}$$

One issue with Variation of Information is that its maximum value is subject to the number of communities in a partition and thus the authors suggest normalizing it by either $\frac{1}{\log N}$

or $\frac{1}{2 \log K^*}$ where K^* is the maximum bound on the number of communities possible. As a metric the VI is quite useful for exploring the space of clusters generated by an algorithm, for example if one wanted to cluster the identified clusters.

Other metrics

There are many other metric for assessing the alignment between two community partitions. There are cases where one knows one of the partitions is the ground truth and might want to penalize successes and misses differently. The Rand Index [189], for instance, is one of the *pair counting* metrics and is defined by:

$$RI = \frac{a + b}{C_2^N} \tag{1.22}$$

where a is the number of pairs of samples that are within the same community in the ground truth that are also in the same community in the predicted partition, b is the number of pairs of samples that are in different communities in the ground truth that are also in different communities in the predicted partition, and C_2^N is the total number of possible pairings of the N samples. This also has a normalized version that is adjusted for chance, the adjusted Rand index (ARI) [234]. Other pair counting methods include the Jaccard Index and the Fowlkes-Mallows method [66].

Other measures have been adapted from the realm of supervised learning. The F_1 score is defined as the harmonic mean between the precision and recall [143]. While both of these have their use, they are largely dependent on having a set of labels to serve as ground truth. For example, with regards to the F_1 score, in order to assess the precision of an identified partition, one has to have a notion of a true positive. This assumes not only are the class labels aligned between the two partitions (*i.e.* that the labels are permuted correctly), but also assigns a privileged status to matching one community label (a positive prediction) over another (a negative prediction). For an overview of metrics see [235] and [146]. Throughout this paper we report the majority of results using AMI.

1.2.4 Benchmarking community detection algorithms

One of the major approaches to validating community detection algorithms is to show that a particular method is able to detect communities on networks generated from synthetic models. For instance, we described in Section 1.2.2 how the SBM is a generative model, meaning that if I arbitrarily fix the parameter of the model, then I can sample random networks according to that distribution. While the statistically optimal approach to identifying communities in a synthetic network is to fit the model that generated the network, synthetic models still provide a useful way to test and compare performance across different algorithms. Furthermore, there are a number of more complicated and realistic generative models for which there is not a tractable approach to inferring the parameters of the model from the observed data. One such widely used synthetic model would be the Lancichinetti-Fortunato-Radicchi (LFR) benchmark model community detection [125]. The LFR model seeks to generate networks with degree distributions and community sizes that are both captured by power laws, hoping to match the more heterogenous distributions found in real world networks. Another flexible set of multilayer models was developed by Bazzi *et al.* to capture how various interlayer topologies can influence community structure across layers [21]. We use both of these models to benchmark our belief propagation approach in Chapter 3. Synthetic models like these are very useful in that because we know the ground truth community assignments used to generate samples from the model, we can reliably assess the performance of our algorithm. Furthermore, we can compare multiple community detection algorithms across a range of parameters for the model to assess the scenarios where different approaches are optimal. The downside of using synthetic data is that most models are drastically more simplistic than real-world datasets and each model assumes a particular notion of communities that might not align with the assumptions of the algorithm. Thus it is good practice to test any algorithm across multiple synthetic benchmark sets, and as wide a range of parameters as possible to understand its specific benefits and limitation.

1.2.5 Assessing results on real world data

Given the limitations of evaluating the performance of a community detection algorithm on synthetic data alone, we also suggest the testing of algorithms across real world datasets as

well. This enables the researcher to see how the algorithm will hold up when applied across data that is noisy with heterogenous distributions, generated from an unknown process. While this sounds at first blush like the ideal way to test methods, the main challenge with using real world data is that in general, we don't know what the underlying structure (if there is any) is for most networks. Usually a network is simply observed or constructed (with partial information) and we can't observe the latent process that gives rise to propensities for groups of nodes to form communities. As a proxy measure for the real world datasets, researchers commonly look for alignment between the detected communities and known node-attributes or metadata. This make sense if there is good reason, *a priori*, to believe that the metadata would be highly influential in contributing to the structure of the network. For instance, in both Chapters 2 and 3 we assess the results of our approach on the NCAA College Football network compiled in [58, 71]. In this network, each node represents the football team for a given university, while each edge represents whether or not two particular teams played each other in the 2000-2001 season. The metadata that we use to assess the results of our algorithm is the knowledge of which conference each team belongs to. Because the overall schedule of games is designed to produce a ranking within each conference, we know that the structure of the network should largely be determined by these group memberships. Thus any results produced from a community detection should strongly align with the conference labels. Similarly, there are other datasets for which there is good reason to believe known metadata reflects the structure of the network. For instance a protein-protein interaction network might have labels denoting the biological function for each gene/protein. It is fairly plausible to believe that interacting genes should be likely to be involved in the same or similar functions. [12]

However, the metadata approach to evaluating community detection algorithms has recently been critiqued by Peel *et al.* in [179]. They point out that failure to identify communities that are well aligned with the metadata of a given network could be caused by: 1) there is no detectable community structure in the network 2), the metadata attribute in question did not significantly contribute to the detected structure within a network, or 3) the algorithm under evaluation simply performed poorly. They frame the challenge of community detection as an

[12] One should be careful that this standard is not applied circularly. For many genes, the function has not been definitely shown, it is just deduced from the set of interactions the gene has.

inverse problem: assuming that there is a process that generates an observed network from the ground-truth community labels, the challenge of community detection is then to find the inverse of that mapping from the network we observe to those community labels. Without knowing either the original process to construct the network, or the original community assignments, there are not enough constraints on this problem to enforce a unique solution. By substituting the metadata labels for the unknown ground-truth community assignments, we provide more constraints to the problem. However, Peel *et al.* argue that this "simultaneously tests the metadata's relevance and the algorithm's performance, with no ability to differentiate between the two" [179]. They suggest a test for assessing how well a given metadata explains the structure of a network under a given statistical model. They also introduce a variant of the SBM called neoSBM that applies a penalty to the standard SBM model for switching a node out of the community label dictated by its metadata. By varying the cost of switching from the metadata labels to the communities implied under the SBM, their method can assess whether the metadata is capturing a completely different aspect of the network's structure than the SBM model. Regardless, one should use caution when assessing the performance of a community detection algorithm with metadata and should not rely exclusively on misalignment between it and detected communities as proof that an algorithm performs poorly.

1.2.6 Significance of community structure

There has also been a push to develop tools for assessing whether or not detectable community structure is present in a network at all. Most algorithms will produce a clustering of the network regardless of whether or not the input network is well partitioned. While many approaches such as modularity do provide a "score" of the clustering, these values are often difficult to interpret and can be misleading. Modularity, for example, can be quite high for certain classes of random graphs including trees [10], graphs with constant average degree [49], as well as even Erdős-Rényi graphs [259]. Reichart and Bornholdt offer a formula for the expected modularity under the partitioning of a random ER graph based on the ground state of a q-state Potts model [192], in addition to other limits on detectability imposed by the "resolution-limit" discussed above [123]. Hard limits on the detectability of communities under the planted partition version stochastic block model have also been derived in [50] as well as an optimal

approach for recovering those communities [153]. Nadakuditi and Newman showed that modularity also worked all the way down to this limit for the planted partition model [160]. Detectability limits have also been explored for networks with heterogenous degree distributions [188], hierarchical structure [180, 204], and in the case of different multilayer aggregation strategies for multiplex multilayer networks [218]. Many of these approaches characterize detectability as the conditions under which the eigenvalues associated with community structure emerge from the bulk structure for a particular matrix representation of the network [160, 204, 218]

Knowledge of these detectability limits can allow one to assess whether or not the observed network and learned parameters are within the detectability regime for a given community model. They also provide important benchmarks to assess whether a given method is performing optimally for synthetic networks generated under the model for which the limits have been derived. However, there is an ongoing need for methods that can assess the significance of detected communities on real world data for more flexible models. The contribution of our thesis towards this problem is detailed in Chapter 3, where we present a modularity-based, belief propagation approach towards detecting significant community structure in multilayer networks.

1.3 Network based approaches in genomics and oncology

The advent of large scale "omics" data in recent years has allowed us to probe the depths of molecular biology in ways that were hitherto impossible. With such large increases in the scale of available information also comes an increased awareness of the complexity of the underlying biology as well as the need for a greater array of analytical tools to make sense of these systems. This is especially true in the realm of oncology where we have seen nation-wide efforts to collect and characterize genomic data from many different cancer types with the ultimate goal of developing personalized therapies. Network analyses have played a key role in being able to synthesize the many disparate molecular sources of information as well as in identification and interpretation of the underlying biological patterns uncovered by these new high throughput assays. One of the aims of this thesis is to highlight the success that applications of networks-based analyses have had in the field of oncology, especially in the interpretation of large

scale genomics data. In this section, we walk the reader through several broad applications of network science to problems in oncology including the representation of the human genome using different sets of networks, the identification of driver mutations and pathways in various cancers, and the characterizations of new molecular subtypes on the basis of multiple data sources.

1.3.1 Representation of the human genome

One of the main ways that network analyses have been applied to genomics research is in representing the vast array of genes/proteins interactions captured by various molecular assays and experiments. Protein-protein interaction (PPI) networks attempt to encapsulate the sum total of interactions between genes based on different lines of evidence. This has been particularly impacted by the ability to conduct different kinds of high throughput assays including genomics, transciptomics, metabolomics, and proteomics. With the development of these assays, there has been a rise in databases and analysis tools that seek to compile/curate known interactions as well as predict new ones. There are many different databases that differ in how evidence is compiled, what lines of evidence are accepted, and what species are included [116].

Broadly speaking, databases can be characterized as either primary databases that seek to compile interactions from individual sources (IntAct [110], BioGRID [170]), or secondary databases that collect and/or computationally predict interactions from multiple primary sources (STRING [216], HumanNet2 [111], HINT [45]). Interactions in the primary databases such as BioGRID are derived from the primary literature (using a mix of data mining and hand curation) [170]. Secondary tools, however, such as HumanNet2 include links inferred from various sources including predicted links from the observed network topology; the mining co-citations in the genomics literature; and the occurrence of *interlogs* in other species [111]. Many of these networks can be treated as inherently multilayered, representing a number of factors. For example, the STRING-db characterizes edges not only on the types of evidence available (co-expression, experimental/biochemical, co-mentioned in PubMed Abstracs, etc.), but also based on the type of functional association (stable physical associations, transient binding, substrate chaining, information relay, etc.) [216]. In some contexts, it is useful to conceptualize these different types of functional relationships as different layers within the network.

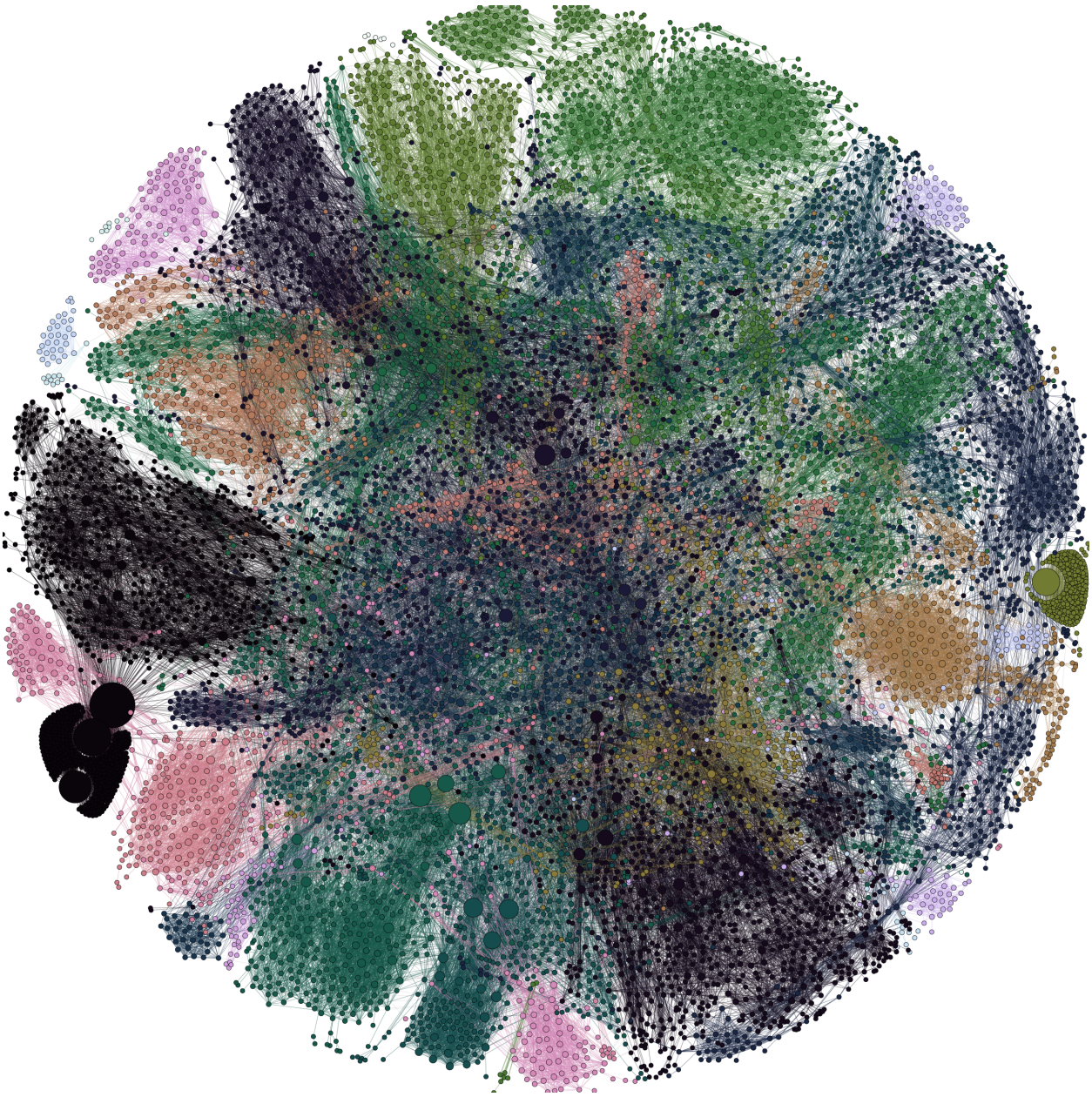

Figure 1.8: A portion of the STRING PPI for the human genome. We have selected the top 83000 edges and removed all nodes with zero degree, leaving 10641 vertices. We have applied the Leiden algorithm [225], to identify 34 communities within the network, which we have denoted using the color of the nodes.

These databases of PPI's have been a boon for the application of networks-based analyses toward many different problems. Mapping the space of functional interactions between genes has provided one of the largest sources of evidence for predicting functions for many proteins. For example, the popular Gene Ontology (GO) annotations can be predicted on the basis of associations within mapped protein-protein interaction networks [52, 257]. In pharmacology, everything from the prediction of possible drug targets to understanding side effect profiles has been augmented by the incorporation of PPI's [24]. These databases are also widely used to infer disease-gene association: genes that are strongly connected in the PPI to known disease causing genes might also be involved in the pathogenesis of a disease [96]. Several of the approaches in the following section incorporate the protein-protein interaction network towards understanding cancer biology.

There has also been a rise in other types of networks that seek to model a more limited set of interactions among genes or other types of regulatory molecules. For example transcription regulatory networks attempt to curate known interactions between genes and regulatory molecules such as transcription factors and microRNA's [80, 140]. There are also networks that represent the metabolic pathways and reactions such as KEGG [106], as well as networks of the many small molecules (proteins, amino acids, carbohydrates, etc.) and their chemical/enzymatic relationship within the human body [33, 248]. See [264] for an overview of the different kinds of genomic networks that have been constructed as well as details about the type of graph structures used to represent them (*e.g.* bipartite, directed, etc.).

1.3.2 Identification of driver mutations

One of the hallmarks of cancer is the dysregulation of normal genomic signals, either through mutations or through aberrant epigenetic changes. Most cancers are characterized by genomic instability, with the resulting cascade of mutations leading to the bypassing of normal regulatory processes. While mutations normally accumulate in cells throughout the body over the course of one's lifetime, with the majority of them being non-impactful, in the case of cancer, mutations are acquired at an increased rate. One of the main challenges in the field of cancer genomics is to distinguish between the "driver" mutations that are contributing to development and survival of a tumor, and the many "passenger" mutations that are present by random chance.

Examining the most frequently mutated genes across many different tumor samples has revealed a few highly mutated driver genes. For example TP53 or PIK3CA were two of the earliest identified driver genes in breast cancer [12, 202] based on high mutation frequencies in these genes. More recent approaches look for genes that are more frequently mutated than expected under the estimated background mutation rate, which can vary across tumor types as well as individual patients [126]. However even the most common mutations in breast cancer only occur in 35% of cases, while most mutations are relatively rare leading to a "fat-tailed" distribution of mutation frequencies [54, 69]. Frequency of mutations in the various genes also differs markedly across cancer types and generally speaking, mutations in several genes are required for cancer to develop and spread [141, 215]. Not only is landscape of mutations quite variable across different tumors; we even see remarkable heterogeneity in the mutations present across the individual cells of the same tumor as new mutations are continually acquired [88]. This can have large implications for the ability of cancer cells to develop resistance to therapies.

In light of the finding that most mutations are relatively rare, there has been a large push to characterize the common genomic pathways that are dysregulated across different cancers. The underlying assumption is that while most individual mutations are rare, alterations at the level of specific functional genomic pathways are more common, leading to the much lower levels of phenotypic heterogeneity observed within cancer subtypes. The hope is that if by leveraging the map of functional associations across all genes, more frequently occurring patterns of genomic disruptions would emerge. Many approaches test for significantly different mutation rates at the level of predefined pathways. Although designed for gene expression analysis, tools such as DAVID [95] or GOstat[22] can also be used on mutational significance test scores. For example, Lin *et al.* derive a group Cancer Mutational Prevalence (CaMP) score to assess the significance of mutation rates for a group of genes and also apply gene set enrichment analysis (GSEA) [214] to the individual gene CaMP scores based on predefined set of pathways [137]. Other approaches such as PathScan define a significance score for each pathway at the level of individual tumors, which are then combined appropriately [246]. These tests overcome many of the limitations of using individual genes by increasing their power to detect significantly increased mutation rates aggregated over genes in each pathway. However, these methods suffer from a number of drawbacks. As the number of gene sets included in the analysis grows, so does the corresponding

threshold for significance; thus smaller gene sets may not be ever be considered enriched. These methods largely ignore the overlap of genes across sets as well as the crosstalk between the multiple pathways needed for cancer to develop. And finally they treat each gene within a group equally, ignoring the gradations in gene centrality within the underlying network structure [190].

Rather than testing for significance in predefined groups of genes, there are a number of approaches that directly leverage the underlying network structure of gene interactions to identify significantly mutated modules. This provides a more unbiased approach that respects the complexity of the interconnected signally pathways. For instance, HotNet2 projects the vector of sample mutations onto the PPI network using a "heat diffusion" process [133]. Using the undirected adjacency for the HINT gene interaction network [45], a directed diffusion matrix is defined as follows:

$$\mathbf{F} = \beta(\mathbf{I} - (1-\beta)\mathbf{W})^{-1} \tag{1.23}$$

where β is an insulation parameter than controls what fraction of a node's heat is retained at steady state, and $\mathbf{W} = \mathbf{A}\mathbf{D}^{-1}$ is the degree normalized adjacency matrix.[13] The heat projected onto each node j from node i is then given by:

$$E_{ij} = F_{ij}h_j \,, \tag{1.24}$$

where the vector $\mathbf{h}$ encodes the mutation frequency for each gene. Basically, this allows for mutations that commonly occur in genes that are well connected within the network to be pooled together in the test for statistical significance. To assess significance of observed subnetworks, they use a two-stage test based on observed heat values across similar subnetworks within many different permuted versions of the PPI. Using this approach, they identified several genomic modules including the MHC class I proteins, cohesin and condensin complexes, and the telomerase complex that appeared to be important in cancer proliferation, even though all of their individual genes are fairly rarely mutated [133]. DawnRank is another network based method to find rare driver mutations. DawnRank looks for an association between individual mutations and alterations in expression levels for genes that are downstream in the PPI network

[13] Normalizing the adjacency matrix by the degrees makes it into a Markov transition matrix that acts on a column as input.

[93]. DawnRank is based on the well known PageRank algorithm[172], though it uses in-degree of each node rather than the out degree. It computes the following iterative equation for the rank score of each gene, j:

$$r_j^{t+1} = (1 - d_j) f_j + d_j \sum_{i=1}^{N} \frac{A_{ji} r_i^t}{k_i^{in}} \tag{1.25}$$

where d_j is the individual gene dampening score, k_i^{in} is the in-degree of node, i, and f_j is the differential expression score for gene j between tumors and normal samples. This equation tells us that the rank for a gene, j, is a weighted combination of its differential expression score, and of the differential expression scores for its neighbors normalized by their degrees. By incorporating differential expression, as well as the topology of the PPI, DawnRank is able to identify "personalized" driver mutations that are present in only a single sample.

Community detection tools have also been applied to reveal driver genes and pathways in the cancer genome. Cantini *et al.* construct a multilayer, gene-gene network with the different layers representing transcription factor co-targeting, microRNA co-targeting, the known physical interactions (the PrePPI network [262]), and finally, the cancer specific co-expression network [32]. They apply several popular multilayer community detection methods to reveal modules of genes that extend across the different network layers. They compared networks constructed from normal tissue samples to those from cancer samples to reveal biological pathways that were enriched in different cancer types. This highlights how community detection tools can be applied directly to network representations of the different genomics data to reveal tumor driver pathways.

These network based approaches will continue to improve as the quality and coverage of the available PPI networks improve. They will also benefit by the introduction of tissue specific PPI networks such as GIANT [79], that will allow for information to be shared between different cancer types in a way that respects tissue-specific differences. There is also an ongoing push to identify driver mutations within the intergenic regions [243]. Interpretation of mutations within these unexpressed, regulatory elements that have recently been mapped by the ENCODE project [57] will no doubt rely on the tools of network analyses to reveal new patterns of genomic reprogramming in cancer.

1.3.3 Classification of cancer molecular subtypes

Cancer is not itself a single disease entity; it is wildly heterogenous, covering an enormous span of clinical presentations, with a wide range of ramifications for each patient. This fact has been recognized for centuries and researchers have sought to develop a schema for classifying cancers in order to better prognosticate, and in some cases, treat the patient. Until recently, researchers largely characterized different cancers on the basis of the organ/tissue from which they arose, and their histological appearances. This is still a mainstay in how cancer types are conceptualized today, with over 200 different organ sites listed on the NCI's website (https://www.cancer.gov/types/by-body-location). However, with the advent of novel molecular assays has come an increased ability to peer into the inner workings of the cell, and map the many different ways that the normal biological functions of the cell can become aberrant. For example, prior to microarray expression assays, breast cancer was largely categorized based on IHC staining for several extracellular markers including the estrogen receptor (ER), the progesterone receptor (PR), and the human epidermal growth factor receptor 2 (HER). However, with the ability to probe tens of thousands of gene expression levels using microarrays, Perou *et al.* were able to identify 5 different classes of breast cancer subtypes as well as the main sets of genes contributing to this variation [185]. Importantly this schema was shown to be highly predictive of overall survival and the rate of metastases [211]. Similar approaches have been applied to other cancer types such as bladder cancer [44, 107], Glioblastoma Multiform (GBM) [233], as well as many others. Classes across different cancer types continued to be refined as more samples are collected and new types of molecular assays (DNA mutations, epigenetic differences, copy number variants, etc.) all split samples along orthogonal biological differences.

Networks provide a natural platform for coherently integrating the information from numerous biological assays. Wang *et al.* created a multiplex, multilayer network of Glioblastoma Multiform (GBM) samples, with each layer representing similarity relationships defined by a different biological assay [239]. They used microRNA, methylation, and mRNA expression data to define proximity between samples in each of the layers, combining an exponential kernel with a K nearest neighbors approach. They identified communities across the various layers using multilayer modularity (see Equation 1.6), which they showed were somewhat distinct from the

canonical subtypes, and importantly, were predictive of difference in survival. One of the issues with their approach, which they note, is the dependency of the number of identified subtypes on the choice of K for the KNN construction of the network, as well as the resolution parameter γ for setting the scale of communities. Presumably these could be tuned to maximize a downstream parameter such as differences in survival, however this would limit the generalizability of the identified subtypes to different datasets.

There are a number of challenges associated with developing robust classification schemata on the basis of high dimensional molecular assays. Like other unsupervised approaches, it can be difficult to validate the identified results. There are several metrics for assessing how tightly clustered a dataset is without reference to any ground truth: compactness, connectedness, or predicted stability [85]. However, there is currently no rigorously developed process for demonstrating the significance of identified clusters. This is particular troubling given that for most datasets the number of measured features vastly outnumbers the sample size (*i.e.* $p \gg n$), making it possible to find discriminating features for random partitions of the data. One common approach to surmounting this problem is to apply network based regularization to the feature space prior to clustering the data. This regularization enforces that genes that are proximal to each other in the PPI network also have similar valued coefficients in the classifier [264]. One way to apply this regularization is through the graph Laplacian, $\mathbf{L}$ (see Section 1.1.1 for additional details about the graph Laplacian).

One such unsupervised approach where graph-based regularization has been successfully applied is called non-negative matrix factorization (NMF). NMF is similar to principal component analysis (PCA) in that it seeks to identify a low rank approximation to a given matrix; unlike PCA, however, the loadings for each factor can only be non-negative (and the factors do not have to be orthogonal to each other) [130]. If we let $\mathbf{X} \in \mathbb{R}^{N \times M}$ represent our genomics dataset (for example, the expression of N samples across M genes), the NMF seeks to minimize the following objective function:

$$\underset{\mathbf{U},\mathbf{V}}{\arg\min} ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||_F^2 \qquad \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \tag{1.26}$$

where $\mathbf{U} \in \mathbb{R}^{+N \times k}$ is the loadings for the samples and $\mathbf{V} \in \mathbb{R}^{+M \times k}$ is the loadings for each of the

features, both of which are constrained to be non-negative. The number of loadings, k, can be chosen to be the lowest integer that achieves an acceptable reconstruction error. In enforcing positivity on the loadings, NMF achieves a parts based representation of the data with more interpretability for each of the feature loadings. In genomic applications, each of k loadings for each samples is referred to as a meta-gene, because each represents a weighted combination of genes that can be summarized by a single value in the reconstruction [30, 251]. Cai *et al.* introduce the concept of regularizing the NMF reconstruction on graph structure features by including the graph Laplacian in the objective function to minimize:

$$\underset{\mathbf{U},\mathbf{V}}{\arg\min} ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||_F^2 + \lambda \text{trace}(V^T \mathbf{L} V) \qquad \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \tag{1.27}$$

where $\mathbf{L} \in \mathbf{R}^{M \times M}$ is the graph Laplacian for the network of the features (such as a PPI network) and λ is a penalization parameter that sets how strongly the regularization is enforced. This penalty enforces a smoothness of the loadings for features that are close to each other in the underlying network structure.

Hofree *et al.* combined this regularized NMF approach with a network-based smoothing approach to identify subtypes with ovarian, lung, and uterine cancers on the basis of their mutational data alone [90]. The discover of cancer subtypes using mutational data alone is difficult because of the very sparse structure of the data: most pairs of samples will share few if any mutations even if very similar genomic pathways are disrupted. To overcome this, Hofree *et al.* allowed the mutation status of each gene to be propagated across its local neighborhood in the PPI network using the approach from [232], which is similar in spirit to the diffusion process from HotNet2 described above. This transforms the data matrix from being sparse and binary to being dense and continuous, and allows information about the relatedness of genes to be incorporated. They then apply network-regularized NMF to this dense mutation data, along with consensus clustering, to identify clinically relevant subtypes that predicted differences in survival. Most importantly their approach allows for the discovery of which subnetworks in the broader PPI network were important in determining the cancer type.

Xi *et al.* used a similar approach to factorize mutation information with regularized NMF [250]. However in addition to regularizing the gene loadings, they also incorporated gene

expression data as additional prior information. They construct a sample-sample network based on the pairwise correlations of gene expression levels. This network is then used to apply regularization to the sample loadings in the same manner as before. This encourages similar loadings for samples that are close in the expression correlation network. In addition to revealing cancer subtypes, they use this approach to identify additional driver mutations on the basis of the strength of the feature loadings in the sparse representation.

While these unsupervised approaches can categorize different samples in an unbiased manner, it is often of interest to identify genomic features that are predictive of an important phenotypic trait, for example whether or not a cancer has metastasized. Networks can be incorporated into these supervised approaches at numerous points in the process. Chuang *et al.* project expression data for each gene onto the PPI network and greedily search for subnetworks that are discriminatory between the metastatic and non-metastatic samples [38]. Aggregated expression across each subnetwork is then used downstream as features in a logistic regression classifier. In other cases, the regularization can be incorporated directly into the supervised model. Chen *et al.* [36] apply a network-based penalty to inform the features learned by a support vector machine (SVM) that has a very similar form to the penalty term in Equation 1.27. More recently, there has been a push to apply the tools of deep learning to graph structured data. Graph convolutional networks (GCN) provide the flexibility of deep architectures to modulate how information is shared across neighborhoods in a network to increase predictive power [113]. Wang *et al.* deployed this framework successfully to integrate multiple types of genomic information to predict survival rates across cancer types [238]. These tools offer exciting new approaches for combining the power of deep learning with the rich prior knowledge of the underlying PPI networks. For example Rhee *et al.* [193] trained a relational network downstream of a PPI based GCN to predict the PAM50 breast cancer subtypes [177] with high accuracy. While the application of deep approaches suffers from lack of interpretability, incorporation of graph-based regularization could enable researchers to identify which parts of the gene networks are most informative for prediction tasks. However, the potential of this field remains largely untapped.

1.3.4 Other network based applications

While the previous sections have focused mainly on methods that incorporate databases of interaction (PPI) networks, there are a number of other network representations that have been used successfully in the analysis of oncology datasets. It is often useful to represent relationships between different classes of objects using a special type of network, a *bipartite* network, demonstrated in Figure 1.9. A bipartite network is one where each node belongs to one of two classes and edges are only allowed between nodes of different classes. For example, Zhao *et al.* create a bipartite network where one class of nodes represents a specific cancer type and the other class represents protein complexes [265]. They connected edges within the network if the complex as a whole is over-expressed in the cancer tissues, but under-expressed in the tissue type of origin compared to other tissue types as a whole. They then cluster the nodes of the network via a hierarchical, agglomerative algorithm (Ward's method), using the similarity of neighborhoods between each cancer node to cluster the cancer types. The bipartite representation is useful in this context because it reveals which sets of altered protein complexes differentiate cancers from the different tissue types.

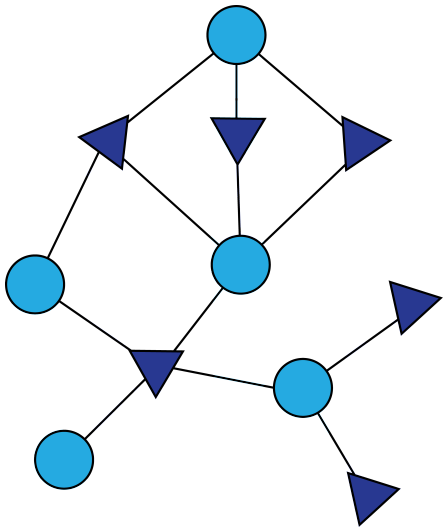

Figure 1.9: Example of a bipartite layout. Node classes are indicated by color and shape. Edges are restricted to nodes of different classes (*i.e.* edges are only present between circles and triangles). This type of network commonly arises when interactions between objects are measured through a different variable of interest. Examples include the network of actors/actresses and which movies they appeared in, researchers and papers they have published, the diseases and symptoms they express, etc.

Another important question that has been investigated using bipartite networks is the

identification of pairs of genes that tend to be significantly co-mutated together within cancers, or conversely, have less co-occurring mutation than expected by chance (mutual exclusivity). It is believed that each cancer has a relatively small number of driver mutations [236] that disrupt a few cancer enabling cellular functions [84]. Driver mutations rarely occur within the same pathway because there is a loss of selective pressure on other genes in the same pathway once an initial mutation has already occurred. Similarly, if there are two separate pathways that are both necessary for cancer to progress, one could expect to see a significant co-occurrence for pairs of genes in the two pathways. Thus the identification of mutually exclusive or co-occurring mutations can help to identify novel driver genes [9, 31, 40, 53, 258]. Many of these approaches rely on incorporating prior information from the known set of gene interactions contained in several PPI networks; however, Muller *et al.* used networks in a different manner with their NetCutter approach to discovering co-occurring/mutually exclusive pairs of mutations [158]. They use a bipartite graph to represent their mutational dataset, using the genes as one class of nodes and the tumor samples as the other. Each edge in the network connects a tumor sample to a gene if that specimen contained a mutation in the corresponding gene. Using this framework, they discuss how various other approaches to identifying significant co-occurrence can be thought of as various null models on the space of bi-partite networks. They suggest that the appropriate null model is the bipartite configuration model, which accounts for variation in both gene and sample mutation rates. Mathematically, this gives rise to the Poisson-binomial distribution to represent the expected number of occurrences between any two pairs of genes. Another paper by Canisius *et al.* also uses a Poisson-binomial model in their DISCOVER method to identify pairs of co-occurring/mutually exclusive genes, though they don't frame this directly as a networks approach [31]. Canisius *et al.* also use a different approach to estimate tumor-sample specific probabilities and don't use an approximation for estimating the Poisson-binomial distribution. Canisius *et al.* find that overall, statistically significant co-occurring mutations are quite rare, while mutually exclusive sets of mutations are more common, suggesting that negative selection is the dominant force acting on the cancer genomes.

Both of these approaches show how reformulating a question in terms of a network can provide new insight into the structure of the problem. The network can reveal how constraints on the data can sometimes produce artificial correlations that need to be accounted for by any model

in order to reveal true biological relationships. We provide a similar cautionary tale in Chapter 4 of this thesis; by treating our underlying dataset with the same bipartite network model as [158], we show how previously observed correlations mostly arise from constraints on the data itself rather than the underlying biology.

In this section we have shown how networks have been used to represent relational data between genes/proteins, as well as how that prior information can be used to inform specific tasks. We examined how networks can be to identify "driver genes" in the context of the fat-tail distribution, and how network based regularization can be used for both subtype identification and for downstream prediction. Finally we examined how other network representations can be useful in identifying protein complex association with different cancer types or revealing mutually exclusive sets of driver mutations. In Chapter 4, we will further demonstrate how a networks-based approach can yield insight into predicting an important genomic readout, tumor mutational burden. We close this chapter with a short outline of the entire thesis with links to the relevant publications for each chapter.

1.4 Outline of thesis

Chapter 1 **Introduction:** We provide the necessary background on networks for the reader to grasp the concepts detailed in the subsequent chapters. We also provide an overview of community detection in networks with a particular focus on modularity-based approaches. We detail some of the drawbacks of these approaches, providing additional justification for methods developed in Chapters 2 and 3. We conclude the chapter with an overview of how network analyses have been used in the field of oncology, setting the stage for Chapter 4.

Chapter 2 **CHAMP:** We present the Convex Hull of Admissible Modularity Partitions (CHAMP) a method to select the optimal subset of partitions of a network out of a larger ensemble. We develop the justification for CHAMP based on current approaches to community detection and showcase in several practical examples how CHAMP help identified the appropriate resolution scales for partitioning a network. We close this Chapter with a discussion of

practical considerations when applying CHAMP and other caveats to be aware of.

Post-Processing Partitions to Identify Domains of Modularity Optimization.
W.H. Weir, S. Emmons, R. Gibson, D. Taylor, P.J. Mucha. Algorithms. 2016.
http://www.mdpi.com/1999-4893/10/3/93

CHAMP package: Convex Hull of Admissible Modularity Partitions in python and matlab.
W.H. Weir, R. Gibson, P.J. Mucha. 2017-2020.
https://github.com/wweir827/CHAMP

Chapter 3 **Multilayer Modularity Belief Propagation:** We explore an alternative approach to optimizing modularity-based on the principles of statistical mechanics and discuss the benefits of such an approach. We introduce *multimodbp*, a belief propagation approach to maximizing multilayer modularity and walk the reader through the justification for our approach. We provide numerous benchmark and real world examples of our model and compare it to other state of the art approaches. We conclude by discussing the benefits of using an ensemble based approach to community detection.

Modularity belief propagation on multilayer networks to detect significant community structure.
W.H. Weir, B. Walker, L. Zdeborovà, P.J. Mucha. *In Submission*. 2019.
https://arxiv.org/pdf/1908.04653.pdf

Modbp package: multilayer modularity belief propagation in python.
B. Walker, W. H. Weir. 2019-2020.
https://github.com/bwalker1/ModularityBP_Cpp

Chapter 4 **The TMB Paradox:** In the final Chapter of this thesis, we apply a networks-based approach to identify genes whose loss or malfunction is associated with elevated levels of tumor mutational burden. In particular we address the question of which DNA damage repair genes are associated with elevated TMB. We show how this challenge can be recast as a networks problem and this yields insight into why other univariate approaches are

beholden to sampling bias. We demonstrate the robustness of across different datasets and showcase the implications our method has for predicting patient response to novel immunotherapy treatments.

CHAPTER 2: CONVEX HULL OF ADMISSIBLE MODULARITY PARTITIONS (CHAMP)

In this chapter we introduce a central problem in modularity-based community, the resolution limit, and discuss our contribution towards identifying the scales over which communities exist in a network. Our method, the Convex Hull of Admissible Modularity Partitions (CHAMP), is not a community detection algorithm *per se*, but rather a meta tool to identify the best partitions of a networks starting from a large diverse set. Given a set of partitions, CHAMP identifies the domain of modularity optimization for each partition —i.e., the part of parameter-space where it has the largest modularity relative to the input set—discarding partitions with zero-sized domains to obtain the subset of partitions that are "admissible" candidate community structures. This subset of the starting partitions represents those that remain potentially optimal over indicated parameter domains. We begin by presenting the motivation for and derivation of our method as well as an efficient algorithmic for applying CHAMP. We demonstrate the utility of CHAMP on several datasets and discuss it's role in community detection in both single-layer and multilayer networks. We discuss some of the drawbacks of the CHAMP approach as a motivation for an additional community detection tool that we will presented in the next chapter.

2.1 Modularity-based detection and the resolution limit

In Section 1.2.1, we introduced the quantity of modularity in it's various forms as a heuristic for optimizing community structure within a network. For single-layer networks, the original formulation of modularity was derived by Newman and Girvan [168] as a score to measure the assortativity of node attributes on a network [162] by counting the number of links between nodes of a certain class and subtracting the expected number of nodes under random rewiring on the network (with the configuration model). Newman and Girvan then suggested that this metric would be an appropriate way to select the cut off level for a hierarchical algorithm

based on successively splitting the network up by removing edges with the highest betweenness [168]. Ironically, here we see that modularity itself is being used a metric for the appropriate scale of drawing communities lines for a different method of fragmenting the network. Eventually, the modularity quantity was optimized directly by spectral [164], and other heuristic approaches (*e.g.* Louvain [27]). For a review of algorithmic approaches to modularity see Section 1.2.1 as well as [62].

One of the main challenges of the modularity-based approach to community detection, is the in ability of the algorithm to resolve smaller communities with in a larger network. Fortunado and Barthélemy demonstrated that modularity score of a given subgraph depends on the size of the network it is a part of it and not just on a "local" comparison of internal and external edges [63]. They demonstrate how this implies that for a ring of cliques, optimization of modularity will force adjacent cliques into a single community, despite the rather obvious natural divide into cliques. This phenomenon had been noted in real world networks previously, especially larger biological networks [157]. See Section 1.2.1 for a more detailed account of why modularity tends to artificially merge communities immersed in a larger network . One solution proposed by Kumpula *et al.* [119] was to use the adjusted formula for modularity developed by Reichardt and Bornholdt [192]:

$$Q(\gamma) = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \gamma P_{ij}\right) \delta(c_i, c_j)\,, \tag{2.1}$$

with the resolution parameter, γ as a free parameter to "balance between missing and existing links" [192]. Typically $P_{ij} = \frac{k_i k_j}{2m}$ is used representing the configuration model null for an undirected network, but there are also other null models that have been suggested. Kumpula *et al.* suggested that such a resolution limit would be inherent to all methods that rely on a null model based on global connectivity probabilities because the expected number of links between any two small sets of nodes is small at baseline; thus it only takes a few spurious edges to give a higher modularity by combining smaller communities [119]. Their conclusion is that there is no single "optimal" resolution for a given network. They suggest looking for communities across different values of γ and examining the hierarchical structure revealed over a range of γ. Traag *et al* suggested an efficient way of scanning the resolution domain when a Constant Potts null model

is used ($P_{ij} = p =$ constant), which can be interpreted as the minimum density threshold of internal edges needed to collapse nodes into a community [223]. It is essentially equivalent to using an Erdös-Renyí null model. They show that under this model, the number of communities is non-decreasing in γ, allowing for an efficient bisectioning method to identify specific values of γ where communities are fractionated further [223]. Several other approaches have been described to address the resolution problem including the addition of weighted, self loops to nodes on the network [7, 78] or the recursive application of modularity to the subgraphs induced by the communities until communities no longer split [124].

Despite the resolution limit as well as other draw backs (many of which are discussed in Sections 1.2.1), it remains a widely used approach to community detection and has been adapted to account for a number of other network models including directed [132], bipartite [14], signed [73, 221], and multilayer networks [156]. Regardless of the particular form, all of the aforementioned methods seek to identify community labels $\{c_i\}$, such that the particular definition of modularity, Q is maximized. We emphasize that throughout this work we will use the term "modularity" in its broadest sense to include any of these generalizations as applied appropriately to a given data set. Such generalizations include the use of resolution parameter γ, or multiple resolution parameters for signed networks, and can include one or more interlayer-coupling parameters for multilayer networks (which we discuss in more detail in Section 2.4). As the number of tunable parameters grows, the difficulty of trying to explore identified communities at multiple scales is compounded.

2.2 Scanning the resolution domain

The space of possible partitions is discrete and grows exponentially fast with the number of nodes within a network. Optimizing over In fact, optimizing Eq 3.10 over the space of possible communities has been proven to be $\mathcal{NP}$-complete [28]. Because identifying globally optimal community structure is computationally intractable for most networks (both for modularity and most other approaches), optimization algorithms are usually "greedy" heuristics that are guaranteed to find local extrema only. To ensure that a larger range of possible solutions is

covered, these methods are usually run repeatedly in a stochastic fashion[1] over a range of γ. The possibly different community structures found by computational heuristics at a particular γ parameter point [(γ, ω) for multilayer networks] are then typically assessed only at that point before moving on to generate results at other parameter values. For instance, one might select the partition with greatest modularity found at that specific value of γ or measure some statistic over the partitions that were generated at that γ (see, e.g., [1]). Thus a scanning of the resolution domain typically involves a combination of adjusting the resolution parameter(s) as well as multiple runs with stochastic initial conditions (as well as other sources of stochasticity within the algorithm).

In order to determine whether the obtained community structures are "robust" to the γ selection in any sense, one might look for stable plateaus in the number of communities (see, e.g., [60, 61, 78, 142]), consider another metric such as significance [224], directly visualize the different community assignments across parameters (as in [136, 142]), or compare obtained communities with other generally-acceptable labels by some measure such as pairwise counting scores (see, e.g., the discussion in [227]) or information-theoretic measures like Variation of Information [146] and Normalized Mutual Information [68]. A more computationally-demanding approach that directly attacks the problem that there is no *a priori* notion of what constitutes a "good" value of modularity is to compare the obtained best modularity at each γ with the distribution of modularities obtained by running community detection across some selected random-graph model, either on realizations from a model or from permutations of the data, repeating this process at different γ to identify parameter values where the obtained communities are strongest relative to the random cases [17]. Additionally, one may use a given set of partitions to generate a new partition by ensemble learning [171] or consensus clustering [17, 103, 124].

In many scenarios, the approaches above can produce very clean results. For example, in Figure 2.1 we have computed 10 partitions each at 300 different values of γ (total of 3000 partitions) using the Leiden algorithm by Traag et al [225] to optimize Equation 3.10. For each value of γ we compute the average, pairwise Variation of Information (VI) [146] between the ten

[1] with random initial conditions or selecting random moves

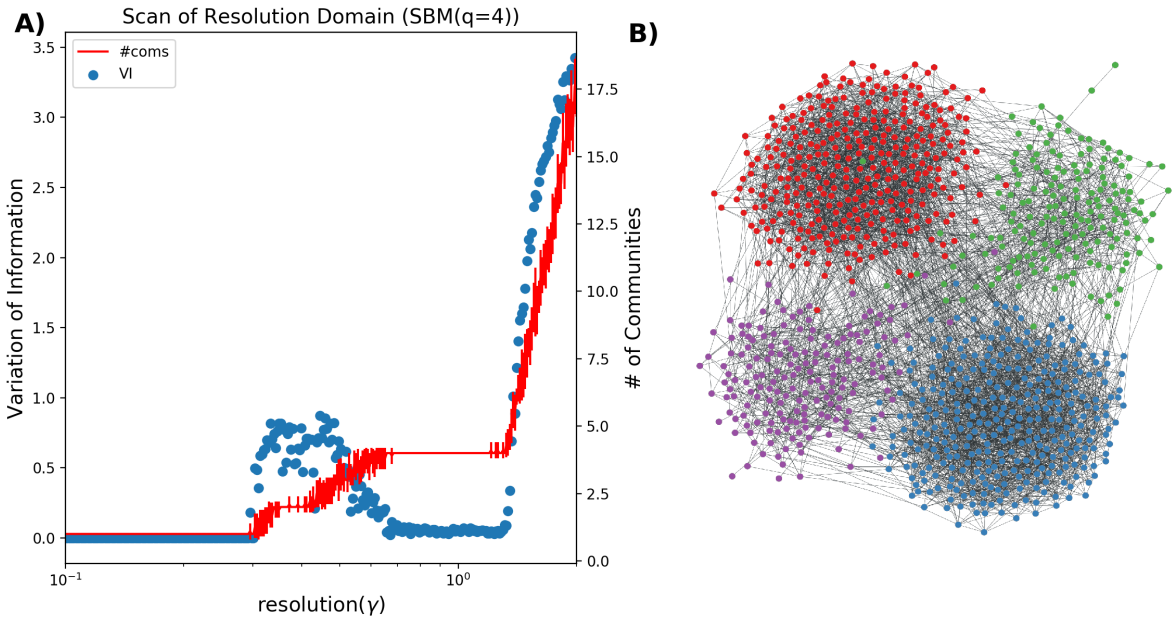

Figure 2.1: Application of modularity-based community detection (using Leiden method [225]) on very detectable 4 community stochastic block model. (non-degree corrected) with $N = 1000$, $< k >= 4$, $\epsilon = .2$, community sizes $= [325, 325, 175, 175]$. **A)** We run the Leiden algorithm 10 times for each value of $\gamma \in [.1, 2]$ in 300 evenly space intervals on a logarithmic scale. **B)** Layout of the network using force directed layout, ForceAtlas2 [98], colored according to ground truth communities.

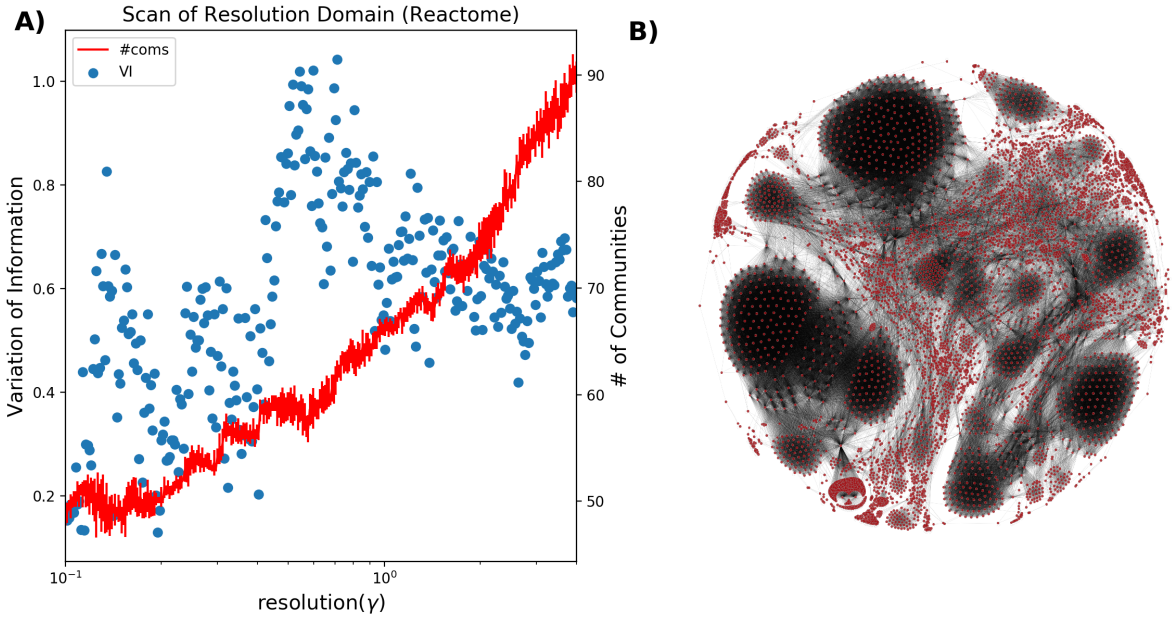

Figure 2.2: Application of Leiden [225] to the human reactome network [105, 120]. We run the Leiden algorithm 10 times for each value of $\gamma \in [.1, 4]$ in 300 evenly space intervals on a logarithmic scale.

partitions as well as the average number of communities. The left panel of Figure 2.1.A clearly shows a range of $\gamma \in [.6, 1.3]$ where both the number of communities identified (red line with error bars) and the VI stabilizes. In the case of the VI we get a dip in the amount of variation indicating that the identified communities in this range are more stable (recall VI is a measure of dissimilarity between partitions). However, we note that while the group of communities at each particular value of γ are fairly stable, the figure is misleading in that it doesn't reflect the change in community structure happening happening as γ is increasing. While we measure the stability of the communities identified are at any single value of γ, we don't capture how different communities are as we adjust γ. For the network in Figure 2.1, the average pairwise difference between communities identified at $\gamma = .6$ and those identified at $\gamma = 1.3$ is $.2$. While relatively small, this is 4 times larger than the average pairwise VI within any single value of γ. Thus, comparing partitions only at the γ for which they were discovered doesn't allow one to really assess stability of partitions with respect to changes in resolution.

Furthermore, for real world networks, especially larger ones, this type of approach often fails to suggest a single good value of γ. In Figure 2.2, we scan the resolution range with the same approach but on a real world network of the human reactome (a type of protein-protein interactions networks) [105, 120]. This network has 6327 nodes and 147547 edges and clearly has modular structure based on the layout in Figure 2.2.B. The partitions vary widely both within and across neighboring values of γ. We propose filtering some of this stochastic variation by only looking at the partitions that are have optimal modularity across different values of γ

Importantly, in all of the aforementioned approaches for exploring the parameter space, the optimal partitions associated with each γ value are typically computed independently of those at other γ values [and, again, in the multilayer case, (γ, ω)]. Variation in the structure of these partitions and their corresponding modularity can arise from both adjusting the input parameters and importantly, from the stochasticity of the algorithm itself. Often, for close enough values of γ, the variation in modularity of identified partitions is driven more by the stochasticity of the algorithm rather than the difference in the value of γ. Because of this independent treatment of the results from different γ values, a large amount of information that might be useful for further assessing the quality of the obtained partitions is typically thrown away. We propose a different approach, which we call CHAMP, that uses the union of all computed partitions to identify the

Convex Hull of Admissible Modularity Partitions in the parameter space. Our CHAMP method [244], allows for identification of the (much) smaller subset of partitions generated by such procedures, which is optimal by the modularity metric itself.

2.3 The CHAMP Algorithm (Convex Hull of Admissible Modularity Partitions)

Instead of associating each partition with a particular value of γ based on which value was supplied to the heuristic that identified it, we instead imagine each partition as representing function of γ over some larger domain. That is each partition gives us a $Q(\gamma)$ based on Equation 3.10, with γ allowed to vary. Thus an ensemble of partitions gives a collection of functions over γ with each function mapping from γ to the modularity of that partition given that choice of γ. The CHAMP framework identifies which of these functions give the maximum value of Q for any given value of γ as well as the values of γ for which each function is optimal. This task is greatly facilitated by the fact that Equation 3.10 is a linear function of γ; Thus each partition can be thought of as representing a line in the (γ, Q) (or a plane in (γ, ω, Q) for multilayer networks). CHAMP identifies the domains of optimality across a set of partitions by ignoring the γ that was used to compute each partition, finding instead the full domain in γ for which each partition is optimal relative to the rest of the input partitions (hereafter, we always use the word "optimal" in this restricted sense relative to the set of partitions at hand).

We find the intersection of the half-spaces above the linear subspaces by computing the convex hull of the dual problem. By identifying the convex hull of the dual problem, we prune that set of partitions to the subset wherein each partition has at least some non-empty domain in the parameter space over which it is has the highest modularity. This pruned subset contains all of the partitions admitted through the dual convex hull calculation. Visually, plotting Q as a function of the parameters, the pruned subset is that which remains in the upper envelope of Q, so that each partition appears along the boundary of the convex space above the envelope in the domain where it provides the optimal Q relative to the input set.

We have visualized the CHAMP approach in Figure 2.3. Originally, each partition is represented by a point in (γ, Q), where the x-coordinate is given by the γ for which the partition was identified by the particular algorithm used (Figure 2.3.A). All algorithms attempt to

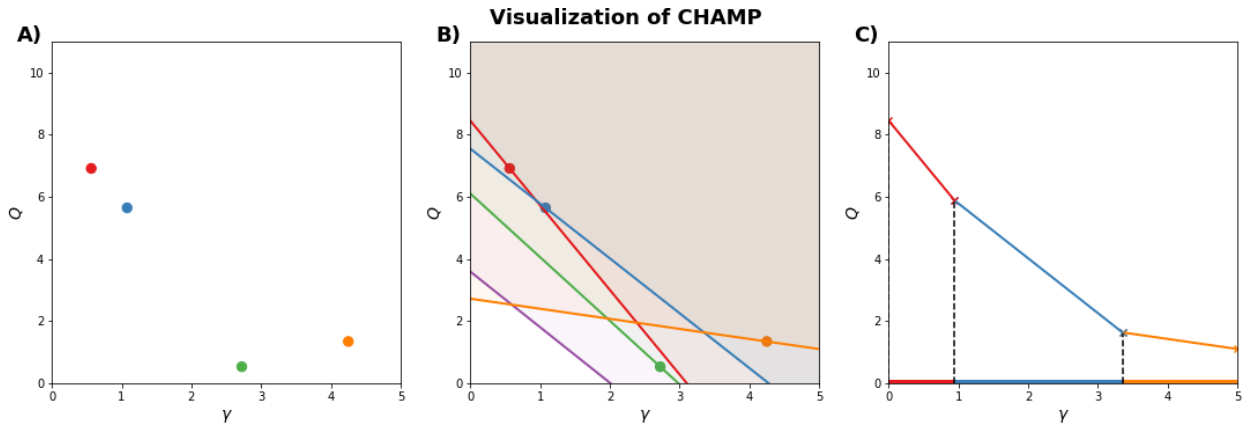

Figure 2.3: Visualization of the CHAMP algorithm. A) Each point represents a partition. The x-coordinate of the point is the resolution at which the partition was obtained by algorithm. **B)** We can think of each partition as defining a line. We want to to find the lines which bound the intersection of the areas above all of the lines (*i.e.* the region shaded brown in the figure). **C)** Only a fraction of the original lines will form this boundary (be in the convex hull) and each line will only optimal along some portion of the γ domain.

maximize modularity for a *fixed* value of γ, so this is a reasonable association to make. However, as we shall show, each partition can also be thought of a linear relationship between γ and Q (Figure 2.3.B), and in this framework we can compare partitions across different values of γ to identify the optimal subset (Figure 2.3.C).

2.3.1 CHAMP FOR SINGLE-LAYER NETWORKS

Consider a non-empty set, Σ of unique network partitions encoded by the node community assignments $\{c_{i\sigma}\}$ with $\sigma \in \{1, \ldots, |\Sigma|\}$. By construction, $\delta(c_{i\sigma}, c_{j\sigma}) = 1$ if nodes i and j are in the same community in partition σ (i.e., $c_{i\sigma} = c_{j\sigma}$), and 0 otherwise. Let $Q_\sigma(\gamma)$ denote the value of Equation 3.10 for given γ under partition σ. Ignoring the constant multiplicative factor in front of the summation (alternatively, absorbing that factor into the normalization of A_{ij} and P_{ij}), Equation (3.10) can be written as

$$\begin{aligned} Q_\sigma(\gamma) &= \sum_{i,j} \left(A_{ij} - \gamma P_{ij}\right) \delta(c_{i\sigma}, c_{j\sigma}) \\ &= \sum_{i,j} A_{ij}\delta(c_{i\sigma}, c_{j\sigma}) - \gamma \sum_{i,j} P_{ij}\delta(c_{i\sigma}, c_{j\sigma}) \\ &= \hat{A}_\sigma - \gamma \hat{P}_\sigma \end{aligned} \tag{2.2}$$

where the quantities $\hat{A}_\sigma$ and $\hat{P}_\sigma$ are the respective within-community sums over A_{ij} and P_{ij} for partition σ. Importantly, $\hat{A}_\sigma$ and $\hat{P}_\sigma$ are scalars that depend only on the network data (i.e., A), null model (i.e., P), and partition σ. Thus, for a given partition σ, Equation (2.2) is a linear function of γ, which can be visualized as a line in the (γ, Q) plane. (See Figure 2.3 as well as Figure 2.5B in Section 2.5.1 for an illustration of lines $\{Q_\sigma(\gamma)\}$ for several partitions of the 2000 NCAA Division I-A college football network [58, 71].)

We now compare the partitions' modularity lines $\{Q_\sigma(\gamma)\}$, seeking to identify the optimal partitions that yield the largest modularity values across the γ values—that is, the upper-envelope boundary $\{Q_\sigma(\gamma)\}$ for the set. We will additionally obtain γ-domains over which a given partition is optimal (discarding partitions that are never optimal). Given a finite set of partitions $\{\sigma\}$, the coefficients $\hat{A}_\sigma$ and $\hat{P}_\sigma$ can be computed individually, independent of how those partitions were obtained. Therefore, a given value of γ admits an optimal partition σ^* corresponding to the maximum $Q_{\sigma^*}(\gamma) \geq Q_\sigma(\gamma)$ from the given set of partitions $\{\sigma\}$. At most values of γ, only a single partition provides the maximum (i.e., "dominant") modularity. When two partitions σ and σ' correspond to identical modularity values [i.e., $Q_\sigma(\gamma) = Q_{\sigma'}(\gamma)$], it is typically because this is the unique intersection of the two corresponding lines. [2]

For a pair of partitions σ and σ', the intersection point $(\gamma_{\sigma\sigma'}, Q_{\sigma\sigma'})$ indicates the resolution $\gamma_{\sigma\sigma'}$ at which one partition becomes more (less) optimal over the other with increasing (decreasing) γ. That is, one partition dominates when $\gamma < \gamma_{\sigma\sigma'}$, while the other dominates when $\gamma > \gamma_{\sigma\sigma'}$. It immediately follows that the γ-domain of optimality for a partition must be simply connected. (We note that in higher dimensions, such as for signed or multilayer networks, the same linearity requires that domains of optimality must be convex [156].).

We leverage these intersections to efficiently identify the upper envelope of modularity for a given set of partitions, and the corresponding dominant partitions (relative to the set) for all $\gamma \geq 0$ as follows. Starting at $\gamma_0 = 0$, the partition with maximum $\hat{A}_\sigma$ is optimal. For networks with a single connected component, this partition is a single community containing all nodes; for multiple disconnected components, any union of connected components gives the same $\hat{A}_\sigma$, but

[2]It is possible to have the case where two different partitions have identical $\hat{A}_\sigma$ and $\hat{P}_\sigma$ coefficients, and thus have equal $Q_\sigma(\gamma)$ for all γ; but in practice we have observed this rarely in our examples (we refer to these as "twin" partitions). We hereafter ignore this possibility; it merely indicates two partitions of equal merit, in the sense of modularity, across all scales.

we select the partition wherein each separate component defines a community. Denoting the optimal partition at γ_0 by σ_0^*, we calculate the intersection points $\{\gamma_{\sigma_0^*\sigma}\}$ with the other partitions $\{\sigma\}$ where $Q_{\sigma_0^*}(\gamma) = Q_\sigma(\gamma)$. Substituting Equation (2.2) into this constraint yields

$$\gamma_{\sigma_p^*\sigma} = \frac{\hat{A}_{\sigma_p^*} - \hat{A}_\sigma}{\hat{P}_{\sigma_p^*} - \hat{P}_\sigma}, \tag{2.3}$$

where $p \geq 0$ for generality. Starting with partition σ_p^* for $p = 0$, we identify the smallest intersection point $\gamma_{\sigma_p^*\sigma} > \gamma_p$, which we define as γ_{p+1}. We denote the associated partition by σ_{p+1}^*. That is, partition σ_p^* is optimal for the γ-domain $\gamma \in [\gamma_p, \gamma_{p+1})$, above which partition σ_{p+1}^* becomes optimal. In the unlikely event that multiple partitions are associated with the γ_{p+1} intersection point, the one with smallest $\hat{P}_\sigma$ becomes σ_{p+1}^*. Setting p to $p+1$, we iteratively repeat this process until there are no intersections points satisfying $\gamma_{\sigma_p^*\sigma} > \gamma_p$. We thus obtain an ordered sequences of optimal partitions, $\{\sigma_p^*\}$, and intersection points $\{\gamma_p\}$ for $p = 0, 1, \ldots$. The optimal modularity curve for $\gamma > 0$, given by the upper envelope of the set $\{Q_\sigma(\gamma)\}$, is then given by the piecewise linear function

$$\tilde{Q}(\gamma) = Q_{\sigma_p^*}(\gamma)\,, \quad \gamma \in [\gamma_p, \gamma_{p+1}). \tag{2.4}$$

Of course, this procedure can be started at any selected γ of interest, and the analogous procedure for identifying intersections for decreasing γ could be used to obtain the upper envelope for $\gamma < 0$; but in practice here we restrict our attention to $\gamma \geq 0$

2.4 Extension of CHAMP to multilayer networks

2.4.1 Multilayer modularity

Despite it's problems, one crucial reason why maximizing modularity remains one of the few approaches for community detection in networks that has been extended in a principled way [156] to multilayer networks [114] with very fast, scalable, and easy to use software packages [104] available. We also call attention to the multilayer extension [48] of Infomap [198] and recent developments extending stochastic block models [SBMs] to multilayer networks (see [82, 212, 218] and, for a general update of developments in SBMs, [3]. In the case of a single

intralayer coupling parameter, the multilayer modularity formula developed by Mucha *et al.*[156] can be written in a similar form to the single layer version using the 'supra-adjacency' [3] representation :

$$Q(\gamma, \omega) = \sum_{i,j} \left(A_{ij} - \gamma P_{ij} + \omega C_{ij}\right) \delta(c_i, c_j) \tag{2.5}$$

where A_{ij}, P_{ij}, and C_{ij} represent the (possibly weighted) edges, null model, and interlayer connections, respectively, between the node-in-a-layer indexed by i and that indexed by j; and c_i indicates the community assignment. Each node in a layer be indexed by a single subscript, i or j.

While multilayer modularity provides a means for community detection in multilayer networks using many of the same heuristics and applying some of the same conventional wisdom developed for single-layer networks, the generalization admits at least one more parameter to control the contribution of interlayer connections to modularity relative to that from intralayer connections, e.g., the interlayer coupling ω in [156]. The same multilayer modularity framework can be applied generally to include multiple interlayer coupling parameters controlling the relative contributions of different parts of the multilayer structure, e.g., for data that is both temporal and multiplex. As such, multilayer modularity requires exploring a two-dimensional parameter space in its simplest setting, and higher dimensions in more general cases. For present purposes, we will here only explicitly consider the case of a single interlayer coupling parameter ω; but this does not put constraints on the coupling topology or relative values, only that there is some selected interlayer coupling tensor that is multiplied by ω. Meanwhile, the CHAMP approach we have presented can be naturally generalized to higher dimensions.

In the case where there is a single interlayer coupling parameter ω, each partition can be identified with a plane in the (γ, ω, Q) space. Likewise the domains of optimality for the optimal set of partitions can be envisioned as simply connected, convex polygons in the space bounding

[3] For example, in the 'supra-adjacency' representation of a simple multilayer network of multislice type where the same N nodes appear in each of L layers, one might order the indices so that $i \in \{1, \ldots, N\}$ corresponds to the first layer, $i \in \{N+1, \ldots, 2N\}$ corresponds to the second layer, and so on. To emphasize that the formulation of CHAMP is independent of the details of the multilayer network under study, we note here that the only distinction used presently is that A_{ij} encodes all of the edges, P_{ij} specifies the within-layer null model contributions, and C_{ij} describes the known interlayer connections. The key fact here is that P_{ij} and C_{ij} make distinct contributions to multilayer modularity, as controlled by two different parameters, γ and ω. As such, we need to extend CHAMP to simultaneously address both parameters. See Section 1.1.3 as well as [47, 114] for broader discussion about different notations and their advantages.

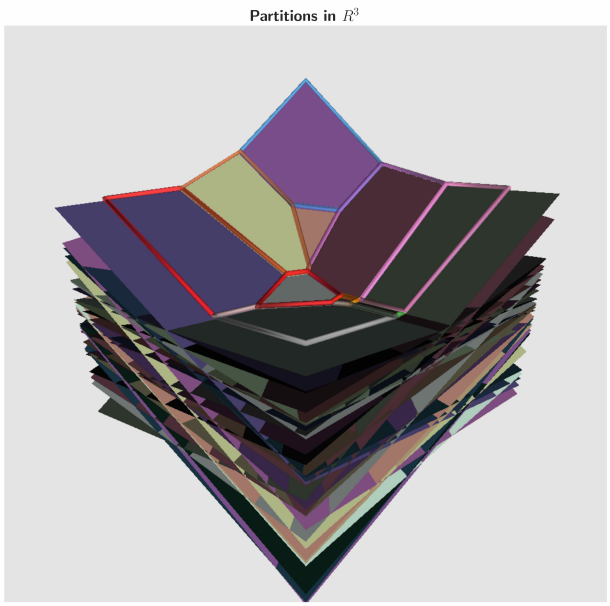

Figure 2.4: Visualization of CHAMP on multilayer networks. Each planes is identified with a single multilayer partition detected using the multilayer modularity framework with set (γ,ω). Note that the surface formed by the boundary of the convex hull is piece-wise, simply connected, convex polygons.

the intersection of the area above all planes as depicted in Figure 2.4. Several of our results in Section 2.5.2 depict this surface projected into the two dimensional, (γ, ω) plane.

2.4.2 Multilayer CHAMP and Qhull

Coupling the communities across layers is conceptually intuitive. Unfortunately, introduction of the additional parameter, ω makes the previous methods for parameter selection via visual inspection difficult to employ in practice and would seem to greatly complicate the challenge of selecting good values of the parameters. (See [17] for one approach to addressing this challenge.)

However, because the multilayer modularity function is linear in the parameters γ and ω, we can again apply the general approach of CHAMP, albeit now in a larger dimensional parameter space. For each partition σ, we again define the scalar quantities $\hat{A}_\sigma$ and $\hat{P}_\sigma$ to be the within-community sums over the adjacency matrix and null model, respectively, and now include a similar sum over the interlayer connections, $\hat{C}_\sigma$:

$$\hat{A}_\sigma = \sum_{i,j} A_{ij}\delta(c_{i\sigma}, c_{j\sigma})\,, \qquad \hat{P}_\sigma = \sum_{i,j} P_{ij}\delta(c_{i\sigma}, c_{j\sigma})\,, \qquad \hat{C}_\sigma = \sum_{i,j} C_{ij}\delta(c_{i\sigma}, c_{j\sigma})\,. \tag{2.6}$$

In this notation, the multilayer modularity of partition σ becomes simply

$$Q_\sigma(\gamma, \omega) = \hat{A}_\sigma - \gamma \hat{P}_\sigma + \omega \hat{C}_\sigma \,. \tag{2.7}$$

Thus, the partition σ is represented by the plane Q_σ in (γ, ω, Q). Analogous to the single-layer case, each point in the two-dimensional (γ, ω) parameter space admits an optimal Q_{σ^*}.

Given a set of partitions $\{\sigma\}$, CHAMP calculates the coefficients of the $Q_\sigma(\gamma, \omega)$ planes in Equation (2.7) and solves a convex hull problem to find the convex intersection of the half-spaces above these partition-representing planes (see Figure 2.4). That is, each partition is represented by a plane dividing (γ, ω, Q) in two, thereby defining a half-space. The intersection of the half-spaces above all of these planes is the convex space of $Q(\gamma, \omega)$ values greater or equal to all observed quality values, with the boundary specifying the maximum modularity surface of the set. In single-layer networks, we considered ordered $\gamma \geq 0$ and iteratively identified the next intersection and associated partition for increasing γ. In the presence of multiple parameter dimensions here, we instead apply the Qhull implementation [2] of Quickhull [13] to solve the dual convex hull problem. In practice, multiple partitions of the network can be identified in parallel, calculating and saving each set of $\hat{A}$, $\hat{P}$, and $\hat{C}$ coefficients. These coefficients defining the planes are then input into Qhull. CHAMP thereby prunes $\{\sigma\}$ to the subset admitted to the convex hull and identifies the convex polygonal domain in (γ, ω) where each partition is optimal (relative to $\{\sigma\}$).

We note that in practice the runtime for finding the pruned subset of admissible partitions and associated domains of optimality is typically insignificant compared to that of identifying the input set of partitions in the first place. In particular, computing the scalar coefficients of the linear subspace of each partition is a direct $O(M)$ calculation for M edges in the network. Meanwhile, the subsequent convex hull problem has no explicit dependence on the network size, depending instead on the number of partitions in the input set.

While we assume here that there is a single interlayer coupling parameter ω, we emphasize again that we do not restrict ourselves here to a particular form of the interlayer coupling, which might connect nearest-neighbor layers, all-to-all layers, connect only some nodes

in one layer to those in another, and might have multiple different weights along different interlayer edges. Rather, we only require here that there is some selected interlayer coupling tensor C that is multiplied by ω. Even more complicated interlayer couplings with multiple parameters (e.g., data that is both multiplex and temporal with the freedom to vary the relative weights between these couplings) can in principle be treated analogous to the above in the appropriate higher-dimensional space. With the notation $\vec{\gamma} = (\gamma, \omega)$ and $\hat{\mathbf{P}}_\sigma = (\hat{P}_\sigma, -\hat{C}_\sigma)$, we can write Equation (2.7) as $Q_\sigma(\vec{\gamma}) = \hat{A}_\sigma - \vec{\gamma} \cdot \hat{\mathbf{P}}_\sigma$, specifying linear subspaces of codimension one in higher-dimensional parameter spaces, given appropriate definitions of $\vec{\gamma}$ and $\hat{\mathbf{P}}_\sigma$. However, we do not go beyond two parameters (γ, ω) in our example results here. While this is possible in theory, in practice, the implementation of Qhull we have used does not suggests using 8 dimensions since the number of facets in the output scales by n, the number of halfspaces raised to $d/2$, the dimension [2]. We have not tested our algorithm for higher than the single interlayer coupling, ω case (2D halfspaces).

For convenience we have implemented and distributed a python package for running and visualizing both the single layer and multilayer CHAMP found at [245].

2.5 Applications of CHAMP

In this section we showcase CHAMP on a few real world datasets, both single-layer as well as multilayer networks, and discuss the interpretation of the results. In Section 2.5.1, we consider a network of NCAA Division I-A college football teams from the 2000 season [58, 71]. We then look at results of applying CHAMP to a Human Protein Reactome (Section 2.5.1) and a Caltech Facebook network [227] (Section 2.5.1). All three of these undirected networks are studied using the Newman-Girvan null model with a resolution parameter as in Equation (3.10). Then, in Section 2.5.2 we apply CHAMP to communities found using the multilayer generalization of modularity in the multilayer network of roll call similarities across time, where each layer is a different two-year Congress [156]. Finally, we explore the stability of these domains under many different runs and explore how the size of the CHAMP set changes as input size grows in Section 2.5.3

For each example, we input into CHAMP a set of partitions identified by the Louvain

heuristic [27], as implemented by [226] for our three single-layer examples and by GenLouvain [101] for our multilayer example. Because of the modest sizes of these example networks, we perform large numbers of runs of the heuristic (between 20,000 and 240,000, as indicated for each example). Each run of the heuristic is performed at a resolution parameter γ (including also a parameter ω in the multilayer example) selected uniformly from a preselected range of the parameter, as indicated for each example. Node indices were randomly permuted for each run to ensure different order of considering nodes in the heuristic, to allow for possibly different partitions to be found at identical parameters. CHAMP makes no requirement that so many partitions be generated, nor about the way in which those partitions were generated, assuming only that multiple partitions have been obtained by one means or another. CHAMP then prunes the input partitions down to the admissible subset; as such, the overall quality of the final subset of course depends on the input set. In practice, one's tolerance for the computational burden will be dictated by the cost of running the community detection heuristics employed on the network of interest. Once the input set of partitions is identified, CHAMP reduces each partition to its scalar coefficients—$\hat{A}_\sigma$, $\hat{P}_\sigma$, and for multilayer networks, $\hat{C}_\sigma$—and then prunes down to the admissible subset in a trivial additional computational cost relative to that already expended to obtain that input set.

Throughout this section, we typically assess the correspondence between identified partitions and the known ground truth using the metric, Adjusted Mutual Information (AMI), unless otherwise noted. See Section 1.2.3 for a more detailed discussion of this and other metrics.

2.5.1 CHAMP on single-layer networks

2000-2001 Division 1-A College Football Network

Figure 2.5A visualizes a computational scan of the γ resolution domain for the Division I-A college football network of 115 nodes representing teams and 613 (unweighted) edges representing that at least one game was played between two teams. Additionally, each team has a label identifying its athletic conference, a subgroup of teams that generally share a geographic region and compete for a conference championship. One would expect that a good partition of the network reflects the conference structure.

For input to CHAMP, we ran the Louvain heuristic [27, 226] 50,000 times on the network. The modularity and number of communities found for each run is plotted at the γ resolution parameter used, which were uniformly spaced on $\gamma \in [0, 6]$. We observe in particular the wide range of γ over which one finds 12-community partitions, but note that the range also includes results with other numbers of communities, with ambiguity about which partition is the better choice.

By considering each partition as a line over the full domain of γ as shown in Figure 2.5B, we find the set of lines that form the convex hull of all the modularity functions and the intervals in which each partition is optimal, indicated by the red step function in Figure 2.5A, with the steps at the transition values of γ indicated by blue triangles in Figure 2.5B. These 50,000 runs of the heuristic generated 384 unique partitions, with the average run time for each cycle of the Louvain heuristic was 0.02 s. After application of CHAMP, there were only 19 partitions in the pruned admissible subset associated with the original parameter search space ($\gamma \in [0, 6]$). Moreover, CHAMP identifies a wide γ-domain of optimality of the 12-community partition, running from $\gamma \doteq 1.45$ to just below 4.

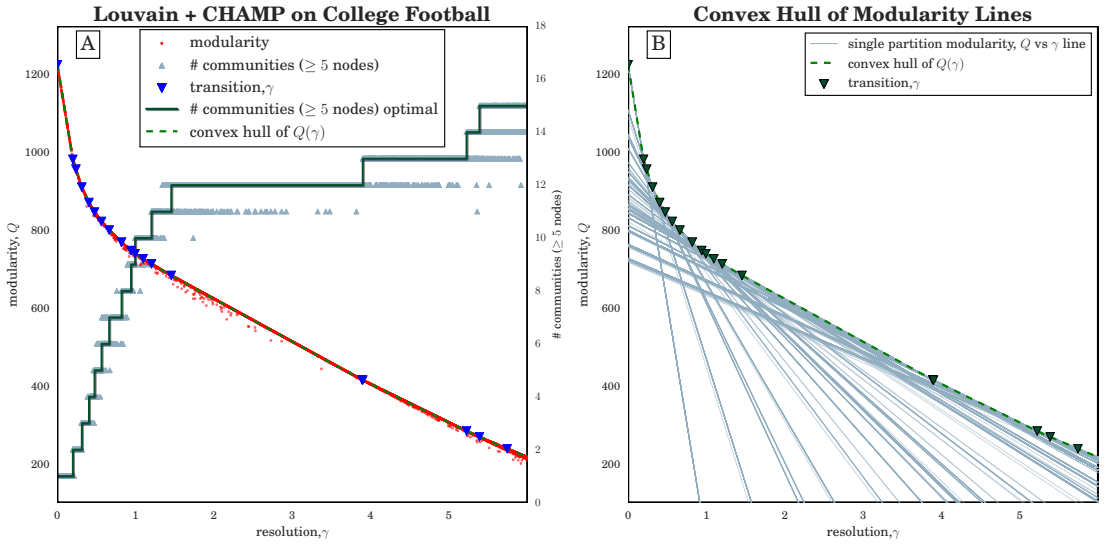

Figure 2.5: CHAMP on the NCAA Football Network. A) Modularity $Q(\gamma)$ given by Equation (3.10) versus resolution parameter γ for $50,000$ runs (10% of results displayed here) of the Louvain algorithm [27, 226] at different γ on the unweighted NCAA Division I-A (2000) college football network [58, 71]. Grey triangles indicate the number of communities that include ≥ 5 nodes in each run, while the green step function shows the number in the optimal partition in each domain; **B**) Graphical depiction of CHAMP algorithm (see Section 2.3). Each line indicates $Q_\sigma(\gamma)$ given by Equation (2.2) for a particular partition σ. Both panels show the convex hull of these lines as the dashed green piecewise-linear curve, with the transition values represented by downward triangles.

This 12-community partition, visualized in Figure 2.6B, aligns very closely with the conference labels of the teams as measured by Adjusted Mutual Information (AMI$\doteq 0.92$). Further increasing γ, we see this 12-community partition domain is followed immediately by a smaller (but still sizeable) domain of optimality for a 13-community partition. Note that while partitions with 11 communities are repeatedly returned by the heuristic, CHAMP indicates the corresponding domain of optimal γ to be quite small.

Figure 2.6A shows the pairwise adjusted mutual information (AMI) of the admissible partitions, as organized by their domains of optimality. That is, the large white blocks on the diagonal of the figure are $\mathrm{AMI} = 1$ agreement between each partition and itself. In particular, we observe that the 12-community partition (visualized in Figure 2.6B) is fairly similar to the next few partitions in increasing γ, suggesting stability of some main features as communities break up into smaller communities with increasing γ. At lower values of $\gamma < 1$, we see another possible grouping of domains with reasonable pairwise AMI to one another but who have much lower AMI with the partitions found at higher γ. These partitions could represent additional large-scale network structure.

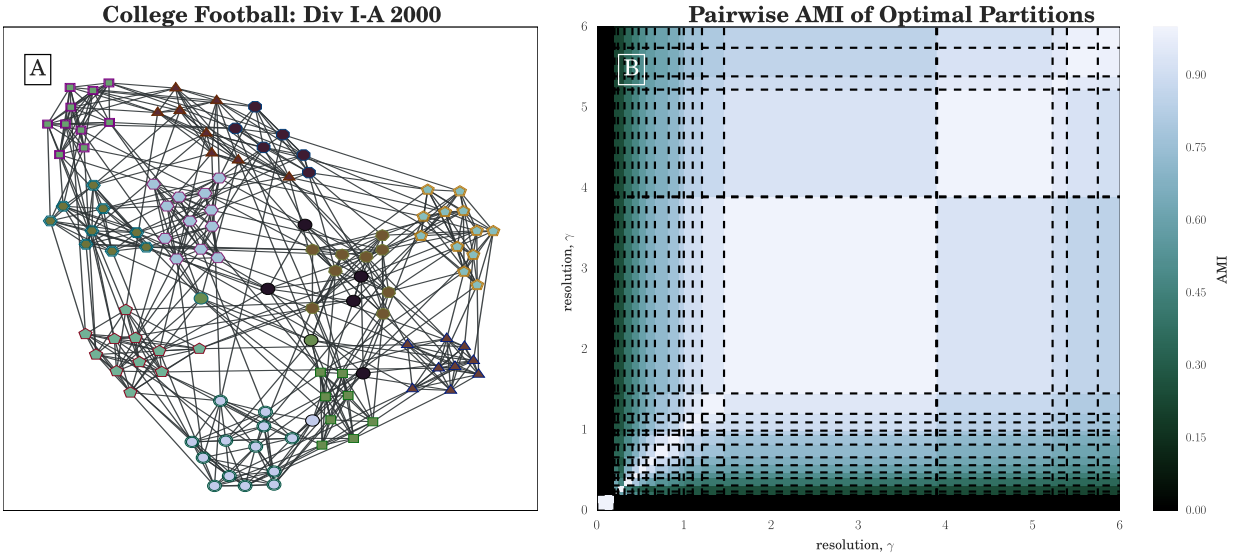

Figure 2.6: Similarity of CHAMP domains for NCAA Football. A) ForceAtlas2 [98] layout, created with [182], of the unweighted NCAA Division I-A (2000) college football network. Nodes are colored according to the dominant 12-community partition with the widest γ-domain $\gamma \in [1.45, 3.89]$, with node shapes and border indicating their conference labels; **B**) Pairwise adjusted mutual information (N=AMI) between all partitions in the admissible subset identified by CHAMP, arranged by their corresponding γ-domains of optimality. Dashed lines indicate the transition values of γ identified by CHAMP.

Human Protein Reactome

We employed CHAMP to map the domains of modularity optimization for a larger example: the undirected (single-layer) network representation of the Human Protein Reactome [105, 120], with 6327 nodes representing human proteins and 147,547 edges signifying common biological reactions. We ran the Louvain heuristic 20,000 times on this network with $\gamma \in [0, 4]$ uniformly spaced, generating 19,980 unique partitions. For this example, each run of Louvain required an average of 2.6 s, generating the input set of partitions in approximately 140 CPU hours. CHAMP pruned this input set of partitions down to 39 admissible partitions in the convex hull over the original parameter search space ($\gamma \in [0, 4]$). Similar to the figures of the previous example, Figure 2.7A shows the spread in the modularities and the numbers of communities identified across all instances of the heuristic, along with the domains of optimization and the number of communities for the admissible subset (see the red step function).

Contrasting Figures 2.5A and 2.7A, we observe in the latter that the red step function decreases with increasing γ at some points. Importantly, these decreases are not because of our choice to plot the number of communities that contain at least 5 nodes. The numbers of communities is provably monotonically non-decreasing with increasing resolution parameter in the special case where the null model $P_{ij} = \gamma$ is a constant independent of i and j [223, 224], but we are unaware of any similarly rigorous condition for the Newman-Girvan null model used in Equation (3.10). Nevertheless, one typically observes the number of communities to be non-decreasing with increasing γ, so the results here may indicate values of the resolution parameter near which additional runs of the heuristic might be more likely to identify higher quality partitions.

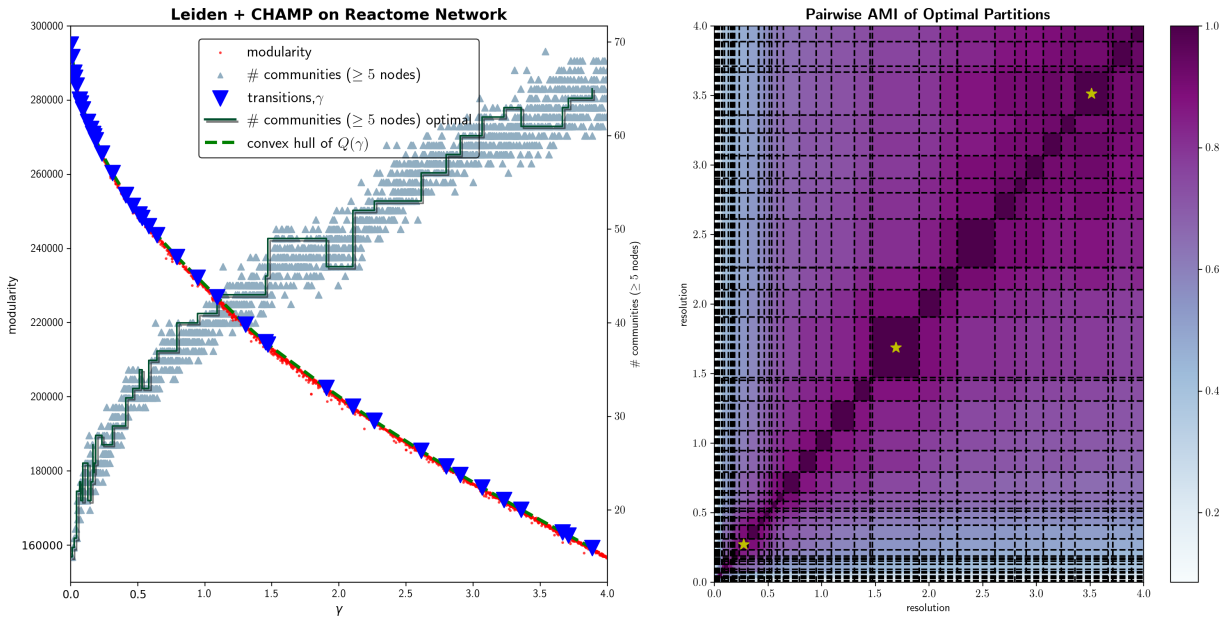

Figure 2.7: CHAMP on the human reactome. A) Modularity $Q(\gamma)$ given by Equation (3.10) v. resolution parameter γ for $20,000$ runs (25% of results shown) of Leiden [225] on the Human Protein Reactome network [105]. Small, grey triangles indicate the number of communities that include ≥ 5 nodes in each run, while the dark green step function shows the number in the optimal partition in each domain. The dashed green curve is the piecewise-linear modularity function for the optimal partitions, with the transition values marked by blue triangles; **B**) Pairwise AMI between all partitions in the admissible subset identified by CHAMP, arranged by their corresponding γ-domains of optimality. Yellow stars denote the domains shown in Figure 2.8.

The number of communities in the initial set of partitions is highly variable, even for small adjustments in γ, as shown by the yellow triangles in Figure 2.7A. It would be difficult to extract any range of stability from such a plot. However, when we consider the admissible subset of partitions, we see a few wide domains of optimality in the figure, the two most prominent being $\gamma \in [1.47, 1.91]$ and $\gamma \in [3.36, 3.67]$. Layouts of the network colored according to the partitions of these two broadest domains are shown in Figure 2.8. The pairwise AMI of the admissible partitions are shown in Figure 2.7B. Unlike the college football network, where pairwise AMI appears to indicate two well separated groups of highly similar partitions, the communities here appear to be diffusely similar throughout. Partitions of adjacent domains are fairly similar but there is no clear divide into groups of partitions.

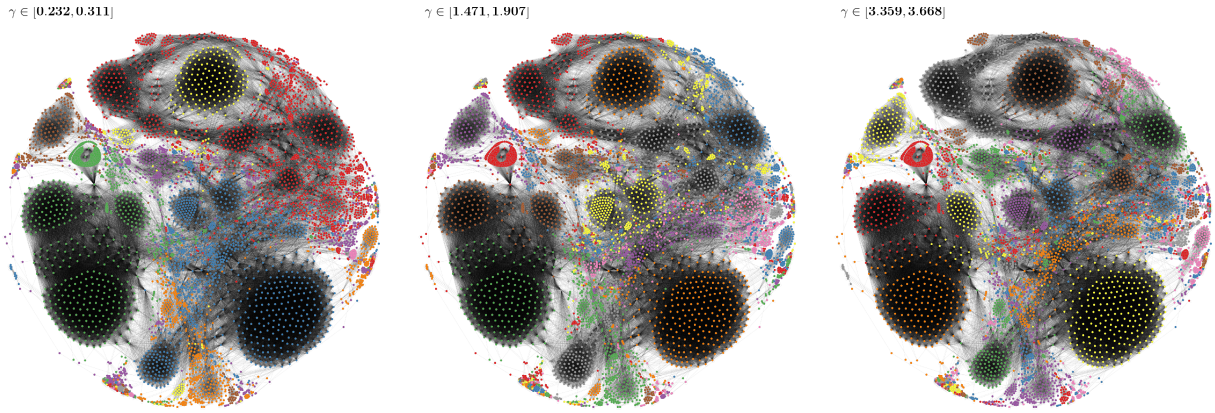

Figure 2.8: Visualization of Reactome communities. ForceAtlas2 layout [98], created with [182], of the Human Reactome Network, colored according to the partitions with the three widest γ-domains of optimization identified by CHAMP from $20,000$ runs of Leiden.

Caltech Facebook Network

As a final single-layer example, we considered the undirected network of Facebook friendships for students at Caltech in September of 2005 [227], the largest connected component of which includes 762 nodes representing Facebook users and 16,651 unweighted edges representing reciprocal friendships.

We used the Louvain algorithm $100,000$ times on $\gamma \in [0, 4]$ uniformly spaced, generating $91,080$ unique partitions. CHAMP pruned this set down to 51 partitions with associated γ-domains of optimality in the original parameter search space ($\gamma \in [0, 4]$). That is, the number of partitions in the pruned subset is 1785 times smaller than that in the set of unique partitions found by our Louvain runs that were input into CHAMP. Each run of Louvain on the Caltech Facebook network required around 0.8 s with all runs representing approximately 20 CPU hours. This output from CHAMP, visualized in Figure 2.9A, does not indicate the same wide domains of optimality for the community structures in this network as with the previous two examples. The pie-chart visualization within Figure 2.9A corresponds to one of the wider domains here narrowly straddling the default $\gamma = 1$ value. This community structure is reasonably well aligned with the House System at Caltech (see also the associated discussion in [227]). At higher values of γ, we expect that the scales of the communities will be subgroups within the Houses. We observe that some of the wider plateaus in the numbers of communities in the figure correspond to multiple different partitions with the same numbers of communities (note the transition values indicated by blue triangles).

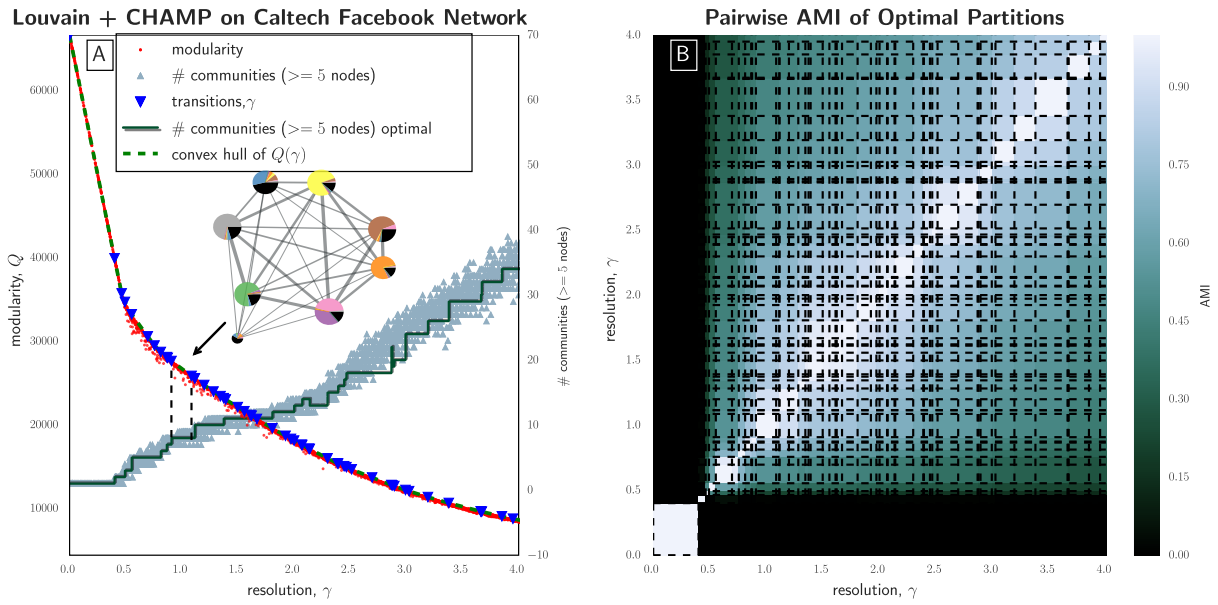

Figure 2.9: CHAMP applied to Caltech Facebook network. A) Modularity $Q(\gamma)$ v. γ for $100,000$ runs (5% of results shown) of Louvain [27, 226] on the Caltech Facebook network [227]. Orange triangles indicate the number of communities that include ≥ 5 nodes in each run, while the red step function shows the number in the optimal partition in each domain. The dashed green curve is the piecewise-linear modularity function for the optimal partitions, with the transition values marked by blue triangles. The condensed layout of communities (created with [182]) here visualizes the optimal partition found for $\gamma \in [0.908, 1.09]$, with each pie-chart corresponding to a community, fractionally colored according to the House membership of the nodes in the community. The AMI between this partition and House labels (including the missing label) is 0.513; **B**) Pairwise AMI between all partitions in the admissible subset identified by CHAMP, arranged by their corresponding γ-domains of optimality.

2.5.2 CHAMP ON MULIT-LAYER NETWORKS

U.S. SENATE ROLL CALL VOTING NETWORK

We demonstrate the use of CHAMP to explore the parameter space for a multilayer network using the roll-call-voting similarity network for the U.S. Senate from 1789 to 2008 (Congresses 1 to 110) as defined in [242] and studied with multilayer modularity in [155, 156]. This data represents the similarities of voting patterns within each two-year Congress between the 1884 distinct U.S. Senators who served across the first 110 Congresses. As in [155, 156], each two-year Congress starting in the early January following the biennial Congressional elections is represented as a layer, with interlayer connections only between the multiple appearances of each Senator when they appear in nearest-neighbor layers; as such, multilayer modularity directly handles additions and removals of Senators over time. Self-loops within each layer are zeroed out, since these only represent perfect agreement of a Senator with herself during the same

two-year period. This representation of the voting data is useful for describing legislative voting activity because the community structures typically group together Senators who vote similarly, providing relatively accessible and intuitive examples of communities that are related to the underlying political alignments as expressed by the Senators through voting, independent of their nominally declared party affiliations. The temporal extents of the communities found by multilayer modularity can then indicate different periods of stability in these political alignments.

We ran the GenLouvain [101] heuristic 240,000 times, on a 600-by-400 uniform grid over $[0.3, 2] \times [0, 2]$ in (γ, ω), generating 197,879 unique partitions of the network. Each run of GenLouvain required approximately 5 s for a total of 340 hours of CPU time. CHAMP pruned this set to 1447 partitions admissible in the convex hull of modularity. We note that there were 267 additional partitions with corresponding domains of optimality that were completely outside the selected parameter range $[0.3, 2] \times [0, 2]$. In Figure 2.10 we visualize the (γ, ω)-domains of optimality within this region of parameter space. In Figure 2.10A, a domain's color indicates the numbers of communities for its corresponding optimal partition, whereas in Figure 2.10B domain color indicates the average AMI between the corresponding partition and the neighboring optimal partitions (weighted by the lengths of borders between domains).

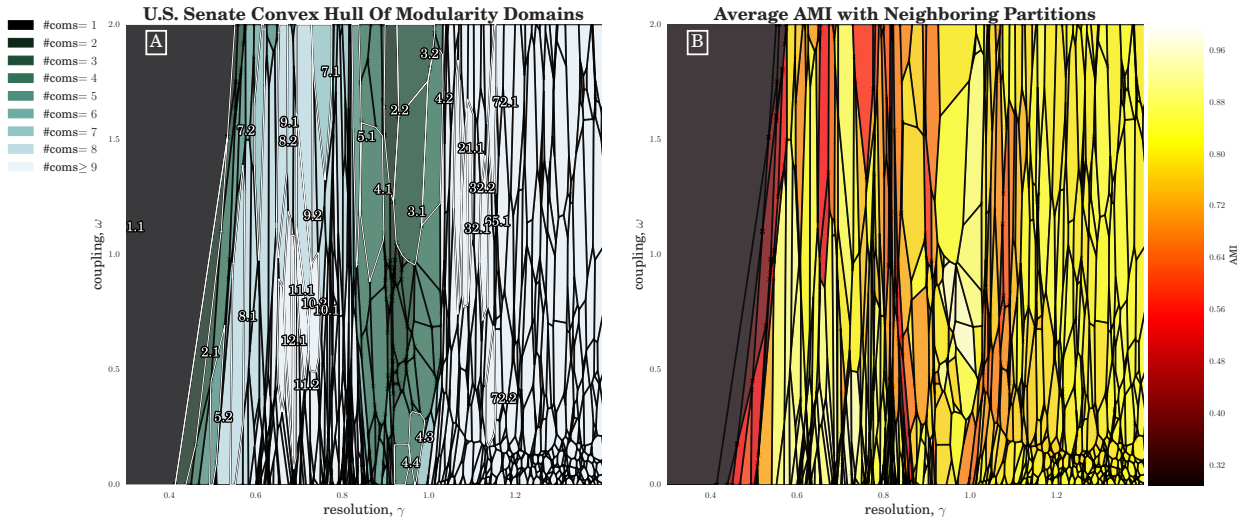

Figure 2.10: CHAMP on the US Senate network. A) Domains of optimization for the pruned set of partitions, colored by the number of communities within each partition. The set of partitions was generated from $240,000$ runs of GenLouvain [101] on a 600×400 uniform grid over $[0.3, 2] \times [0, 2]$ in (γ, ω). The largest partitions are labeled "$X.Y$" with X the number of communities with ≥ 5 nodes and Y the rank of the domain area (that is, in terms of size) for that given number of communities (e.g., "5.2" is the second-largest domain corresponding to 5-community partitions). The partitions of each labeled domain are visualized in Appendix 2.1; **B**) Weighted-average AMI of each partition with its neighboring domains' partitions, weighted by the length of the borders between neighboring domains.

The trivial 1-community partition dominates the left of the panels in Figure 2.10 at small γ. Increasing γ outside of this domain, most of the (non-trivial) domains here appear to be relatively long in the ω direction and much narrower in γ. Interestingly, we observe a range of γ from roughly 0.8 to just above 1 where the domains visually widen in the γ direction while also corresponding to a smaller number of communities than partitions below $\gamma \approx 0.8$. Near $\omega = 1$, the widths in γ of the domains appear larger than those at smaller ω, suggesting perhaps that the stability of identified communities is being enhanced by coupling between the layers. As γ increases only slightly past 1, the number of communities in each partition rapidly increases, with the majority of partitions past $\gamma = 1.2$ having over 100 communities. At the lower right corner we see the domains are small and highly fragmented in both the γ and ω directions.

We also aim to identify parameter regions corresponding to similar partitions. For single-layer networks, we directly visualized the whole set of pairwise AMI's ordered by γ. Given two parameters here, we calculate the weighted average AMI of each partition with its neighbors, with weight proportional to the length of the border with the neighboring domain along which the two partitions have the same value of multilayer modularity. The resulting neighbor-averaged AMI of each partition is shown by color in Figure 2.10B. We again observe at least three distinct regions of high pairwise similarity, separated by much lower neighbor-averaged AMI, aligned with the different regions in Figure 2.10A discussed above: (1) the region below $\gamma \approx 0.8$; (2) the region just below $\gamma = 1$, with particularly high neighbor-averaged AMI for $\omega \in [0.6, 0.9]$; and (3) the many-community partitions for $\gamma > 1.2$.

Indeed, we see a shift in the types of partitions with increasing γ across this $\gamma \approx 0.8$ transition boundary. The qualitative difference in community structure between these regions is demonstrated in Figure 2.11, highlighting in Figure 2.10A the two partitions labeled 5.1 (Figure 2.11A) and 8.1 (Figure 2.11C). Recall, that these are the partitions with the largest domains of optimality with 5 and with 8 communities, respectively. Most of the Congress layers in the 8-community partition include only a single community label per Congress (see Figure 2.11D). In contrast, the 5-community partition divides the Senators both across time and within each Congress, typically into 2 communities in each Congress. These intralayer divisions that extend across time are additionally highlighted by the individual Senator layout in Figure 2.11A showing distinct branches, because the Senators have been sorted here first by community label and then,

within each community by time. Layouts for the other domains labeled in white in Figure 2.10A further demonstrate qualitatively similar patterns, as shown in Appendix 2.1.

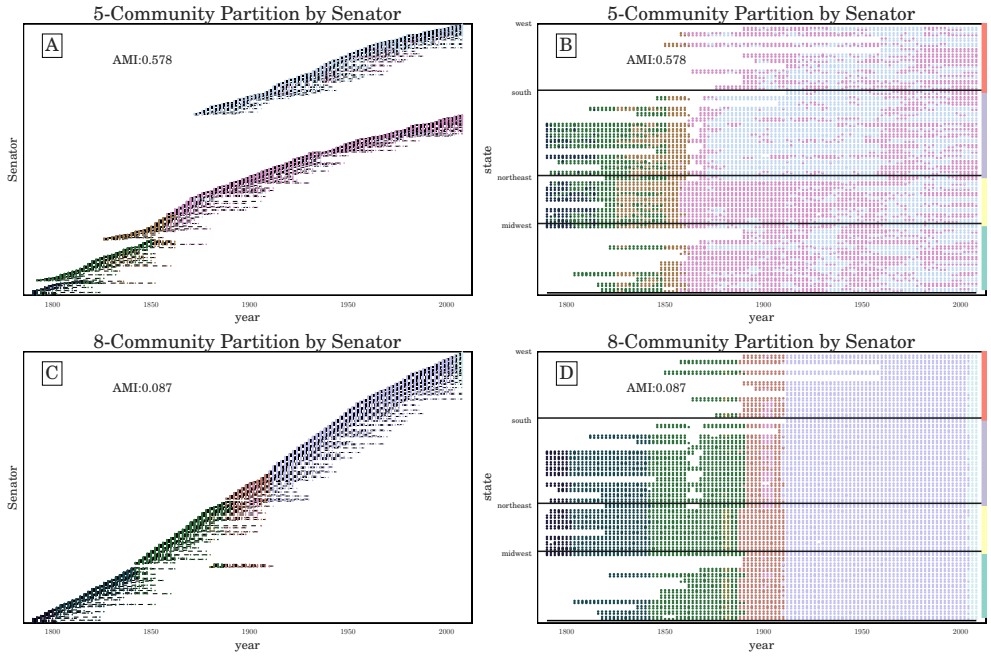

Figure 2.11: Time-varying community structure for the U.S. Senate from 1789 to 2008 according to the (**A**,**B**) 5-community and (**C**,**D**) 8-community partitions with widest domains of optimality (see labels 5.1 and 8.1 in Figure 2.10A); (**A**,**C**) The vertical axis indicates individual Senators, sorted by community label and time. The AMI reported here is the average over layers (Congresses) of the AMIs in each layer between the identified communities in that layer and political party labels. (This layer-averaged AMI is shown for all partitions in the convex hull over the originally searched parameter range in Figure 2.12.) (**B**,**D**) The vertical axis indicates the state of a Senator, sorted according to geographic region, and the horizontal axis represents time (two-year Congresses).

In Figure 2.12, we again visualize the domains of optimality in the (γ, ω) parameter space, now color-coded by the layer-averaged AMI between each partition and the known political affiliations of Senators. Specifically, we compute for each layer the AMI between the community labels $\{c_{i\sigma}\}$ and the Senators' party affiliations, and then we average the AMIs across layers (i.e., across Congresses). The central, broadest domains have the highest AMI with the mostly 2-party system seen throughout the different session of Congress, consistent with our observations above. For the most part, partitions with neighboring domains have fairly similar structure within the layers. There are a few places in the Figure where a darker border represents a transition in the qualitative features of the community structure, such as the transition region around $\gamma \approx 0.8$ discussed above.

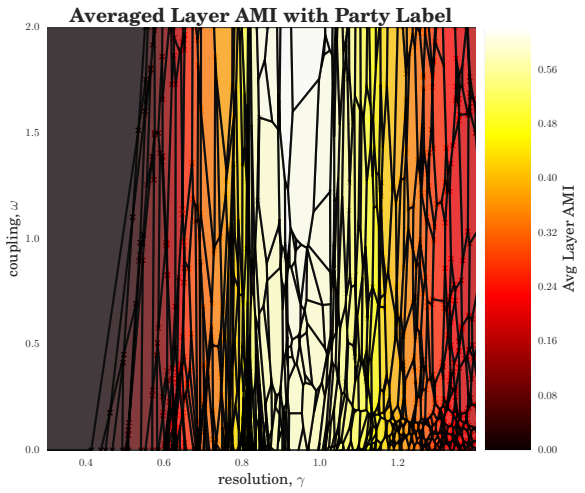

Figure 2.12: The domains of optimality for the time-varying U.S. Senate roll-call similarity network (as in Figure 2.10), colored by the layer-averaged AMI between the political-party affiliations of Senators and the community labels $\{c_{i\sigma}\}$ for that layer.

2.5.3 Stability of CHAMP domains

Although the set of partitions that are identified by CHAMP are maximizers of modularity with respect to the starting ensemble of partitions, for most networks of reasonable size, it is quite probable that there will be optimal partitions that were not included in the original set. The CHAMP approach is beholden to the ability of the community detection heuristic it is used with to identify high modularity partitions (*e.g.* Louvain, Leiden, etc.). One question that naturally arises is to what extent does the size of the CHAMP set depend on the size of the starting input and how consistent are the domains that are identified? That is, how large does my input set need to be to be reasonably certain one has captured enough partitions to well approximate the ideal CHAMP set. Although it is always possible that there is a single partition one has not yet uncovered that could dominate many of the partitions already in the CHAMP set, we find in practice that the domains are relatively stable over repeated sweeps of our algorithm, and that the number of partitions added to the dominant set typically decays logarithmically with the size of the input set, indicating that past a certain point, continuing to search for additional partitions yields diminishing gains.

In Figure 2.14, we explore the stability of the CHAMP set for the reactome network [105, 120] by varying the number of runs of Leiden used as input for computing the CHAMP set.

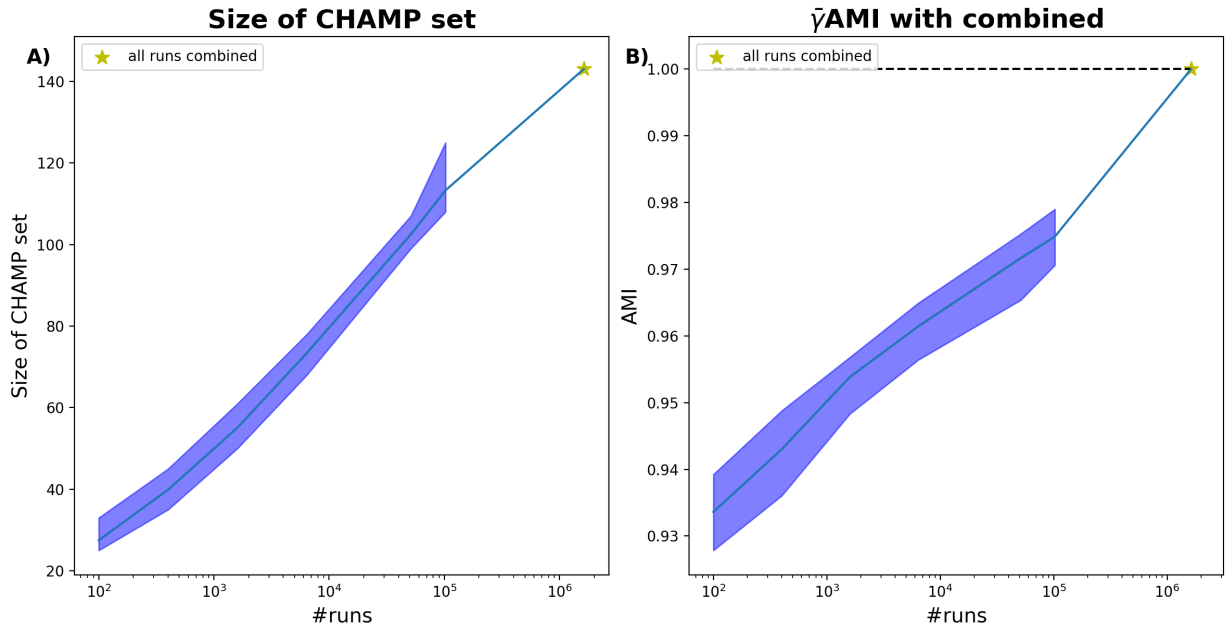

Figure 2.13: Size and consistency of the CHAMP sets for reactome network[105, 120]. A) The total size of the CHAMP set for each partition ensemble of r runs, averaged over 10 trials. Size of baseline set of all partitions indicated by gold star. **B)** The average AMI between the CHAMP set for each partition ensemble of r runs and the baseline ensemble, weighted by the size of the domain, and averaged over 10 trials (see Equation 2.8). Baseline partition has average AMI of 1 by construction.

Specifically, for each value of $r \in \{100, 400, 1600, 6400, 51200, 102400\}$, we create 10 independent ensembles of partition by running Leiden r times across $\gamma \in [0, 4]$, uniformly distributed. We compare the CHAMP set for each of these individual ensembles to the CHAMP set of the union of all ensembles (which is a total of $10 \times \sum r = 1621000$ input partitions), which we refer to as the baseline partition ensemble. In Figure 2.13.B, we show that the number of unique partitions in the CHAMP set grows logarithmically with the size of the input ensembles (the point representing baseline partition set is denoted by a star). In Figure 2.14.B, we calculate the average AMI of the partitions in each individual CHAMP set with the CHAMP set for the baseline, weighted by the overlap of the domains of the each partition:

$$\bar{\gamma}\text{AMI} = \frac{1}{\bar{\gamma}} \sum_{c_i \in \Sigma^*_{\text{baseline}}} \sum_{c_j \in \Sigma^*_{r_k}} AMI(c_i, c_j) * \bar{\gamma}_{c_i} \cap \bar{\gamma}_{c_j} \ , \tag{2.8}$$

where $\Sigma^*_{\text{baseline}}$ and $\Sigma^*_{r_k}$ represent the CHAMP set of partitions for the baseline partition set and the k^{th} set of r runs respectively, $\bar{\gamma}_{c_i}$ represents the domain for the c_i partition, and $\bar{\gamma}$ represents the length of the whole γ domain over which all runs occurred (in this case 4). This gives us a way

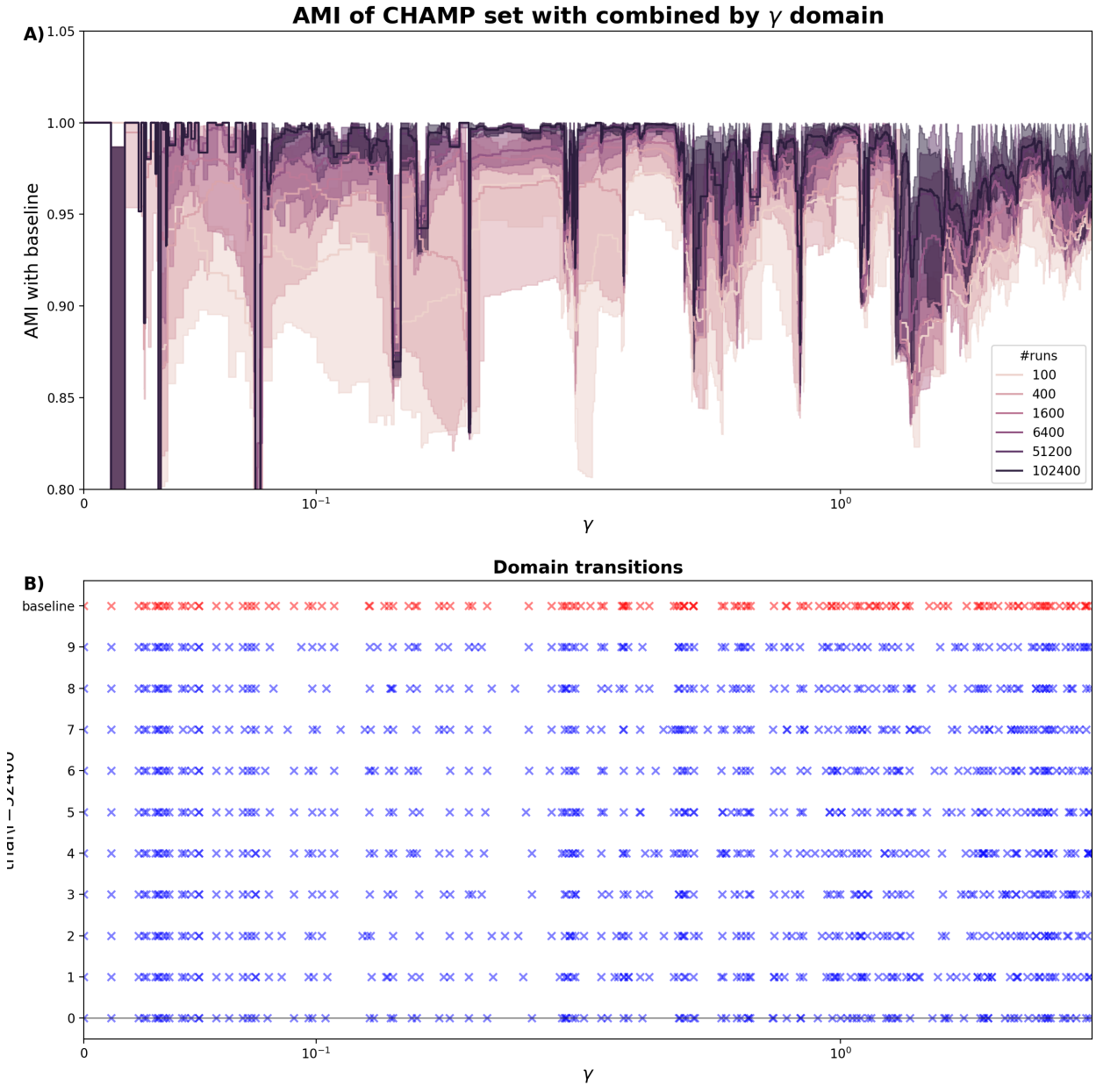

Figure 2.14: Exploring the stability of the CHAMP sets for reactome network[105, 120]. A) We compute the AMI for the intersection of each domain between the partitions for the baseline set of all partitions, and each individual set with r runs. We have averaged the individual step functions over 10 independent trails, each with r runs. **B)** Location of transitions between dominant domains for each of the 10 trials with 102400 runs of Leiden, uniformly spaced across $\gamma = [0, 4]$, as well as the transitions for the baseline combined set (1621000 total runs) shown in red.

to compare partitions in the CHAMP set between different runs on the basis of the overlap between their domains of dominance. In Figure 2.14.A, we plot the AMI for all non-empty intersections $\bar{\gamma}_i \cap \bar{\gamma}_j$ between the CHAMP set for the baseline partitions and each of set of r runs, averaged over the 10 trials with 95% confidence intervals shaded. Equation 2.8 gives us the average AMI value for each of these curves.

In Figure 2.13.A we see that the size of the CHAMP set continues to grow logarithmically with the size of the input set. Figure 2.13.B shows that although increasing the number of runs better approximates the baseline CHAMP set, even a small set of runs well approximates the domains of the much larger baseline set. We see that even for only 100 runs, we achieve an average $\bar{\gamma}$AMI $\geq .93$ with the baseline CHAMP set. This continues to increase logarithmically with the size of the input set, at least with the set sizes considered here. The baseline CHAMP set is well approximated by each of the sets of 102400, despite each of these having less than one tenth of the number of total partitions in the baseline. Thus while the size of the CHAMP set appears to grow logarithmically with the size of the input set, we see that newly added partitions are either very similar to already discovered partitions or have increasingly small domains of dominance. In Figure 2.14.A, we show how the concordance between each of the smaller subsets of runs and the baseline set varies across the γ domain. We see that there are regions where the CHAMP set is more consistently identified (for example around $\gamma = .4$ or $\gamma = 1.0$). In Figure 2.14.B, we show how the domains of dominance align across the various runs at $r = 102400$, as well as the baseline. We see several locations where the domains identified are quite consistent across all of the runs such as at $\gamma = .4$. These results demonstrate that CHAMP is a relatively effect tool for exploring the space of partitions even without a near exhaustive set of input partitions.

2.6 CHAMP Discussion

There are a number of features of CHAMP that make it a useful tool for community detection, as we have demonstrated by way of our real world examples. By eliminating partitions that are non-admissible to the convex hull, CHAMP can greatly reduce the number of partitions remaining for consideration. By assessing the sizes of the domains of optimality of the partitions in the pruned admissible subset, and through direct pairwise comparisons of partitions in the

admissible subset, CHAMP provides a framework for identifying stable parameter domains that signal robust community structures in the network.

The set of input partitions can be obtained as a result of a community-detection method across a range of parameter choices (as we explored here) or from the comparison of different community-detection methods. Ideally the input set contains near-optimal partitions with relevance for the application at hand. Because each partition is allowed to compete across the whole space of resolution and coupling parameters, CHAMP can surmount some of the pathologies associated with modularity-based community detection heuristics. For example, CHAMP has uncovered several cases where there is a parameter range over which Louvain consistently identifies suboptimal partitions compared to partitions that Louvain itself identifies at other parameter values. In our study of the Human Protein Reactome network (see section 2.5.1), we have seen that the stochasticity over multiple runs of the heuristic makes finding a plateau in the number of communities challenging; nevertheless, CHAMP is able to identify regions where a single partition is intrinsically stable, regardless of how frequently a particular detection algorithm uncovers such a partition. By identifying a manageable-sized and organized subset of admissible partitions with CHAMP, one can then apply a pairwise measure of similarity such as AMI to adjacent partitions to identify shifts in the landscape of optimal community structure.

We in no way claim that CHAMP resolves all of the problems with modularity-based methods
(see, e.g., the discussion in [64]). And CHAMP is certainly not the only way to try to process different results across various resolution parameters (see again the Introduction). However, by taking advantage of the underlying properties of modularity, including the fact that each partition defines a linear function for Q in terms of the resolution and interlayer coupling parameters, CHAMP provides a principled method built directly on the definition of modularity to make better sense of the parameter space when modularity methods are employed. In particular, many of the various other proposed approaches assess each partition at the particular parameter value input into the community detection heuristic that found the partition, that is treating each partition as a single point in (γ, Q). In contrast, CHAMP returns to the underlying definition of modularity with a resolution parameter to recognize that each partition here is more

completely represented as a line in (γ, Q) [in the multilayer case, as a plane in (γ, ω, Q)]. The single point is on that line but does not completely explore the potential of that partition to compete against the other identified partitions. By using the full linear subspace associated with each partition, CHAMP prunes away the vast majority of partitions in practice.

Importantly, CHAMP itself is not a method for partitioning a network, and as such its ability to highlight partitions is limited by the set of partitions given as input to the algorithm. Given the many available heuristics, the computational complexity of maximizing modularity [28], and the potentially large number of near-optimal partitions [76], it is possible that interesting and important community features may be missing from the provided input set. CHAMP as developed here is restricted to processing hard partitions of nodes into community labels, whereas overlapping communities and background nodes (those not belonging to any community) can be important for some applications. One may also reasonably worry about the potential value of partitions in the input set that are near-optimal over a wide domain of the parameters but yet never achieve admission to the convex hull itself and are thus discarded by the algorithm.

With the introduction of CHAMP presented here, we have left open many other possible uses of this general approach that may be worth exploring. Although we apply Louvain to discover partitions, CHAMP is agnostic to the detection method used to generate the set of partitions. The partitions input into CHAMP do not even need to be generated by modularity-maximizing heuristics; for example, one may also include new partitions as generated by ensemble learning [171] or consensus clustering [17, 103, 124]. By comparing the results between sets of partitions generated by different methods, CHAMP might be useful as an additional method for making comparisons between these methods.

In Section 2.5.3 we have shown how the identified domains are consistent from run to run and that it takes relatively few runs to well approximate the final CHAMP set obtained after many additional trials. While the size of the CHAMP set grows with the number of domains as input, the larger domains tend to remain relatively fixed, with the identity of newly added partitions fairly consistent with those that they replace. The number of initial partitions needed to get a good mapping of the parameter space undoubtedly depends on the structure of the network and the computational heuristics used. It may also be possible to use a variant of CHAMP to

iteratively steer the parameters at which additional partitions might be sought.

Of course, even with a resolution parameter, modularity may not be a good measure for what constitutes a good "community" in some networks, and one could investigate whether other quality functions with parameters might be explored with an analogous approach. Even within the consideration of modularity, it would be interesting to generalize the approach of CHAMP to exploring different scales as resolved with different self-loop weights as proposed in [7] (see also [78] for an application of this approach). Unlike the resolution and coupling parameters used here, changing the self-loop weight makes a nonlinear change to modularity. Nevertheless, we believe it may be possible to extend CHAMP to the self-loop method for resolving different scales. It would also be useful to extend CHAMP to methods for community structures with overlap and with background nodes.

In further developing CHAMP, it is important to recognize the inability of many community-detection algorithms to assess the reliability of identified communities versus apparent structures arising in random network models. The particular value of modularity, for example, does not immediately indicate whether an identified partition is significant; in fact, the modularities of many classes of random networks such as trees of fixed degree can be quite high in the asymptotic limit [10, 49]. Thus, it may be interesting to use CHAMP to further explore and characterize the domains of optimization for partitions of such random networks, to determine the extent to which leveraging such partition stability information can address questions about detected structures and random noise.

In summary, we have presented the CHAMP algorithm as a post-processing tool for pruning a set of network partitions down to the admissible subset in the convex hull that optimizes modularity at different parameters. We have demonstrated the utility of CHAMP on various single-layer networks and on a multilayer network, identifying partitions and their associated domains of optimality in the parameter space. Further research may focus on how the sizes of these domains and the comparisons between domains can be best used to ascertain confidence in identified community structures, to explore subgraphs of a network, and to further process the admissible subset for consensus clustering, as well as other uses of the pruned subset identified by CHAMP.

CHAPTER 3: MULTILAYER MODULARITY BELIEF PROPAGATION

In Chapter 3 we present an alternative approach to maximizing modularity directly on a network. Instead, we apply the tools of statistical physics to identify how strongly each node is associated with a possible community and whether or not community structure is present in the network. We begin by providing background on the general algorithmic approach we have employed here, known as "belief propagation". Then we detail the specific belief propagation approach to modularity maximization developed by Zhang and Moore [259]. Our main contribution in this chapter is the extension of the belief propagation algorithm to multilayer networks, which we refer to as *multimodbp*. As part of this extension, we have introduce a resolution parameter, γ, and we show how this can improve performance of the algorithm, especially in the context of single-layer networks. In Section 3.3.1 we demonstrate the improved performance of our method on several synthetic models of communities in single layer networks as well as two real world networks. Then, in Section 3.3.2, we showcase our method on the Dynamic Stochastic Block Model, a synthetic model of multilayer communities, as well as on a synthetic model of mesoscale communities for different multilayer topologies. We also apply our approach on two real world datasets. Through these examples, we demonstrate how the application of multilayer modularity belief propagation can provide additional information about the structure of multilayer networks and suggest some practical considerations in applying the tool. Finally, we close this chapter with a discussion of our contribution, and additional details concerning the implementation of *multimodb* Section 3.4.

3.1 Introduction to belief propagation

The belief propagation approach (also known as the cavity method in physics, and sum-product message passing) is a widely used algorithm for calculating the marginals of a joint distribution where the variables can be expressed in a graphical model. One general way to

represent such a model is through a factor graph[1] for example seen in Figure 3.1. A factor graph is a bipartite network representation of our model with two classes of nodes : variable nodes, $\{S_i\} = \mathcal{V}$ and factor nodes, $\{f_a = f(\subset \mathcal{V}\} = \mathcal{F}$. Each variable node, $S_i \in \sigma$ can occupy one of several states in the state space (which could be discrete or continuous), while each factor can take one or more variables as inputs. We write the joint distribution of our model as the product of all the factor nodes as follows

$$p(\{S_i\}) = \frac{1}{\mathcal{Z}} \prod_a f_a(\{S_i\}_{i \in \partial f_a}) \,, \tag{3.1}$$

where ∂f_j represents the neighbors of f_j in the factor graph (*i.e.* the subset of the variable nodes that f_j depends on). We note that the functions represented by the factor nodes do not themselves have to be probability densities as we include a normalization constant

$$\mathcal{Z} = \sum_{\{S_i\}} \prod_j f_a(\{S_i\}_{i \in \partial a}) \,, \tag{3.2}$$

where the sum is over all possible combinations of values of the input variables, which grows exponentially in the number of variables in our model. As such, while Eq 3.1 is rather simple to write, calculating $\mathcal{Z}$, known as the partition function, involves summing over all possible combinations of variables in the state space which grows exponentially in the number of variable nodes. Factors graphs are quite flexible in that relationships of arbitrary order can be represented by different factors (in the example in Figure 3.1 we have shown a maximum of order 3). There are several tools are available to compute the marginals of Eq 3.1, including Markov Chain Monte Carlo sampling, Gibb's sampling, the class of algorithms known as belief propagation. However, belief propagation offers several unique advantages including computational efficiency and the tractability of asymptotic analysis [147].

Given our set of variable nodes and factor nodes, we can derive the rules of the belief propagation algorithm by assuming that our network is a tree and computing the marginals at

[1]There are several other ways to represent the graphical models for which belief propagation can be applied, most notably bayesian networks and markov random fields. Any model represented by one of these can be converted to an equivalent factor graph (see [253] for procedure) and it is a matter of convenience which representation to use. The form of the belief propagation equations will look different depending on the representation used.

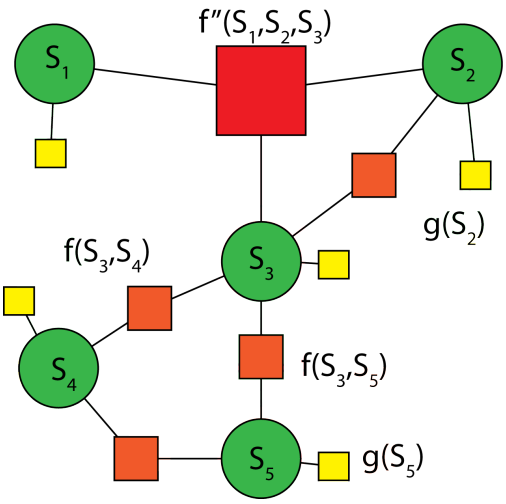

Figure 3.1: Example representation of a factor graph. Each node is of one of two classes: variables represented by green circles or factors, represented by colored squares. Each variable node can occupy one of several states. Each factors node one more more of the variables as input to determine its value. The overall probability of a given state of the model is the product over all of the factor nodes.

each node (with all other marginals held fixed). In the case that our network is a tree, the belief propagation updates are tantamount to writing down the factorized joint distribution and give exact results [175]. Remarkably, the belief propagation algorithm also works well in the case of graphs with loops as long as the graph meets certain, fairly non-restrictive conditions [147, 159, 178, 253, 254, 256].[2] Because factor graphs are bipartite, we will have two different rules for updating messages: one for messages from the variable nodes to the factors, and another for sending messages from the factors to the variable nodes. These update rules have the following recursive form:

$$\chi_{S_j}^{j\to a} = \frac{1}{\mathcal{Z}^{j\to a}} g_j(S_j) \prod_{b\in\partial j\setminus a} \psi_{S_j}^{b\to j} \tag{3.3}$$

$$\psi_{S_i}^{a\to i} = \frac{1}{\mathcal{Z}^{a\to i}} \sum_{\{S_j\}_{j\in\partial a\setminus i}} f_a(\{S_j\}_{j\in\partial a}) \prod_{j\in\partial a\setminus i} \chi_{S_j}^{j\to a}\,, \tag{3.4}$$

where $\chi_{s_j}^{j\to a}$ is the message that variable node j sends to factor node a, and $\psi_{s_i}^{a\to i}$ is the message that variable node i receives concerning its own state from factor node a. If we restrict our factor graph to allow only pairwise interactions between the variable nodes (*i.e.* each f_a only depends on 2 variable nodes and thus only has two neighbors), we can simplify the belief propagation

[2] They should be locally tree-like, with correlations between nodes decaying $\mathcal{O}(\log N)$.

equations by substituting the definition of $\psi_{s_i}^{a\to i}$ where it appears in the formula for $\chi_{s_j}^{j\to a}$:

$$\begin{aligned}\chi_{s_j}^{j\to a} &= \\ &= \frac{1}{\mathcal{Z}^{j\to a}} g_j(S_j) \prod_{b\in\partial j\setminus a} \psi_{S_j}^{b\to j} \\ &= \frac{1}{\mathcal{Z}^{j\to a}} g_j(S_j) \prod_{b\in\partial j\setminus a} \sum_{\{S_k\}_{k\in\partial a\setminus j}} f_b(\{S_k\}_{k\in\partial b}) \prod_{k\in\partial b\setminus j} \chi_{s_k}^{k\to b}\end{aligned} \tag{3.5}$$

Note that now, $\partial a \setminus j$ is only the single node that is connected by each of the factors. Thus we can switch from $j \to a$ to $j \to i$ because each factor node defines a pair of variable nodes. We also use ψ to denote the single set of messages in this case

$$\psi_{s_j}^{j\to i} = \frac{1}{\mathcal{Z}^{j\to i}} g_j(S_j) \prod_{k\in\partial j\setminus i} \sum_{\{S_k\}} f(S_k, S_j) \psi_{S_k}^{k\to j} \ . \tag{3.6}$$

If our model is a tree, we can initialize the states on leaves of the tree, and calculate the beliefs proceeding up the root to obtain the marginals for each node exactly. In the case where there are loops in the graph, we can initialize the nodes beliefs randomly[3] and then apply Eq 3.6 iteratively in a random sequential order to all of the nodes until the beliefs have converged [256]. We then use the beliefs to calculate the estimate of the marginal each node, $\hat{p}(S_j)$ by taking the product of all incoming beliefs to node j:

$$\hat{p}(S_j) = \frac{1}{\mathcal{Z}^{j}} g_j(S_j) \prod_{k\in\partial j} \sum_{\{S_k\}} f(S_k, S_j) \psi_{S_k}^{k\to j} \ , \tag{3.7}$$

where we note that the product is over all incoming beliefs. One can think of the BP estimate of the marginals as the belief that a node sends to itself about it's state. Belief propagation also gives us an estimation to the value of $\mathcal{Z}$, the partition function, which is quite useful in determining the statistical properties of our model. We can use our belief propagation approximation to estimate our partition function with the *Bethe free energy* approximation:

$$f^{bethe} = -\frac{1}{N} \ln \mathcal{Z} = \sum_i \log \mathcal{Z}^i - \sum_{ij} \log \mathcal{Z}^{ij} \tag{3.8}$$

[3]In this chapter we initialize them to a random perturbation about the factorized form: $\frac{1}{q} + \epsilon$

where $Z^{ij} = \sum_{\{S_i,S_j\}} f(S_i,S_j)\psi_{S_i}^{i\to j}\psi_{S_j}^{j\to i}$ is the partial partition for the belief propagation estimation of the pairwise marginals. One can show that any fixed point for the belief propagation equations will also be a local stationary point for Eq 3.8 [253, 254]. As we will see in the next section we can write down a graphical model for modularity on a network using Eq 3.1. This allows for a different perspective on optimizing modularity with some unique advantages. Later we will extend this approach to the multilayer formulation for modularity (see Sections 1.2.1 and 2.4.1 for more complete background to multilayer modularity).

3.2 Belief propagation approach to modularity

In general, maximizing modularity over the combinatorially large space of possible partitions is NP-hard [28]. Several fast and efficient algorithms exist for locally optimizing modularity, including Louvain [27] and the *GenLouvain* [104] extension for optimizing multilayer modularity. One of the main problems with optimizing modularity as a means of community detection is that partitions of high modularity often exist even in randomly generated networks without underlying structure (see for example [10, 49]). Zhang and Moore [259] were able to surmount several of the issues with modularity-based methods by treating modularity optimization in terms of the statistical physics of the spin-glass system with Hamiltonian $\mathcal{H} = -mQ(\{c_i\})$, where $\{c_i\} = [c_1,\ldots,c_N]$ with $c_i \in \{1,...,q\}$ indicating the assignment of node i (of N) to one of q communities. As such, the distribution of states of the system is given by the Boltzmann distribution

$$P(\{c_i\}) \propto e^{-\beta\mathcal{H}(\{c_i\})} \tag{3.9}$$

where β represents the nondimensional inverse temperature that sets the sharpness of the energy landscape. Maximizing the *joint* distribution $P(\{c_i\})$ is then equivalent to globally optimizing modularity and identifying the ground state of the system. Instead of searching for a global modularity maximum, Zhang and Moore attempt to solve for the marginals of each node, $P(c_i = q)$, in the finite temperature regime. By looking for a "consensus of good partitions" rather than seeking a single "best" partition, the algorithm converges to non-trivial structures above a certain temperature only if there is broad underlying structure within the network. If it exists, this parameter regime where belief propagation converges to non-trivial structure is called the

retrieval phase. In particular, Zhang and Moore demonstrated that the algorithm's convergence properties distinguish between synthetic networks with or without known underlying structure, even when the nominal modularity values of the identified partitions are quite similar. In this sense, a belief propagation approach is able to detect when a particular network has "significant structure". We note that throughout this chapter we use the term 'significant' to mean having an identifiable region in the β domain where the algorithm converges to non-trivial community structure, i.e., the marginals are not all approximately equal to $\frac{1}{q}$. From a statistical physics perspective, this means that for a given network there exists a particular phase in the state space (referred to as the retrieval phase by Zhang and Moore) where the beliefs converge to a non-trivial solution. The relationship between convergence of belief propagation and the detectability of communities in SBMs has been explored analytically [50, 51, 153] and empirically [70, 259]. Later, we will empirically demonstrate this for the multilayer modularity belief propagation. We emphasize that this differs from the standard notion of 'statistical significance' — specifically, we do not assess the value of a statistic compared to any particular model. That is, we do not assume any explicit model of communities here in using the modularity objective function maximization approach. There are a number of community detection approaches that employ a statistically based approach (*e.g* the stochastic block model (SBM) and all of its variants). However, we highlight that in the modularity belief propagation approach, there is no explicit model of communities. Beyond providing this notion of significance of structure, maximization of the marginals has the additional benefit of producing an interpretation of a soft partition wherein nodes are partially assigned across multiple communities. That is, the marginals reveal which node labels the algorithm is most uncertain about. Moreover, we can use the average entropy across all of the node labels as a measure of confidence in the predicted structure.

While there are several other tools available to compute the marginals of Eq 3.9, including Markov Chain Monte Carlo sampling, Gibb's sampling, belief propagation offers several unique advantages including computational efficiency and the tractability of asymptotic analysis [147]. Belief propagation is a general algorithm for calculating the marginals of a joint distribution by a series of iterative updates. Belief propagation was initially developed for trees [175] for which it is an exact algorithm, but has been shown to provide good approximations on graphs with loops (*i.e.* "loopy" belief propagation) [147, 178] assuming loops are small and short range correlations

that decay exponentially [256]. A belief propagation was first successfully applied to community detection in solving the stochastic block model in [86] and improved upon in [51] using an expectation maximization process to update the model parameters, and rigorously analyzed in [50]. Zhang and Moore introduced the belief propagation updates for single layer modularity maximization [259]. Recall the original formula by Newman and Girvan for modularity:

$$Q(c) = \frac{1}{2m}\sum_{i,j}(A_{ij} - \frac{k_i k_j}{2m})\delta_{c_i,c_j} \tag{3.10}$$

where A_{ij} is the possibly weighted adjacency, $c_i \in \{1, ..., q\}$ denotes the community assignment of node i, and $k_i = \sum_j A_{ij}$ givens the degree/strength of node i, and $m = \sum_{i<j} A_{ij}$ the number of edges in the graph. In writing this as a factor graph, we must distinguish between edges in the original network and interactions in our factor graph. In the case of modularity, the sum is over all pairs of nodes (rather than only the edges) because pairs of nodes within the same community still contribute to the score through the null model term, $\frac{k_i k_j}{2m}$. Thus, the set of edges in our factor graph includes those in the original graph, but also the interactions described by the null model, as depicted in Figure 3.2.

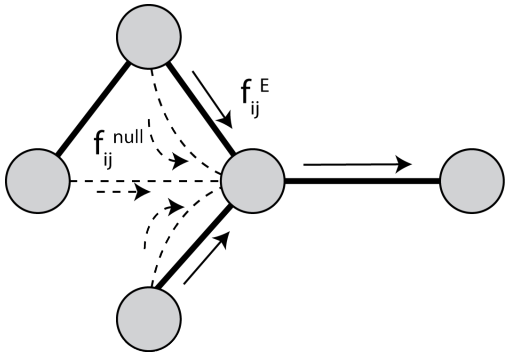

Figure 3.2: Schematic of modularity belief propagation. We have split the contributions to the modularity into two kinds of interactions: strong interactions represented by edges in the original graph (shown as dark, solid lines in figure) and weaker, all-to-all connections given by the null model term (shown as dashed lines). Beliefs (shown as arrows) are summed from all interacting nodes except the one who is receiving the message (far right node).

We could imagine Equation 3.10 as defining the contribution to the energy for a single type of interaction between nodes with corresponding interaction term: $f_{ij} = e^{\beta(A_{ij} - \frac{k_i k_j}{2m})\delta_{st}}$, where we have dropped the factor of $\frac{1}{2m}$ because only the relative energies matter for the probability of occupying a given state, and switched notations from $c_i, c_j \leftrightarrow s, t$ to match the belief

propagation notation. We note that $s, t \in \{1, \ldots, q\}$ are used to index the marginals, while c_i, c_j represented the community assignment of specific nodes. However, this leads to a much more computationally intensive form of belief propagation where updates for each node depend weakly on every other node in the graph requiring $\mathcal{O}(qn^2)$ computations for each round of updates. Instead we represent two different kinds of interactions in between our nodes shown in Figure 3.2: those corresponding to the edges in the original graph with contribution : $f_{ij}^{\mathcal{E}} = e^{\beta A_{ij}\delta_{st}}$ and the weak but dense interactions between all pairs of nodes in the graph that arise from the null-model term : $f_{ij}^{\text{null}} = e^{-\beta\frac{k_i k_j}{2m}\delta_{st}}$. By splitting our interaction into contributions from the separate terms in Equation 3.10, we can factor out the contribution from these, giving us the belief update equations introduced by Zhang and Moore [259]:

$$\psi_t^{i\to k} = \frac{1}{Z_{i\to k}} \prod_{j\in\partial i\setminus k} \sum_{s=1}^{q} e^{\beta\delta_{st}} \psi_s^{j\to i} \prod_{j\neq i\setminus k} \sum_{s=1}^{q} e^{-\beta(d_i d_j/2m)\delta_{st}} \psi_s^{j\to i}. \tag{3.11}$$

where the first product is only over the neighbors of node i in the network excluding k (*i.e.* $j \mid (i,j) \in \mathcal{E}$). Both terms can be simplified as follows:

$$\sum_{s=1}^{q} e^{\beta\delta_{st}} \psi_s^{j\to i} = e^{\beta}\psi_t^{j\to i} + \sum_{s\neq t}^{q} \psi_s^{j\to i} = (e^{\beta}-1)\psi_t^{j\to i} + 1 \tag{3.12}$$

$$\sum_{s=1}^{q} e^{-\beta\frac{k_i k_j}{2m}\delta_{st}} \psi_s^{j\to i} = e^{-\beta\frac{k_i k_j}{2m}}\psi_t^{j\to i} + \sum_{s\neq t}^{q} \psi_s^{j\to i} = (e^{-\beta\frac{k_i k_j}{2m}}-1)\psi_t^{j\to i} + 1 \tag{3.13}$$

Further simplification of the second term occurs by replacing all of the weak interactions with a single field term that is updated after each round of belief propagation updates. In the event that the network is sparse, the degree of any given node will be small compared to square root of the total number of edges edges ($k_i, k_j << \sqrt{2M}$). In this case we can approximate the message a

node sends to a non-neighboring nodes (along a dashed line in Figure 3.2) with its own marginal:

$$\psi_t^i = \psi_t^{i\to k} \sum_s e^{-\beta \frac{k_i k_j}{2m} \delta_{st}} \psi_s^{k\to i} \tag{3.14}$$

$$= \psi_t^{i\to k} (\sum_{s\neq t} \psi_s^{k\to i} + e^{-\beta \frac{k_i k_j}{2m}} \psi_t^{k\to i}) \tag{3.15}$$

$$\approx \psi_t^{i\to k} (\sum_{s\neq t} \psi_s^{k\to i} + (1 - \beta \frac{k_i k_j}{2m}) \psi_t^{k\to i}) \tag{3.16}$$

$$= \psi_t^{i\to k} (\sum_{s} \psi_s^{k\to i} - \beta \frac{k_i k_j}{2m} \psi_t^{k\to i}) \tag{3.17}$$

$$= \psi_t^{i\to k} (1 - \beta \frac{k_i k_j}{2m} \psi_t^{k\to i}) \tag{3.18}$$

$$\approx \psi_t^{i\to k} \tag{3.19}$$

Thus we can write our contribution from the null-model as follows:

$$\prod_{j\neq k} \sum_s e^{-\beta \frac{k_i k_j}{2m}} \delta_{st} = \prod_{j\neq k} ((e^{-\beta \frac{k_i k_j}{2m}} - 1) \psi_t^{j\to i} + 1) \tag{3.20}$$

$$\approx \prod_{j\neq k} ((e^{-\beta \frac{k_i k_j}{2m}} - 1) \psi_t^{j} + 1) \tag{3.21}$$

$$\approx \prod_{j\neq k} (\beta \frac{k_i k_j}{2m} \psi_t^{j} + 1) \tag{3.22}$$

$$\approx \prod_{j\neq k} (e^{\beta \frac{k_i k_j}{2m} \psi_t^{j}}) \tag{3.23}$$

$$= exp\left(-\beta \frac{k_i}{2m} \sum_j k_j \psi_t^j \right) \tag{3.24}$$

$$= exp\left(-\beta \frac{k_i}{2m} \theta_t \right) \tag{3.25}$$

where $\theta_t = \sum_j \psi_t^j d_j$ and is treated as constant for each round of belief propagation and then updated accordingly with each node's marginal. This "field trick" originally applied in [50] and [51] is made possible by splitting off the contributions from the edges of the network into a separate term from the interactions that come from the null-model term in the modularity formula. This reduces the computational complexity to a much more manageable $\mathcal{O}(qm)$.

Combining these simplifications gives Zhang and Moore's original update equation:

$$\psi_t^{i\to k} \propto exp\left(\frac{-\beta k_i}{2m}\theta_t + \sum_{j\in\partial i\setminus k} \log\left(1+(e^\beta-1)\psi_t^{j\to i}\right)\right) \tag{3.26}$$

Fixed points of the Eq 3.26 are also stationary points of the Bethe free energy

$$f_{\text{Bethe}} = -\frac{1}{N\beta}\left(\sum_{i\in\mathcal{V}}\log Z_i - \sum_{(i,j)\in\mathcal{E}}\log Z_{ij} + \frac{\beta}{4m}\sum_t \theta_t^2\right), \tag{3.27}$$

where $\mathcal{V}$ is the set of N nodes, $\mathcal{E}$ is the set of edges, and $Z_{ij} = \sum_{st} e^{\beta\delta_{st}}\psi_s^i\psi_t^j$ is the normalization constant for the pairwise joint marginals.

Computing marginals for each node, Zhang and Moore defined a "retrieval partition" assigning the community for each node according to its greatest marginal $c_i = \arg\max_t \psi_t^i$, with randomly broken ties. Retrieval modularity can be computed from the retrieval partition using Eq 3.28. We note that while this approach uses the modularity score to establish the energy landscape over which optimization is performed, ultimately the belief propagation minimizes the free energy; while lower free energy often corresponds to higher modularity for the retrieved partition, this relationship is in no way required and indeed is sometimes violated.

3.2.1 Extension of belief propagation to multilayer modularity

Recall the equation for the multilayer extension of modularity developed by Mucha *et al.* [156] written in the supra-adjacency form[4]

$$Q(\gamma,\omega) = \sum_{i,j}\left(A_{ij} - \gamma P_{ij} + \omega C_{ij}\right)\delta(c_i,c_j) \tag{3.28}$$

where i and j each index distinct node-layer objects, possibly in different layers, $\mathbf{A}$ is the supra-adjacency encoding the intralayer edges, $\mathbf{P}$ describes the expected number of intralayer edges based on the selected random model(s), and $\mathbf{C}$ encodes the interlayer connections. The

[4]In the supra-adjacency representation, a single block diagonal matrix is used to represent all intralayer connections, each block representing a single-layer, with no connections between the blocks. A different matrix, $\mathbf{C}$ encodes the interlayer connections. Note that $\dim(\mathbf{A}) = \dim(\mathbf{C}) = \dim(\mathbf{P})$ For introduction to multilayer networks and explanation of the notation, see Section 1.1.3.

normalizing factor traditionally written in front of the summation above has been absorbed here into the constituent terms for notational convenience. We here assume for simplicity the Newman-Girvan model for undirected edges within each layer, writing the null model contribution (prior to absorbing the normalizing factor) as

$$P_{ij} = \begin{cases} \frac{d_i d_j}{2m_{l_i}} & l_i = l_j \\ 0 & l_i \neq l_j \end{cases} \tag{3.29}$$

where l_i is the layer containing node-layer i, (*i.e.* $i \in \mathcal{V}_{l_i}$), $d_i = \sum_j A_{ij}$, and $m_{l_i} = \sum_{i,j \in \mathcal{V}_{l_i}} A_{ij}$ is the total weight of edges in layer l_i. We enforce on $\mathbf{A}$ that a given node-layer i only participates in intralayer edges within its own layer (by definition). In the case where edge weights are binary ($A_{ij} \in \{0, 1\}$), d_i is the degree of node i. For weighted networks, A_{ij} is continuous and $d_i = \sum_j A_{ij}$ is called the 'strength' of node i. Similar null models are available for bipartite graphs, directed networks, and networks with signed edges (see, e.g., the supplement of [156] for references to appropriate forms for P_{ij} in different contexts).

We have employed a very similar approach as Zhang and Moore, however now we use the formula for multilayer modularity in Equation 3.28 as the Hamiltonian to represent interactions in our model. We now use $i, j,$ and k to index the node-layers in our multilayer network. For more details about the multilayer notation used here, please see Section 1.1.3 in the introductory chapter. First, we account for the additional contribution of the interlayer edges in a similar manner to the intralayer edges: $f_{ij}^{\text{multi}} = e^{\beta \tilde{A_{ij}}}$, where $\tilde{A}_{ij} = A_{ij}\delta(l_i, l_j) + \omega C_{ij}(1 - \delta(l_i, l_j))$ is the appropriate weight for the inter/intralayer edge the message is traveling along. We note that we have allowed for weights along the intralayer edges in the same method as Shi *et al.* [208] while interlayer edges are of uniform weight which is incorporated into ω. The block description of A_{ij} and C_{ij} considered here makes the $\delta(\cdot, \cdot)$ indicators in $\tilde{A}_{ij}$ unnecessary; but we include them to help clarify the notation in terms of the layers containing i and j. Thus the contributions from the edges in the network as now given by :

$$\prod_{j \in \partial i \setminus k} \sum_{s=1}^{q} e^{\beta \tilde{A}_{ij}} \psi_s^{j \to i} = \prod_{j \in \partial i \setminus k} \left((e^{\beta \tilde{A}_{ij}} - 1)\psi_t^{j \to i}) + 1 \right) . \tag{3.30}$$

This product is over all node-layers in the neighborhood of i including its interlayer neighbors (but excluding node-layer k). There are also several modifications we have made to the contribution from the null-model in multilayer modularity. In the null model for multilayer modularity given by Equation 3.28, only pairs of node-layers that are within the same layer contribute to P_{ij} with the denominator being the total number of edges within i's layer, m_{l_i} (see Equation 3.29) . Thus the product in the null-model component only goes over the indices of node-layers within the same layer as node-layer,i :

$$\prod_{j \neq k} \sum_{s} e^{-\beta \frac{k_i k_j}{2m}} \delta_{st} \Rightarrow \prod_{j \in \mathcal{V}_{l_i} \backslash k,i} \sum_{s} e^{-\beta \frac{k_i k_j}{2m_{l_i}}} \delta_{st} \tag{3.31}$$

(3.32)

We have also incorporated a resolution parameter, γ to the contribution from the null model. The resolution parameter balances between the contribution of edges that are internal to communities and the strength of the field from the null-model. This gives the null-model interaction: $f_{ij}^{\text{null}} = e^{-\beta\gamma \frac{k_i k_j}{2m_{l_i}}}$. We can still incorporate the field trick detailed above as long as $k_i, k_j << \sqrt{2m_{l_i}/\gamma}$. We have found that the algorithm generally doesn't converge if γ is too large and have generally used $\gamma \leq 3$ for experiments in this manuscript. We can apply the same line of reasoning as Equation 3.20 above, substituting ψ_t^j for $\psi_t^{j \to i}$ and Taylor's theorem to arrive at :

$$\prod_{j \in \mathcal{V}_{l_i} \backslash k} \sum_{s} e^{-\beta\gamma \frac{k_i k_j}{2m_{l_i}}} \delta_{st} \approx exp\left(-\beta \frac{k_i}{2m_{l_i}} \theta_{l_i}^t\right) , \tag{3.33}$$

where $\theta_{l_i}^c = \sum_{j \in \mathcal{V}_{\updownarrow\rangle}} \psi_t^j k_j$, is the layer specific field term that is treated as constant for each round of message passing, then updated according to the new marginals. These modification combined give us the update equations 3.34 for *multimodbp*. We note that the message passed from node-layer i to node-layer k, $\psi_t^{i \to k}$ does *not* depend on the type of edge (i, k). Node-layer i integrates information from its neighboring nodes-layers (except node-layer k), handling both edge weights and types appropriately, and passes this information to node-layer k. The edge type (and weight) between node-layer i and node-layer k only comes into play when node-layer k integrates all the information coming in from its neighboring nodes. Thus we can write the belief

propagation equations for multilayer modularity as follows :

$$\psi_t^{i\to k} \propto \exp\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i} + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right] \tag{3.34}$$

The fixed point of the above iterative equations are minimizers for the following Bethe free energy equation, as derived in Section 3.1 of the supplement:

$$f_{\text{Bethe}} = -\frac{1}{N\beta}\left(\sum_i \log Z_i - \sum_{i,j\in\mathcal{E}}\log Z_{ij} + \sum_l \frac{\beta}{4m_l}\sum_t(\theta_t^l)^2\right) \tag{3.35}$$

where $Z_{ij} = \sum_{st} e^{\beta\delta_{st}}\psi_s^i\psi_t^j$ is the normalization factor for the pairwise joint marginals.

While we demonstrate *multimodbp* below in the context of the specific multilayer topology corresponding to a multiplex network, our formulation is flexible enough to handle any type of multilayer network consisting of two classes of edges (i.e., intralayer and interlayer edges). In particular, we remark that, similar to the weights in A_{ij}, the contribution from C_{ij} is explicitly included here, allowing for different interlayer weights. In principle, the method could also be extended to networks with multiple types of edges, such as encountered in representing network data that is both longitudinal and multiplex, with each new edge type introducing its own coupling parameter, ω_i. There are several other details concerning the implementation of the belief propagation approach which we leave for the interested reader at the end of this chapter, including the selection of the inverse temperature parameter β in Section 3.5.1 as well as how we can use the algorithm to identify the appropriate number of communities in Section 3.5.2. Next, we will demonstrate how the changes we have made affect the performance of modularity belief propagation in the case of single-layer networks as well as the interpretation of the results.

3.3 Multimodbp results

3.3.1 Single-layer networks

We begin by examining how our modifications affect the ability of *modbp* to detect communities within synthetically generated data in the single-layer case. For single-layer networks, our method is equivalent to Zhang and Moore's except two main differences (see also

3.5. Additional Methods for Chapter 3): First, we have included a resolution parameter γ that adjusts the relative balance of the terms in the update equation. Like other implementations of modularity, this effectively controls the size of the identified partitions. Second, we have set an upper limit $q_{\max}$ on the number of communities and incorporated the approach from [206] to select an effective number of communities based on the overlap of the marginals (see Section 3.5.2).

SINGLE-LAYER STOCHASTIC BLOCK MODEL

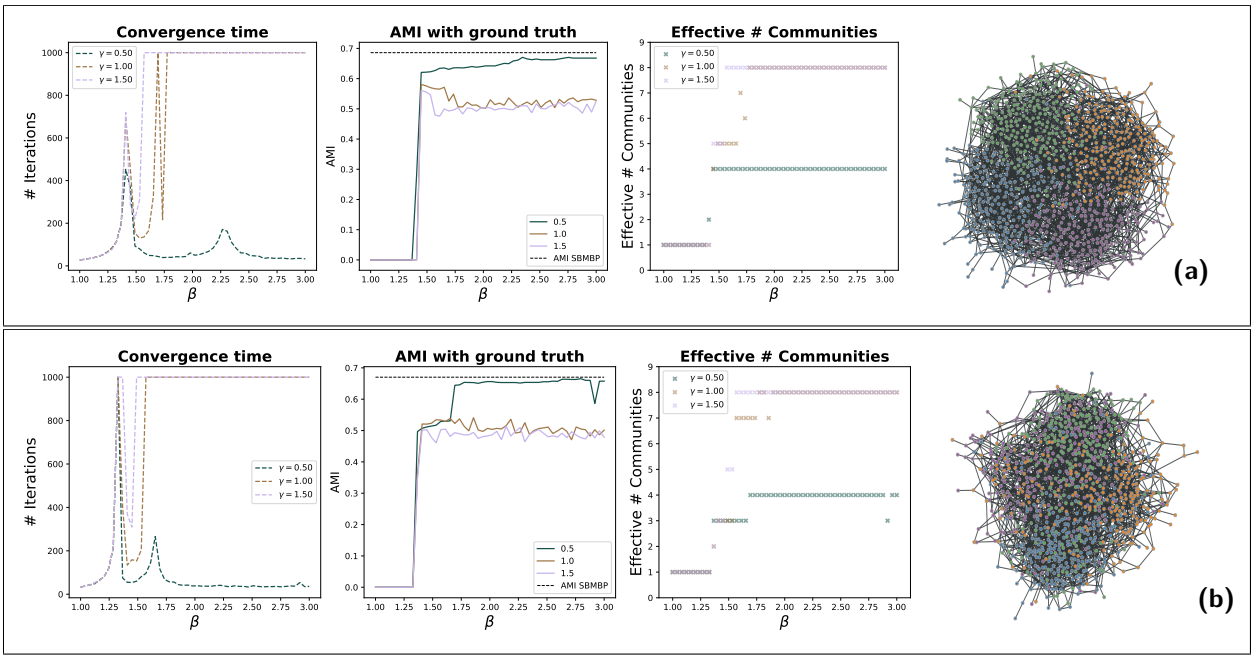

Figure 3.3: Demonstration of *multimodbp* on two realizations of the original SBM model (non-degree corrected). From left to right, the plots show the retrieval modularity, number of iterations to convergence, and the AMI of the retrieval partition with known community assignments and the effective number of communities. **(a)** 4 community SBM with $n = 1000$, $\epsilon = \frac{p_{\text{out}}}{p_{\text{in}}} = .1$, $c_{\text{avg}} = 4$, and even community sizes and **(b)** and 4 community SBM with $n = 1000$, $\epsilon = .1$, $c_{avg} = 4$, with uneven community sizes ($\nu = [300, 200, 300, 200]$). For each network we also show the performance of the *sbmbp* with parameters for the SBM supplied (middle plot, dotted black line. See Section 3.3.1 for details of *sbmbp* method.)

We examine the behavior of *multimodbp* on instances of a four-community stochastic block model (SBM) (using the original, non-degree corrected SBM) for different values of the resolution parameter γ. First, we show that in the setting with several smaller communities, a lower value of γ produces a much wider retrieval phase and thus makes detection of communities more robust to selection of β. To investigate this robustness, we generated a single realization of

an SBM and scanned a range of β values to characterize the behavior of the algorithm seen in Figure 3.3. For an SBM network with four even-sized communities, Figure 3.3a shows that the retrieval phase for both $\gamma = 1.0$ and $\gamma = 1.5$ are very narrow (leftmost panel) with a small corresponding peak in the AMI of detected communities (middle panel). In contrast, for $\gamma = 0.5$ the retrieval phase widens out with a broader and higher set of AMI values for the detected communities. Furthermore, the number of communities identified for $\gamma = 0.5$ plateaus at the correct number, 4 as shown in far right panel of Figure 3.3a)

We also tested the performance of the algorithm in the case where the sizes of the planted communities were uneven, shown in Figure 3.3b. The relative performance for varying γ is even more disparate in this case. There is a small retrieval phase for $\gamma = 1$, but it is much smaller than that of $\gamma = 0.5$ and the AMI is again consistently lower. For $\gamma = 0.5$ we actually detect two retrieval phases. In the first retrieval phase ($\beta \in [1.4, 2.0]$), only nodes within the two larger communities are labeled correctly. Then, as β increases ($\beta \in [1.75, 3.0]$), the smaller two communities also become identifiable. This is consistent with the multiphase behavior observed in [206], though we note that in their example, the phase transition is observed for the default value of $\gamma = 1$. In both of these examples the AMI of the identified partition by *multimodbp* is close to the result achieved by a belief propagation implementation of the SBM model, which has been shown to achieve the optimal bounds for this model [50, 51].

In both of these experiments the value of β^* marking the transition from the paramagnetic phase to either the retrieval phase or the spin-glass phase is independent of value chosen for the resolution parameter, γ. However, the width of the retrieval phase is dependent on the particular value of the resolution parameter, γ (see upper left panel in Figure 3.3a. Thus the detection of significant communities in this case relies on the appropriate selection of the value of γ.

Comparison of *multimodbp* with SBMBP on LFR benchmark networks

We compare the performance of our algorithm *multimodbp*, with a belief propagation approach to fit the Stochastic Block Model (SBM) developed and implemented in Ref. [50], which we refer to as *sbmbp*. This Expectation-Maximization (EM) implementation of *sbmbp* alternates between iteratively updating the marginals using belief propagation with fixed SBM parameters, and updating the SBM parameters using likelihood maximization for the fixed marginals. Their

implementation requires setting a fixed q however, so for testing we ran *sbmbp* across a range of q values ($q \in 2, 3, ..., 8$) and selected the partition with the lowest free energy density.

Our test dataset is the Lancichinetti-Fortunato-Radicchi (LFR) benchmark generator [125], an algorithm developed to generate networks with more diverse community structures. We tested our *multimodbp* with several values of the resolution parameter γ against *sbmbp* across a range of parameters of the LFR model. We vary the LFR mixing parameter μ, which sets the detectability of the underlying communities. The LFR algorithm also has a parameter $\hat{\gamma}$ to set the exponent of the power law for the degree distribution and a parameter $\hat{\beta}$ to set the exponent of the community size distribution. We tested both algorithms for two sets of $(\hat{\gamma}, \hat{\beta})$ in Figure 3.4. Figure 3.4 shows that the modularity based approach outperforms the stochastic block model

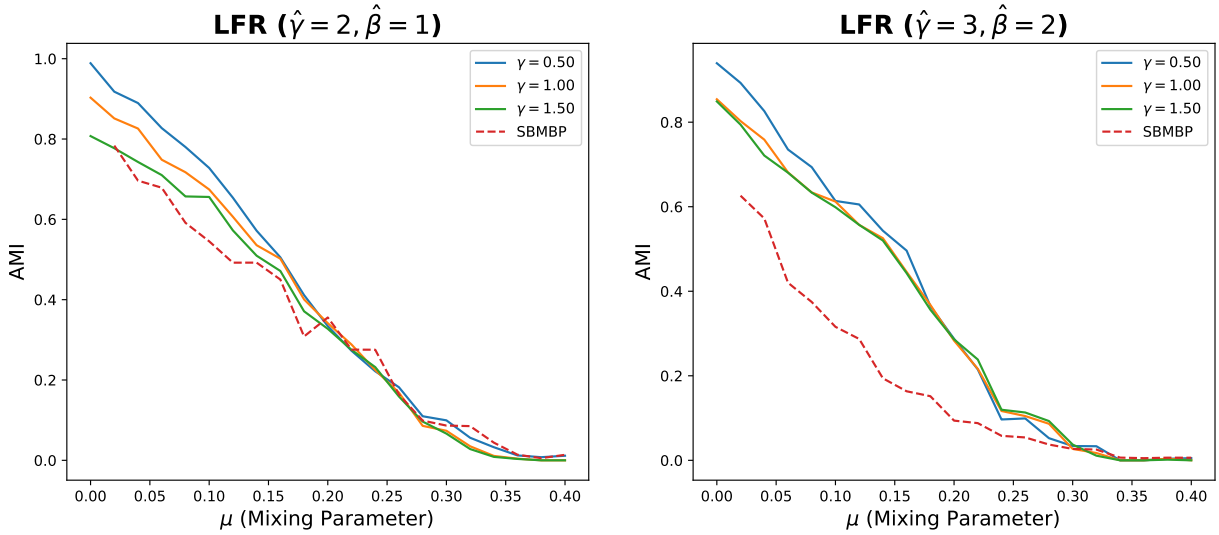

Figure 3.4: Performance of *multimodbp* and *sbmbp* over many LFR benchmark realizations with a range of values for the mixing parameter μ. Each point represents an average over 100 realizations of LFR with 1000 nodes, an average degree of 3 (with a max of 10), and other parameters set to default values.

across a range of μ, the mixing parameter, all the way down to the detectability limit. The flexibility of the modularity approach allows for better identification of communities with for real world degree distribution (since the classic SBM assume homogenous degree distribution within a community). The comparison was done using *sbmbp*'s EM approach which is not well suited to determine the number of communities. In contrast, using our approach as described in Section 3.5.2, the *multimodbp* algorithm was able to identify the correct number of communities

and get more accurate community assignment using a resolution parameter value of $\gamma = 0.5$ (though other values of γ also performed well).

NCAA Division I-A College Football network

We now demonstrate that inclusion of the resolution parameter γ in the modularity objective function can significantly improve performance on real-world networks. As an example of a real-world network with stable community structure we selected the 2000-2001 NCAA Division I-A College football network, which has 115 nodes representing teams (schools) and 613 unweighted edges connecting teams that played at least one game [58, 71]. Our previous work suggests that modularity optimization produces the best community partition in a range $\gamma \in [1.4, 4]$ [244, 245]. To investigate how the value of γ affects the retrieval phase, we ran *multimodbp* for a range of values of the parameter γ and examined the minimal number of iterations for which non-trivial structure was identified, shown in Figure 3.5. For each value of γ, *multimodbp* was run over 30 evenly-spaced values over $\beta \in [0.5, 4.5]$. For each value of γ we show the minimum number of iterations over all values of β for which non-trivial structure was identified and the AMI of the partition of the corresponding partition (the partition identified with the minimum number of iterations). Runs that did not converge after 500 iterations suggest that for that value of γ the retrieval phase was either very small or nonexistent. It is possible that a retrieval phase exists outside the chosen range for β, though we verified for a few arbitrary values of γ that the algorithm did not have a retrieval phase. Furthermore, Figure 3.5 demonstrates that the AMI of the retrieval partition increases as a function of γ from $\gamma = 1$ up until it plateaus from $\gamma = [1.7, 3.4]$ at a stable 11 community partition (shown in the far right panel). In Figure 3.6, we show the algorithm convergence properties as well as performance for a few values of γ on this network. We also compare the performance of the *multimodbp* algorithm with the *sbmbp* approach, showing that even when the SBM approach identifies the correct number of communities (middle panel dashed line), *multimodbp* still achieves more accurate identification of the underlying community structure (right panel).

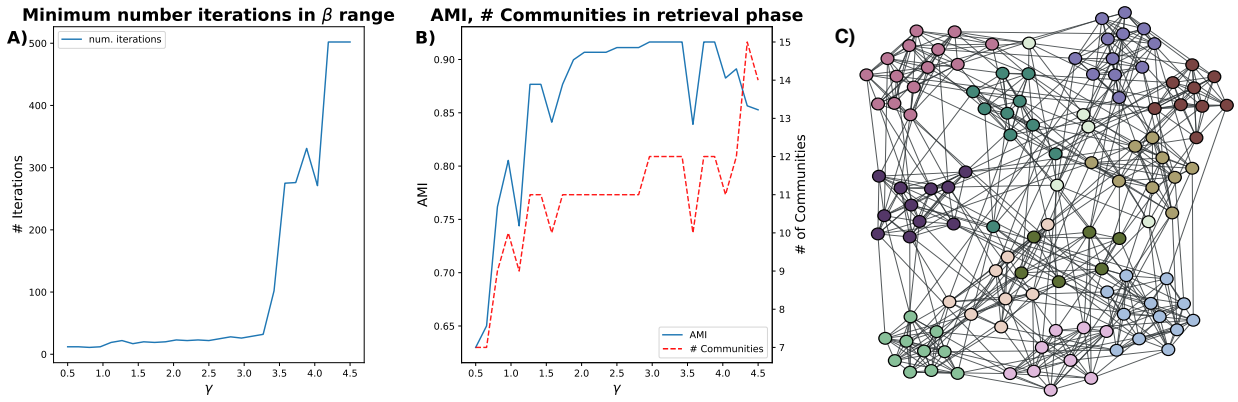

Figure 3.5: Testing *multimodbp* on the 2000-2001 Division I-A College Football network [58, 71]. **A)** The average number of iterations until convergence in the retrieval phase across a range of γ values. **B)** The average number of communities detected in the retrieval phase as γ increases and the corresponding adjusted mutual information (AMI) of those partitions. **C)** ForceAtlas2 [98] layout of the football network with each node colored according to a partition identified using $\gamma = 3.0$, demonstrating excellent alignment to the conference structure.

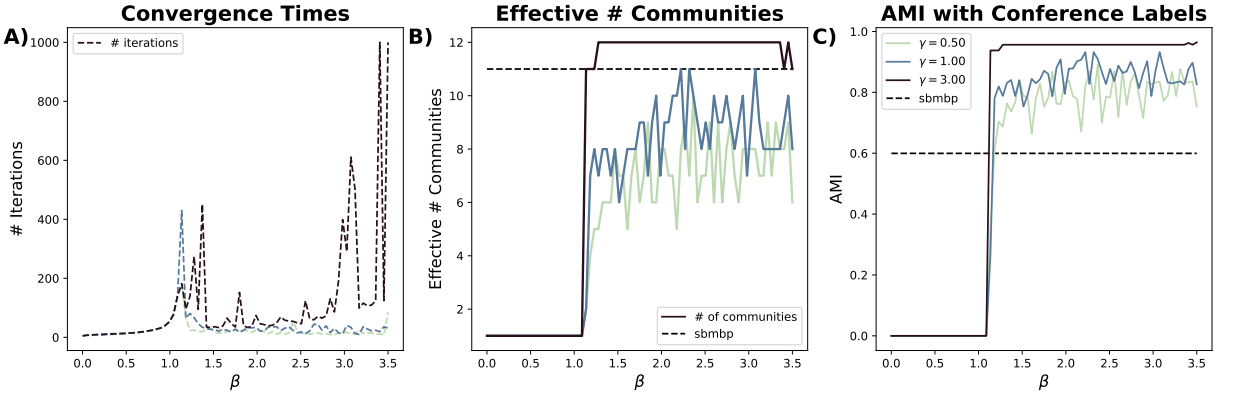

Figure 3.6: Performance characteristics of the algorithm for 3 different values of γ on the 2000-2001 NCAA Division I-A College football network. A) Although all three values of γ produce a wide retrieval phase, the communities identified within each retrieval phase are different. **B)** The number of non-redundant communities is higher as γ increases with $\gamma = 3$ producing the number of communities that lines up well with the ground truth (the conferences) in this example, with **C)** showing corresponding higher values of AMI for $\gamma = 3$. Horizontal black dashed line shows that *sbmbp* identifies correct number of communities in **B)** but has less agreement with the known conference structure of the network.

3.3.2 Multilayer layer results

Dynamic stochastic block model

We test the multilayer functionality of *multimodbp* by application to a multilayer SBM called the dynamic stochastic block model (DSBM) as described in [70]. The DSBM represents a temporal multilayer network where each node in the network is represented by a single node-layer within each layer. The correspondence between identified node-layers is represented by a single interlayer edge between adjacent layers. In the DSBM, each layer is drawn from a regular stochastic block model with q communities and edge probabilities described by probabilities p_{in} within communities and p_{out} between communities. Each node-layer's community assignment has a fixed probability η of remaining the same between subsequent layers (and $1-\eta$ probability of choosing a new community). Conditioned on the node community assignments, each layer's edges are independent of all other layers. For a fixed average degree c, the strength of community structure within each layer is given by the parameter $\epsilon = p_{\text{out}}/p_{\text{in}}$.

In Figure 3.7 we show the average $\langle \text{AMI} \rangle$ score of the *multimodbp* algorithm on the dynamic stochastic block model for a range of parameters. We consider DSBM networks created using values of ϵ and η ranging from 0 to 1. For each choice of ϵ and η we created 50 networks and computed the $\langle \text{AMI} \rangle$ between partitions identified using *multimodbp* and the ground truth. Because the value of q is usually not known beforehand, for each (γ, ω) point we scan a range of possible values of β^* corresponding respectively to possible values of q as given by Eq 3.41 with $q_{\max} = 4$ set to twice the true number of communities (2). For each trial, we select the partition with the highest retrieval modularity among all that converged.

We apply the *multimodbp* algorithm in this analysis with several choices of the resolution parameter, γ (columns in Figure 3.7) and coupling parameter, ω (rows of Figure 3.7). Figure 3.7 shows that incorporation of a resolution parameter makes a large difference for detectability of community structure based on the DSBM parameters used to generated the network. For lower values of ϵ (i.e., increased intralayer community signal) with with frequent community switching (decreased η), $\gamma = 0.5$ clearly outperforms the higher values of γ. However, for $\gamma = .5$ the algorithm fails to utilize information across the layers and performance drops off as ω is

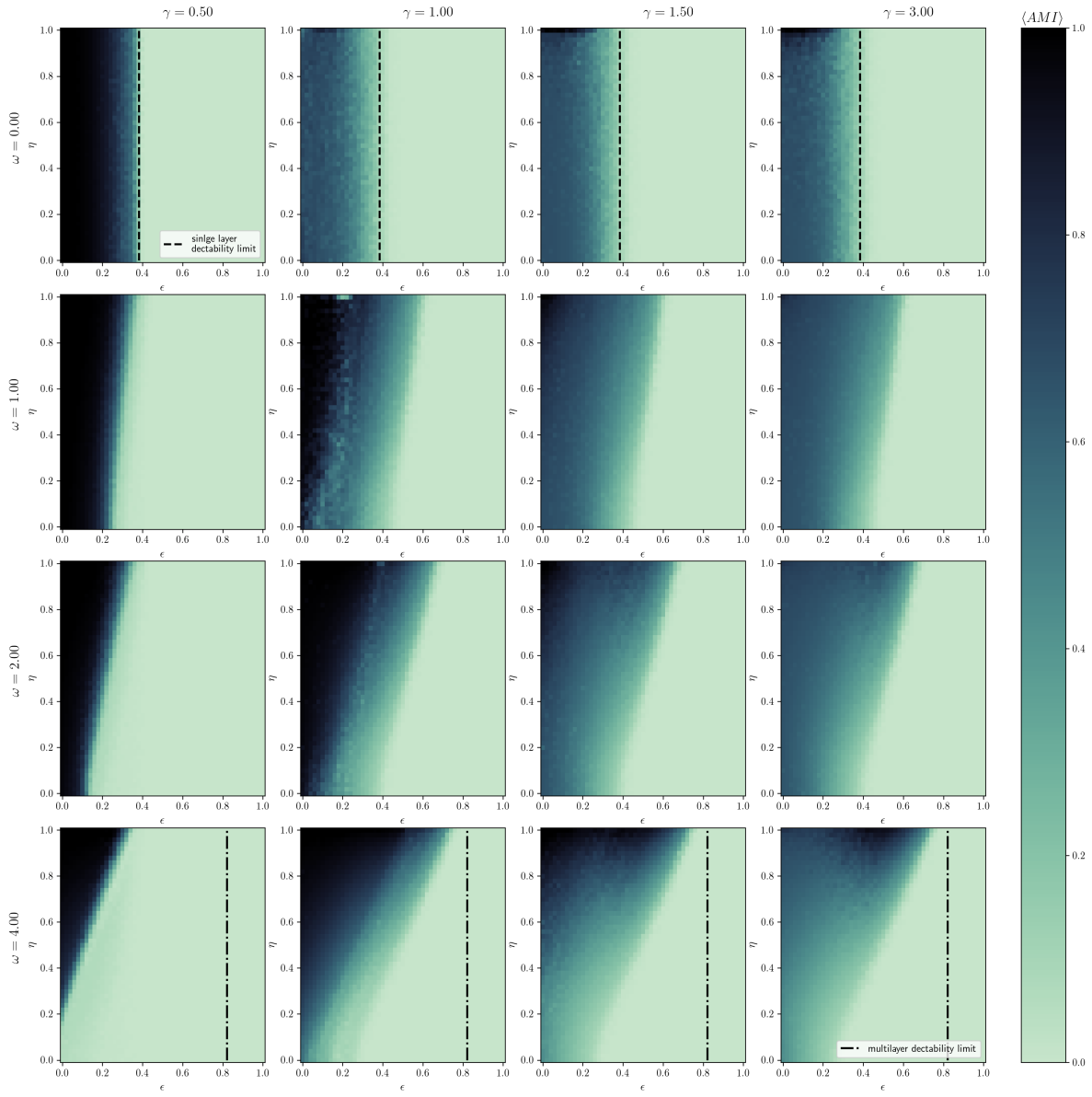

Figure 3.7: Accuracy of the multilayer modbp algorithm on a DSBM. We test *multimodbp* across different values of model parameters ϵ, and η (x and y axes respectively) and for *multimodbp* parameters γ and ω (moving horizontally and vertically vertically across panels). For these generated networks, $N = 250$, $n_{layers} = 20$, $c = 10$, and $q_{true} = 2$.

increased. For $(\gamma = .5, \omega = 0)$, *multimodbp* performs quite well all the way down to the limit of detection for single layer networks, given by the condition $N(p_{\text{in}} - p_{\text{out}}) > q\sqrt{c}$ in [51] and [160] (depicted by the vertical dashed line at $\epsilon = .38$ in upper row of Figure 3.7).

For higher values of γ we get better aggregation of information across the layers with increasing ω. At $\gamma = 1.0, \omega = 4.0$ the range of detection is increased to a maximum of $\epsilon = .75$ for $\eta = 1.0$ (no community switching). This behavior is consistent with the limits of detectability that are achieved through aggregation of layers as discussed in [218]. They derive a modified limit of detectability in the case where each layer is drawn from the same 2-block SBM with the community labels fixed throughout the layers, unlike our model where each nodes' community assignment is allowed to vary. They compute a detectability threshold of $NL(p_{\text{in}} - p_{\text{out}}) = \sqrt{4NL\rho(1-\rho)}$ where $\rho = \frac{1}{2}(p_{\text{in}} + p_{\text{out}})$. For parameters used in this experiment the theoretical detectability limit is $\epsilon \approx .82$ (shown by the dashed lines in Figure 3.7). These results demonstrate how the additional flexibility provided by tuning γ and ω allows for achieving near optimal performance depending on the parameters of the underlying model.

Comparison with *GenLouvain* on Heterogeneous Benchmarking Networks

To assess the performance of *multimodbp* on more realistic synthetic data, as well as on different multilayer topologies, we have applied it to the generative models described in [21] and implemented in MATLAB [99] and python [100]. In [21], Bassi *et al.* present a multilayer generative model that allows for the coupling of mesoscale structures across a variety of interlayer topologies. In their approach, a multilayer partition is sampled from a distribution defined by a given null model as well as the specified interlayer dependencies. For the multilayer networks shown here, communities assignments are drawn from a Dirichlet distribution in an arbitrary starting layer, and then either copied or resampled based on the interlayer coupling probability (p) in the other layers. For a complete description see [21] . Then the interlayer edges are drawn independently for each layer conditioned on the assigned communities. After a multilayer community partition has been sampled, the edges within each layer are sampled according to a degree corrected stochastic block model (DCSBM) conditioned only on the community

assignments within each layer[5]. That is the edges within each layer are independent of each other given the community assignments. Interlayer dependencies are introduced only through the probability that a given node keeps the same community assignment from layer to layer. Within each layer, the strength of community structure is given by the mixing parameter, $\mu \in [0, 1]$. If $\mu = 0$, communities are perfectly separated (no edges between) while if $\mu = 1$ edges are placed without regard to the communities. We specify the interlayer coupling topology and parameters for each experiment below and the parameters for sampling the intralayer edges from the DCSBM. For each experiment in this section, we have used the same parameters detailed in Section V.A of [21] for the corresponding multilayer topologies. We have compared *multimodbp* to *GenLouvain* [104] across a range values for the interlayer coupling parameter, ω (keeping $\gamma = 1.0$). Within the multiplex experiments detailed below, we found that as the number of layers became deeper, the all-to-all connections between node-layers representing a single node quickly became stuck in a local minimum where node-layers with interlayer connections were all strongly forced into the same community, washing out the weak community information from the intralayer neighbors. To surmount this, we used a rudimentary spectral clustering approach on the modularity matrix to initialize the beliefs as was suggested in [260]. We compute the top $k = q_{max} - 1$ eigenvectors of $\mathbf{B} = \mathbf{A} - \gamma\mathbf{P} + \omega\mathbf{C}$, the modularity matrix, and use the K-means algorithm to find q_{max} different clusters. This does not add significant additional runtime as computing the leading eigenvectors of sparse matrices can be done efficiently. All incoming beliefs to a given node are then initialized to a soft version of the identified spectral partition where the belief representing the node's associated community is set to be some factor (5 in this case) times larger than the other beliefs. We found that in general that starting *multimodbp* with even relatively weak alignment from the spectral partitioning greatly improved the results for the highers values of ω. We show in supplement Figure 24 that *multimodbp* improves on the baseline provided by the spectral initialization (even when the spectral clustering performs quite poorly). All of the results of *GenLouvain* were obtained using the iterated approach where results from each run are used to initialized the communities for the next round until no improvements in

[5]Other models could be used for sampling the intralayer connections. The network generation process described by Bazzi *et al.* is modular in nature allowing for a large combination of inter and intralayer structures. We have chosen the DCSBM for comparability with the results in [21]

modularity can be obtained. We also used the random move setting which allows the algorithm to break out of local optimum. For each combination of $\mu, p,$ and ω in the experiments below, we run each approach once on 100 independently sampled networks from the benchmark model. Our results below for *GenLouvain* closely mirror the findings from [21].

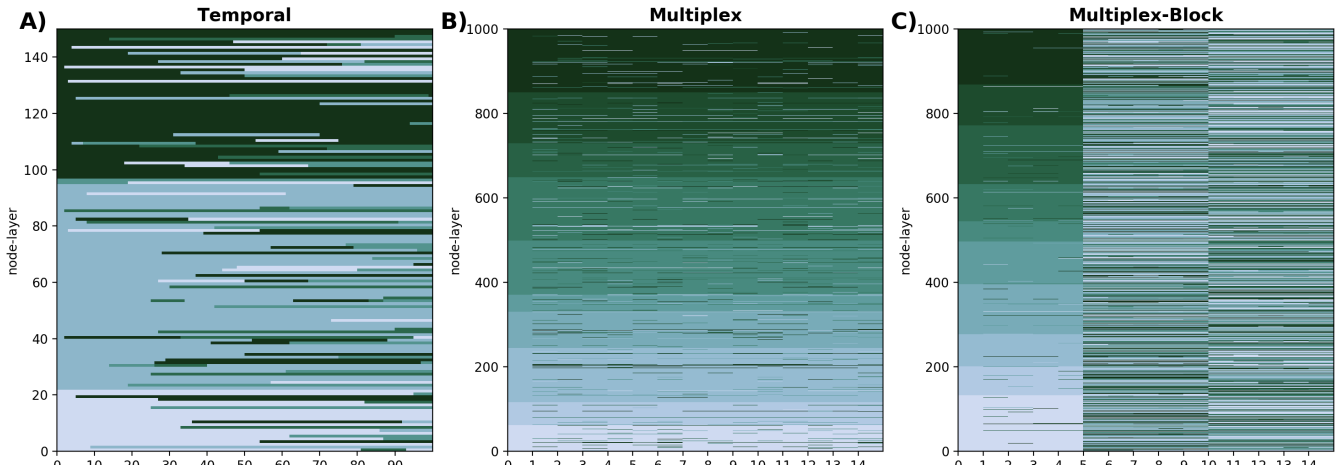

Figure 3.8: Graphical representation of the community structures for networks samples from different interlayer topologies available with the multilayer-generative model in [21]. In each subfigure, each row represents a particular node, with each column representing a layer of the network. Each node-layer is colored according to its multilayer partition. Thus we can see how the different communities persist across the layers of the network.

TEMPORAL MULTILAYER NETWORK: This network has a similar interlayer topology as the DSBM detailed in Section 3.3.2 with an ordering on the layers and each node-layer connected to only the node-layers in the layers adjacent to it. That is

$$C_{ij}^{temporal} = \begin{cases} 1 & \text{if } l_j = l_i \pm 1 \text{ AND } i \cong j \\ 0 & \text{otherwise} \end{cases} . \tag{3.36}$$

Similarly to the DSBM, each node has a probability, p of copying its identified node-layer's community in the preceding layer. The major difference with this experiment is that community assignments are drawn from a more realistic Dirichlet null distribution with $\theta = 1, n_{set} = 5,$ and $q = 1$ (rather than a uniform size distribution) and that the intralayer connections are drawn from an SBM with degree correction (DCSBM) with : $\eta_k = -2, k_{max} = 30, k_{min} = 3$. Each sampled network has 150 node-layers in each layer with 100

layers for a total of 15000 node-layers. Visualization of an example temporal network is shown in Figure 3.8.A. We have run both *multimodbp* across a range of p and μ and compared how the increasing the interlayer coupling parameter ω affects the performance of the algorithm. In Figure 3.9.A (the top two rows) we see that *multimodbp* with the spectral initialization tends to outperform *GenLouvain* for a wide variety of model parameters. We see that the peak AMI obtained for $\mu = .8$ is higher for *multimodbp* across most values of p in some cases notably so ($\langle \text{AMI} \rangle \approx .8$ vs $\langle \text{AMI} \rangle \approx .4$ at $p = .99$). Thus *multimodb* is better able to utilize the information across the adjacent layers to inform community prediction.

UNIFORM MULTIPLEX AND BLOCK MULTIPLEX: We also sampled graphs from two different multiplex interlayer topologies. Unlike in the temporal multilayer networks, in the multiplex topology, there is no inherent ordering to the layers. Each node-layer is connected with interlayer edges to all other node-layers in the identified set:

$$C_{ij}^{multiplex} = \begin{cases} 1 & \text{if } i \cong j \\ 0 & \text{otherwise} \end{cases} . \tag{3.37}$$

In the uniform multiplex, a node-layer's community assignment is copied with a given probability p to all of its identified node-layers. A visualization of an example network with this structure is shown in Figure 3.8.B. In contrast, for the block multiplex, we divide the layers into a specified number of blocks, and only copy the layer assignments with probability p for layers within a given block. Within each block the structure is the same as the uniform multiplex, however there is a complete discontinuity in node-layer community assignments from block to block. Note that while the interlayer coupling probabilities are set to 0 between blocks in the model, the interlayer edges between blocks are still present in the network. Figure 3.9.C shows an example of a multiplex block network. For both of these examples we use a Dirichlet null model with $\theta = 1, n_{set} = 10,$ and $q = 1$ and generate the intralayer edges from the DCSBM with parameters: $\eta_k = -2, k_{min} = 3,$ and $k_{max} = 150$.

We compare *multimodbp* with *GenLouvain* on the uniform multiplex benchmark

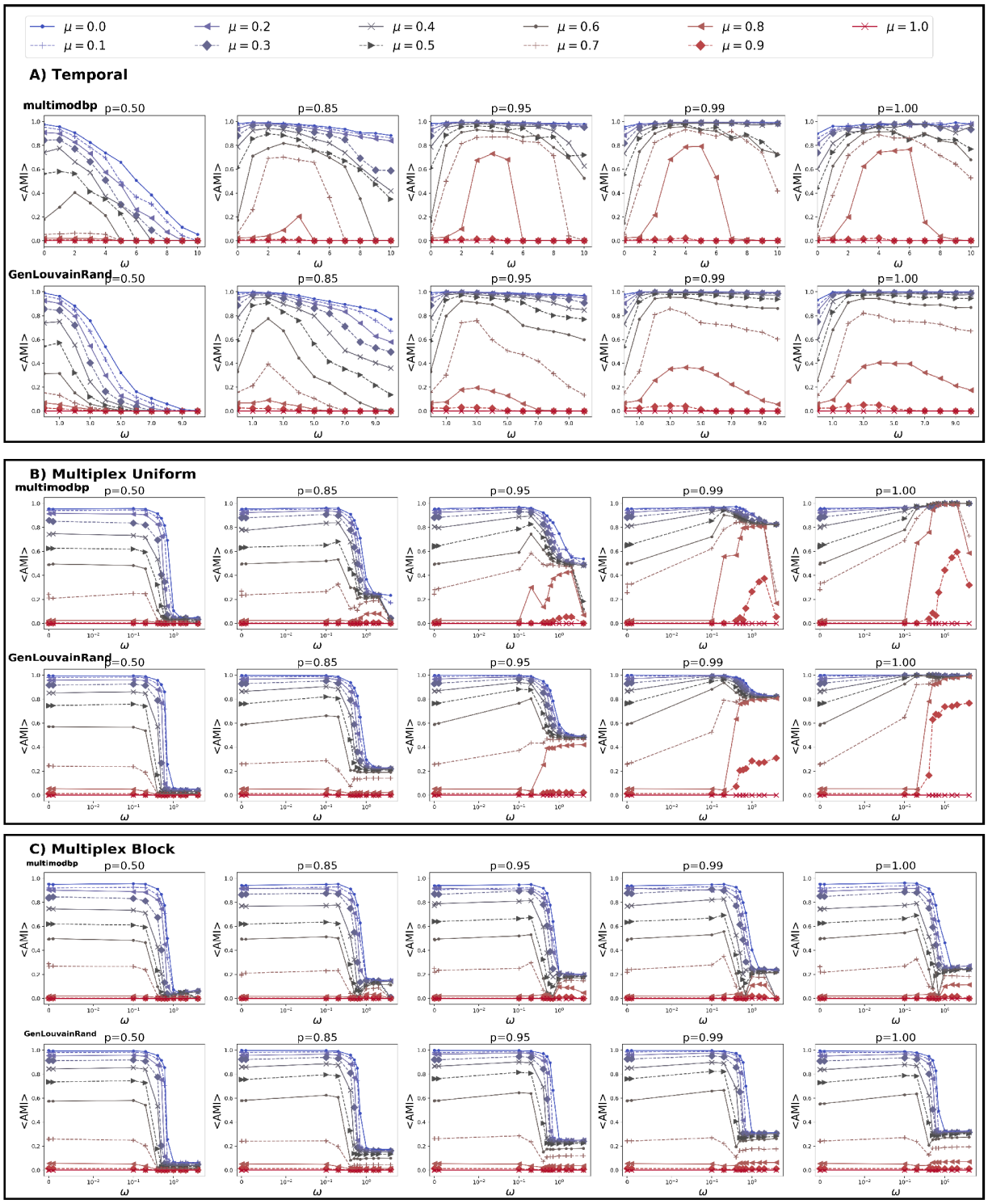

Figure 3.9: (*Caption on next page.*)

Figure 3.9: (Previous page.) **Comparison of *multimodb* with *GenLouvain* on multilayer benchmarks.** We compare the performance of *multimodb* (top rows of each panel) and *GenLouvain* [104] (bottom row of each panel) across a range of multilayer benchmark networks developed by Bazzi *et al.* [21]. For each model we vary both μ, the intralayer mixing parameter (strength of communities) denoted by the different markers and colors. From left to right, across the subfigures, we vary the persistence of communities across layers from $p = .5$ to $p = 1.0$. Each points represents the average $\langle \text{AMI} \rangle$ over 100 independent realizations of the model. **A)** Temporal network topology with ordered layers and interlayer connections only present between adjacent layers. Multilayer community partitions are drawn from Dirichlet distribution with $\theta = 1, n_{set} = 5,$ and $q = 1$ and intralayer edges are samples from a DCSB with : $\eta_k = -2, k_{max} = 30, k_{min} = 3$. Each network has 100 node-layers in each layer with 150 layers for a total of 15000 node-layers. **B)** Uniform multiplex multilayer network with unordered layers and all to all interlayer connections among identified node-layers across all layers. Multilayer partitions are sampled from Dirichlet distribution with $\theta = 1, n_{set} = 10,$ and $q = 1$ and intralayer connections are drawn from DCSB with $\eta_k = -2, k_{min} = 3,$ and $k_{max} = 150$. Each network has 1000 node-layers in each layer with 15 layers for a total of 15000 node-layers. **C)** Block multiplex model with the same parameters as the uniform multiplex models however we introduce a discontinuity between each block of 5 layers where community labels are completely independent.

networks in Figure 3.9.B (middle two rows). We find that in most of the parameter regimes, performance is relatively comparable between the two methods, with *GenLouvain* having a slight edge overall, especially for higher values of μ and lower values of ω. However, for some parameters on the block multiplex networks in Figure 3.9.C, *multimodbp* tends to have the edge over GenLouvain, especially at lower values of μ (see $\mu = .8, p = .95$). Overall, we see that *multimodbp* is able to utilize information across layers to detect community structure where it would be undetectable if each layer was considered independently. These benchmarks demonstrate that *multimodb* performs comparably and in some cases outperforms one of the leading multilayer community detection methods, GenLouvain.

Furthermore, the convergence properties of *multimodbp* provide additional information about whether there is significant community structure within a network. In Figure 3.10 we show that *multimodbp* stops converging when the planted structure is undetectable. In contrast, the modularity of the communities detected by *GenLouvain* remains relatively stable, even as the communities become increasing undetectable for higher values of μ. Normal modularity is thus able to overfit the noise within a network and cannot reliably assess its own performance. In networks without known communities (*e.g* most real-world networks) *multimodbp* can better assess whether there community structure is actually present. In addition, while we have assessed the performance of our algorithm in the previous sections using a hard partitioning of the network, one of the advantages of our method is the ability to generate a soft partitioning by using the computed marginals for each node-layer. In the next section we showcase how the

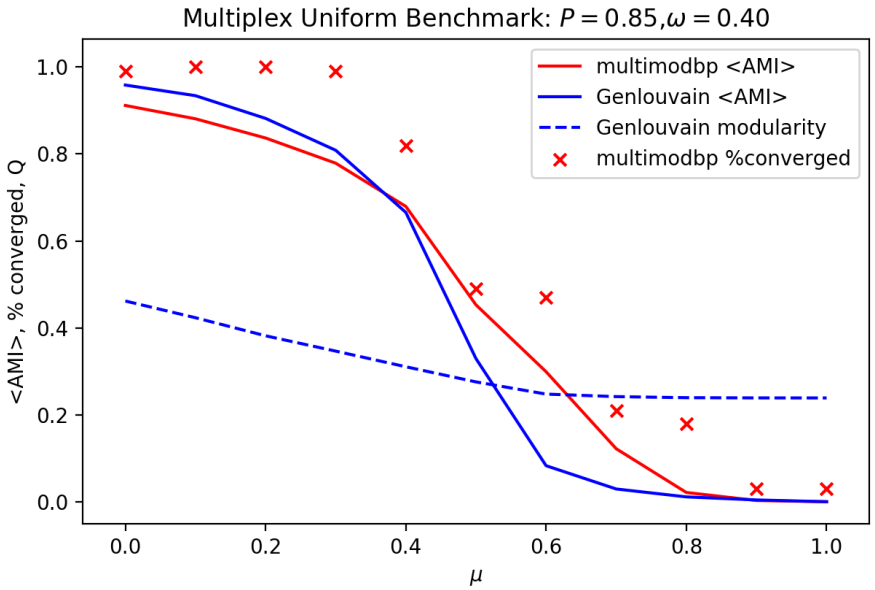

Figure 3.10: Detectability of communities in the uniform multiplex benchmarking network (with $p = .85$) as μ is varied. We plot the average $\langle \text{AMI} \rangle$ of the detected communities for both *multimodbp* (solid red line) and for *GenLouvain* (solid blue line). We also show the average modularity of the partitions identified by *GenLouvain* (dashed blue line) as well as percentage of trials that converged to a non-trivial solution for *multimodbp*

information can be used to interpret the structure of two real world multilayer networks.

Real world multilayer networks

We conclude our results by demonstrating the inferences that can be made on real-world networks using the additional information provided by *multimodbp*. We begin with the US Senate voting similarity network as introduced by [242] and analyzed in [155]. This dataset represents the voting similarity patterns of 1,884 U.S. Senators over 110 Congresses starting in 1789. Each 2-year Congress beginning in the January following an election is represented as a layer within this network. A node within a layer represents a Senator serving in that Congress with Senators serving in consecutive Congresses linked through interlayer edges. In the analysis performed here, the network was modified to sparsify the intralayer connections by taking the K-nearest neighbors (KNN) of each Senator based on voting correlations (using K=10) while keeping the edges with the original weights based on voting correlations.

In Figure 3.11.A we show the correspondence between the retrieval modularity, the Bethe free energy (Eq 3.35), and the AMI with the political party labels of partitions identified across a

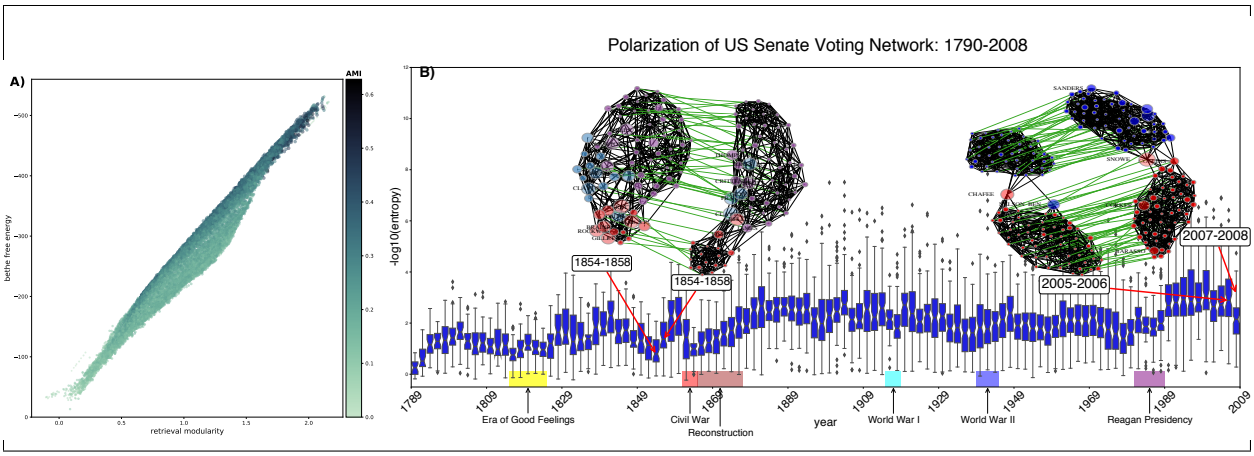

Figure 3.11: Application of *multimodbp* to US Senate voting network. We ran *multimodbp* on the US Senate voting similarity network comprised of 1884 Senators across the first 110 Congresses [155, 242]. **A)** The relationship between the retrieval modularity (x-axis) and the Bethe free energy is given by equation Eq 3.35. The Bethe free energy correlates strongly with modularity of a partition, and the partitions with the lowest free energy tend to correspond best with the underlying party structure. **B)** We examined the distribution of the average Senator entropy for each Congress (layer) in the network. Inset graphs depict how changes in average entropy correspond with network structure and the overall level of polarization within the network. Node size depicts the average entropy level of Senators with "high entropy". Senators labeled

range of the (γ,ω) parameter space. Each point represents a partition identified using *multimodbp*. The belief propagation algorithm fixed points are actually minimizers of the Bethe free energy (rather than optimizers of the retrieval modularity). We see in general that partitions that minimize the Bethe free energy produce high retrieval modularities. Optimizing the Bethe free energy also produces partitions that accurately reflect the underlying known structure in the data set (*i.e.*, the political party affiliations of the Senators), shown by the color of the scatter points in Figure 3.11. We show a comparison of these partitions with the real party layouts in Figure 23. It appears that the most appropriate choice (in this sense) of the *multimodbp* parameters are around $(\omega = 6, \gamma = 0.5)$.

One of the main benefits of using the belief propagation approach for community detection is that we can obtain a measure of how confident we are in the predicted community for each node. In Figure 3.11.B, we show the distribution of Senator entropies for each Congress, averaged over the top 200 partitions identified (by AMI with parties). On the y-axis we plot the distribution of $-\log_{10}(\text{entropy})$ across all Senators as a measure of how strongly identified the communities are and thus how polarized Congress is along party lines. We have highlighted several periods of American history such as the *Era of Good Feelings* with corresponding low polarization/high entropy, or the high level of polarization immediately preceding the Civil War.

The insets show how the corresponding changes in entropy from Congress to Congress are reflected in the community structure of the graph. This is consistent with the increasing level of polarization identified by Moody *et al.* in their study of this data set[152]. Our method gives the further benefit of providing a node level metric to identify how strongly a node is connected with its community. In Figure 3.11.B we have labeled the "high entropy" Senators, those whose voting patterns indicate a measure of bipartisanship (or independence from the party as in the case of Bernie Sanders in the 2007–08 Congress). Thus node-level information contained in the marginals allows for an extra layer of interpretability about the community structure. This has allowed us to assess performance of *multimodbp* on a real world network with an approach that is orthogonal to comparing detected communities with the meta data on the network. This is particularly notable given the difficulties with assessing community detection approach solely on the basis of alignment with metadata attributes as is discussed in [179].

The second real-world network that we have analyzed is the Lazega Lawyer network introduced by [127]. We scan the (γ, ω) parameter range $[0, 3] \times [0, 3]$ and select the partition with the highest retrieval modularity, $Q(\{\hat{t}\})$ at each point. In Figure 3.12, we show the number of iterations taken by the converged partitions for different parameter choices of (γ, ω). Within the lower right quadrant (high γ, low ω) the algorithm only converged for a small range of β values. In the top row, middle panel, we see that for this network three communities were chosen for a large portion of the parameter space, although the structure of the identified partitions varied widely. In the top right panel of Figure 3.12.A, we look at the average entropy per node across the parameter space to identify regions where node ambiguity is minimized. These suggest another way to identify regions of the (ω, γ) with corresponding strong community structure. We see that there are a few partitions with quite low entropy for $\gamma < 1$, and that average entropy tends to increase past this threshold. The region where the algorithm converged for very few values of β (lower right corner) also tends to have the highest entropy. In the bottom two rows we have explored how each partition overlays with a particular metadata attribute within a given layer. For instance the panel titled "office-friends" shows the AMI of all partitions with the office attribute only within the friends layer. We see that within different parts of the parameter space, different features of the metadata align more closely with the partitions identified. For instance

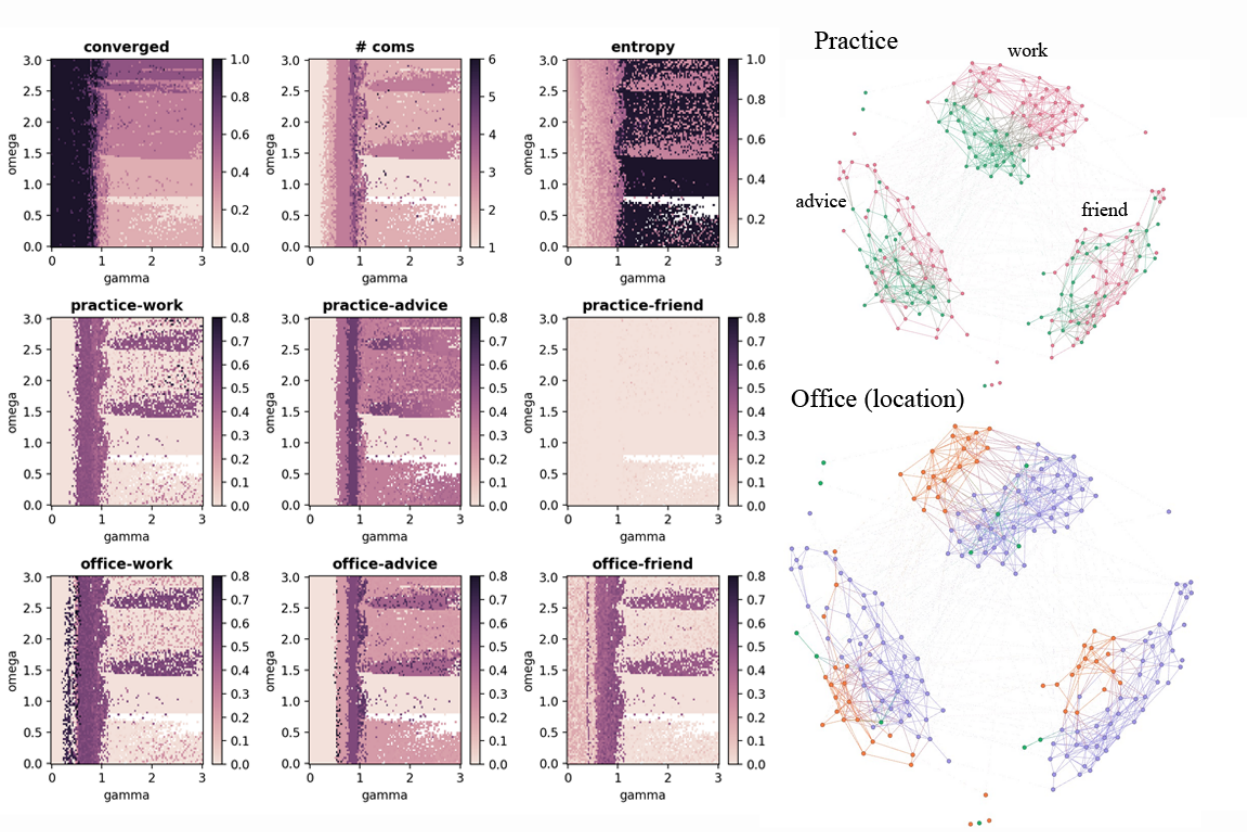

Figure 3.12: Several visualizations of the Lazega Lawyer network [127]. On the right we show several characteristics of partitions identified with *multimodbp* at various values of γ (x-axes) and ω (y-axes). In the top row, from left to right, we show how many times the algorithm converged over 10 runs at different β values, the number of communities identified by the best run for each set of parameters (based on lowest Bethe free energy), and the average entropy of the marginals across all of the nodes for each of these partitions. In the next two rows we show the AMI of the identified partition within a single-layer and a specified metadata attribute. For example in the left most panel of the second row, we show how the "practice" (which type of law practiced by each node) attribute lines up with the partitioning of work layer. To the right in **B)** we show the three layers of the network (advice, work, friends) colored by two of the metadata attributes, practice (which specialty of law each person is involved in) and office (which is the location the person works in). Showing the partitions in this manner demonstrates how different metadata attributes affect the community structure in the different layers and how this is best captured by *multimodbp* for different values of γ and ω.

there is a narrow band from $\gamma \in (0.5, 0.9)$ for which the practice attribute strongly aligns with the community structure of the advice network. For higher values of γ the advice layer switches to being more aligned with the office metadata. Similarly, within the "work" layer, we see that the practice attribute contributes most significantly to the structure identified at a similar $\gamma \in (0.5, 0.9)$ regime, however the office attribute actually contributes more at higher γ and lower ω. Our results for this network complement those derived in [179] suggesting that no single metadata attribute explains the structure of this network. These results highlight the need to explore and summarize partitions across different parameter ranges.

3.4 Discussion: Benefits of an ensemble based approach

We have presented *multimodbp*, an extension of the modularity-based belief propagation framework to multilayer modularity. Like the original belief propagation framework for modularity [259], there are a number of features of *multimodbp* that make it a useful tool for identifying community structure in real-world multilayer networks. At its core, modularity and its multilayer extension are objective functions for assessing community structure and do not allow for true statistical inference[6](cf. generative approaches like the stochastic block model, e.g., [81, 184, 212, 218] for example). However, by formulating multilayer modularity optimization from the perspective of a Boltzmann ensemble, we can obtain an estimate of the uncertainty of assignment at each node from its marginal. The marginals reflect how much shifting a node from one community to another changes the modularity and thus is a measure of how strongly a node prefers a certain community. In this sense we can find a "soft" partitioning of the nodes, in which one node may belong to multiple communities, along with confidence levels corresponding to each community. We have shown in two real world examples how knowledge at the node level about the confidence in the community prediction can inform interpretation of the community structure of the graph. Most modularity-based algorithms do not allow for overlapping communities with a few notable exceptions including OverMod [23] and the fuzzy c-means [263], both of which require an initial disjoint partitioning of the network in order to identify overlaps.

[6]Newman has shown that optimizing modularity is equivalent to the MLE for a planted partition of the degree corrected stochastic block model for a certain value of γ [165]. Likewise Pamfil *et al.* showed an equivalence for multilayer modularity for a multilayer SBM with both temporal and multiplex topologies [174].

Other versions of overlapping modularity-like approaches include [87, 138]. Our approach is useful in that it can be used for either a hard or soft partitioning of the network depending on the desired context.

Meanwhile, although the method of Zhang and Moore allows for the selection of the number of communities by identifying the value of q for which the retrieval modularity plateaus [259], we have shown that this approach fails to perform optimally in a number of cases. This underscores the need for greater flexibility as provided by incorporation of the resolution parameter γ. Rather than searching along the domain of q, we allow q to float (up to a certain point $q_{\max}$) and search along the γ domain to characterize network structure. The flexibility added by the resolution parameter becomes even more important in the multilayer context. We have shown that performance of *multimodbp* is optimized by different combinations of (γ, ω) in different parameter regimes of the dynamic stochastic block model. This is consistent with the work of Newman who demonstrated a link between the resolution parameter γ of modularity and the p_{in} and p_{out} parameters of the degree-corrected stochastic block model [165]. Recently, Pamfil *et al.* extended this approach to multilayer modularity, deriving a similar mapping between the coupling parameter, ω, and the parameters of a model very similar to the DSBM studied here [174].

One of the greatest benefits of the *multimodbp* approach is that the convergence of the algorithm to non-trivial solutions reveals the existence of significant community structure above what would be expected at random. Several prior works have shown that even in randomly-generated networks without underlying structure, modularity optimization heuristics are capable of finding high-modularity partitions [10, 49, 259]. For this reason alone we believe an extension of modularity belief propagation for multilayer networks provides a valuable new tool for network analysis. We have shown that our algorithm performs comparably to *GenLouvain* across a range of multilayer topologies and that its convergence properties can be useful in determining whether significant community structure is present.

There remain a number of technical challenges for implementing *multimodbp* at scale. The runtime of the algorithm depends greatly on the number of iterations of belief propagation that are required to run before convergence. As described in Zhang and Moore, this tends to spike as you approach the retrieval phase, and the formula for β^* we have used tends to yield values

slightly above where this spike occurs. Ideally, one could have an adaptive solution, identifying a value of β for which the algorithm appears to be converging quickly early on and adjusting β once the algorithm is closer to converging. Eventually, we would like to devise an automatic method of selecting an appropriate value for β based on a preliminary scan of convergence rates across the β domain, similarly to how we iteratively select the appropriate number of communities as the algorithm runs. Another issue is the dependency of the runtime and memory of the algorithm on the number of marginals being optimized. We try to reduce the dimension of the marginals after the algorithm has run, by attempting to combine redundant dimensions (those that are highly correlated). One could imagine attempting such a reduction earlier on after a few course-grained runs of the algorithm to produce additional performance gains.

To facilitate use of (and possible improvements on) our method, we have written and distributed a Python package available on PyPI [237].

3.5 Additional Methods for Chapter 3

3.5.1 Selection of β

By analyzing the linearized stability of the fixed point to small, uncorrelated perturbations, Zhang and Moore provided a heuristic for selecting an appropriate value of $\beta = \beta^*$ at which point the trivial, factorized solution ($\psi_t^{j\to i} = 1/q$ for all beliefs) is no longer stable, assuming a random distribution of edges conditioned on the degree distribution. If significant structure is *not* present within the network, for values of $\beta > \beta^*$, the algorithm enters the 'spin-glass' phase in which convergence never occurs. In contrast, if the network has detectable community structure, then there is a range of values, $\beta^R < \beta < \beta^{SG}$ where a retrieval state has lower free energy than the trivial solution and is stable. Typically, β^* is greater than β^R and is within the retrieval phase. We demonstrate empirically that β^* is indeed within the retrieval phase in supplement Figures 16, 17, and 20. However, in principle for real-world networks, β^* could exist outside of the retrieval phase, in which case it would be necessary to scan a wider range of β values.

Practically this can be used to eliminate or at least reduce one of the free parameters involved in running the algorithm. Shi *et al.* [208] recently expanded the stability analysis around the fixed point for the case where random weights are added on the edges. We have adopted their

heuristic for selecting β^* in the multilayer context, arguing that in the limit of small perturbations, the intralayer field term does not contribute to the linearized form of the update equations. The linear stability of the factorized solution is characterized by the derivatives of the messages with respect to each other at the fixed point ($1/q$). To identify β^*, the critical value for instability with respect to random, uncorrelated perturbations, we linearize the *multimodbp* update equations (Eq 3.34) and then analyze the stability of the equations under repeated iteration. We use the notation from Zhang and Moore and Shi *et al.* Suppose that each belief is perturbed by a small random amount, $\psi_{c_j}^{i\to j} = \frac{1}{q} + \epsilon_{c_j}^{i\to j}$, these perturbations will propagate to first order by :

$$\epsilon_{c_i}^{i\to j} = \sum_{k\in\partial i\setminus j} \sum_{c_k} T_{c_i,c_k}^{i\to j,k\to i} \epsilon_{c_k}^{k\to i} \,, \tag{3.38}$$

where

$$T_{c_i,c_k}^{i\to j,k\to i} = \left.\frac{\partial \psi_{c_i}^{i\to j}}{\partial \psi_{c_k}^{k\to i}}\right|_{1/q} . \tag{3.39}$$

We provide a derivation for the form of $T_{c_i,c_k}^{i\to j,k\to i}$ in the supplement, Section 3.3 and show that its largest eigenvalue is:

$$\eta_{ij} = \frac{e^{\beta\tilde{A}_{ij}} - 1}{e^{\beta\tilde{A}_{ij}} + q - 1} \,, \tag{3.40}$$

where $\tilde{A}_{ij} = A_{ij}\delta(l_i, l_j) + \omega C_{ij}(1 - \delta(l_i, l_j))$ defines the appropriate weight and connectivity between nodes i and j. Shi *et al.* show that the message will only remains stable if the variance of the perturbations remains less than one over an arbitrary length path in the graph, providing the following equation:

$$\left\langle \left(\frac{e^{\beta^*\tilde{A}_{ij}} - 1}{e^{\beta^*\tilde{A}_{ij}} + q - 1} \right)^2 \right\rangle_{ij} \hat{c} = 1 \,, \tag{3.41}$$

where $\hat{c} = \frac{<d^2>}{c} - 1$ is the average excess degree of the network, and the expectation is taken over all non-zero edge-weights. We can solve this equation to identify the appropriate β^* that appropriately incorporates both the weights on the edges of the networks as well as the interlayer coupling ω. We use a root finder to solve Equation 3.41 for $\beta^*(\tilde{A}_{ij}|q, \omega)$.

We have found that this heuristic works well in identifying values of β for which our method converges. We note that β^* represents the boundary for stability of the solution for uncorrelated perturbations in the beliefs. In the case when detectable community structure

exists, the messages become correlated with each other and the transition from the trivial paramagnetic phase to the retrieval phase is generally lower than β^* [259]. Thus, choosing values of β near β^* works well in practice. Additionally, Schülke *el al.* showed that in many networks there can be multiple zones of the retrieval phase corresponding to detecting communities at different scales [206]. Therefore, in our experiments, we run the algorithm for a range of $\{\beta_q^*\} = [\beta^*(q=2), \ldots \beta^*(q=q_{\max})]$, where $q_{\max}$ is some reasonable upper limit for the number of communities in a particular network. We have found that this approach identifies a reasonable retrieval phase for the networks tried in this chapter. For example in Figure 20, we show how several of the $\{\beta_q^*\}$ consistently lie within the retrieval phase for the US Senate voting network discussed in Section 3.3.2.

We emphasize that like the original Zhang and Moore approach as well that that by Shi *et al.*, our heuristic assumes a sparse, tree-like network as well as randomly distributed edges and edge weights and provides no guarantees that β^* will be found within the retrieval phase. For certain networks, scanning a larger range of β will be necessary, though in practice we have found that the approach above is fairly robust. We note that while it is possible that a fairly small retrieval phase could be missed by such an approach, in our experiments this approach for selecting β^* has identified values of β for which the algorithm converges close to the known detection limit (see Figure 3.7). In running the algorithm, we also set un upper limit to the number of message passing iterations allowed before we say that the algorithm has not converged. We generally select this to be several hundred times the number of iterations at which the algorithm converges to the fixed point ($\psi_t^i = 1/q, \forall i, t$) for smaller values of β.

3.5.2 Selection of number of communities, q

One critical issue with many community detection algorithms is in selecting the appropriate number of communities. In the context of modularity, adjusting the resolution parameter γ can reveal communities of different scale and size, overcoming the "resolution limit of detection" first raised by [63]. Since then there have been several approaches showing how the scale of the community structures identified varies with the resolution parameter (see, e.g., the discussion and references in [244]).

Zhang and Moore do not include a resolution parameter in deriving their *modbp*

algorithm (thereby implicitly setting $\gamma = 1$ in Eq 3.26), instead suggesting an alternative approach for selecting the appropriate number of communities. They show in several examples that the maximum modularity achieved in the retrieval phase of the algorithm peaks at certain numbers of communities. They suggest that this peak identifies the correct value for q, the number of communities, where there is no additional increase in the retrieval modularity $Q(\{c_i\})$. However, this approach requires running *modbp* for many possible values of q, and then choosing an arbitrary threshold when modularity is no longer sufficiently increasing to establish the correct value of q. In many cases, selecting an exact value of q is made difficult because of fluctuations in the retrieval modularity near the β^* value derived by Zhang and Moore. Figure 19 in the supplement illustrates how choosing q is challenging in practice by these considerations. Meanwhile, selecting the number of communities in this manner implicitly uses the value $\gamma = 1$, which has been shown to return non-ideal partitions in synthetic and real-world networks (see, e.g., [7, 63, 165, 224]). We show in Section 3.3.1 the positive impact of using different values for γ on several different networks.

There have been two other approaches to selecting the appropriate number of communities using *modbp* without having to run the algorithm at many values of q. Both approaches involve selecting a $q_{\max}$, the largest possible number of communities, and then using similarities in the marginal probabilities of assignments to evaluate the true number of communities. Lai *et al.* [121] noted that in the event that q is too large, many of the marginal community assignments will be highly correlated, and highly correlated states (community assignments) can be condensed into a single group. Similarly, Ref. [206] condenses the community assignments on the basis of the average distance between the marginals across all nodes in the network. In practice, we have found that for the default resolution ($\gamma = 1$), choosing the number of communities this way all but obliterates the retrieval phase if $q_{\max}$ is chosen to be too much larger than the actual number. We have implemented the method in Ref. [206], letting the number of communities float up to a pre-specified $q_{\max}$ (See Section 3.5.1), and condensing together communities that have closely aligned marginals. We show that incorporation of a resolution parameter γ restores the width of the retrieval phase and returns values closer to the correct number of communities. As previously mentioned, because we do not specify a single value of q; rather, we run the algorithm across a range of $\beta = [\beta^*(c, q = 2), \ldots, \beta^*(c, q = q_{\max})]$

where the formula for $\beta^*(c, q)$ is given obtain using Eq. 3.41. We have found that this provides a reasonable range of β values to search within and that performance of the algorithm does not depend on the precise value of β, as long as it is within the retrieval phase.

3.5.3 Cross-layer community alignment

In running *multimodbp* at low levels of interlayer coupling (ω) on multilayer networks with both temporal and multiplex coupling topology (e.g the dynamic stochastic block model described in Section 3.3.2 and the multiplex networks in Section 3.3.2), we frequently observed that the intralayer marginals would rapidly converge to communities that remained misaligned between layers. Such misalignment would then typically lead to "fragmented" partitions as shown in Figure 21 as well as a lower AMI. For these partitions, within any single-layer the AMI of the partition with the ground truth with that layer would be very high, but the total AMI over the entire multilayer data would become much lower. To correct for this issue, we implemented a greedy heuristic to explicitly permute the community assignments within certain layers in order to maximize local alignment between neighboring layers. Specifically, we identify the layer x that has the greatest number of nodes (of those present in both layers) that change community identity from the previous layer, y. We then find the matching of community labels in x that best matches those observed in y; that is, we minimize the total number of mismatches across layers x and y:

$$C(x, y) = \sum_{i \in \mathcal{V}_x} \sum_{j \in \mathcal{V}_y} [(c_i \neq c_j) \wedge \mathbb{I}((i, j) \in \mathcal{E}_{inter})] \tag{3.42}$$

Once the optimal bipartite matching has been identified [118], the community labels in layer x and every subsequent layer are rearranged according to that matching (with community labels in subsequent layers that are not present in either layer x or y remaining unchanged). We then repeat this procedure until no further labels are changed (*i.e.* the optimal matching is the identity at the layer where the greatest change occurs). We note that this procedure does not alter the community structure identified within any particular layer, maintaining nodes that have been grouped together. Rather, this procedure aligns the community labels between layers in a way that always increases the retrieval modularity, thereby improving the computed results. This approach assumes a notion of persistent community across inherently *ordered* layers which is

appropriate in the temporal multilayer setting. In the multiplex case, for each layer we permute communities in order to minimize that layers differences across with all other layers $\sum_{y \neq x} C(x, y)$, cycling through the layers in random order until no permutations are found. This heuristic is the same as the *interlayer merging* developed by Bazzi *et al.* to overcome a similar problem encountered when optimizing multilayer modularity with the *GenLouvain* algorithm [20].

CHAPTER 4: THE TMB PARADOX

In this chapter, we discuss a particular question in cancer genomics whose solution is revealed more readily when formulated as a network. We dive back into the realm of oncology here and examine the following challenge: when looking across the mutational landscape of many different tumors, can one identify which mutations are associated with increased levels of mutational burden overall. One senses intuitively that there is a bit of a chicken and egg problem here because the cause (mutations in specific genes) is also part of the effect we are measuring (mutations across all genes). We will see how this conundrum leads to a phenomenon which we have dubbed, the "Tumor Mutational Burden (TMB) paradox". We begin with a discussion of the relevance of this question to personalized medicine with a new class of immune based therapies. We detail the previous, univariate approaches and why these give inappropriate results. We reveal how a networks based approach can reveal new insight to the problem and provides a test for such association. We close by showcasing how our results can increase predictive power on several clinical datasets.

4.1 Introduction to TMB Paradox

4.1.1 Immune Checkpoint Blockade (ICB) therapy and TMB

Immune Checkpoint Blockade (ICB) has revolutionized the treatment of many solid tumors achieving remarkable remissions in some cancers, while largely sparing patients from the more toxic side effects of traditional chemotherapy. ICB uses monoclonal antibodies targeting cell surface proteins including cytotoxic T-lymphocyte antigen 4 (CTLA-4), programmed cell death 1 (PD-1), and programmed cell death ligand 1 (PD-L1), all of which serve as inhibitory pathways to activating an adaptive anti-tumor immune response. While some patients achieve a durable response to ICB, the majority of patients either have or develop primary and secondary resistance respectively. While several genomic markers are available to predict which patients

will respond, there is still a large degree of heterogeneity that is not explained by existing markers of response. Continuing to refine and develop novel predictive biomarkers will allow us to apply these drugs more sparingly among patients, as well as understand the basic mechanisms underlying tumor susceptibility.

Tumor mutational burden (TMB), usually defined as the number of synonymous and non-synonymous mutations per megabase sequenced has consistently been associated with response to ICB [35, 89]. A number of recent studies have shown [109] that increased levels of tumor mutational burden are predictive of response to immune checkpoint blockade (ICB) [35, 77, 144, 148, 154, 187, 195, 210, 230, 249, 252] The basis of TMB's correlation with ICB effectiveness is believed to be that higher levels of TMB correspond to more neoantigens, altered expressed proteins that are not recognized as self by the immune system, triggering an antigen-driven immune response [128, 129, 154]. However, TMB levels only explain a fraction of the variation in patient response to ICB [109]. TMB levels can range widely both across and within tumor types [252]. For example, pediatric tumors tend to have very low median levels of TMB, while carcinogen induced tumors such as melanoma, lung, and bladder tumors have much higher median TMB levels [154]. Tumor types with higher levels of TMB typically have better response rates. However, there are a few tumor types such as renal cell carcinoma that have good response rates despite having generally low levels of mutation [252]. Even within high TMB cases, response rates are below 40% of patients, with many high TMB patients failing to respond. Conversely, there are also many low TMB patients that do respond to ICB therapy, suggesting that TMB alone fails to capture the complexity of ICB response and that other predictive biomarkers are needed.

4.1.2 Tumor Mutational Burden and DNA Damage Repair

The level of tumor mutational burden is thought to represent the balance between a tumor's exposure to a mutagenic process (UV radiation, carcinogen, replicative stress) and the integrity of the cellular DNA Damage Repair (DDR) pathways. Multiple studies have now confirmed an association between the inactivation of a DDR gene and increased levels of TMB. Studies have documented this association with individual genes (i.e. POLE) [34, 205], individual DDR pathways such as MMR [35, 67, 266], co-mutations across multiple pathways [240], or the

presence of any DDR mutation [144, 161, 266]. All of these reports have noted a strong correlation between mutations in the DDR genes/pathways examined with higher levels of TMB and in some cases ICB outcome. In contrast to these previous reports, a recent study of DDR mutations across all TCGA tumors found only two DDR genes where mutations were associated with a higher level of TMB than other genes in the cohort [115].

There have been several studies that have reported a link between mutations in DDR pathways and response to ICB therapy. Recently, pembroluzimab was approved by the FDA for treatment in cancers with loss of the mismatch-repair (MMR) machinery, the first cancer drug approved for use on the basis of a biomarker alone, regardless of cancer type [134]. Other research has shown a link between ICB response and mutations in specific genes such as POLE [145], as well as BRCA2 [97]. Importantly, models that include mutations in any DDR genes have had increased predictive power of ICB response over TMB alone [219]. However, Mariathasan *et al.* found that while DDR mutations were able to predict response to ICB, in aggregate they did not provide any explanatory power beyond TMB [144]. It is believed that alterations in DNA Damage Repair could mediate the production of neo-antigens in a way that is not captured by TMB alone [109, 154]. Thus, firmly establishing the causal link between alterations in DNA Damage Repair pathways and elevated TMB has import prognostic and therapeutic implications.

4.1.3 Univariate approach to associating TMB with mutations

Most of the studies examining the link between elevated TMB and mutations in the DDR pathways rely on a univariate test such as the t-test or Mann-Whitney-U test to determine statistical significance. In these univariate tests, the in-group is defined as all samples with a deleterious mutation in a given gene (or pathway), which we refer to as a gene's mutated sample set. The out-group is represented by all other samples in the cohort. However, this approach is confounded by a bi-directional feedback loop between our readout of interest (levels of TMB) and the variable we are trying to relate it to (mutations in DDR genes). If there is a mutation in a gene that leads to higher levels of genomic instability and increased mutation frequency (e.g. a DDR gene), we are more likely to observe mutations within that gene as well as the other genes we are testing (such as other DDR pathways). This leads to a bias in the selection of samples for the comparison groups of the univariate approach.

Furthermore, aggregating mutations across genes within a pathway or multiple pathways increases the effect of this bias. We suggest a networks based approach that accounts for this issue, and demonstrate our approach on the TCGA pan-cancer dataset, identifying the DDR pathways that are associated with higher levels of TMB. We validate these findings in an independent, novel dataset from an academic-industry collaboration, the Precision Medicine Exchange Consortium (PMEC). The PMEC dataset complements the TCGA data well in that many of the gene and tumor-level characteristics agree (see supplemental Figure 25) and that it is similar in size and diversity of tumor types. We highlight how the results suggested by our approach differ markedly from the univariate approach described above. Finally, we find that mutations in the DDR genes that are not associated with higher levels of TMB are independently predictive of immune checkpoint blockade (ICB) therapy response in two clinical data sets.

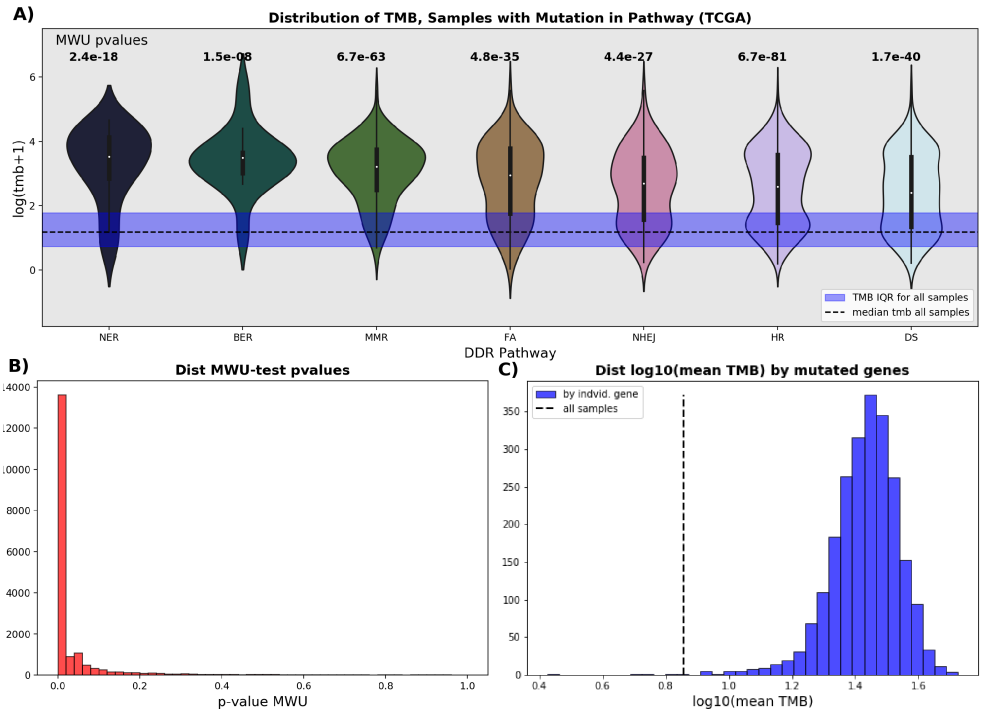

Figure 4.1: Application of univariate approach to DDR pathways and comparison against all genes A) We show the distribution of Tumor mutational burdens (TMB) for all samples with a mutation in each of the DNA Damage Repair (DDR) pathways. Dashed line shows the median TMB for all TCGA samples (including those with mutations in the DDR pathways) with light blue line showing the interquartile range. Mann-Whitney U p-values are calculated by comparing the distribution of TMB for samples with any mutation in the genes that define a pathway (counting only once if they have multiple mutations) to the distribution of all samples, again including those with mutations in the DDR pathway. B) Distribution of Mann-Whitney U test p-values (without multiple test correction) across all genes in TCGA. For each gene MWU test compares distribution of TMB values for samples with a mutation in the gene vs all samples in cohort. C) Distribution of mean TMB values for mutated sample set for each gene compared with the overall mean TMB for the cohort (dashed black line).

The most straightforward analysis to identify the genes in which a disrupting mutation leads to higher levels of TMB is to compare the distribution of TMB in samples with a mutation in a given gene to those without using a statistic like a t-test or the non-parametric Mann-Whitney U (MWU) test. Because the mutational frequency of most individual genes is quite low, greater power can be achieved if we combine mutations across identified genomic pathways, again comparing samples with a mutation in each pathway to those without any such mutation. Depending on the number of genes to be considered, there are often a number of samples with mutations in multiple genes or pathways. These samples might simply be ignored, or considered as part of multiple in-groups for the analysis. Throughout this chapter we have used the core DNA Damage Repair pathway assignments as defined by Knijnenbrug *et al.* [115], shown in

Table 4.1.[1] In Figure 4.1 we show how this analysis finds that all of the DNA Damage repair pathways are highly associated with elevated levels of TMB. We have computed Mann-Whitney-U p-values here using a more conservative approach: we use all samples for the out-group in the comparison, including those with mutations in the DDR pathway we are assessing. This has the benefit that each DDR pathway is compared to the same out-group. Despite this we find very significant p-values ranging from $1.5e{-}08$ to $6.7e{-}81$. We found similar results using the mutation data in the PMEC dataset, although a generally lower significance level (p-values in the range of $1.2e{-}07$ to $2.6e{-}55$)

We also examined how the DDR pathway genes compared to the overall set of genes. Figure 4.1.B shows that the vast majority of genes were found to have a significant association with elevated TMB based on the MWU test (around 70% of genes have p-value <.01.) This is reinforced by Knijnenburg *et al.*'s finding that most of the DNA Damage repair genes were uniquely associated with higher levels of TMB as compared to other genes (see Figure S2.K)[115]. Surprisingly, when we looked at the distribution of mean TMB values of samples harboring a mutation in each gene in Figure 4.1.C, we found that the majority of genes have a mean TMB well above the average TMB of the whole cohort. Because it seems paradoxical that almost all individual genes have an associated higher median or mean TMB than the median or mean of all samples, we have labeled this phenomenon, "The TMB Paradox". We refer to finding as the TMB Paradox: almost all genes have a higher median/mean TMB than if one considers the dataset as a whole. As the t-test and Mann-Whitney U test are testing for differences in the central tendencies (i.e. the mean or the median respectively) between two distributions, this observation suggests that the majority of genes will have a highly statistically significant association with elevated TMBs.

We also wondered whether or not the current test was biased towards significance for genes with a greater mutational frequency.[2] Figure 4.2 shows the mean level of tumor mutational

[1]The presence of a given (usually highly mutated) samples across multiple in-groups is particularly problematic when testing at the pathway level because often times a given gene is in multiple pathways. While the DDR classification schema we have used assigns each gene to only one pathway, given the extensive crosstalk between the different pathways, many schemas (including the inclusive schema from [115]) contain overlapping memberships.

[2]The mutation frequency for a given gene is the proportion of samples with a mutation in that gene for a given cohort.

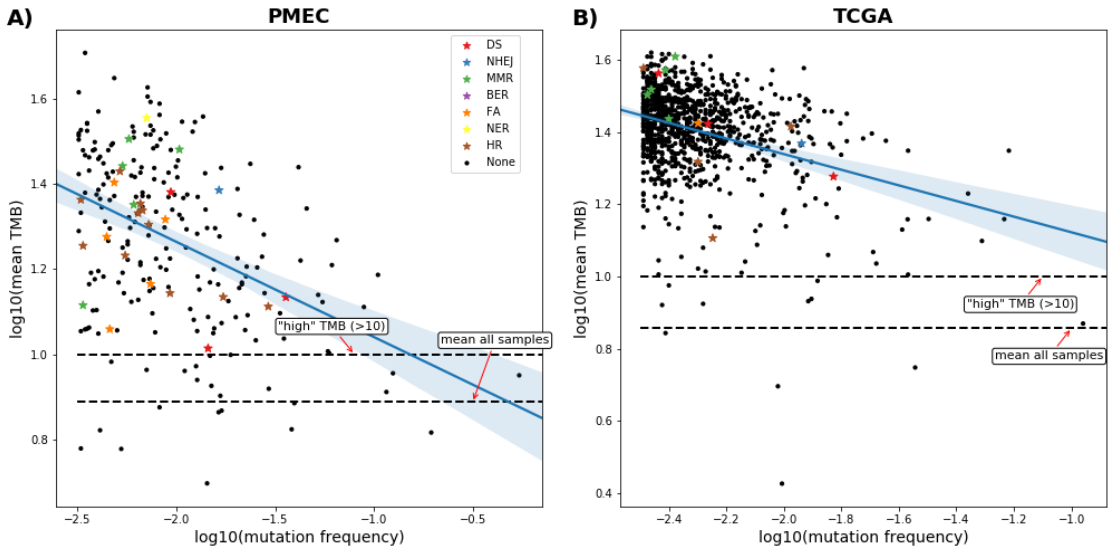

Figure 4.2: Comparison of mutation frequency with mean TMB of mutated sample set. Gene level mutation frequency (x-axis) plotted against the mean TMB for all samples with a mutation in each gene. (y-axis) for the PMEC **A)** and the TCGA **B)**. Each scatter point represents a unique gene in the dataset, with DNA Damage Repair genes starred and colored according to their pathway (see legend in panel A). Dashed horizontal lines depict the threshold for high TMB (> 10) as well as the mean TMB for all samples within the data set (lower line in each panel). Regression curve show slight negative relationship between mutation frequency and the median level of TMB for samples with a mutation in that gene.

burden for samples with a mutation in each gene versus the mutational frequency for the corresponding genes. We have plotted all genes (including the DDR genes with colored stars) for both PMEC and TCGA. We see that there is a slight negative trend in both datasets with higher mutation frequencies predicting a lower mean TMB. We believe that this weak association represents yet another artifact of the univariate approach rather than a true biological trend. Figure 4.2 also reinforces that the mutated sample set for most genes has a mean TMB that is above what would classify as "high" TMB (> 10) and nearly all genes have mean TMB above the mean TMB for all samples as a whole.

This finding is similar to the well known "friendship paradox" in network science [59]. The friendship paradox states that for most nodes in a network, their neighbors will on average have a higher degree than the node itself does. By re-casting our data into a network of a particular form, we can show how the TMB paradox is a case of the friendship paradox and how this form of oversampling bias arises. We propose an appropriate statistical test for whether mutations in a given gene are associated with higher levels of TMB. We expect that a statistical test built on a network analysis that corrects for the TMB Paradox would 1) yield a uniform

distribution of p-values, with only a small subset of genes found significant; 2) exhibit no relationship between mutation frequency of a gene and TMB; and 3) be independent of gene size.

4.2 Networks-based approach to identify gene-TMB associations

4.2.1 Bipartite network representation for mutations data

To explain why the TMB Paradox arises in the previously described tests, we first show how our mutational data can be recast as a network. In Figure 4.3, we demonstrate how a matrix that encodes which genes are mutated in each sample can be converted into a bipartite network. A network is a collection of nodes and edges that represent pairwise interactions between these nodes. Networks provide a powerful framework for analyzing complex systems and have been successfully applied in numerous contexts in oncology including cancer subtyping [90, 139], identification of driver mutations [39, 217, 220, 258], and the identification of dysregulated pathways associated with cancer development and prognosis [112, 133, 231]. See Section 1.3 for a more detailed review of networks based methods in oncology.

A bipartite network is a network where the nodes are in one of two classes and edges are only allowed between nodes of different classes [167]. In the context of this chapter, the two classes of nodes are the genes and the tumor samples as shown in Figure 4.3. For each of the

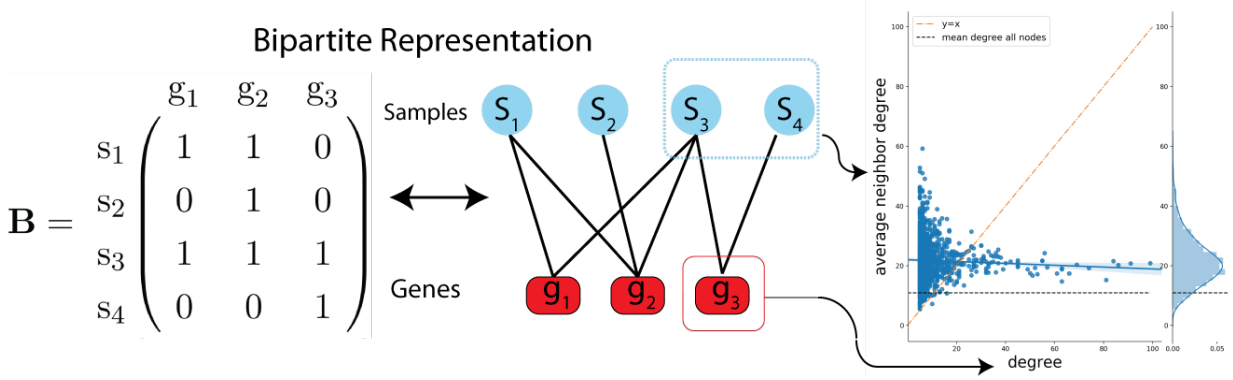

Figure 4.3: Schematic representation of converting our mutational data in matrix from, B to a bipartite network. The two classes of nodes are the genes and the samples. Each sample is connected to a gene if that sample has a mutation within that gene. For simplicity, we consider an unweighted bipartite network since it is rare for a sample to have multiple mutations in the same gene. Far right panel depicts the friendship paradox for a randomly generated (non-bipartite) network using a common, synthetic benchmark model (Lancichinetti, Fortunato, & Radicchi, 2008) . Each scatter plot represents a node in the network, with the x-axis showing the node's degree, and the y-axis showing the average neighbor degree. The scatter plots that are above the y=x line (dashed orange line) have a higher average neighbor degree than their own degree. We see that this is the case for most nodes in the network.

datasets, we represent the mutational data as an N (genes) by M (samples) binary matrix with each entry denoting whether a particular sample has a mutation within a given gene by a 1 and a zero otherwise. Figure 4.3 shows the equivalence between a representative mutational matrix, $\mathbf{B}$, and the corresponding bipartite network. We emphasize that, by construction, edges can only occur between the different classes of nodes, in this case between a gene and a tumor sample.

In the bipartite network representation of the mutation data, a gene's mutation frequency is proportional to its node's degree (*i.e.* the number of neighbors it has), since by definition each edge represents a mutation of that gene in a sample. This quantity is directly proportional to the mutation frequency of the gene (modulo M, the number of samples observed, which is the same for every gene). Likewise, each tumor sample's total mutational count will be proportional to its degree in the network. We have found empirically that the total number of mutations for a given sample is very tightly correlated ($R^2 = .91$ for TCGA) with the TMB of the sample (which incorporates other factors such as sequencing depth).

The friendship paradox states that for the majority of nodes in a network, their degree will be less than the average degree of their neighbors. This is referred to as the friendship paradox because in social networks representing friendships (edges) between individuals (nodes), it implies that for most individuals in the network, their friends will on average have more friends than they do. This phenomenon arises because nodes with large degrees are counted in the average neighbor degree for many different individuals; they are oversampled and play an outsized role in computing the average neighbor's degrees. We offer a proof the friendship paradox in the supplemental section 4.1.

In our bipartite model, the average neighbor degree[3] for any given gene is the average degree of the samples with a mutation in that gene. For example, in Figure 4.3 the circled node labeled g_3 would have an average neighbor degree of $(1 + 3)/2 = 2$. In the case of the bipartite network representing our mutational data, the average neighbor degree for each gene is the mean TMB of the samples with a mutation in that gene. In network science, the average degree of a

[3]The distribution of neighboring degrees is closely related to a quantity known as the excess degree. The excess degree is the distribution of degrees you get from picking a random edge in the network and looking at the degree of one of the nodes attached to that edge, not counting the edge you travelled along [167]. Therefore the average excess degree is really the average degree of the neighbors minus 1.

node's neighbors is related to called the excess degree [167].[4]

This phenomenon applies to all networks with non-trivial degree distributions[5] and becomes more pronounced as the variance of the degree distribution gets larger. That is as the degree distribution becomes more fat-tailed, the divergence between the average neighbor degree and the average degree increases [59]. On the far right panel of Figure 4.3, we demonstrate this phenomenon for a randomly generated network using the Lancichinetti-Fortunado-Radicchi (LFR) generative model for random networks [125]. The far right of the plot depicts the marginal distribution of the neighboring degrees, with the mean degree of the network represented by the horizontal, dashed line. This plot shows that most of the nodes in the network have a significantly higher mean neighbor degree than their degree.

The phenomenon we have described in our mutational datasets, shown in Figures 4.1 and 4.2, is really a special case of the friendship paradox. In our bipartite network representation of our data, the mutational frequency of the genes in our network is equated with the degree of the gene nodes in our network. Likewise the mean TMB of the samples mutated in each gene is represented by the average neighbor degree of the gene nodes. Application of the friendship paradox dictates that most genes will be associated with a higher level of TMB than the distribution of the samples as a whole. This phenomenon is augmented by the fact that the distribution of mutations across the tumor samples is fat tailed. Most of the samples have relatively few mutations (in the TCGA dataset, 78% of the tumors have less mutations than the average), while a few samples are highly mutated. In recognizing this TMB paradox at play, we are in a position to test for associations between mutations in each genes and elevated TMB in a way that respects the underlying constraints of the network data.

4.2.2 Sampling from the bipartite configuration model

In recognizing the effect that the distribution of mutation frequency (for both tumor samples and genes) plays in biasing our test statistic, we construct a novel test that corrects for

[4]The excess degree is actually the degree of the neighbor minus the edge travelled along to reach the neighbor. Therefore the average excess degree is really the average degree of the neighbors minus 1.

[5]For example a ring or other k-regular graph would not exhibit this. See [173] for more examples of special graphs where the paradox does not hold.

this TMB paradox. Our test consists of comparing the observed mean TMB for each gene or pathway's mutated sample set against the expected distribution under random sampling of bipartite networks that match the degree distribution of the original dataset. The null model for networks in which all networks with a given degree sequence are uniformly likely is known as the configuration model [167], which has also been extended to bipartite graphs [203]. A useful way to visualize the bipartite configuration model is demonstrated in Figure 4.4.A: we cut across the edges in the original network and reconnect the "stubs" at random with each possible sets of pairings respecting the bipartite structure of the original network and being equally likely under the model.

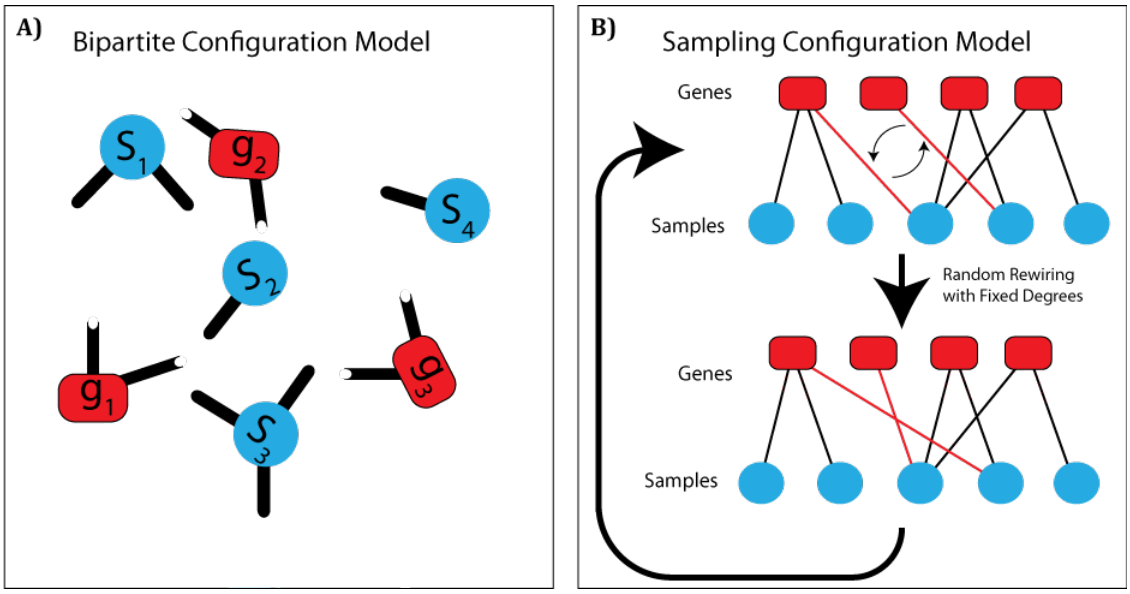

Figure 4.4: Sampling from the bipartite configuration model A) Depiction of the configuration null model for a given network. Edges are 'cut' in two leaving stubs, which can be connected to any other stub as long as the bipartite structure is maintained. In the configuration model, each valid arrangement is equally likely to occur. **B)** We can sample from the configuration model by repeated rewiring of the network. The samples from the model will be independent as long as a sufficient number of rewires has occurred between each sample.

In Figure 4.4.B we demonstrate a method for sampling from this space of network using a rewiring procedure as analyzed in [65]. We note that we sample the bipartite configuration model using a series of rewiring steps rather than the more direct "stub matching" approach to ensure that we sample the appropriate, more restricted model without self-loops and multi-edges [65].[6]

[6]Although the bipartite model without self-loops and multi-edges can be directly sampled with "stub matching" with rejection sampling, this can take exponential time. See [65] for more detailed discussion of algorithms to sample the

We iterate the following steps:

1. Select two edges at random in the network.
2. Assert that each edge is connected to a distinct pair of nodes. If the two edges involve either the same sample node, or the same gene node, repeat 1.
3. Swap which gene is connected to which sample.
4. Repeat 1-3

This process creates a Markov chain that, if run for long enough, will sample all of the possible networks under our model with uniform probability [65]. Our test statistic compares the observed mean TMB of a gene's mutated sample set, to the distribution of mean TMB's from many independently drawn mutated sample sets for that gene from the bipartite configuration model.

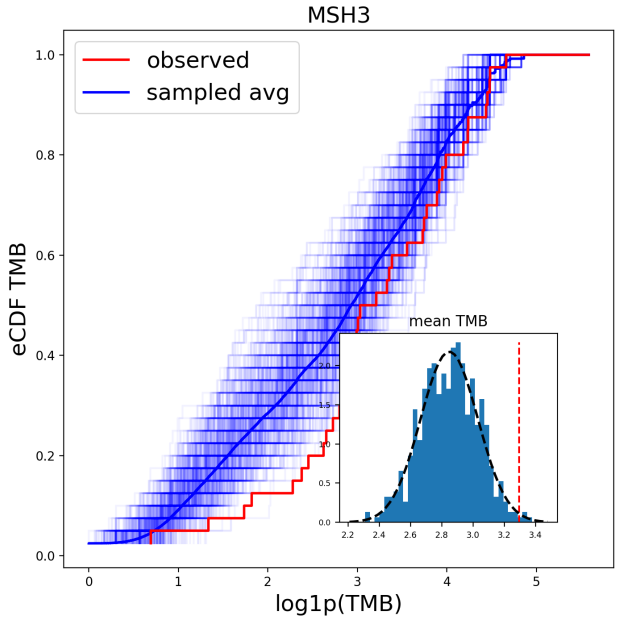

Figure 4.5: Sampling the distribution of mean TMBs from the bipartite null model. Each blue step function represents the empirical cumulative distribution function for the TMB of all samples with a mutation in the selected gene (MSH3) in a single network drawn from the null model. The red line shows the observed distribution of TMB and the inset shows the distribution of the means of each set of sampled TMBs with a fitted Gaussian overlaid. The red dashed line in the inset represents the observed mean TMB for that gene.

As demonstrated in Figure 4.5, we compute a z-score for the observed mean level of TMB compared with the distribution of the means across many random networks under the

various versions of the configuration model.

assumption that the sampled means follow a normal distribution.[7] The z-score can then be used to assess whether the statistic is significant at a given threshold with the appropriate correction for multiple testing. We can compute our statistic for both an individual gene as well as a group of genes such as a DNA Damage Repair pathway. In the next section we apply our network permutation test to two large datasets and compare the results with the univariate approach. In so doing, we demonstrate that only two of the major DNA damage repair pathways are associated with elevated levels of TMB.

4.3 TMB Paradox Results

4.3.1 Application of bipartite permutation test to DDR pathways

We have applied our test to the TCGA mutational data as well as to the PMEC data, computing gene level and DDR pathway level z-scores and comparing them with the Mann-Whitney U test p-values for reference. Overall, we found that only mutations in the mismatch repair (MMR) and nucleotide excision repair (NER) pathways were associated with higher levels of tumor mutational burden in both datasets. Figure 4.6 shows the distributions of TMB's for each of the pathways across 500 permuted networks from the bipartite configuration model.

None of the other pathways showed a strong association with elevated tumor mutational burdens, with the exception of non-homologous end joining (NHEJ) pathway in the PMEC dataset. However, only a single NHEJ gene, PRKDC, was available in the much smaller set of genes present in PMEC. Interestingly, mutations in the damage-sensing (DS) pathway were associated with lower levels of TMB. This could arise if mutations in this pathway were mutually exclusive with mutations in a high TMB associated pathway such as NER. However, this warrants further investigation as to the biological interpretation of this result.

We examined the correlation between the z-scores obtained via our permutation test using the full TCGA data (consisting of about 18000 genes across 9500 tumor samples) with the

[7]We compute a z-score and convert it to a p-value as an approximation to the true significance of the test due to computational constraints. Under the assumption that our samples are independent, by the Central Limit Theorem, the distribution of the mean TMB should be normally distributed. One could also draw a sufficiently large number of samples from the null distribution to compute an empirical p-value.

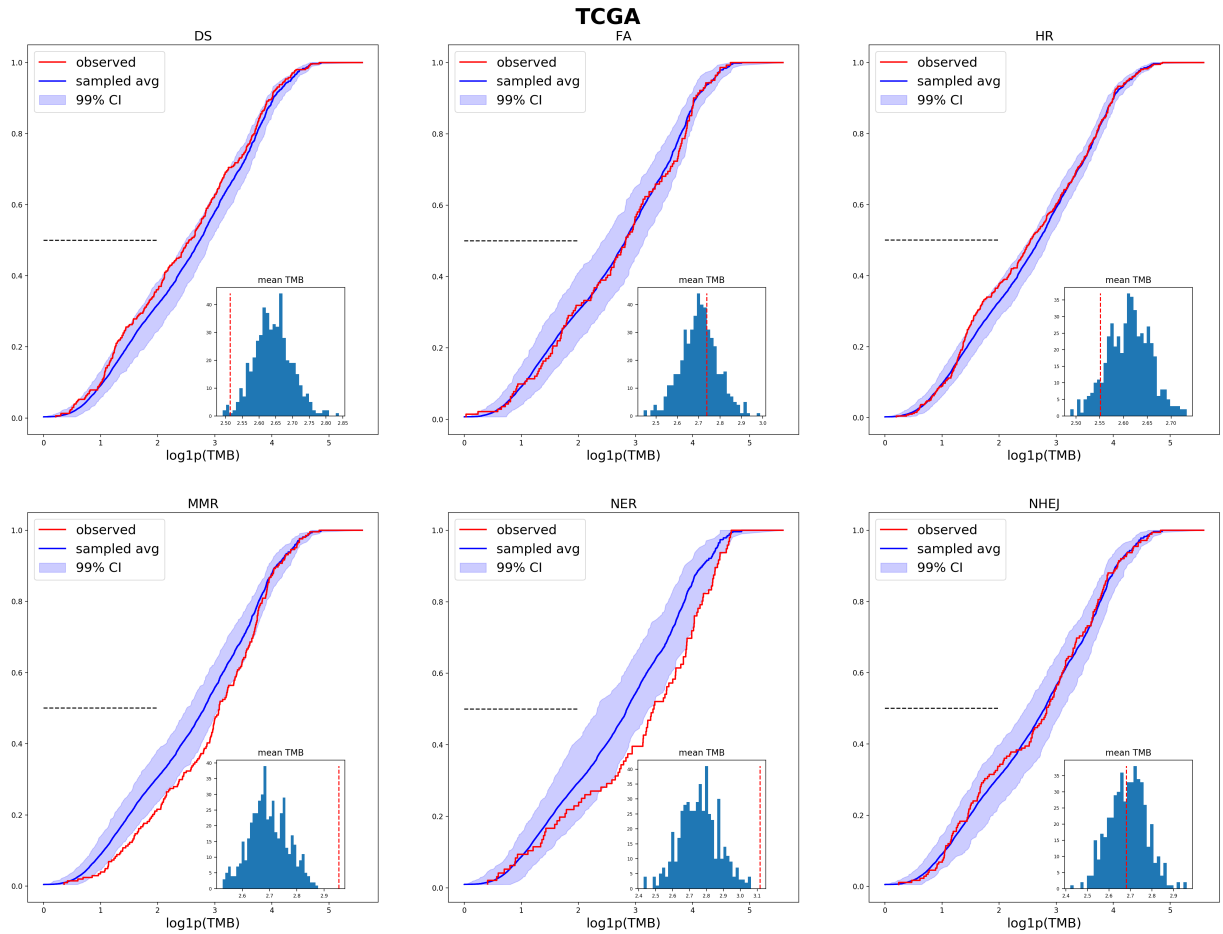

Figure 4.6: Application of bipartite configuration test to the DNA Damage Repair pathways in the TCGA data. Each figure shows the observed cumulative distribution of TMB for samples with any mutation in the genes of the specified pathway (with samples with multiple mutations counted only once) by the red solid line. The blue line shows the average cumulative distribution across 500 sampled networks, with the light blue band showing the 99% confidence interval. The horizontal line at $y = .5$ denotes the median TMB for the distributions. The inset figure in each panel shows a histogram of the means of the sampled distributions of TMB for samples with a mutation in the corresponding DDR pathway. The vertical red dashed line depicts the observed mean TMB in the actual data set. Z-scores were constructed by comparing the observed mean TMB to the sampled means.

z-scores obtained on the PMEC dataset, which only consists of about 500 genes total. In Figure 4.7.A, we show that the obtained scores are consistent for the DDR pathway genes between the two datasets ($R^2 = .46$), with good consistency between the pathway level scores. Similar agreement between gene-level z-scores across the two datasets is found using the entire set of common genes ($R^2 = .37$) shown in supplemental Figure 26.

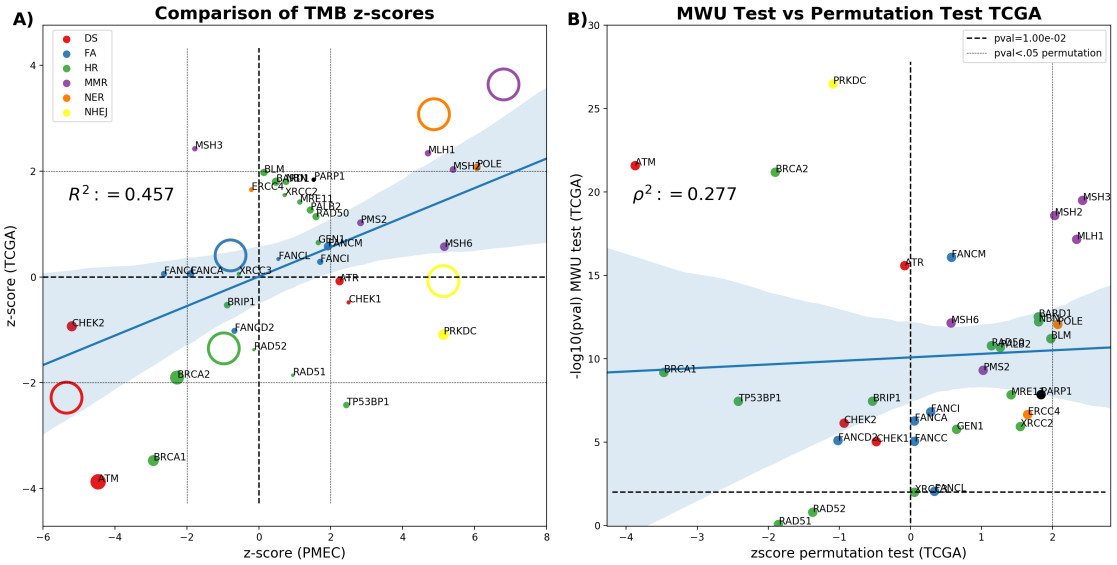

Figure 4.7: Comparisons of permutation test applied to DDR genes. A) We compare z-scores obtained for each of the DDR genes (solid dots) and the DDR pathways (open circles) using both the PMEC dataset (x-axis) as well as the TCGA mutation data (y-axis). Solid gray lines indicate boundaries for a $p < .05$ significance threshold for each test. **B)** We compare the z-scores obtained for the permutation test in the TCGA data with the p-values for the corresponding Mann-Whitney U test in the same dataset. Genes are colored in both plots according to their DDR pathway.

We also looked at the consistency between the z-scores of network permutation test and the p-values of the Mann-Whitney U test[8] shown in Figure 4.7.B. As previously mentioned, the MWU test predicts a positive association between TMB and mutations in the vast majority of genes. However using the network based test, only a few of the individual DDR genes (with z-score >2) have enough evidence to suggest a positive association between mutations in the gene and elevated levels of TMB, including POLE and several of the MMR genes. Several other genes that would be ranked highly under the MWU are no longer significant or have a negative

[8]The Mann-Whitney U statistic tests whether a value drawn from one distribution is likely to be larger than that from the second. We perform the two-side version of the test here and report the p-values. In all cases the distribution of TMB for samples with a mutation in the gene was higher than the non-mutated samples, even though the opposite direction of effect would also have given a small p-value.

association under the permutation test, suggesting that there is a confounding effect from co-mutations with other genes. We see a similar relationship between the permutation test z-scores and the MWU p-values when looking across all genes in the TCGA set (see supplemental Figure 26) though with a slightly stronger correlation ($\rho = .51$). We note, however, that the relationship is clearly not linear.

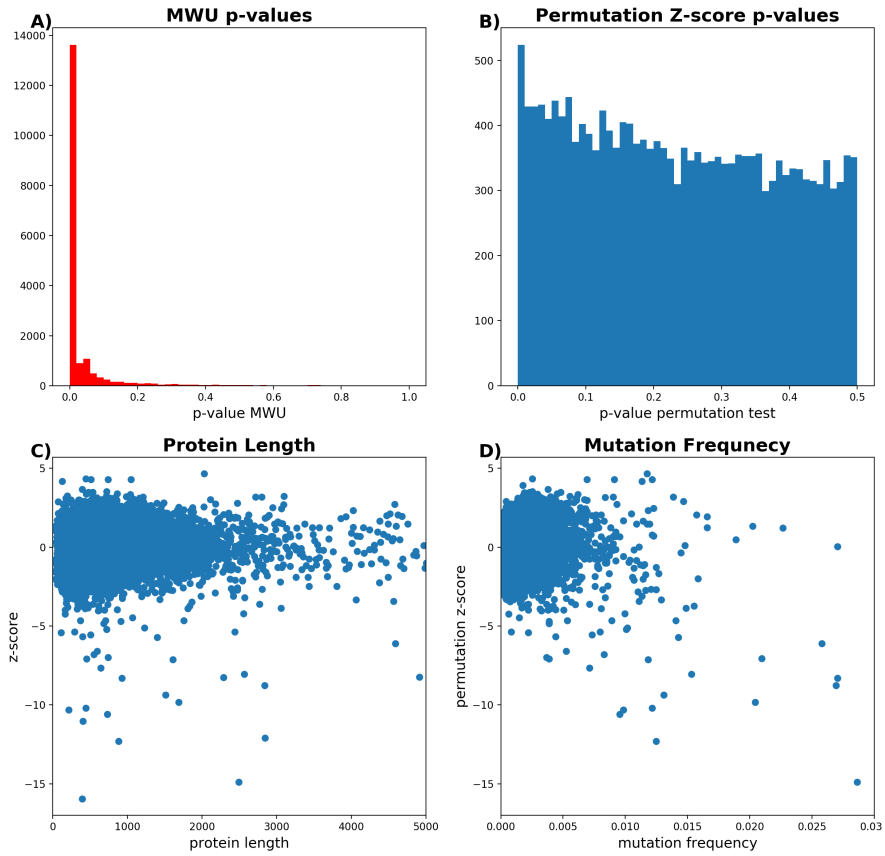

Figure 4.8: Characterization of network permutation test Distributions of p-values for the **A)** Mann-Whitney U test as well as the **B)** network permutation test applied to all $18,000$ genes in the TCGA dataset. **C)** Z-score values and protein length (number of amino acids) show no relationship. **D)** Mutation frequency for each gene in TCGA plotted against its z-score based on the permutation test.

If we compare the distribution of p-values implied by the network permutation test to the distribution generated by the MWU test (Figures 4.8.A and B) we find that the network based test produces a more uniform distribution of p-values, which is what one would expect assuming that the null hypothesis holds for most genes (i.e. that mutations in the majority of genes are not correlated with higher levels of TMB). We also see that our z-score is independent of both the length of the encoded protein and the mutation frequency of the gene across all tumors

(Figure 4.8.C and D). Thus our network based permutation test is able to properly account for the co-occurrence of mutations that drastically biases other approaches.

4.3.2 GO-term analysis reveals low z-score genes enriched for chromatin remodeling and negative regulation of cell proliferation

We looked for enrichment of Gene Ontology (GO) terms [52] associated with the genes with the 50 highest and lowest z-scores using the full TCGA dataset. We found that the genes with the lowest z-scores were highly enriched for a number of biological processes, most prominently negative enrichment of cell proliferation and in chromatin remodeling shown in Figure 4.9. This finding could be explained by the fact that cancer types with low overall levels of mutational burden are often driven by epigenomic changes and disruption in chromatin structure [75]. For example, the gene ARTX, a member of the SWI/SNF chromatin remodeling complex, had a very low z-score ($z = -14.6$). ARTX is commonly mutated in Glioma ($\approx 20\%$ in Samstein *et al.* dataset and 40% in the TCGA dataset), which tends to have one of the lowest levels of TMB across all tumor types [5]. We did not find any gene ontology terms enriched in the genes with the 50 highest z-scores by our permutation test.

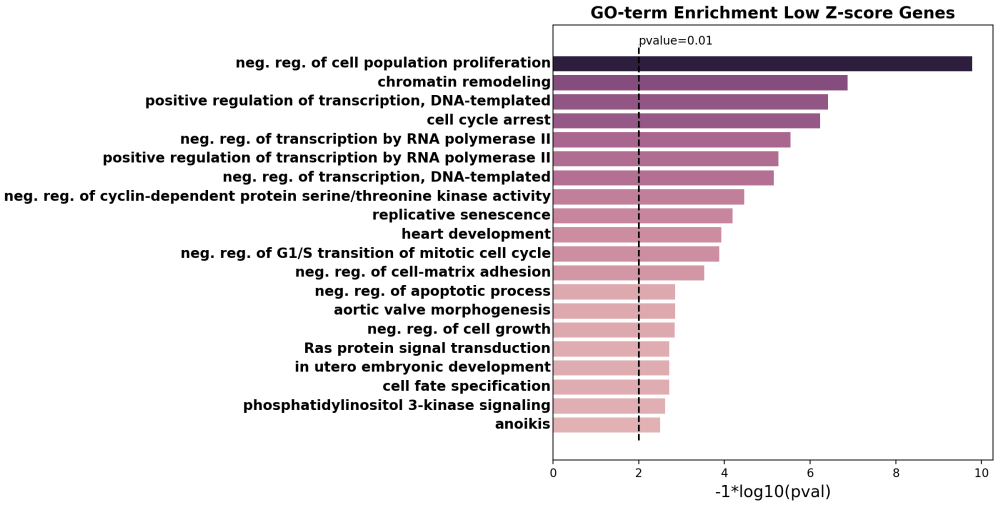

Figure 4.9: Gene Ontology Enrichment analysis for genes with the lowest z-scores. Bars represent the $-\log_{10}$ of the p-values for the corresponding GO term with multiple test correction applied (using Bonferroni multiple test correction).

4.3.3 Mutations in low z-score DDR genes predict ICB therapy response

The original impetus for our analysis was the question of whether mutations in DNA Damage Repair genes (or other genetic pathways) have predictive power for response to ICB therapy that is independent of their association with elevated TMB levels. Our method equips us to approach this question because we can identify which groups of DDR genes are actually associated with elevated levels of TMB. Using two independent clinical datasets (see Section 4.5.1 for full description of each), we find that DDR mutations not associated with elevated levels of TMB can help predict response for ICB therapy.

We split samples within both datasets into those with a high z-score DDR mutation and those samples with a low z-score DDR mutation (see Section 4.5.5 for the list of genes within each set). In cases where samples had mutations in both low and high z-score DDR genes, samples were considered in the high z-score group (and not in the low z-score group).

We tested whether having a mutation in a low z-score (z-score<0) DDR gene in combination with high TMB had an effect on overall survival using the data from the IMVigor210 trial as released in [144]. Figures 9.A and B show that having a mutation in a low z-score DDR gene has a significant negative interaction with elevated TMB. Samples with a mutation and high TMB have increased survival above those that have high TMB but no mutation. However, the effect is reversed for samples with low TMB, with mutated samples having much worse overall survival rates. In contrast, mutations in genes associated with high TMB (z-score > 0) had no effect on overall survival as shown in Supplement Figure 27.

The difference in effect on survival between mutations in low z-score DDR genes and those with high z-scores was also borne out in the objective response data, shown in Figures 4.10.C and D respectively. Among tumors with elevated levels of TMB, there was a significant difference (by Fisher's exact test) in response rates between samples with a mutation in the low z-score DDR genes and those without (WT). However, no such difference was seen when the samples were segregated by the presence of a high z-score DDR mutation. This suggests that DDR genes that are not associated with elevated levels of TMB could have additional power to predict which samples will respond to ICB therapies.

To validate our findings in the IMVigor210 dataset, we conducted a similar analysis using

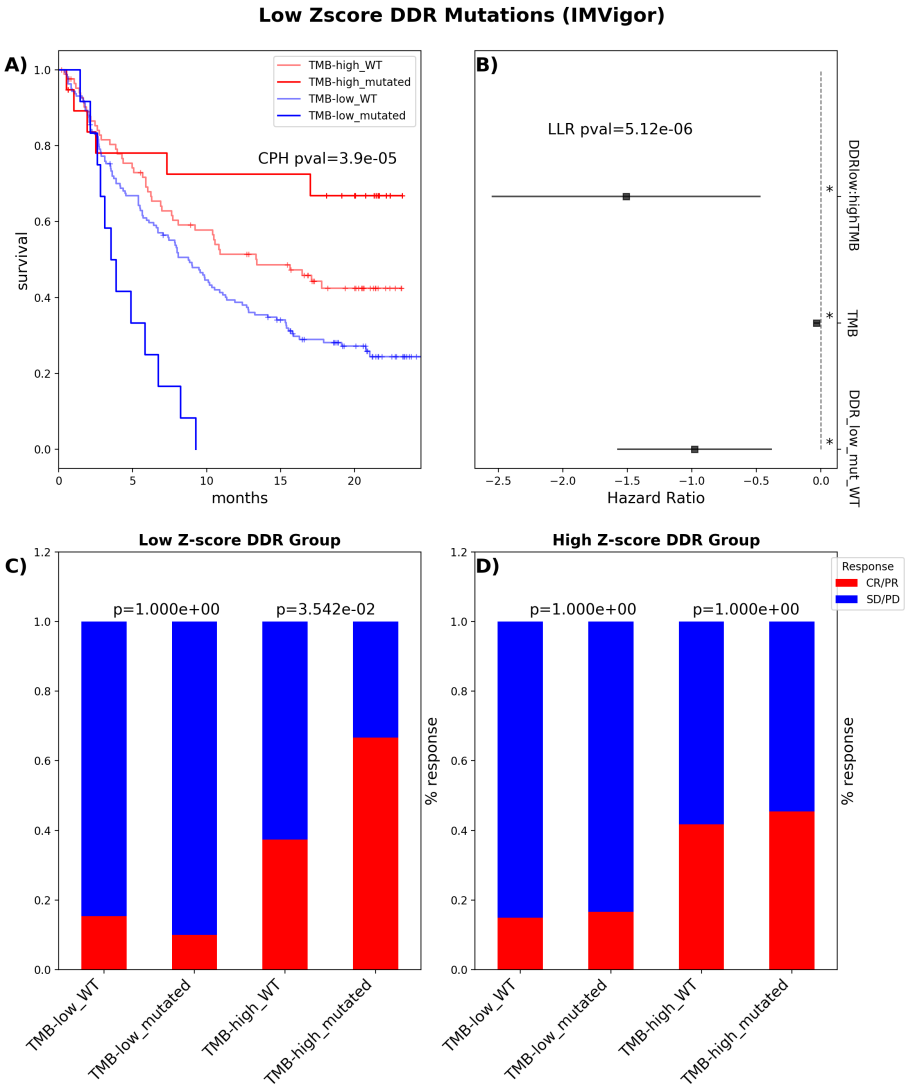

Figure 4.10: Effect of Low Zscore DDR mutations on survival for the IMVigor dataset. A) Splitting the samples into groups based on high (> 10) or low TMB (< 10) and mutated or WT for the low z-score DNA Damage Repair Genes. Samples are groupled based on whether or not they had a mutation in a low z-score DDR gene. We tested a Cox proportional hazard model for differences in overall survival between these four groups. **B)** Forest plot showing the estimated coefficients for a CPH model testing jointly testing TMB, mutation in low-zscore DDR genes, as well as an interaction term between the two variables (denoted by DDR-low:highTMB). We note here that TMB is treated as a continuous variable. **C)** and **D)** show the percentage of clinical response rates across the samples segregated by low TMB with no mutation in low z-score DDR, low TMB with a mutation, high TMB with no mutation, and high TMB with a mutation in order. Red bars denote the percentage of samples in each group that had a complete or partial response while blue bars denote the fraction that had stable or progressive disease. P-values were assessed using Fisher's exact test.

data from a large, multi-cancer cohort with targeted sequencing compiled by Samstein *et al.*[201].

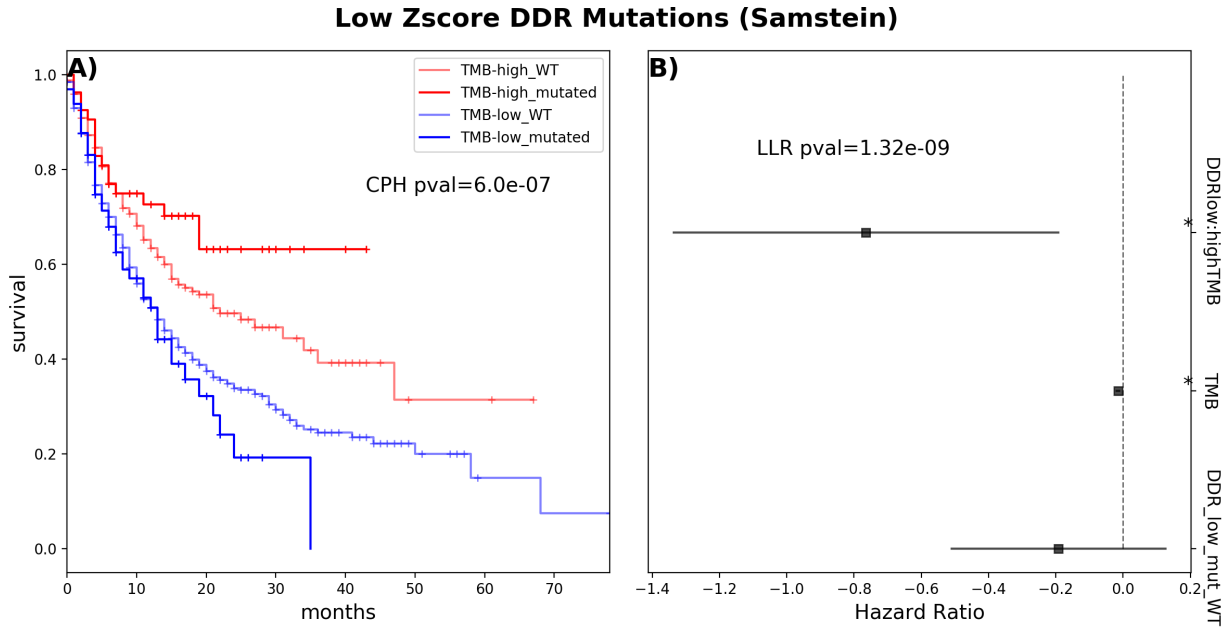

Figure 4.11: Effect of Low z-score DDR mutations on survival for the Samstein dataset. A) Splitting the samples into groups based on high (> 10) or low TMB (< 10) and mutated or WT for the low z-score DNA Damage Repair Genes. We group samples based on whether or not they had a mutation in a low z-score DDR gene. We tested a Cox proportional hazard model for differences in overall survival between these four groups, each treated as a dummy variable. **B)** Forest plot showing the estimated coefficients for a CPH model testing jointly testing TMB, mutation in low-zscore DDR genes, as well as an interaction term between the two variables (denoted by DDRlow:highTMB). We note here that TMB is treated as a continuous variable.

We found a similar effect in predicting survival with the Samstein *et al.* cohort: a significant negative interaction effect between TMB and mutation in the low z-score DDR genes as shown in Figure 4.11. We see a similar pattern in survival between the groups in both datasets, with mutations in the low z-score DDR genes having a positive effect for high TMB samples and a negative effect for low TMB samples. Having a low z-score DDR mutation was not in and of itself significant in this cohort. However this is not altogether surprising given the negative interaction term when conditioned on TMB. This is an example of the well known Simpson's paradox whereby a significant difference can be masked when groups are aggregated [209]. As was seen in the IMVigor210 data, this effect disappears when samples are segregated by the presence of a high z-score DDR gene mutation, shown in supplemental Figure 27.C-D.

We also tested for this interaction effect in the Samstein *et al.* data with groups defined based on mutation in DDR genes with very low Mann-Whitney-U scores. These would be the genes that the MWU test would characterize as being the least associated with elevated TMB.

Supplement figure 28 shows that there was no discernible difference between the samples separated by mutations defined by the MWU test. Thus our approach is able to uniquely identify a set of genes with clinical relevance.

4.4 TMB Paradox Discussion

We have presented a novel, networks based test for an association between Tumor Mutational Burden and mutations in specific genes or pathways. Our approach considers the co-mutational structure of the data jointly in that we use all of the data at once to define the null model to test associations against. By reformulating our mutational data as a bipartite network, we are able to appropriately account for the outsized role that highly mutated samples play in univariate tests for statistical association. We have shown that this oversampling phenomenon is a special case of the friendship paradox from the network science literature, and we have presented a network permutation test that allows us to assess the true significance of associations between mutations and TMB. Our test results in a much more principled and appropriate assessment of the genome as a whole as compared to the univariate tests such as the t-test or the Mann-Whitney U. Our network based approach suggests that relatively few genes are actually associated with higher levels of TMB.

We have showcased the power of our method by applying our test to the genes in the DNA Damage Repair pathways. Our results are consistent between two large, multi-cancer datasets (TCGA and PMEC) and show that the only pathways in which mutations are actually associated with elevated levels of TMB are the mismatch repair and nucleotide excision repair pathways. This confirms the results from other studies that have examined the link between DDR pathways and TMB in the context of a broader set of genes [115].

We examined whether mutations in DNA Damage Repair genes that were independent of elevated TMB were predictive of overall response in two separate patient cohorts treated with immune checkpoint therapy and with clinical sequencing. We found that there is a strong interaction between TMB as a covariate and having a mutation in a DDR gene with a low z-score and that this interaction appears to be negative. This effect is seen within both datasets and is reinforced in the clinical response data for the IMVigor210. By distinguishing which genes are

actually associated with high levels of TMB, we have identified a group DDR genes in which having a mutation is predictive of better outcomes for samples with high levels of TMB. We hypothesize that mutations with this group of low z-score DDR genes might illicit an immune response in a mechanism that is not captured by TMB alone, though this hypothesis requires further investigation.

One way in which our null model might not be realistic is in how we have ignored the tumor types of the samples in our null model. The likelihood of connecting any samples with any gene is simply proportional to the product of their degrees in the bipartite network. However, it is well established that different cancer types have different profiles of genes that are commonly mutated. One way we could improve our test is by restricting that edges can only be swapped among samples with the same cancer type. In addition, we do not include multi-edges in the network for samples that have multiple mutations in the same genes and do not include larger deletions or amplifications in constructing our network. We could also look at smaller genomic units than a single gene such as the exons that compose a specific domain or other regulatory regions. Such analyses will become more and more feasible as the number of samples in large-scale clinical datasets is expanded. All of these effects could make our null model more realistic and would be interesting to investigate in future lines of research. In addition, further investigation is required into the mechanisms underlying the observed interaction between TMB and mutations in the low z-score DDR genes. This line of inquiry could provide insight into how ICB therapy could be rescued in combination with other targeted approaches.

In summary, we have identified a sampling bias in the current testing methods to identify an association between elevated levels of TMB and mutation in specific genes or pathways. By recognizing this bias as a special case of the friendship paradox, we have developed a networks based test for significant associations. We have shown that our method gives consistent results in multiple, large-scale genomic datasets and we have identified two DNA Damage Repair pathways where mutations are associated with elevated TMB. We demonstrate the clinical significance of our findings on two large datasets with annotated clinical outcomes. We found concordance in how the genes identified by our approach were able to predict survival in both datasets. We have released all of our analyses as part of a python package, including several notebooks to replicate all of our figures.

4.5 TMB Paradox Additional Methods

4.5.1 Description of datasets used

The Cancer Genome Atlas (TCGA) Pan-cancer MC3 Dataset

Throughout this chapter we have relied on several datasets for development and validation of our approach. Our primary dataset for developing our method was the TCGA-pancan unified ensemble MC3 variant call set, with Whole Exome Sequence (WES) tumor samples from all TCGA centers re-called in a uniform pipeline. This dataset includes 3.6 million variants from 10,295 tumor samples. This dataset is described in [55]. We filtered this dataset down to the most deleterious variants, keeping only those with the following high consequence annotations: stop lost, stop gained, transcript ablation, start lost, and frameshift variant. This resulted in 209,612 remaining variants in 9539 different samples in 18,322 different genes.

Precision Medicine Exchange Consortium (PMEC)

The Precision Medicine Exchange Consortium is a partnership between 12 universities and Foundation Medicine to provide clinical sequencing of a large cohort of tumors. The goal of the consortium is to pool together genomic sequencing data as well as large-scale clinical annotation to advance research in personalized medicine. The PMEC dataset consist of 12,657 unique tumor specimens with targeted sequencing of 557 cancer-associated genes. Specimens within the PMEC dataset were delineated into 330 different cancer types and subtypes, which were grouped into 40 different main cancer types that aligned with the TCGA nomenclature. Clinical outcomes have not yet been annotated for this dataset.

IMVigor210

The IMVigor210 trial is a Phase II single arm study examining the response of patients with locally advanced or metastatic urothelial bladder cancer to atezoliziumab (anti PD-L1). A full description of the characteristics of the patient cohort can be found in [11, 197]. We have used the publicly available dataset released by Mariathansan *et al.* which can be accessed at http://research-pub.gene.com/IMvigor210CoreBiologies/. The cohort consists of 260

patients with 1249 short variants across 160 different genes. We did not filter any of the mutations from this cohort.

Samstein *et al.* Cohort

To validate our clinical findings, we used a large, multi-trial cohort consisting of 1662 patients treated with different Immune Checkpoint Blockade (ICB) therapies and with targeted clinical sequencing, first compiled and analyzed in [201]. Sequencing was performed using the Memorial Sloan Kettering Integrated Mutation Profiling of Actionable Cancer Targets (MSK-IMPACT) panel [37]. We downloaded the data from cbioportal using the following link: https://www.cbioportal.org/study/summary?id=tmb_mskcc_2018. We filtered down to the 1307 patients that received anti PD-1/PD-L1 therapies and kept the variants with one of the following high impact consequences: missense mutation, nonsense mutation, frameshift deletion, frameshift insertion, translation start site, or nonstop mutation. This resulted in a total of 19,057 variants in 468 different genes.

4.5.2 Defining the DNA Damage Repair Pathways

We have relied on the core DNA Damage Repair pathways defined by Knijnenburg *et al.* [115] to conduct all of our pathway level analysis. The pathways are defined as follows:

BER	NER	MMR	FA	HR		NHEJ	DS
PARP1	CUL5	EXO1	FANCA	MRE11	EME1	LIG4	ATM
POLB	ERCC1	MLH1	FANCB	NBN	GEN1	NHEJ1	ATR
APEX1	ERCC2	MLH3	FANCC	RAD50	MUS81	POLL	ATRIP
APEX2	ERCC4	MSH2	FANCD2	TP53BP1	PALB2	POLM	CHEK1
FEN1	ERCC5	MSH3	FANCI	XRCC2	RAD51	PRKDC	CHEK2
TDG	ERCC6	MSH6	FANCL	XRCC3	RAD52	XRCC4	MDC1
TDP1	POLE	PMS1	FANCM	BARD1	RBBP8	XRCC5	RNMT
UNG	POLE3	PMS2	UBE2T	BLM	SHFM1	XRCC6	TOPBP1
	XPA			BRCA1	SLX1A		TREX1
	XPC			BRCA2	TOP3A		
				BRIP1			

Table 4.1: Core DDR Pathways defined by [115].

4.5.3 Sampling bipartite configuration model

Our bipartite permutation test compares the observed distribution of TMB for samples with a mutation in a given gene or pathway, with the expected distribution for bipartite networks with the degree sequence fixed. To obtain samples from this bipartite configuration model, we follow the network rewiring approach developed in [65]. To rewire the network, we select two edges in the network at random, $(g_i, s_x) \in \mathcal{E}$ and $(g_j, s_y) \in \mathcal{E}$. We check whether or not this represents a valid switch that would not create a multi-edge in the permuted graph. That is we assert that $g_i \neq g_j$ and $s_x \neq s_y$. If this condition is not met, we simply resample from all edges with replacement. Once the sampled edges represent a valid swap, we add the new edges (g_i, s_y) and (g_j, s_x) to the network, removing the original sampled edges. This constitutes a single rewire of the network. To obtain a single random sample, we rewire the network many times in series, keeping track of the new edges so that the network becomes progressively unrecognizable from the original network. The series of rewired network is known as a Markov chain. Each network is independent of all of the previous networks conditioned on its immediate predecessor. Furthermore, if run long enough, the process will generate all possible networks under our model with uniform probability. Prior to generating samples from our Markov chain, we conduct "burn-in" rewires in order to get the process into a region of high likelihood under the model and to get samples that are largely independent from our observed data. We select the number of rewires, R, between each sampled network to be equal to the total number of edges in the network, $R = M$, with a burn-in of $2M$. This ensures that most edges in the network will have been re-wired between each sample.

4.5.4 Equations for sampling

Let the gene g_i have degree k_i. Let $\partial {g_i}^{obs} = \{s | (s, g_i) \in \mathcal{E}^{obs}\}$ denote the set of samples connected to g_i in the bipartite representation of the original data, $\mathcal{G}^{obs}$ (that is the neighbors of g_i). Let $T_{obs} = \mathbb{E}_{s \in \partial {g_i}^{obs}} (TMB_s)$ be the average TMB for all samples connected to g_i in the observed dataset. We derive a z-score for a significant association between TMB and g_i as follows:

1. We sample R independent realizations of the bipartite network with fixed degrees sequences.

2. For each sampled network, $\mathcal{G}_r$, we compute $T_r = \mathbb{E}_{s \in \partial g_i{}^r}\,(TMB_s)$, the average TMB for all neighbors of g_i in each sampled bipartite network $\mathcal{G}_r$.
3. We compute the z-score for the observed graph using :

$$z_i = \frac{T_{obs} - \mathbb{E}_r T_r}{\hat{\sigma}_r(T_r)}$$

where $\hat{\sigma}_r(T_r)$ is the empirical standard deviation for the distribution of sampled $\{T_r\}$.

4.5.5 Survival Analyses

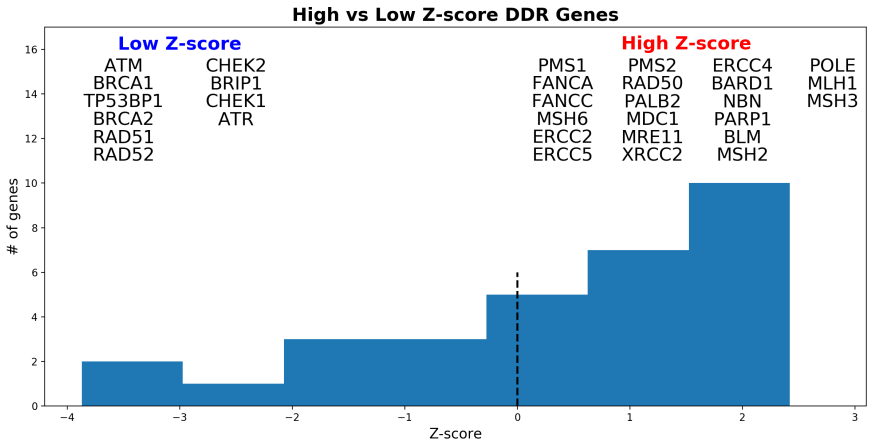

Figure 4.12: Splitting DDR genes on the basis of z-score. We split the DDR genes on the basis of having a high (>0) or low (<0) z-score for both the IMVigor and the Samstein *et al.* datasets. Here we show the distribution of z-scores as well as which genes were placed in each category.

All survival analyses were conducted using the Cox-proportional hazard model with a log-likelihood ratio test (LLR) to test for the overall significance of the model and a t-test to assess the significance of individual variables in the model. For the depiction of Kaplan-Meyer curves, TMB is treated as a binary variable with a threshold of TMB> 10 defining the high TMB group. Samples were divided into groups based on the presence of a mutation within a high z-score DDR genes or low z-score DDR genes, shown in Figure 4.12. The low z-score DDR genes included: ATM, ATR, BRCA1, BRCA2, BRIP1, CHEK1, CHEK2, RAD51, RAD52, TP53BP1. The high z-score genes were: BARD1, BLM, ERCC2, ERCC4, ERCC5, FANCA, FANCC, MDC1, MLH1, MRE11, MSH2, MSH3, MSH6, NBN, PALB2, PARP1, PMS1, PMS2, POLE, RAD50, XRCC2. Samples with mutations in both sets of genes were considered in the high z-score category and excluded from

the low z-score category. The genes that defined the low MWU test set were: RAD51, RAD52, ERCC5, ERCC2, CHEK1, FANCC, PMS1, XRCC2, CHEK2, and FANCA.

APPENDIX

2 Chapter 2 CHAMP Supplement

2.1 Chapter 2 supplemental figures

In this Appendix, we include visualizations of each of the partitions with domains of optimization labeled in white text in Figure 2.10A. In Figure 13, the Senators are plotted according to their states. In Figure 14, the individual Senators have been sorted according to community assignment and, within communities, time of first appearance in the Senate.

We call particular attention to the qualitative difference between the community structures with domains above and below the transition around $\gamma \approx 0.8$. Below $\gamma \approx 0.8$, each Congress layer has only a single community, with the communities broken up across time. In the region just above this transition, the typical Congress layer has two communities, with the community structure corresponding to an evolving two-party system over time.

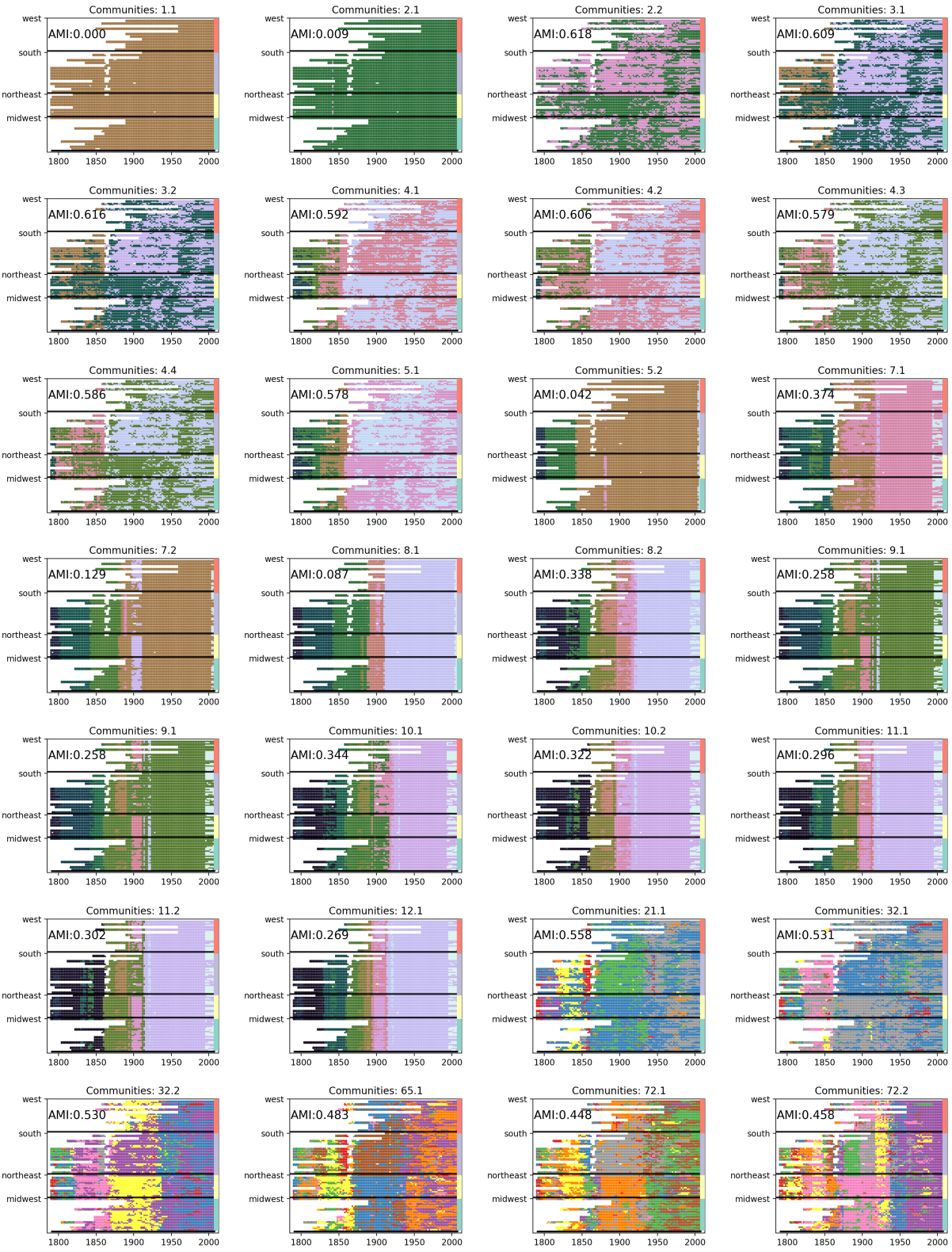

Figure 13: Visualizations of partitions labeled in white in Figure 2.10.A, with Senators grouped according to their state. The listed AMI is the average over layers of the AMI in each layer (Congress) between the communities and political party affiliations for that Congress. Partitions are labeled "$X.Y$" with X the number of communities with ≥ 5 nodes and Y the rank of the domain area for that number of communities.

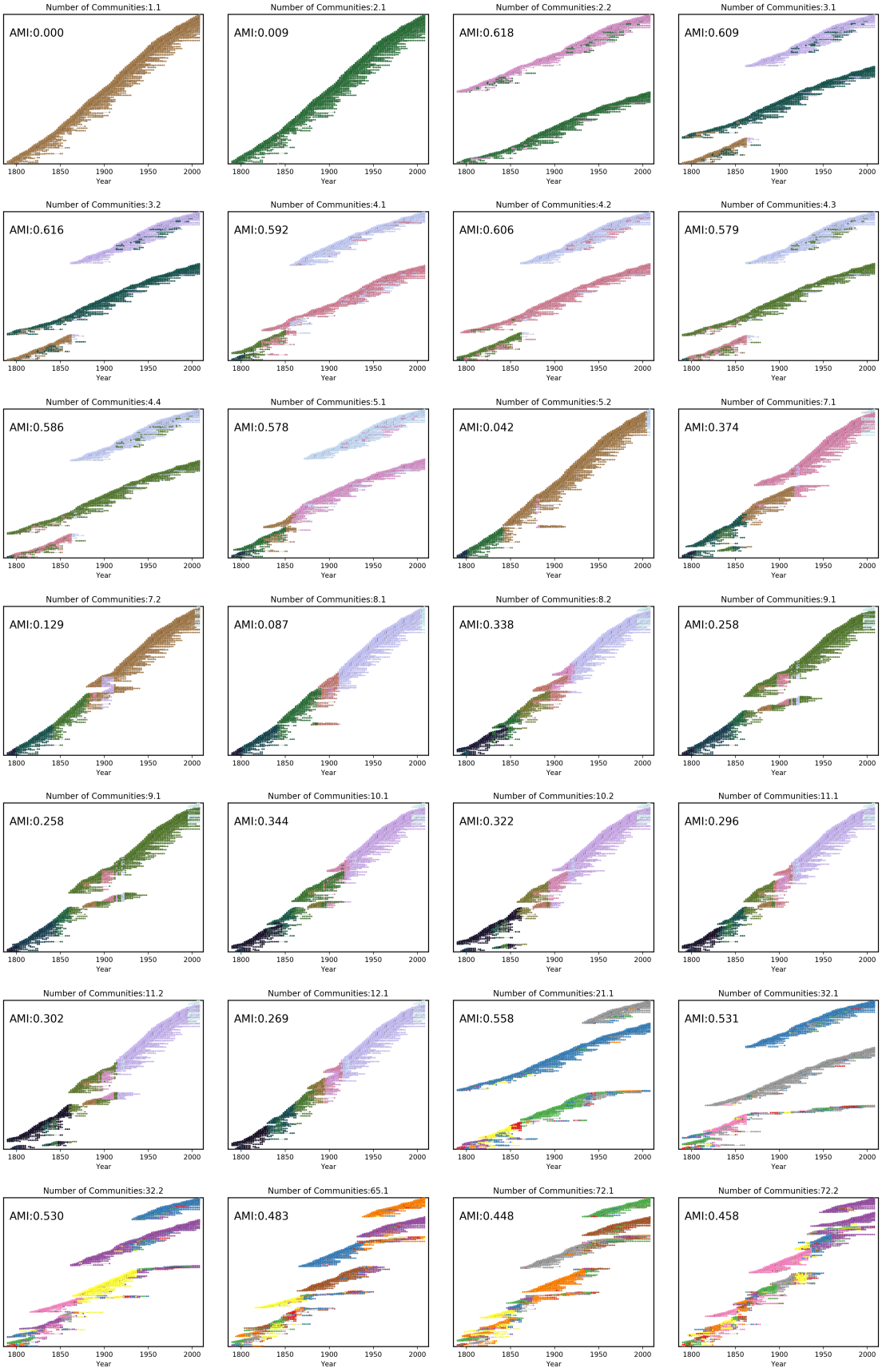

Figure 14: (*Caption on next page.*)

Figure 14: (Previous page.) **Visualizations of partitions labeled in white in Figure 2.10.A**, with Senators sorted by their most frequent community label (with the labels sorted by last appearance in time), and within communities by first appearance. The listed AMI is the average over layers of the AMI in each layer (Congress) between the communities and political party affiliations in that Congress.

3 Chapter 3 Multilayer Modularity Belief Propagation Supplement

3.1 Derivation of Bethe Free Energy

We derive here the formula for the free energy of the single layer model given in Zhang and Moore [259]. In the next section we will show how this naturally extends to the multilayer case with interlayer edges. For any model which has only pairwise interactions, the formula for the Bethe free energy approximation is given by

$$f_{\text{bethe}} = -\frac{1}{N\beta}\left(\sum_i \log Z_i - \sum_{i,j\in\mathcal{E}} \log Z_{ij}\right) . \tag{S1}$$

In the modularity model, there are really two types of edge interactions: those that are given explicitly by the underlying graph (i.e. the $A_{i,j}\delta c_i, c_j$ term), and the pairwise interaction term that comes from the null model (i.e. $P_{i,j} = \frac{k_i,k_j}{2m}\delta_{c_i,c_j}$). We can split these two apart:

$$f_{\text{bethe}} = -\frac{1}{N\beta}\left(\sum_i \log Z_i - \sum_{i,j\in\mathcal{E}} \log Z_{ij} - \sum_{i\neq j} \log \hat{Z}_{ij}\right) \tag{S2}$$

where we refer to the edges in the underlying graph as $\mathcal{E}$ and split out the non-edge interactions into another term with normalization $\hat{Z}^{ij}$. We write out the joint distribution for the "non-edges":

$$\psi^{ij}_{st} = \frac{1}{\hat{Z}_{ij}} e^{-\beta(d_i d_j/2m)\delta_{st}} \psi^i_s \psi^j_t \tag{S3}$$

We use this to compute $\hat{Z}_{ij}$:

$$\hat{Z}_{ij} = \sum_t \sum_s e^{-\beta(d_i d_j/2m)\delta_{st}} \psi^i_s \psi^j_t \,, \tag{S4}$$

$$\sum_{i<j} \log \hat{Z}_{ij} = \sum_{i<j} \log \sum_t \sum_s e^{-\beta(d_i d_j/2m)\delta_{st}} \psi_s^i \psi_t^j \tag{S5}$$

$$\approx \sum_{i<j} \log \left(\sum_t \sum_s 1 - \beta(d_i d_j/2m)\delta_{st} \psi_s^i \psi_t^j \right) \tag{S6}$$

$$\approx \sum_{i<j} \left(\sum_t \sum_s -\beta(d_i d_j/2m)\delta_{st} \psi_s^i \psi_t^j \right) \tag{S7}$$

$$= \sum_t \sum_{i<j} -\beta(d_i d_j/2m) \psi_t^i \psi_t^j \tag{S8}$$

$$= -\frac{\beta}{4m} \sum_t \sum_{i \neq j} d_i d_j \psi_t^i \psi_t^j \tag{S9}$$

$$= -\frac{\beta}{4m} \sum_t \theta_t^2 \,. \tag{S10}$$

This gives us the expected full formula,

$$f_{\text{bethe}} = -\frac{1}{N\beta} \left(\sum_i \log Z_i - \sum_{i,j \in \mathcal{E}} \log Z_{ij} + \frac{\beta}{4m} \sum_t \theta_t^2 \right) \tag{S11}$$

3.2 Multilayer Bethe Free Energy

We now extend the Bethe Free Energy equation to multilayer networks. The formula for multilayer modularity for undirected networks is given by Equation 3.28 in main text:

$$Q(\gamma, \omega) = \sum_{i,j} \left(A_{ij} - \gamma P_{ij} + \omega C_{ij} \right) \delta(c_i, c_j) \tag{S12}$$

As before we only have pairwise interactions within the model. However, note that in the multilayer formulation there are now both intra- and interlayer edges. We can split the edge term $\mathcal{E}$ of f_{bethe} into the contributions from interlayer and the intralayer edges:

$$\sum_{i,j \in \mathcal{E}} \log Z_{ij} = \sum_{i,j \in \mathcal{E}_{inter}} \log Z_{ij}^{inter} + \sum_{i,j \in \mathcal{E}_{inter}} \log Z_{ij}^{intra} \,. \tag{S13}$$

Where $\mathcal{E}_{intra}$ and $\mathcal{E}_{inter}$ are given by the non-zero elements of A_{ij} and C_{ij} respectively. For the non-edge term in the multilayer case, we note that the non-edge interaction terms are all restricted to within a given layer. This means that nodes within different layers of the model only

interact through the interlayer edge term and not through the null model term:

$$\frac{1}{N\beta}\sum_{i\neq j}\log\hat{Z}_{ij}=\frac{1}{N\beta}\sum_{l}\sum_{i\neq j,i,j\in l}\log\hat{Z}_{ij}^{l} \tag{S14}$$

We can therefore split this term into a sum over the contributions from each of the layers with a similar form as from before:

$$\sum_{l}\sum_{i\neq j,i,j\in l}\log\hat{Z}_{ij}^{l}=-\sum_{l}\frac{\beta}{4m_l}\sum_{t}(\theta_t^l)^2 \tag{S15}$$

and we can write the full Bethe free energy as

$$f_{\text{bethe}}=-\frac{1}{N\beta}\left(\sum_{i}\log Z_i-\sum_{i,j\in\mathcal{E}_{inter}}\log Z_{ij}^{inter}-\sum_{i,j\in\mathcal{E}_{inter}}\log Z_{ij}^{intra}+\sum_{l}\frac{\beta}{4m_l}\sum_{t}(\theta_t^l)^2\right) \tag{S16}$$

where the Z_{ij}^{inter} can be computed from the pairwise marginals of the interlayer interactions:

$$\psi_{s,t}^{i,j}=\frac{1}{Z_{ij}^{inter}}e^{\beta\omega\delta_{s,t}}\psi_s^{i\to j}\psi_t^{j\to i}\,. \tag{S17}$$

3.3 Derivation of Eigenvalues for Linearization of Modularity Belief Propagation

In this paper, we have used the approach by Shi *et al.* to identify the value of β^* where the trivial solution is no longer stable [208] which is an extension of the reasoning to the original approach presented in Zhang and Moore [259] (see Section 3.5.1 in main text). Here we present the form of the linearized approximation of the messages as well as its eigenvalue.

Consider the update equation for *multimodbp*:

$$\psi_t^{i\to k}\propto\exp\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i}+\sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right]. \tag{S18}$$

We compute the derivative $\frac{\partial \psi_t^{i\to k}}{\partial \psi_s^{j\to i}}$ assuming both (i,j) and (i,k) are edges:

$$\frac{\partial \psi_t^{i\to k}}{\partial \psi_s^{j\to i}} = \frac{\partial}{\partial \psi_s^{j\to i}}\left[\frac{1}{Z}\exp\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i} + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right]\right] \tag{S19}$$

$$= \frac{1}{Z}\frac{\partial}{\partial \psi_s^{j\to i}}\exp\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i} + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right] \tag{S20}$$

$$+ \exp\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i} + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right]\frac{\partial}{\partial \psi_s^{j\to i}}\left[\frac{1}{Z}\right]. \tag{S21}$$

We will consider each of these two derivatives separately. To help condense notation we define $F_{\text{arg}} = \gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i} + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)$. First,

$$\frac{\partial}{\partial \psi_s^{j\to i}}\exp\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i} + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right] \tag{S22}$$

$$= \exp\left[F_{\text{arg}}\right]\frac{\partial}{\partial \psi_s^{j\to i}}\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i} + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right]. \tag{S23}$$

The derivative of the first term here,

$$\frac{\partial}{\partial \psi_s^{j\to i}}\gamma\frac{\beta d_i}{2m}\theta_t^{l_i}$$

is $O\left(\frac{d_i d_j}{2m}\right)$, which we can ignore given our assumption that the network is sparse ($d_i \ll \sqrt{m}$ for all i).

We are then left with

$$\frac{\partial}{\partial \psi_s^{j\to i}}\left[\sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right].$$

The only term in this sum that will lead to a non-zero derivative is if $s=t$, leading to

$$\frac{\partial}{\partial \psi_s^{j\to i}}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\delta_{st} = \delta_{st}\frac{e^{\tilde{A}_{ij}\beta}-1}{1+\psi_s^{j\to i}\left(e^{\tilde{A}_{ij}\beta}-1\right)}. \tag{S24}$$

Evaluating at the fixed point, and combining with the previous $\frac{1}{Z}\exp(F_{\text{arg}})\big|_{\frac{1}{q}} = \frac{1}{q}$, this term

becomes

$$\delta_{st}\frac{e^{\tilde{A}_{ij}\beta}-1}{q+e^{\tilde{A}_{ij}\beta}-1}\,. \tag{S25}$$

Next we move on to the second term from the previous product rule expansion (Eq S19):

$$\frac{\partial}{\partial\psi_s^{j\to i}}\frac{1}{Z}=-\frac{1}{Z^2}\frac{\partial Z}{\partial\psi_s^{j\to i}}=-\frac{1}{Z^2}\frac{\partial}{\partial\psi_s^{j\to i}}\left[\sum_t\exp\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i}+\sum_{j\in\partial_i\backslash k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right]\right]\,.$$

we follow the same line of reasoning as before, dropping the θ^{l_i} term to arrive at

$$\approx-\frac{\exp(F_{\text{arg}})}{Z^2}\frac{e^{\tilde{A}_{ij}\beta}-1}{1+\psi_s^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)}\,. \tag{S26}$$

If we bring the extra $\exp(F_{\text{arg}})$ from before back in, and evaluate at the fixed point, this leads to

$$=-\frac{1}{q}\frac{e^{\tilde{A}_{ij}\beta}-1}{q+e^{\tilde{A}_{ij}\beta}-1}\,. \tag{S27}$$

In total, we find the linear approximation of the messages is given by the $q\times q$ matrix:

$$T_{st}^{i\to k,j\to i}=\frac{e^{\tilde{A}_{ij}\beta}-1}{q+e^{\tilde{A}_{ij}\beta}-1}\left(\delta_{st}-\frac{1}{q}\right)\,. \tag{S28}$$

The leading eigenvalue of this matrix is given by

$$\eta_{ij}=\frac{e^{\beta\tilde{A}_{ij}}-1}{e^{\beta\tilde{A}_{ij}}+q-1}\,. \tag{S29}$$

3.4 Testing selection of β^*

As part of testing the formula for $\beta^*(q,w)$, we look at the effect of adding normally distributed edges weights on an Erdös-Rényi graph shown in Figure 15. For the Erdös-Rényi graph with normally distributed weights, Equation 3.41 gives a very good estimate of where the divergence occurs, while the unmodified equation becomes less accurates as the weights differ from 1.

In the far right panel of Figure 15, we show that formula derived by Shi *et al.* well predicts

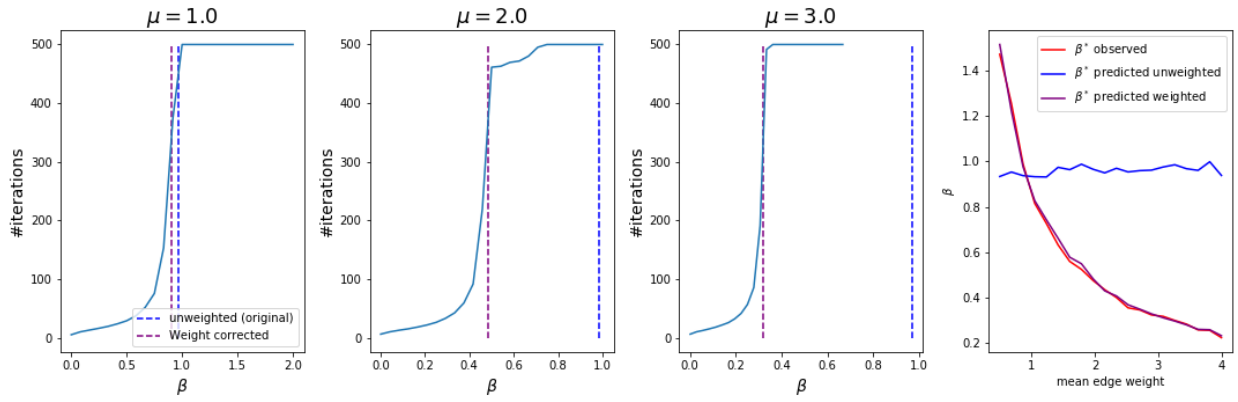

Figure 15: Stability boundary for Erdös-Rényi graph with weights assigned randomly from a $\mathcal{N}(\mu, \sigma = .5)$ normal distribution. Left three plots depict convergence curves of the algorithm for three different means of the normally distributed edge weights (μ =1,2, and 3 respectively). Each curve represents the average over 10 realizations of the ER random graph. The unweighted prediction for β^* is given by the black dashed line, while the weight adjusted prediction is given by the dashed green line. On far right plot β^* was empirically determined for several different mean weights (red line) and compared with the predicted values (blue line) showing good agreement.

the point where the trivial solution is no longer stable (shown by the red line). Below in Figure 16, we also demonstrate that for a 2 community SBM the modified formula for β^* occurs within the retrieval phase, detecting the communities with high accuracy.

We have used Equation 3.41 to identify the value of β to run the algorithm at in all of the experiments within this manuscript. Since *a priori* the number of communities, q, isn't known in advance, we run the algorithm at several values $\beta = [\beta^*(q = 2, c, \langle w \rangle), ...\beta^*(q = q_{max}, c, \langle w \rangle)]$ for a range of expected numbers of communities, $[2, q_{max}]$. We reiterate that the heuristic derived works well in most cases, but makes no guarantees that β^* will be inside the retrieval phase for all degree distributions and distribution of edge weights. For some networks scanning a range of β values might be required.

In Figure 17, we also show that the retrieval phase of multilayer networks also varies with the strength of the coupling parameter, ω. The β^* predicted by Equation 3.41 consistently lies within the retrieval phase even as ω increases (in contrast to the value of β given from the unmodified equation).

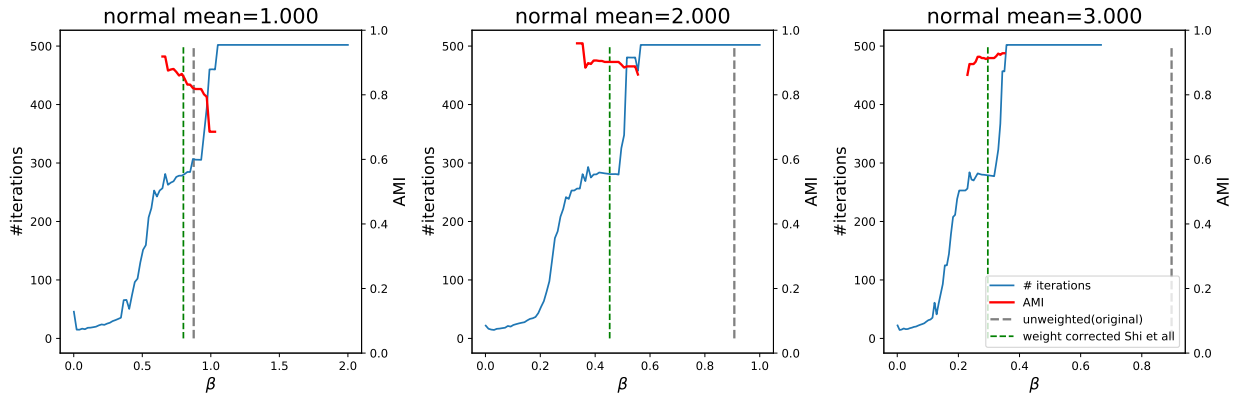

Figure 16: Stability boundary for 2 community stochastic block model graph with weights assigned randomly from a $\mathcal{N}(\mu, \sigma = .5)$ normal distribution. SBM's had $n = 200$ nodes with mean degree, $c = 6$, and $\epsilon = \frac{p_{out}}{p_{in}} = .1$. Each convergence curve was averaged over 10 realizations of the SBM model with different means of the normally distributed edge weights (μ =1,2, and 3 respectively). The unweighted prediction for β^* is given by the black dashed line, while the weight adjusted prediction is given by the dashed green line. Red curve shows the adjusted mutual information with the underlying ground truth. On far right plot β^* was empirically determined for several different mean weights (red line) and compared with the predicted values (blue line) showing good agreement.

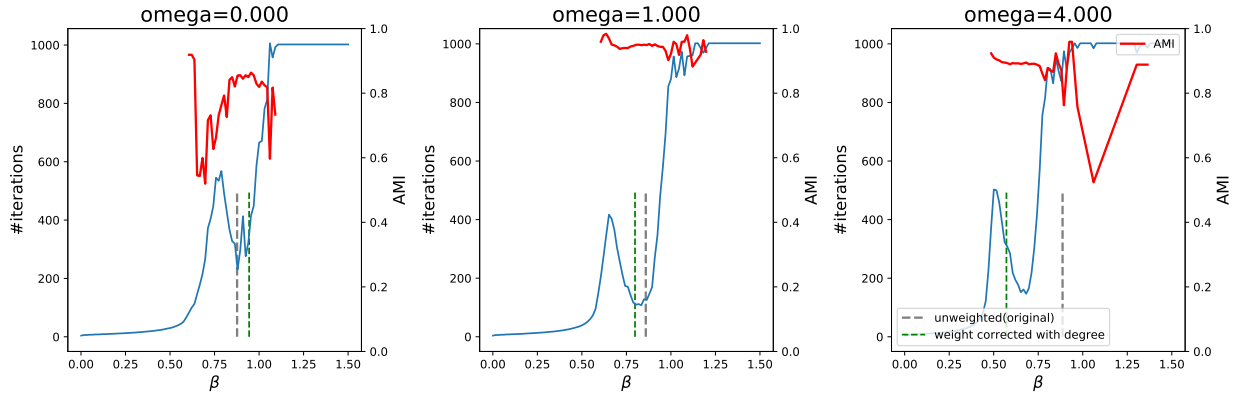

Figure 17: Stability boundary for 2 community unweighted multilayer dynamic stochastic block model graph. Network had $n = 100$ node within each layer with mean degree $c = 6$ and $\epsilon = \frac{p_{out}}{p_{in}} = .1$. Each convergence curve was averaged over 10 realizations of the SBM model with the algorithm run with different interlayer edge couplings (ω =0, 1, and 2 respectively). The unweighted prediction for β^* is given by the black dashed line, while the weight adjusted prediction is given by the dashed green line. Red curve shows the adjusted mutual information with the underlying ground truth. In the far right plot β^* was empirically determined for several different mean weights (red line) and compared with the predicted values (blue line) showing good agreement.

3.5 Chapter 2 supplemental figures

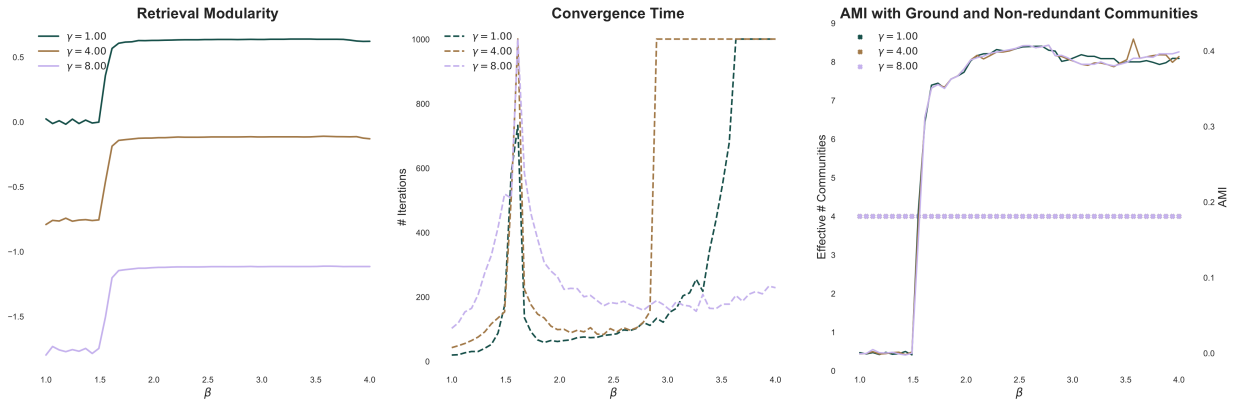

Figure 18: Effect of varying γ with q remaining fixed We compare the performance of the algorithm for a wide range of γ values in the event that the number of communities is fixed at the correct number ($q = 4$). Here we do not allow q to float as described in Section 3.5.2

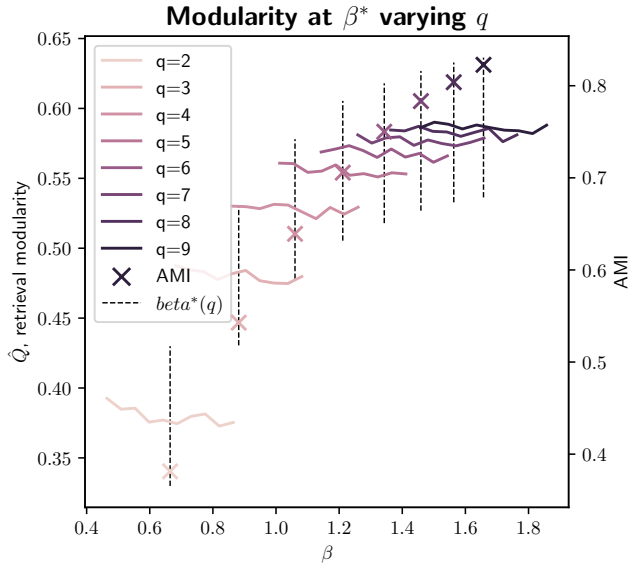

Figure 19: Attempting to select the appropriate value of q on the American football network. Using the method recommended by Zhang and Moore to select the appropriate value of q for the American NCAA Div-IA College Football Network [58, 71]. Each colored line corresponds to running *modbp* for a given value of q across a window of β around $\beta^*(q)$ (shown by black dashed line). Using this method would suggest an appropriate $q \in [6-8]$ depending on the threshold selected. We note that here, we do not collapse community labels as described in Section 3.5.2; for each run a single fixed value of q is used as well as the default resolution ($\gamma = 1$). AMI with the school conferences is denoted for each q by the colored "X".

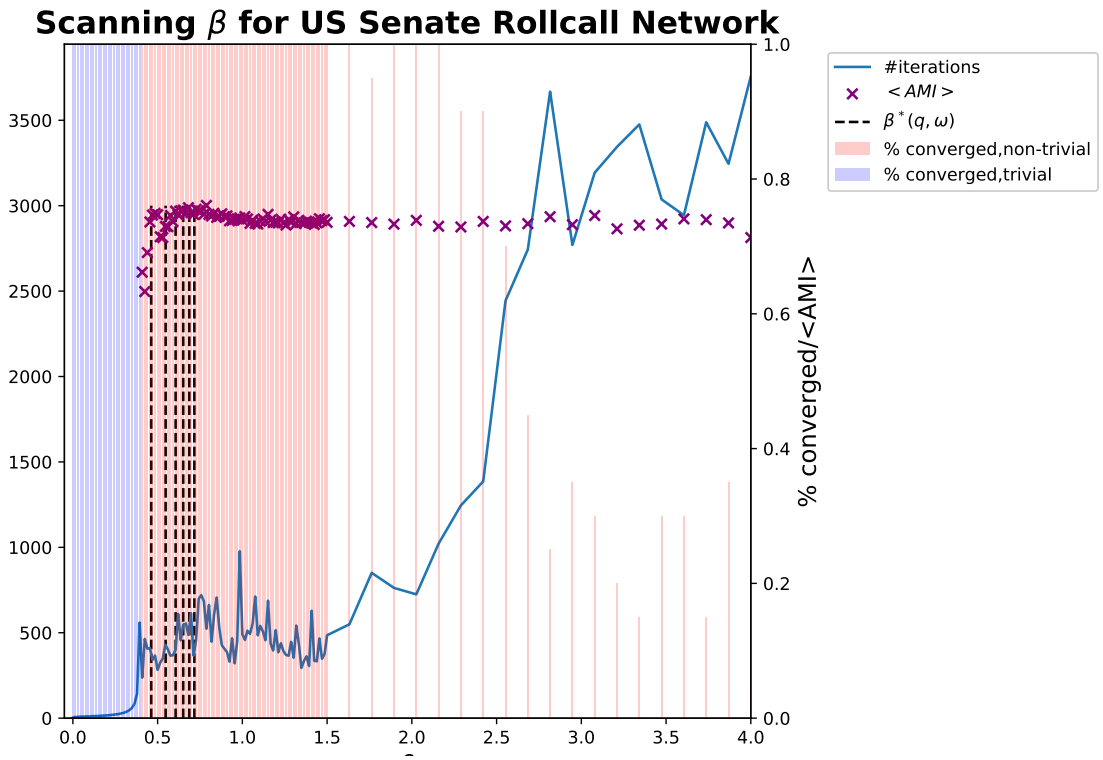

Figure 20: Scanning the β domain for the US Senate Rollcall dataset. We run *multimodbp* on the US Senate Voting similarity network [242], using the KNN (k=10) as described in Section 3.3.2 of the main text. We ran *multimodbp* for a maximum of 4000 iterations across 100 evenly spaced values of $\beta \in [0, 1]$. For each value of β we ran *multimodbp* 5 different times. We show that Shi *et al*'s approach to selecting β^* [208] identifies regions where the algorithm is in the retrieval phase (*i.e* converges to non-trivial partitions). Vertical dashed black lines show calculated value for $\beta^*(q)$ for $q = [4, 6, 8, 10, 12, 14]$. Vertical blue and red bars denote the percentage of runs for that value of β that ultimately converged (percentage is shown by the proportion of the space under the number of iterations curve occupied by the bar). Bar color denotes whether the identified partitions were trivial ($\psi_t^i = \frac{1}{q} \forall i, t$) We see that several of these lie within the observed retrieval phase ($q = [8, 10, 12, 14]$) .

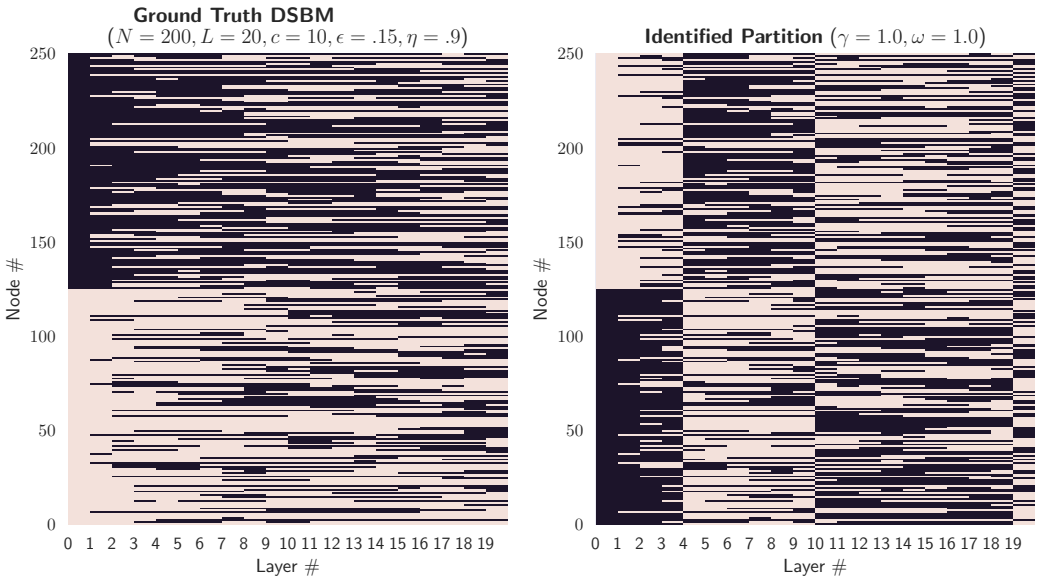

Figure 21: Fragmentation of identified communities across layers. Demonstration of layer "splitting" on the multilayer dynamic stochastic block model (DSBM). Left shows the ground truth planted community assignments while the right shows the communities identified by *multimodbp* without the cross layer assignment procedure. We reiterate that this cross layer label permuting preserves all identified structure within a layer and always results in higher modularity.

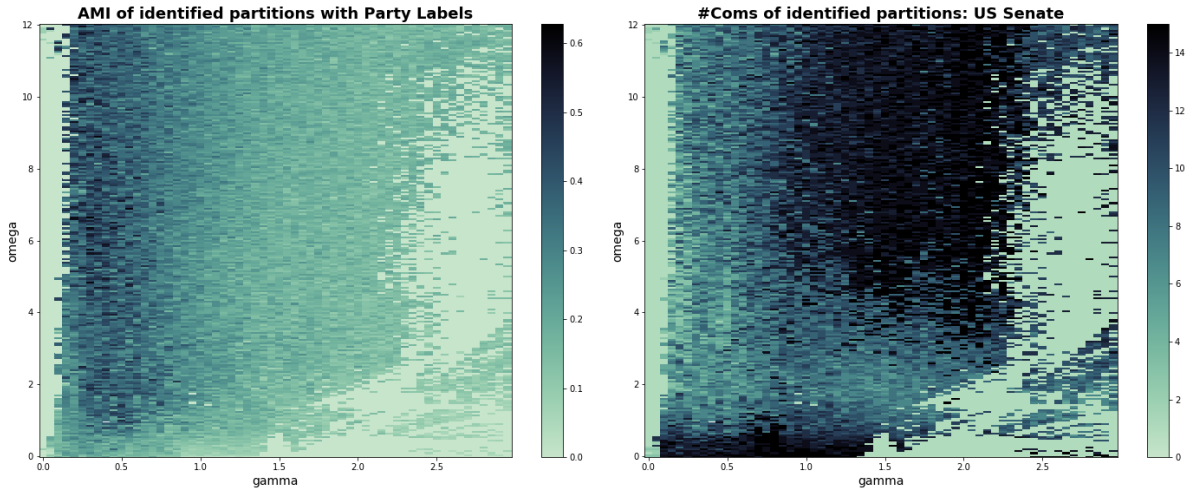

Figure 22: *multimodbp* applied to the US Senate Voting similarity network [242]. Left: AMI of identified partitions with the political party labels using *multimodbp* across a range of γ (x-axis) and ω values. Right: the number of communities identified by the algorithm as a function of the parameters (γ, ω).

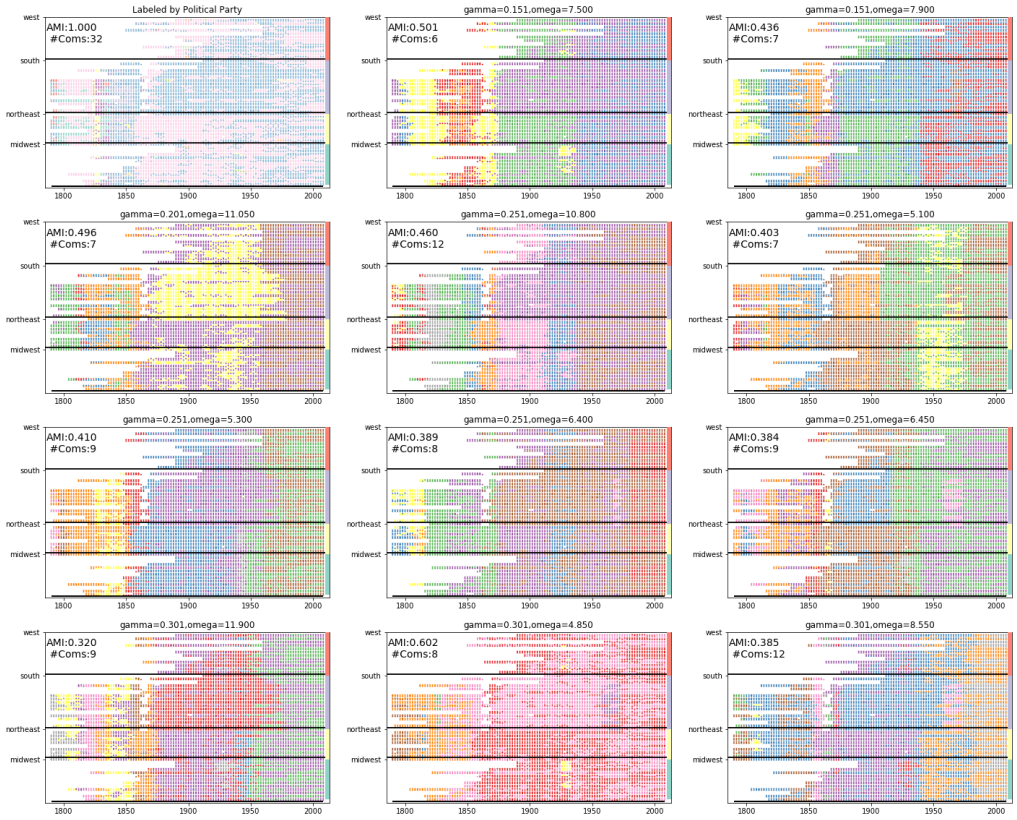

Figure 23: Community structure for lowest free energy partitions identified by *multimodbp*. Top identified partitions based on minimization of the Bethe free energy on the US Senate voting similarity network. In each, each row represents the Senator for a particular State, organized by region, while the x-axis denotes the year of each Congress. Nodes are colored according to their identified partition, while the top left figure is colored by the political party affiliation of each senator.

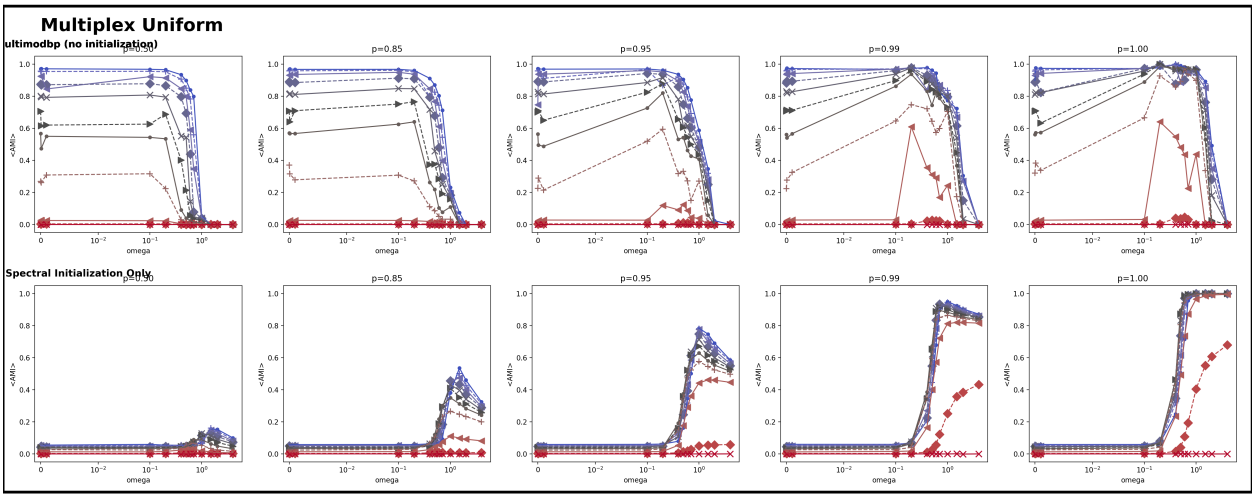

Figure 24: Multiplex benchmark without spectral initialization and only using spectral method. Top row: the performance of *multimodbp* on the uniform multiplex network (as specified in Section 3.3.2 of the main text) *without* the spectral initialization detailed in the main text. Performance of the algorithm at higher omega trails off abruptly. For comparison with *multimodbp* with spectral initialization, see Figure 3.9 in main text. Bottom row: performance of just the spectral initialization (without *multimodbp*). The spectral initialization's performance tends to be better at higher values of omega, complementing the deficiencies in *multimodbp*.

4 Chapter 4 TMB Paradox Supplement

4.1 Proof of the friendship paradox

We can represent a network of N nodes and m edges with an $N \times N$ adjacency matrix, $\mathbf{A}$, where the entries of $\mathbf{A}$ are defined as follows

$$A_{ij} = \begin{cases} 1 \text{ if } (i,j) \in \mathcal{E} \\ 0 \text{ otherwise} \end{cases}$$

where we use $\mathcal{E}$ to denote the set of edges present in the graph, indexed by the pair of nodes connected by each edge. Bipartite networks can be represented by an adjacency matrix as well, with all edges (i,j), connecting only nodes in different classes. The degree of each node, k_i is given by the number of edges connected to node i : $k_i = \sum_j A_{ij}$. We let p_k give us the fraction of nodes that have degree, k, also known as we the degree distribution. This represents the probability that a randomly chosen node will have degree k. The excess degree distribution is the distribution of degrees of a random chosen neighbor of a node. This is equivalent to the probability of that at the end of a randomly chosen edge will have degree k. We show that the

average excess degree is necessary greater than or equal to the average degree of a network, using a proof largely taken from [167].

We begin by computing the probability that after following a randomly chosen edge in our network, we arrive at a node with degree k. Since each edge connects 2 nodes, there $2m$ possible choices of nodes. The probability of ending at any particular node with degree k is $\frac{k}{2m}$. We have $N \times p_k$ nodes with degree k, and thus the probability of following an edge to a node of degree k is given by :

$$\frac{k}{2m} N p_k = \frac{k}{< k >} p_k$$

where we have used the fact that $\frac{2m}{n} =< k >$ gives us the average degree for the network. Thus the excess degree distribution is weighted by a factor of k. We are more likely to choose a higher degree vertex by virtue of the fact that it has more edges coming off of it. We can compute the average excess degree by:

$$\sum_k k \frac{k}{< k >} p_k = \sum_k \frac{k^2}{< k >} p_k = \frac{< k^2 >}{< k >}$$

We can compute the difference between the average excess degree and the average degree:

$$\frac{< k^2 >}{< k >} - < k >= \frac{< k^2 >}{< k >} - \frac{< k >^2}{< k >} =$$
$$\frac{1}{< k >} \left(< k^2 > - < k >^2 \right) = \frac{1}{< k >} Var(k)$$

where $Var(k)$ is the variance of the degree distribution. This is strictly non-negative and is zero in the case where all nodes have the same degree. Since the difference between the average excess degree and the average degree is always non-negative, the average excess degree must always be greater than the average degree. Furthermore we see that this difference is proportional to the variance of the degree distribution, which means that the more heavy tailed the degree distribution is, the larger the difference in the average excess degree and the average degree. Each of the lines of logic we have used above holds for a bipartite network as well, and thus we have our explanation of how the TMB paradox arises.

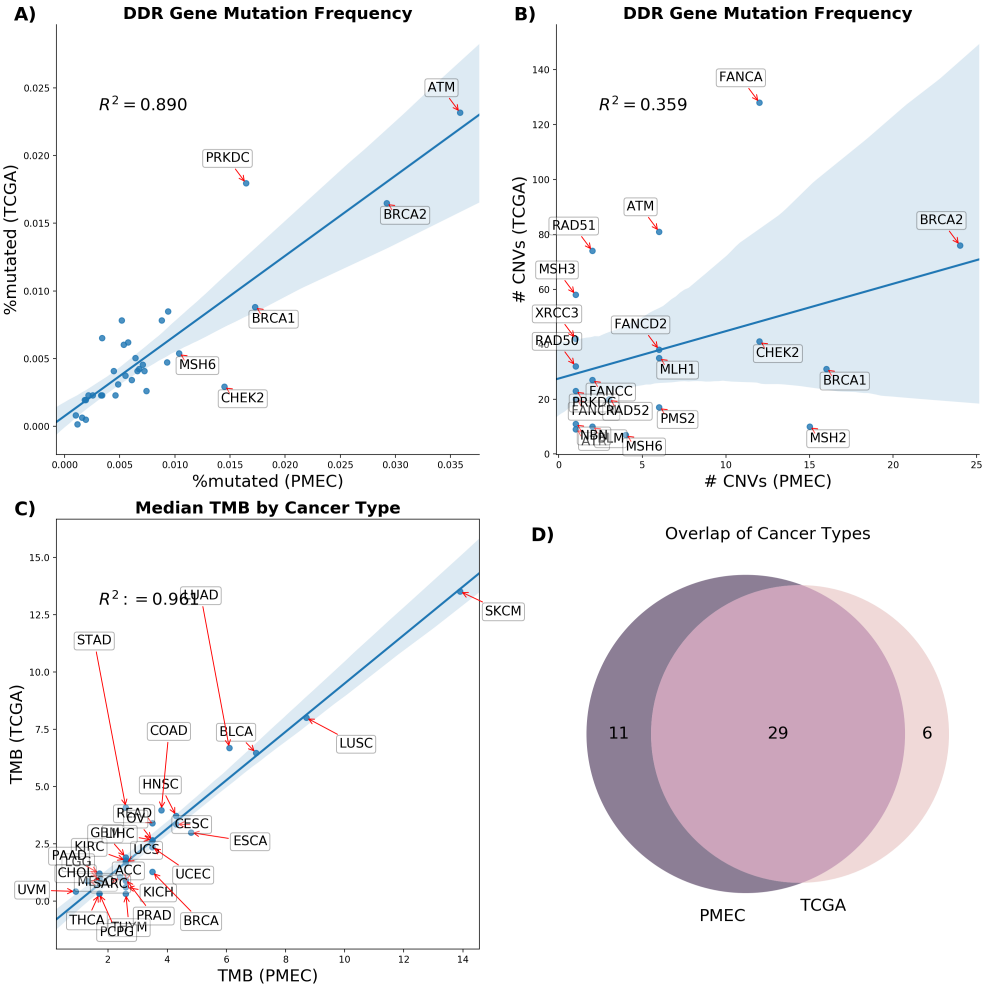

Figure 25: Comparson of general characteristics of PMEC vs TCGA. A Scatter of the short mutation (SNV+indel) frequency for the DDR genes in PMEC (x-axis) vs TCGA(y-axis). **B)** Scatter of the CNV frequency for the DDR genes in PMEC (x-axis) vs TCGA(y-axis). **C)** Scatter of median TMB levels by cancer type for PMEC vs TCGA. **D)** Venn diagram of overlap between broad cancer types in TCGA vs PMEC.

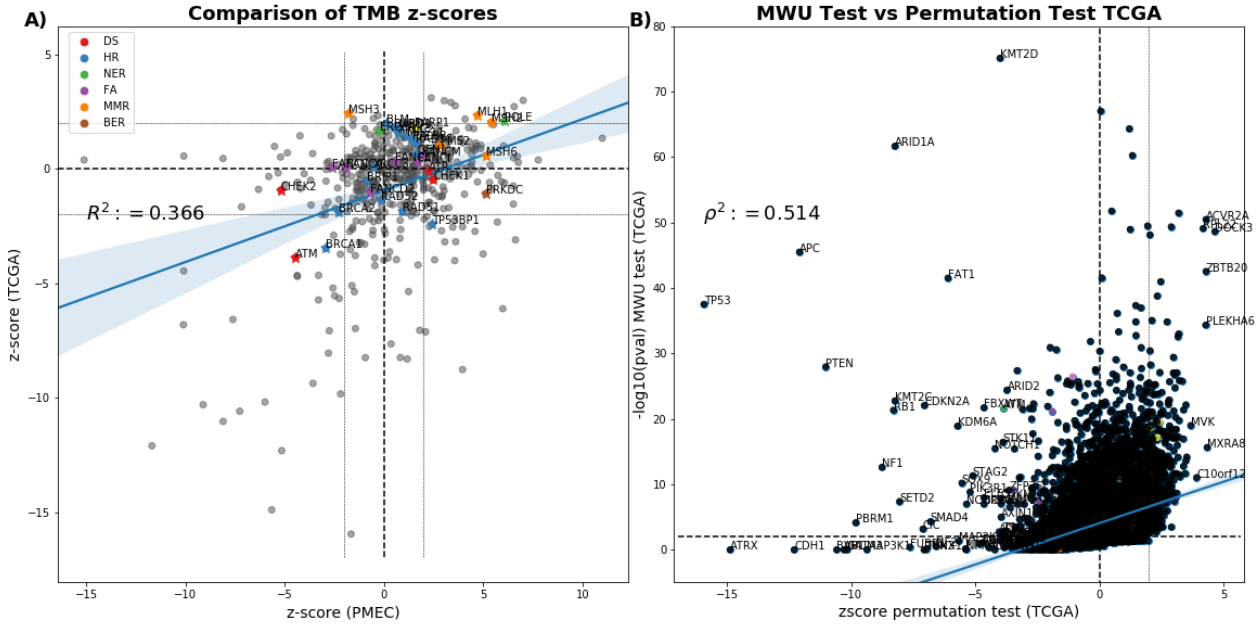

Figure 26: Permutation test z-scores for all PMEC genes. A We scatter the z-scores for the permutation test for the 481 genes in PMEC for TCGA vs PMEC. Note that scores for the TCGA dataset were derived using the full 18K genes. **B)** We also show how the $-\log_{10}$ p-value for the Mann-Whitney U test compares to the z-score for the full TCGA dataset using all 18K genes.

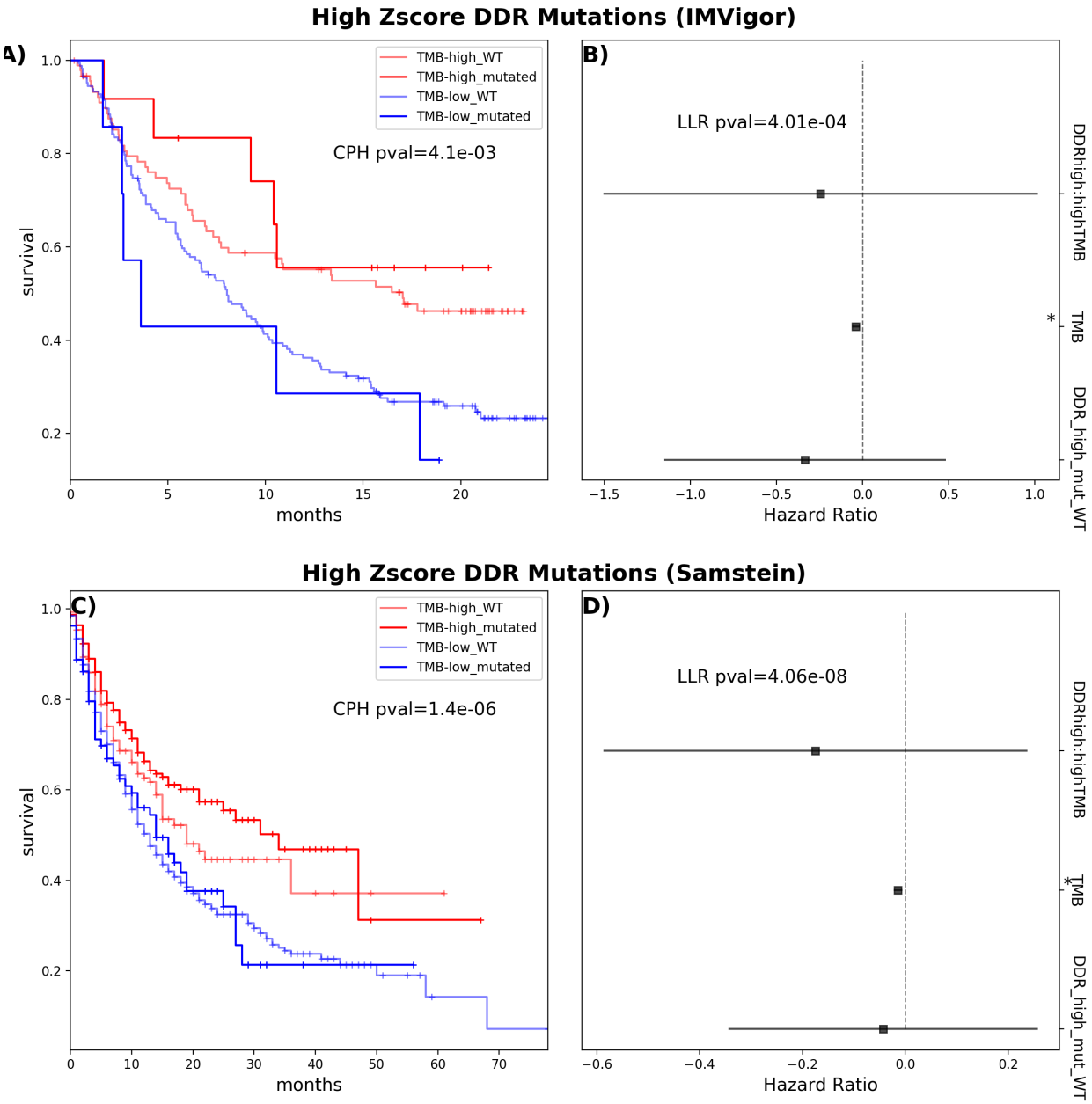

Figure 27: Testing for association with survival in samples with a high z-score DDR mutation. A) Kaplan-Meyer curve for IMVigor samples binned according to high vs low TMB and mutated or WT in high z-score DDR genes. **B)** Cox-proportional hazard model for IMVigor fitting TMB (as continuous variable), along with mutation in high z-score DDR genes, as well as a cross term between TMB and mutation in high z-score DDR genes. **C)** and **D)** Analogous plots for the Samstein *et al.* dataset.

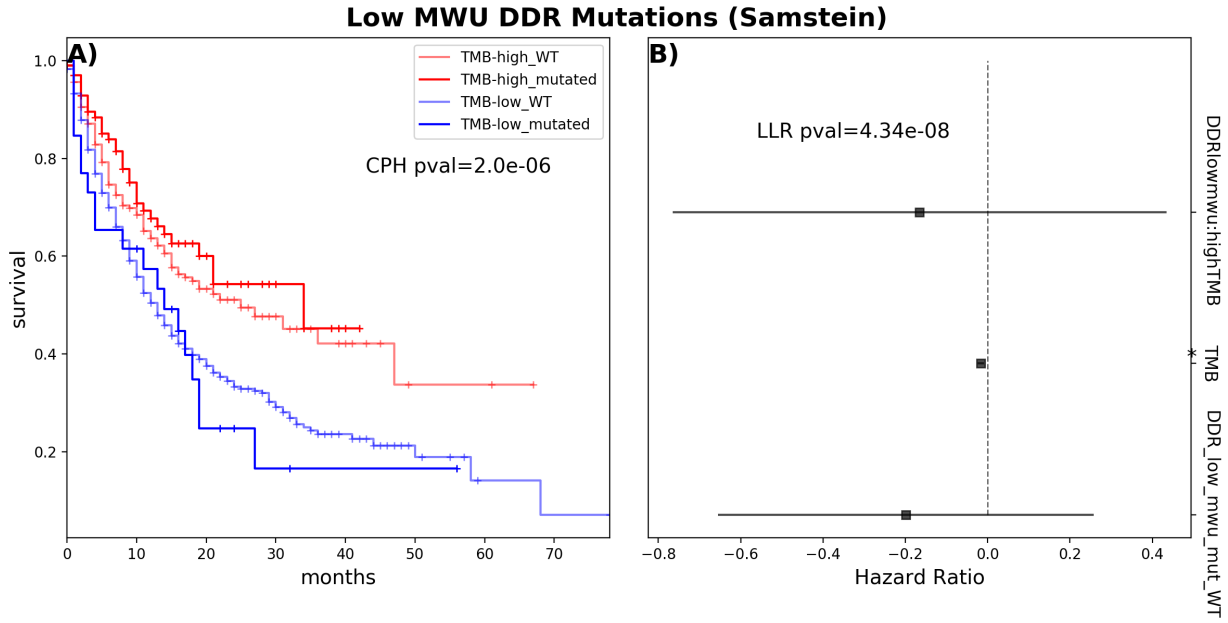

Figure 28: Testing for association with survival in samples with a low MWU score DDR mutations. **A** Kaplan-Meyer curve for Samstein *et al.* samples binned according to high vs low TMB and mutated or WT in low MWU test DDR genes. **B)** Cox-proportional hazard model for IMVigor fitting TMB (as continuous variable), along with mutation in low MWU test DDR genes, as well as a cross term between TMB and mutation in low MWU test DDR genes.

BIBLIOGRAPHY

[1] Detecting communities using Pajek / PajekXXL. URL http://mrvar.fdv.uni-lj.si/pajek/community/CommunityDrawExample.htm.

[2] http://www.qhull.org/.

[3] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

[4] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002.

[5] Ludmil B Alexandrov, Serena Nik-Zainal, David C Wedge, Samuel A J R Aparicio, Sam Behjati, Andrew V Biankin, Graham R Bignell, Niccolò Bolli, Ake Borg, Anne-Lise Børresen-Dale, Sandrine Boyault, Birgit Burkhardt, Adam P Butler, Carlos Caldas, Helen R Davies, Christine Desmedt, Roland Eils, Jórunn Erla Eyfjörd, John A Foekens, Mel Greaves, Fumie Hosoda, Barbara Hutter, Tomislav Ilicic, Sandrine Imbeaud, Marcin Imielinski, Natalie Jäger, David T W Jones, David Jones, Stian Knappskog, Marcel Kool, Sunil R Lakhani, Carlos López-Otín, Sancha Martin, Nikhil C Munshi, Hiromi Nakamura, Paul A Northcott, Marina Pajic, Elli Papaemmanuil, Angelo Paradiso, John V Pearson, Xose S Puente, Keiran Raine, Manasa Ramakrishna, Andrea L Richardson, Julia Richter, Philip Rosenstiel, Matthias Schlesner, Ton N Schumacher, Paul N Span, Jon W Teague, Yasushi Totoki, Andrew N J Tutt, Rafael Valdés-Mas, Marit M van Buuren, Laura van t Veer, Anne Vincent-Salomon, Nicola Waddell, Lucy R Yates, Australian Pancreatic Cancer Genome Initiative, ICGC Breast Cancer Consortium, ICGC MMML-Seq Consortium, ICGC PedBrain, Jessica Zucman-Rossi, P Andrew Futreal, Ultan McDermott, Peter Lichter, Matthew Meyerson, Sean M Grimmond, Reiner Siebert, Elías Campo, Tatsuhiro Shibata, Stefan M Pfister, Peter J Campbell, and Michael R Stratton. Signatures of mutational processes in human cancer. *Nature*, 500(7463):415 EP —-421, August 2013.

[6] Stefano Allesina and Mercedes Pascual. Food web models: a plea for groups. *Ecology Letters*, 12(7):652–662, July 2009.

[7] A Arenas, A Fernández, and S Gómez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, May 2008.

[8] Alex Arenas, Jordi Duch, Alberto Fernández, and Sergio Gómez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9(6):176, 2007.

[9] Özgün Babur, Mithat Gönen, Bulent Arman Aksoy, Nikolaus Schultz, Giovanni Ciriello, Chris Sander, and Emek Demir. Systematic identification of cancer driving signaling pathways based on mutual exclusivity of genomic alterations. *Genome Biology*, 16(1):45, 2015.

[10] James P Bagrow. Communities and bottlenecks: trees and treelike networks have high modularity. *Physical Review E*, 85(6 Pt 2):066118, June 2012.

[11] Arjun V Balar, Matthew D Galsky, Jonathan E Rosenberg, Thomas Powles, Daniel P Petrylak, Joaquim Bellmunt, Yohann Loriot, Andrea Necchi, Jean Hoffman-Censits,

Jose Luis Perez-Gracia, Nancy A Dawson, Michiel S van der Heijden, Robert Dreicer, Sandy Srinivas, Margitta M Retz, Richard W Joseph, Alexandra Drakaki, Ulka N Vaishampayan, Srikala S Sridhar, David I Quinn, Ignacio Durán, David R Shaffer, Bernhard J Eigl, Petros D Grivas, Evan Y Yu, Shi Li, Edward E Kadel, Zachary Boyd, Richard Bourgon, Priti S Hegde, Sanjeev Mariathasan, AnnChristine Thåström, Oyewale O Abidoye, Gregg D Fine, and Dean F Bajorin. Atezolizumab as first-line treatment in cisplatin-ineligible patients with locally advanced and metastatic urothelial carcinoma: a single-arm, multicentre, phase 2 trial. *The Lancet*, 389(10064):67–76, 2017.

[12] Shantanu Banerji, Kristian Cibulskis, Claudia Rangel-Escareno, Kristin K Brown, Scott L Carter, Abbie M Frederick, Michael S Lawrence, Andrey Y Sivachenko, Carrie Sougnez, Lihua Zou, Maria L Cortes, Juan C Fernandez-Lopez, Shouyong Peng, Kristin G Ardlie, Daniel Auclair, Veronica Bautista-Piña, Fujiko Duke, Joshua Francis, Joonil Jung, Antonio Maffuz-Aziz, Robert C Onofrio, Melissa Parkin, Nam H Pho, Valeria Quintanar-Jurado, Alex H Ramos, Rosa Rebollar-Vega, Sergio Rodriguez-Cuevas, Sandra L Romero-Cordoba, Steven E Schumacher, Nicolas Stransky, Kristin M Thompson, Laura Uribe-Figueroa, Jose Baselga, Rameen Beroukhim, Kornelia Polyak, Dennis C Sgroi, Andrea L Richardson, Gerardo Jimenez-Sanchez, Eric S Lander, Stacey B Gabriel, Levi A Garraway, Todd R Golub, Jorge Melendez-Zajgla, Alex Toker, Gad Getz, Alfredo Hidalgo-Miranda, and Matthew Meyerson. Sequence analysis of mutations and translocations across breast cancer subtypes. *Nature Publishing Group*, 486(7403):405–409, 2012.

[13] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, 1996.

[14] Michael J Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6):P09008, December 2007.

[15] Matteo Barigozzi, Giorgio Fagiolo, and Giuseppe Mangioni. Identifying the community structure of the international-trade multi-network. *Physica A: Statistical Mechanics and its Applications*, 390(11):2051–2066, 2011.

[16] D S Bassett, N F Wymbs, M A Porter, Peter J Mucha, J M Carlson, and S T Grafton. Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, 108(18):7641–7646, May 2011.

[17] Danielle S Bassett, Mason A Porter, Nicholas F Wymbs, Scott T Grafton, Jean M Carlson, and Peter J Mucha. Robust detection of dynamic community structure in networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(1):013142, March 2013.

[18] Danielle S Bassett, Nicholas F Wymbs, M Puck Rombach, Mason A Porter, Peter J Mucha, and Scott T Grafton. Task-Based Core-Periphery Organization of Human Brain Dynamics. *PLoS Computational Biology*, 9(9):e1003171 EP –, September 2013.

[19] Danielle S Bassett, Nicholas F Wymbs, Mason A Porter, Peter J Mucha, and Scott T Grafton. Cross-linked structure of network evolution. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 24(1):013112, January 2014.

[20] M Bazzi, M Porter, S Williams, M McDonald, D Fenn, and S Howison. Community Detection in Temporal Multilayer Networks, with an Application to Correlation Networks. *Multiscale Modeling & Simulation*, 14(1):1–41, January 2016.

[21] Marya Bazzi, Lucas G S Jeub, Alex Arenas, Sam D Howison, and Mason A Porter. Generative Benchmark Models for Mesoscale Structure in Multilayer Networks. August 2019.

[22] Tim Beißbarth and Terence P Speed. GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics*, 20(9):1464–1465, February 2004.

[23] Laura Bennett, Aristotelis Kittas, Songsong Liu, Lazaros G Papageorgiou, and Sophia Tsoka. Community Structure Detection for Overlapping Modules through Mathematical Programming in Protein Interaction Networks. *PLOS ONE*, 9(11):e112821, November 2014.

[24] Ellen L Berg. Systems biology in drug discovery and development. *Drug Discovery Today*, 19(2):113–125, 2014.

[25] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. Finding Redundant and Complementary Communities in Multidimensional Networks. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 2181–2184, New York, NY, USA, 2011. Association for Computing Machinery.

[26] Ginestra Bianconi. Statistical mechanics of multiplex networks: Entropy and overlap. *Physical Review E*, 87(6):062806, June 2013.

[27] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[28] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, February 2008.

[29] Anna D Broido and Aaron Clauset. Scale-free networks are rare. *Nature Communications*, 10(1):1017–10, 2019.

[30] Jean-Philippe Brunet, Pablo Tamayo, Todd R Golub, and Jill P Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, 2004.

[31] Sander Canisius, John W M Martens, and Lodewyk F A Wessels. A novel independence test for somatic alterations in cancer shows that biology drives mutual exclusivity but chance explains most co-occurrence. *Genome Biology*, 17(1):261, December 2016.

[32] Laura Cantini, Enzo Medico, Santo Fortunato, and Michele Caselle. Detection of gene communities in multi-networks reveals cancer drivers. *Scientific reports*, 5(1):1–10, December 2015.

[33] Ron Caspi, Tomer Altman, Kate Dreher, Carol A Fulcher, Pallavi Subhraveti, Ingrid M Keseler, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Lukas A Mueller, Quang Ong, Suzanne Paley, Anuradha Pujar, Alexander G Shearer, Michael Travers, Deepika Weerasinghe, Peifen Zhang, and Peter D Karp. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic acids research*, 40(D1):D742–D753, November 2011.

[34] Enrico Castellucci, Tianfang He, D Yitzchak Goldstein, Balazs Halmos, and Jennifer Chuy. DNA Polymerase ϵ Deficiency Leading to an Ultramutator Phenotype: A Novel Clinically Relevant Entity. *The oncologist*, 22(5):497–502, May 2017.

[35] Zachary R Chalmers, Caitlin F Connelly, David Fabrizio, Laurie Gay, Siraj M Ali, Riley Ennis, Alexa Schrock, Brittany Campbell, Adam Shlien, Juliann Chmielecki, Franklin Huang, Yuting He, James Sun, Uri Tabori, Mark Kennedy, Daniel S Lieber, Steven Roels, Jared White, Geoffrey A Otto, Jeffrey S Ross, Levi Garraway, Vincent A Miller, Phillip J Stephens, and Garrett M Frampton. Analysis of 100,000 human cancer genomes reveals the landscape of tumor mutational burden. *Genome medicine*, 9(1):34, April 2017.

[36] Li Chen, Jianhua Xuan, Rebecca B Riggins, Robert Clarke, and Yue Wang. Identifying cancer biomarkers by network-constrained support vector machines. *BMC Systems Biology*, 5(1):161–2009, 2011.

[37] Donavan T Cheng, Talia N Mitchell, Ahmet Zehir, Ronak H Shah, Ryma Benayed, Aijazuddin Syed, Raghu Chandramohan, Zhen Yu Liu, Helen H Won, Sasinya N Scott, A Rose Brannon, Catherine O'Reilly, Justyna Sadowska, Jacklyn Casanova, Angela Yannes, Jaclyn F Hechtman, Jinjuan Yao, Wei Song, Dara S Ross, Alifya Oultache, Snjezana Dogan, Laetitia Borsu, Meera Hameed, Khedoudja Nafa, Maria E Arcila, Marc Ladanyi, and Michael F Berger. Memorial Sloan Kettering-Integrated Mutation Profiling of Actionable Cancer Targets (MSK-IMPACT): A Hybridization Capture-Based Next-Generation Sequencing Clinical Assay for Solid Tumor Molecular Oncology. *The Journal of Molecular Diagnostics*, 17(3):251–264, 2015.

[38] Han-Yu Chuang, Eunjung Lee, Yu-Tsueng Liu, Doheon Lee, and Trey Ideker. Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, 3 (1):D418, 2007.

[39] Giovanni Ciriello, Ethan Cerami, Chris Sander, and Nikolaus Schultz. Mutual exclusivity analysis identifies oncogenic network modules. *Genome research*, 22(2):398–406, February 2012.

[40] Giovanni Ciriello, Ethan Cerami, Chris Sander, and Nikolaus Schultz. Mutual exclusivity analysis identifies oncogenic network modules. *Genome research*, 22(2):398–406, February 2012.

[41] Aaron Clauset, Cosma Rohilla Shalizi, and M E J Newman. Power-Law Distributions in Empirical Data. *SIAM Review*, 51(4):661–703, November 2009.

[42] Phillip E C Compeau. How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology*, 29(11):987–991, November 2011.

[43] Skyler J Cranmer, Elizabeth J Menninga, and Peter J Mucha. Kantian fractionalization predicts the conflict propensity of the international system. *Proceedings of the National Academy of Sciences*, 112(38):11812–11816, 2015.

[44] Jeffrey S Damrauer, Katherine A Hoadley, David D Chism, Cheng Fan, Christopher J Tiganelli, Sara E Wobker, Jen Jen Yeh, Matthew I Milowsky, Gopa Iyer, Joel S Parker, and William Y Kim. Intrinsic subtypes of high-grade bladder cancer reflect the hallmarks of breast cancer biology. *Proceedings of the National Academy of Sciences*, 111(8): 3110–3115, 2014.

[45] Jishnu Das and Haiyuan Yu. HINT: High-quality protein interactomes and their applications in understanding human disease. *BMC Systems Biology*, 6(1):92–12, 2012.

[46] J J Daudin, F Picard, and S Robin. A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183, 2008.

[47] Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathematical Formulation of Multilayer Networks. *Physical Review X*, 3(4):1082, December 2013.

[48] Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. Identifying Modular Flows on Multilayer Networks Reveals Highly Overlapping Organization in Interconnected Systems. *Physical Review X*, 5(1):011027, March 2015.

[49] Fabien de Montgolfier, Mauricio Soto, and Laurent Viennot. Asymptotic modularity of some graph classes. *International Symposium on Algorithms and Computation*, pages 435–444, 2011.

[50] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, December 2011.

[51] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and Phase Transitions in the Detection of Modules in Sparse Networks. *Physical Review Letters*, 107(6):065701, August 2011.

[52] Minghua Deng, Zhidong Tu, Fengzhu Sun, and Ting Chen. Mapping gene ontology to proteins based on protein–protein interaction data. *Bioinformatics*, 20(6):895–902, April 2004.

[53] Yulan Deng, Shangyi Luo, Chunyu Deng, Tao Luo, Wenkang Yin, Hongyi Zhang, Yong Zhang, Xinxin Zhang, Yujia Lan, Yanyan Ping, Yun Xiao, and Xia Li. Identifying mutual exclusivity across cancer genomes: computational approaches to discover genetic interaction and reveal tumor vulnerability. *Briefings in Bioinformatics*, 45:1113, August 2017.

[54] Li Ding, Michael C Wendl, Daniel C Koboldt, and Elaine R Mardis. Analysis of next-generation genomic data in cancer: accomplishments and challenges. *Human Molecular Genetics*, 19(R2):R188–R196, September 2010.

[55] Kyle Ellrott, Matthew H Bailey, Gordon Saksena, Kyle R Covington, Cyriac Kandoth, Chip Stewart, Julian Hess, Singer Ma, Kami E Chiotti, Michael McLellan, Carolyn Hutter, David Wheeler, Li Ding, Samantha J Caesar-Johnson, John A Demchok, Ina Felau, Melpomeni Kasapi, Martin L Ferguson, Carolyn M Hutter, Heidi J Sofia, Roy Tarnuzzer, Zhining Wang, Liming Yang, Jean C Zenklusen, Jiashan Julia Zhang, Sudha Chudamani, Jia Liu, Laxmi Lolla, Rashi Naresh, Todd Pihl, Qiang Sun, Yunhu Wan, Ye Wu, Juok Cho, Timothy DeFreitas, Scott Frazer, Nils Gehlenborg, Gad Getz, David I Heiman, Jaegil Kim, Michael S Lawrence, Pei Lin, Sam Meier, Michael S Noble, Doug Voet, Hailei Zhang, Brady Bernard, Nyasha Chambwe, Varsha Dhankani, Theo Knijnenburg, Roger Kramer, Kalle Leinonen, Yuexin Liu, Michael Miller, Sheila Reynolds, Ilya Shmulevich, Vesteinn Thorsson, Wei Zhang, Rehan Akbani, Bradley M Broom, Apurva M Hegde, Zhenlin Ju, Rupa S Kanchi,

Anil Korkut, Jun Li, Han Liang, Shiyun Ling, Wenbin Liu, Yiling Lu, Gordon B Mills, Kwok-Shing Ng, Arvind Rao, Michael Ryan, Jing Wang, John N Weinstein, Jiexin Zhang, Adam Abeshouse, Joshua Armenia, Debyani Chakravarty, Walid K Chatila, Ino de Bruijn, Jianjiong Gao, Benjamin E Gross, Zachary J Heins, Ritika Kundra, Konnor La, Marc Ladanyi, Augustin Luna, Moriah G Nissan, Angelica Ochoa, Sarah M Phillips, Ed Reznik, Francisco Sanchez-Vega, Chris Sander, Nikolaus Schultz, Robert Sheridan, S Onur Sumer, Yichao Sun, Barry S Taylor, Jioajiao Wang, Hongxin Zhang, Pavana Anur, Myron Peto, Paul Spellman, Christopher Benz, Joshua M Stuart, Christopher K Wong, Christina Yau, D Neil Hayes, Parker, Matthew D Wilkerson, Adrian Ally, Miruna Balasundaram, Reanne Bowlby, Denise Brooks, Rebecca Carlsen, Eric Chuah, Noreen Dhalla, Robert Holt, Steven J M Jones, Katayoon Kasaian, Darlene Lee, Yussanne Ma, Marco A Marra, Michael Mayo, Richard A Moore, Andrew J Mungall, Karen Mungall, A Gordon Robertson, Sara Sadeghi, Jacqueline E Schein, Payal Sipahimalani, Angela Tam, Nina Thiessen, Kane Tse, Tina Wong, Ashton C Berger, Rameen Beroukhim, Andrew D Cherniack, Carrie Cibulskis, Stacey B Gabriel, Galen F Gao, Gavin Ha, Matthew Meyerson, Steven E Schumacher, Juliann Shih, Melanie H Kucherlapati, Raju S Kucherlapati, Stephen Baylin, Leslie Cope, Ludmila Danilova, Moiz S Bootwalla, Phillip H Lai, Dennis T Maglinte, David J Van Den Berg, Daniel J Weisenberger, J Todd Auman, Saianand Balu, Tom Bodenheimer, Cheng Fan, Katherine A Hoadley, Alan P Hoyle, Stuart R Jefferys, Corbin D Jones, Shaowu Meng, Piotr A Mieczkowski, Lisle E Mose, Amy H Perou, Charles M Perou, Jeffrey Roach, Yan Shi, Janae V Simons, Tara Skelly, Matthew G Soloway, Donghui Tan, Umadevi Veluvolu, Huihui Fan, Toshinori Hinoue, Peter W Laird, Hui Shen, Wanding Zhou, Michelle Bellair, Kyle Chang, Kyle Covington, Chad J Creighton, Huyen Dinh, HarshaVardhan Doddapaneni, Lawrence A Donehower, Jennifer Drummond, Richard A Gibbs, Robert Glenn, Walker Hale, Yi Han, Jianhong Hu, Viktoriya Korchina, Sandra Lee, Lora Lewis, Wei Li, Xiuping Liu, Margaret Morgan, Donna Morton, Donna Muzny, Jireh Santibanez, Margi Sheth, Eve Shinbrot, Linghua Wang, Min Wang, David A Wheeler, Liu Xi, Fengmei Zhao, Elizabeth L Appelbaum, Matthew Bailey, Matthew G Cordes, Catrina C Fronick, Lucinda A Fulton, Robert S Fulton, Elaine R Mardis, Michael D McLellan, Christopher A Miller, Heather K Schmidt, Richard K Wilson, Daniel Crain, Erin Curley, Johanna Gardner, Kevin Lau, David Mallery, Scott Morris, Joseph Paulauskis, Robert Penny, Candace Shelton, Troy Shelton, Mark Sherman, Eric Thompson, Peggy Yena, Jay Bowen, Julie M Gastier-Foster, Mark Gerken, Kristen M Leraas, Tara M Lichtenberg, Nilsa C Ramirez, Lisa Wise, Erik Zmuda, Niall Corcoran, Tony Costello, and Christopher and... Hovens. Scalable Open Science Approach for Mutation Calling of Tumor Exomes Using Multiple Genomic Pipelines. *Cell Systems*, 6(3):271–281.e7, 2018.

[56] Scott Emmons and Peter J Mucha. Map equation with metadata: Varying the role of attributes in community detection. *Physical Review E*, 100(2):022301, August 2019.

[57] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, September 2012.

[58] Tim S Evans. Clique graphs and overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(12):P12037, 2010.

[59] Scott L Feld. Why Your Friends Have More Friends Than You Do. *American Journal of Sociology*, 96(6):1464–1477, 1991.

[60] Daniel J Fenn, Mason A Porter, Mark McDonald, Stacy Williams, Neil F Johnson, and Nick S Jones. Dynamic communities in multichannel data: An application to the foreign exchange market during the 2007–2008 credit crisis. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(3):033119, August 2009.

[61] Daniel J Fenn, Mason A Porter, Peter J Mucha, Mark McDonald, Stacy Williams, Neil F Johnson, and Nick S Jones. Dynamical clustering of exchange rates. *Quantitative Finance*, 12(10):1493–1520, October 2012.

[62] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

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

[64] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016.

[65] Bailey K Fosdick, Daniel B Larremore, Joel Nishimura, and Johan Ugander. Configuring Random Graph Models with Fixed Degree Sequences. *SIAM Review*, 60(2):315–355, January 2018.

[66] E B Fowlkes and C L Mallows. A Method for Comparing Two Hierarchical Clusterings. *Journal of the american statistical association*, 78(383):553, September 1983.

[67] G M Frampton, D A Fabrizio, Z R Chalmers, J X Sun, V A Miller, and P J Stephens. Assessment and comparison of tumor mutational burden and microsatellite instability status in $>$ 40,000 cancer genomes. *Annals of Oncology*, 27(suppl6), October 2016.

[68] Ana L.N. Fred and Anil K Jain. Robust data clustering. In *Computer Vision and Pattern Recognition, . Proceedings. IEEE Computer Society Conference on*, pages II–128–II–133 vol. 2. IEEE, 2003.

[69] Levi A Garraway and Eric S Lander. Lessons from the Cancer Genome. *Cell*, 153(1):17–37, March 2013.

[70] Amir Ghasemian, Pan Zhang, Aaron Clauset, Cristopher Moore, and Leto Peel. Detectability Thresholds and Optimal Algorithms for Community Structure in Dynamic Networks. *Physical Review X*, 6(3):031005, July 2016.

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

[72] Olivier Goldschmidt and Dorit S Hochbaum. A Polynomial Algorithm for the k-cut Problem for Fixed k. *Mathematics of OR*, 19(1):24–37, February 1994.

[73] Sergio Gómez, Pablo Jensen, and Alex Arenas. Analysis of community structure in networks of correlated data. *Physical Review E*, 80(1):583, July 2009.

[74] Sergio Gómez, Pablo Jensen, and Alex Arenas. Analysis of community structure in networks of correlated data. *Physical Review E*, 80(1):016114, July 2009.

[75] Abel Gonzalez-Perez, Alba Jene-Sanz, and Nuria Lopez-Bigas. The mutational landscape of chromatin regulatory factors across 4,623 tumor samples. *Genome Biology*, 14(9): r106–15, 2013.

[76] Benjamin H Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 81(4 Pt 2):046106, April 2010.

[77] Aaron M Goodman, Shumei Kato, Lyudmila Bazhenova, Sandip P Patel, Garrett M Frampton, Vincent Miller, Philip J Stephens, Gregory A Daniels, and Razelle Kurzrock. Tumor Mutational Burden as an Independent Predictor of Response to Immunotherapy in Diverse Cancers. *Molecular cancer therapeutics*, 16(11):2598–2608, November 2017.

[78] Clara Granell, Sergio Gómez, and Alex Arenas. Mesoscopic analysis of networks: Applications to exploratory analysis and data clustering. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(1):016102, March 2011.

[79] Casey S Greene, Arjun Krishnan, Aaron K Wong, Emanuela Ricciotti, Rene A Zelaya, Daniel S Himmelstein, Ran Zhang, Boris M Hartmann, Elena Zaslavsky, Stuart C Sealfon, Daniel I Chasman, Garret A FitzGerald, Kara Dolinski, Tilo Grosser, and Olga G Troyanskaya. Understanding multicellular function and disease with human tissue-specific networks. *Nature genetics*, 47(6):569–576C, June 2015.

[80] Heonjong Han, Hongseok Shim, Donghyun Shin, Jung Eun Shim, Yunhee Ko, Junha Shin, Hanhae Kim, Ara Cho, Eiru Kim, Tak Lee, Hyojin Kim, Kyungsoo Kim, Sunmo Yang, Dasom Bae, Ayoung Yun, Sunphil Kim, Chan Yeong Kim, Hyeon Jin Cho, Byunghee Kang, Susie Shin, and Insuk Lee. TRRUST: a reference database of human transcriptional regulatory interactions. *Scientific reports*, 5(1):11432–11, 2015.

[81] Qiuyi Han, Kevin S Xu 0001, and Edoardo M Airoldi. Consistent estimation of dynamic and multi-layer block models. *ICML*, 2015.

[82] Qiuyi Han, Kevin Xu, and Edoardo Airoldi. Consistent estimation of dynamic and multi-layer block models. In *International Conference on Machine Learning*, pages 1511–1520, 2015.

[83] Qiuyi Han, Kevin Xu, and Edoardo Airoldi. Consistent estimation of dynamic and multi-layer block models. pages 1511–1520, 2015.

[84] Douglas Hanahan and Robert A Weinberg. Hallmarks of Cancer: The Next Generation. *Cell*, 144(5):646–674, March 2011.

[85] Julia Handl, Joshua Knowles, and Douglas B Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, August 2005.

[86] M B Hastings. Community detection as an inference problem. *Physical Review E*, 74(3): 035102, September 2006.

[87] Timothy C Havens, James C Bezdek, Christopher Leckie, Kotagiri Ramamohanarao, and Marimuthu Palaniswami. A Soft Modularity Function For Detecting Fuzzy Communities in Social Networks. *IEEE Transactions on Fuzzy Systems*, 21(6):1170–1175, December 2013.

[88] Kunihiko Hinohara and Kornelia Polyak. Intratumoral Heterogeneity: More Than Just Mutations. *Trends in Cell Biology*, 29(7):569–579, July 2019.

[89] Katherine A Hoadley, Toshinori Hinoue, Denise M Wolf, Esther Drill, Ronglai Shen, Alison M Taylor, Michael S Lawrence, Vesteinn Thorsson, Rehan Akbani, Francisco Sanchez-Vega, Christopher K Wong, Christina Yau, Reanne Bowlby, A Gordon Robertson, Andrew D Cherniack, Maciej Wiznerowicz, Houtan Noushmehr, Alexander J Lazar, Joshua M Stuart, and Peter W Laird. Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell*, 173(2):291–304, 2018.

[90] Matan Hofree, John P Shen, Hannah Carter, Andrew Gross, and Trey Ideker. Network-based stratification of tumor mutations. *Nature Methods*, 10(11):1108–1115, November 2013.

[91] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, June 1983.

[92] Petter Holme. Rare and everywhere: Perspectives on scale-free networks. *Nature Communications*, 10(1):1016–3, 2019.

[93] J P Hou and J Ma. DawnRank: discovering personalized driver genes in cancer. *Genome medicine*, 6(7):204, 2014.

[94] Desislava Hristova, Mirco Musolesi, and Cecilia Mascolo. Keep Your Friends Close and Your Facebook Friends Closer: A Multiplex Network Approach to the Analysis of Offline and Online Social Ties. *Eighth International AAAI Conference on Weblogs and Social Media*, May 2014.

[95] Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1):1–13, November 2008.

[96] Justin K Huang, Daniel E Carlin, Michael Ku Yu, Wei Zhang, Jason F Kreisberg, Pablo Tamayo, and Trey Ideker. Systematic Evaluation of Molecular Networks for Discovery of Disease Genes. *Cell Systems*, 6(4):484–495.e5, April 2018.

[97] Willy Hugo, Jesse M Zaretsky, Lu Sun, Chunying Song, Blanca Homet Moreno, Siwen Hu-Lieskovan, Beata Berent-Maoz, Jia Pang, Bartosz Chmielowski, Grace Cherry, Elizabeth Seja, Shirley Lomeli, Xiangju Kong, Mark C Kelley, Jeffrey A Sosman, Douglas B Johnson, Antoni Ribas, and Roger S Lo. Genomic and Transcriptomic Features of Response to Anti-PD-1 Therapy in Metastatic Melanoma. *Cell*, 165(1):35–44, 2016.

[98] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLOS ONE*, 9(6):1–12, June 2014.

[99] LGS Jeub and M Bazzi. A generative model for mesoscale structure in multilayer networks implemented in MATLAB. Technical report, 2019.

[100] Lucas G S Jeub. A Python framework for generating multilayer networks with planted mesoscale structure. 2019.

[101] Lucas G. S. Jeub, Marya Bazzi, Inderjit S. Jutla, and Peter J. Mucha. A generalized Louvain method for community detection implemented in matlab, 2011–2020. http://netwiki.amath.unc.edu/GenLouvain.

[102] Lucas G S Jeub, Prakash Balachandran, Mason A Porter, Peter J Mucha, and Michael W Mahoney. Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *Physical Review E*, 91(1):012821, January 2015.

[103] Lucas G S Jeub, Olaf Sporns, and Santo Fortunato. Multiresolution Consensus Clustering in Networks. *Scientific reports*, 8(1):3259, 2018.

[104] Lucas GS Jeub, Marya Bazzi, Jutla S Inderjit, and Peter J Mucha. A generalized Louvain method for community detection implemented in MATLAB. *URL http://github.com/GenLouvain/GenLouvain*, 2011.

[105] G Joshi-Tope, M Gillespie, I Vastrik, P D'Eustachio, E Schmidt, B de Bono, B Jassal, G R Gopinath, G R Wu, L Matthews, S Lewis, E Birney, and L Stein. Reactome: a knowledgebase of biological pathways. *Nucleic acids research*, 33(Database issue): D428–32, January 2005.

[106] Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic acids research*, 45(D1):D353–D361, November 2016.

[107] Jordan Kardos, Shengjie Chai, Lisle E Mose, Sara R Selitsky, Bhavani Krishnan, Ryoichi Saito, Michael D Iglesia, Matthew I Milowsky, Joel S Parker, William Y Kim, and Benjamin G Vincent. Claudin-low bladder tumors are immune infiltrated and actively immune suppressed. *JCI insight*, 1(3), March 2016.

[108] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[109] Tanya E Keenan, Kelly P Burke, and Eliezer M Van Allen. Genomic correlates of response to immune checkpoint blockade. *Nature Medicine*, 25(3):389–402, 2019.

[110] Samuel Kerrien, Bruno Aranda, Lionel Breuza, Alan Bridge, Fiona Broackes-Carter, Carol Chen, Margaret Duesbury, Marine Dumousseau, Marc Feuermann, Ursula Hinz, Christine Jandrasits, Rafael C Jimenez, Jyoti Khadake, Usha Mahadevan, Patrick Masson, Ivo Pedruzzi, Eric Pfeiffenberger, Pablo Porras, Arathi Raghunath, Bernd Roechert, Sandra Orchard, and Henning Hermjakob. The IntAct molecular interaction database in 2012. *Nucleic acids research*, 40(D1):D841–D846, November 2011.

[111] Chan Yeong Kim, Sunmo Yang, Insuk Lee, Sohyun Hwang, Edward M Marcotte, Eiru Kim, and Traver Hart. HumanNet v2: human gene networks for disease research. *Nucleic acids research*, 47(D1):D573–D580, November 2018.

[112] Yoo-Ah Kim, Stefan Wuchty, and Teresa M Przytycka. Simultaneous Identification of Causal Genes and Dys-Regulated Pathways in Complex Diseases. In Bonnie Berger, editor, *Research in Computational Molecular Biology*, pages 263–280. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[113] Thomas N Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv*, 2016.

[114] Mikko Kivelä, Alex Arenas, Marc Barthélemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, September 2014.

[115] Theo A Knijnenburg, Michael T Zimmermann, Nyasha Chambwe, Gregory P Way, Casey S Greene, Bin Feng, Chase Miller, Yang Shen, Mostafa Karimi, Haoran Chen, Pora Kim, Peilin Jia, Shaojun Zhang, Jianfang Liu, Hai Hu, Matthew H Bailey, Christina Yau, Denise Wolf, Zhongming Zhao, John N Weinstein, Lei Li, David A Wheeler, Samantha J Caesar-Johnson, John A Demchok, Ina Felau, Melpomeni Kasapi, Martin L Ferguson, Carolyn M Hutter, Heidi J Sofia, Roy Tarnuzzer, Zhining Wang, Liming Yang, Jean C Zenklusen, Jiashan Julia Zhang, Sudha Chudamani, Jia Liu, Laxmi Lolla, Rashi Naresh, Todd Pihl, Qiang Sun, Yunhu Wan, Ye Wu, Juok Cho, Timothy DeFreitas, Scott Frazer, Nils Gehlenborg, Gad Getz, David I Heiman, Jaegil Kim, Michael S Lawrence, Pei Lin, Sam Meier, Michael S Noble, Gordon Saksena, Doug Voet, Hailei Zhang, Brady Bernard, Varsha Dhankani, Theo Knijnenburg, Roger Kramer, Kalle Leinonen, Yuexin Liu, Michael Miller, Sheila Reynolds, Ilya Shmulevich, Vesteinn Thorsson, Wei Zhang, Rehan Akbani, Bradley M Broom, Apurva M Hegde, Zhenlin Ju, Rupa S Kanchi, Anil Korkut, Jun Li, Han Liang, Shiyun Ling, Wenbin Liu, Yiling Lu, Gordon B Mills, Kwok-Shing Ng, Arvind Rao, Michael Ryan, Jing Wang, Jiexin Zhang, Adam Abeshouse, Joshua Armenia, Debyani Chakravarty, Walid K Chatila, Ino de Bruijn, Jianjiong Gao, Benjamin E Gross, Zachary J Heins, Ritika Kundra, Konnor La, Marc Ladanyi, Augustin Luna, Moriah G Nissan, Angelica Ochoa, Sarah M Phillips, Ed Reznik, Francisco Sanchez-Vega, Chris Sander, Nikolaus Schultz, Robert Sheridan, S Onur Sumer, Yichao Sun, Barry S Taylor, Jioajiao Wang, Hongxin Zhang, Pavana Anur, Myron Peto, Paul Spellman, Christopher Benz, Joshua M Stuart, Christopher K Wong, D Neil Hayes, Joel S Parker, Matthew D Wilkerson, Adrian Ally, Miruna Balasundaram, Reanne Bowlby, Denise Brooks, Rebecca Carlsen, Eric Chuah, Noreen Dhalla, Robert Holt, Steven J M Jones, Katayoon Kasaian, Darlene Lee, Yussanne Ma, Marco A Marra, Michael Mayo, Richard A Moore, Andrew J Mungall, Karen Mungall, A Gordon Robertson, Sara Sadeghi, Jacqueline E Schein, Payal Sipahimalani, Angela Tam, Nina Thiessen, Kane Tse, Tina Wong, Ashton C Berger, Rameen Beroukhim, Andrew D Cherniack, Carrie Cibulskis, Stacey B Gabriel, Galen F Gao, Gavin Ha, Matthew Meyerson, Steven E Schumacher, Juliann Shih, Melanie H Kucherlapati, Raju S Kucherlapati, Stephen Baylin, Leslie Cope, Ludmila Danilova, Moiz S Bootwalla, Phillip H Lai, Dennis T Maglinte, David J Van Den Berg, Daniel J Weisenberger, J Todd Auman, Saianand Balu, Tom Bodenheimer, Cheng Fan, Katherine A Hoadley, Alan P Hoyle, Stuart R Jefferys, Corbin D Jones, Shaowu Meng, Piotr A Mieczkowski, Lisle E Mose, Amy H Perou, Charles M Perou, Jeffrey Roach, Yan Shi, Janae V Simons, Tara Skelly, Matthew G Soloway, Donghui Tan, Umadevi Veluvolu, Huihui Fan, Toshinori Hinoue, Peter W Laird, Hui Shen, Wanding Zhou, Michelle Bellair, Kyle Chang, Kyle Covington, Chad J Creighton, Huyen Dinh, HarshaVardhan Doddapaneni, Lawrence A Donehower, Jennifer Drummond, Richard A Gibbs, Robert Glenn, Walker Hale, Yi Han, Jianhong Hu, Viktoriya Korchina, Sandra Lee, Lora Lewis, Wei Li, Xiuping Liu, Margaret Morgan, Donna Morton, Donna Muzny, Jireh Santibanez, Margi Sheth, Eve Shinbrot, Linghua Wang, Min Wang, Liu Xi, Fengmei Zhao, Julian Hess, Elizabeth L Appelbaum, Matthew Bailey, Matthew G Cordes, Li Ding, Catrina C Fronick, Lucinda A Fulton, Robert S Fulton, Cyriac Kandoth, Elaine R Mardis, Michael D McLellan, Christopher A Miller, Heather K Schmidt, Richard K Wilson, Daniel Crain, Erin Curley, Johanna Gardner, Kevin Lau, David Mallery, Scott Morris, Joseph Paulauskis, Robert Penny, Candace Shelton, Troy Shelton, Mark Sherman, Eric Thompson, Peggy Yena, Jay Bowen, Julie M Gastier-Foster, Mark Gerken, and ... Leraas. Genomic and Molecular Landscape of DNA Damage Repair Deficiency across The Cancer Genome Atlas. *Cell reports*, 23(1):239–254.e6, April 2018.

[116] Gavin C K W Koh, Pablo Porras, Bruno Aranda, Henning Hermjakob, and Sandra E Orchard. Analyzing Protein–Protein Interaction Networks. *Journal of Proteome Research*, 11(4):2014–2031, March 2012.

[117] Eric D Kolaczyk and Gábor Csárdi. *Statistical analysis of network data with R*. Use R! Springer, New York, NY, 2014.

[118] H W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, March 1955.

[119] J M Kumpula, J Saramäki, K Kaski, and J Kertész. Limited resolution in complex network community detection with Potts model approach. *The European Physical Journal B*, 56 (1):41–45, 2007.

[120] Jérôme Kunegis. *KONECT: the Koblenz network collection*. the Koblenz network collection. ACM, New York, New York, USA, May 2013.

[121] Darong Lai, Xin Shu, and Christine Nardini. Correlation enhanced modularity-based belief propagation method for community detection in networks. *Journal of Statistical Mechanics: Theory and Experiment*, 05(5):053301–, May 2016.

[122] Renaud Lambiotte, Jean-Charles Delvenne, and Mauricio Barahona. Laplacian dynamics and Multiscale Modular structure in Networks. *arxiv.org*, 1(2):76–90, October 2009.

[123] Andrea Lancichinetti and Santo Fortunato. Limits of modularity maximization in community detection. *Physical Review E*, 84(6):066122, December 2011.

[124] Andrea Lancichinetti and Santo Fortunato. Consensus clustering in complex networks. *Scientific reports*, 2(1):336 EP –, March 2012.

[125] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, October 2008.

[126] Michael S Lawrence, Petar Stojanov, Paz Polak, Gregory V Kryukov, Kristian Cibulskis, Andrey Sivachenko, Scott L Carter, Chip Stewart, Craig H Mermel, Steven A Roberts, Adam Kiezun, Peter S Hammerman, Aaron McKenna, Yotam Drier, Lihua Zou, Alex H Ramos, Trevor J Pugh, Nicolas Stransky, Elena Helman, Jaegil Kim, Carrie Sougnez, Lauren Ambrogio, Elizabeth Nickerson, Erica Shefler, Maria L Cortes, Daniel Auclair, Gordon Saksena, Douglas Voet, Michael Noble, Daniel DiCara, Pei Lin, Lee Lichtenstein, David I Heiman, Timothy Fennell, Marcin Imielinski, Bryan Hernandez, Eran Hodis, Sylvan Baca, Austin M Dulak, Jens Lohr, Dan-Avi Landau, Catherine J Wu, Jorge Melendez-Zajgla, Alfredo Hidalgo-Miranda, Amnon Koren, Steven A McCarroll, Jaume Mora, Ryan S Lee, Brian Crompton, Robert Onofrio, Melissa Parkin, Wendy Winckler, Kristin Ardlie, Stacey B Gabriel, Charles W M Roberts, Jaclyn A Biegel, Kimberly Stegmaier, Adam J Bass, Levi A Garraway, Matthew Meyerson, Todd R Golub, Dmitry A Gordenin, Shamil Sunyaev, Eric S Lander, and Gad Getz. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, 499(7457):214–8, July 2013.

[127] Emmanuel Lazega and others. *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand, 2001.

[128] Dung T Le, Jennifer N Uram, Hao Wang, Bjarne R Bartlett, Holly Kemberling, Aleksandra D Eyring, Andrew D Skora, Brandon S Luber, Nilofer S Azad, Dan Laheru, Barbara Biedrzycki, Ross C Donehower, Atif Zaheer, George A Fisher, Todd S Crocenzi, James J Lee, Steven M Duffy, Richard M Goldberg, Albert de la Chapelle, Minori Koshiji, Feriyl Bhaijee, Thomas Huebner, Ralph H Hruban, Laura D Wood, Nathan Cuka, Drew M Pardoll, Nickolas Papadopoulos, Kenneth W Kinzler, Shibin Zhou, Toby C Cornish, Janis M Taube, Robert A Anders, James R Eshleman, Bert Vogelstein, and Luis A Diaz. PD-1 Blockade in Tumors with Mismatch-Repair Deficiency. *New England Journal of Medicine*, 372(26):2509–2520, May 2015.

[129] Dung T Le, Jennifer N Durham, Kellie N Smith, Hao Wang, Bjarne R Bartlett, Laveet K Aulakh, Steve Lu, Holly Kemberling, Cara Wilt, Brandon S Luber, Fay Wong, Nilofer S Azad, Agnieszka A Rucki, Dan Laheru, Ross Donehower, Atif Zaheer, George A Fisher, Todd S Crocenzi, James J Lee, Tim F Greten, Austin G Duffy, Kristen K Ciombor, Aleksandra D Eyring, Bao H Lam, Andrew Joe, S Peter Kang, Matthias Holdhoff, Ludmila Danilova, Leslie Cope, Christian Meyer, Shibin Zhou, Richard M Goldberg, Deborah K Armstrong, Katherine M Bever, Amanda N Fader, Janis Taube, Franck Housseau, David Spetzler, Nianqing Xiao, Drew M Pardoll, Nickolas Papadopoulos, Kenneth W Kinzler, James R Eshleman, Bert Vogelstein, Robert A Anders, and Luis A Diaz. Mismatch repair deficiency predicts response of solid tumors to PD-1 blockade. *Science*, 357(6349): 409–413, July 2017.

[130] D D Lee and H S Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[131] Eun Lee, Scott Emmons, Ryan Gibson, James Moody, and Peter J Mucha. Concurrency and reachability in tree-like temporal networks. *arXiv preprint arXiv:1905.08580*, 462 (7276):1044–1047, 2019.

[132] E A Leicht and M E J Newman. Community Structure in Directed Networks. *Physical Review Letters*, 100(11):118703, March 2008.

[133] Mark D M Leiserson, Fabio Vandin, Hsin-Ta Wu, Jason R Dobson, Jonathan V Eldridge, Jacob L Thomas, Alexandra Papoutsaki, Younhun Kim, Beifang Niu, Michael McLellan, Michael S Lawrence, Abel Gonzalez-Perez, David Tamborero, Yuwei Cheng, Gregory A Ryslik, Nuria Lopez-Bigas, Gad Getz, Li Ding, and Benjamin J Raphael. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nature genetics*, 47(2):106–114, February 2015.

[134] Steven Lemery, Patricia Keegan, and Richard Pazdur. First FDA Approval Agnostic of Cancer Site - When a Biomarker Defines the Indication. *The New England journal of medicine*, 377(15):1409–1412, October 2017.

[135] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker Graphs: An Approach to Modeling Networks. *Journal of Machine Learning Research*, 11(Feb):985–1042, 2010.

[136] Anna CF Lewis, Nick S Jones, Mason A Porter, and Charlotte M Deane. The function of communities in protein interaction networks at multiple scales. *BMC Systems Biology*, 4 (1):100, 2010.

[137] Jimmy Lin, Christine M Gan, Xiaosong Zhang, Siân Jones, Tobias Sjöblom, Laura D Wood, D Williams Parsons, Nickolas Papadopoulos, Kenneth W Kinzler, Bert Vogelstein, Giovanni Parmigiani, and Victor E Velculescu. A multidimensional analysis of genes mutated in breast and colorectal cancers. *Genome research*, 17(9):1304–1318, September 2007.

[138] Jian Liu. Fuzzy modularity and fuzzy community structure in networks. 77(4):547–557, October 2010.

[139] Yiyi Liu, Quanquan Gu, Jack P Hou, Jiawei Han, and Jian Ma. A network-assisted co-clustering algorithm to discover cancer subtypes based on gene expression. *BMC bioinformatics*, 15:37, February 2014.

[140] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*, 2015, September 2015.

[141] Keith R Loeb and Lawrence A Loeb. Significance of multiple mutations in cancer. *Carcinogenesis*, 21(3):379–385, March 2000.

[142] Kevin T Macon, Peter J Mucha, and Mason A Porter. Community structure in the United Nations General Assembly. *Physica A: Statistical Mechanics and its Applications*, 391(1): 343–361, 2012.

[143] Christopher D Manning, Prabhakar Raghavan, and Hinrich Sch u tze. *Introduction to Information Retrieval*. Cambridge University Press, USA, 2008.

[144] Sanjeev Mariathasan, Shannon J Turley, Dorothee Nickles, Alessandra Castiglioni, Kobe Yuen, Yulei Wang, Edward E Kadel Iii, Hartmut Koeppen, Jillian L Astarita, Rafael Cubas, Suchit Jhunjhunwala, Romain Banchereau, Yagai Yang, Yinghui Guan, Cecile Chalouni, James Ziai, Yasin Şenbabaoğlu, Stephen Santoro, Daniel Sheinson, Jeffrey Hung, Jennifer M Giltnane, Andrew A Pierce, Kathryn Mesh, Steve Lianoglou, Johannes Riegler, Richard A D Carano, Pontus Eriksson, Mattias Höglund, Loan Somarriba, Daniel L Halligan, Michiel S van der Heijden, Yohann Loriot, Jonathan E Rosenberg, Lawrence Fong, Ira Mellman, Daniel S Chen, Marjorie Green, Christina Derleth, Gregg D Fine, Priti S Hegde, Richard Bourgon, and Thomas Powles. TGFβ attenuates tumour response to PD-L1 blockade by contributing to exclusion of T cells. *Nature Publishing Group*, 515 (7693):563–548, February 2018.

[145] Janice M Mehnert, Anshuman Panda, Hua Zhong, Kim Hirshfield, Sherri Damare, Katherine Lane, Levi Sokol, Mark N Stein, Lorna Rodriguez-Rodriquez, Howard L Kaufman, Siraj Ali, Jeffrey S Ross, Dean C Pavlick, Gyan Bhanot, Eileen P White, Robert S DiPaola, Ann Lovell, Jonathan Cheng, and Shridar Ganesan. Immune activation and response to pembrolizumab in POLE-mutant endometrial cancer. *Journal of Clinical Investigation*, 126(6):2334–2340, June 2016.

[146] Marina Meilă. Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 98(5):873–895, 2007.

[147] Marc Mézard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

[148] Diana Miao, Claire A Margolis, Wenhua Gao, Martin H Voss, Wei Li, Dylan J Martini, Craig Norton, Dominick Bossé, Stephanie M Wankowicz, Dana Cullen, Christine Horak, Megan Wind-Rotolo, Adam Tracy, Marios Giannakis, Frank Stephen Hodi, Charles G Drake, Mark W Ball, Mohamad E Allaf, Alexandra Snyder, Matthew D Hellmann, Thai Ho, Robert J Motzer, Sabina Signoretti, William G Kaelin Jr., Toni K Choueiri, and Eliezer M Van Allen. Genomic correlates of response to immune checkpoint therapies in clear cell renal cell carcinoma. *Science*, pages eaan5951–11, January 2018.

[149] Vincent Miele, Catherine Matias, Stéphane Robin, and Stéphane Dray. Nine quick tips for analyzing network data. *PLoS Computational Biology*, 15(12):e1007434 EP –, December 2019.

[150] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[151] James Clyde Mitchell. *Social networks in urban situations: analyses of personal relationships in Central African towns*. Manchester University Press, 1969.

[152] James Moody and Peter J Mucha. Portrait of political party polarization. *Network Science*, 1(1):119–121, 2013.

[153] Elchanan Mossel, Joe Neeman, and Allan Sly. Belief propagation, robust reconstruction and optimal recovery of block models. In *International Conference on Machine Learning*, pages 356–370. May 2014.

[154] Kent W Mouw, Michael S Goldberg, Panagiotis A Konstantinopoulos, and Alan D D'Andrea. DNA Damage and Repair Biomarkers of Immunotherapy Response. *Cancer discovery*, 7(7):675–693, July 2017.

[155] Peter J Mucha and Mason A Porter. Communities in multislice voting networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(4):041108, December 2010.

[156] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, May 2010.

[157] Stefanie Muff, Francesco Rao, and Amedeo Caflisch. Local modularity measure for network clusterizations. *Physical Review E*, 72(5):251, November 2005.

[158] Heiko Müller and Francesco Mancuso. Identification and analysis of co-occurrence networks with NetCutter. *PLOS ONE*, 3(9):e3178, September 2008.

[159] Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *arXiv preprint arXiv:1301.6725*, 2013.

[160] Raj Rao Nadakuditi and M E J Newman. Graph Spectra and the Detectability of Community Structure in Networks. *Physical Review Letters*, 108(18):188701, May 2012.

[161] Andrea Necchi, Andrea Anichini, Daniele Raggi, Alberto Briganti, Simona Massa, Roberta Lucianò, Maurizio Colecchia, Patrizia Giannatempo, Roberta Mortarini, Marco Bianchi, Elena Farè, Francesco Monopoli, Renzo Colombo, Andrea Gallina, Andrea Salonia, Antonella Messina, Siraj M Ali, Russell Madison, Jeffrey S Ross, Jon H Chung, Roberto Salvioni, Luigi Mariani, and Francesco Montorsi. Pembrolizumab as neoadjuvant therapy

before radical cystectomy in patients with muscle-invasive urothelial bladder carcinoma (PURE-01): An open-label, single-arm, phase II study. *Journal of Clinical Oncology*, 36 (34):3353–3360, December 2018.

[162] M E J Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, February 2003.

[163] M E J Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 74(3 Pt 2): 036104, September 2006.

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

[165] M E J Newman. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E*, 94(5):052315, November 2016.

[166] M E J Newman and Aaron Clauset. Structure and inference in annotated networks. *Nature Communications*, 7(1):11863–11, 2016.

[167] Mark Newman. *Networks: an introduction*. Oxford Univ. Press, New York, NY, 2010.

[168] MEJ Newman and M Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):268, 2004.

[169] Heidi K Norton. Detecting hierarchical genome folding with network modularity. *Nature Methods*, 15(2):119–122, January 2018.

[170] Rose Oughtred, Chris Stark, Bobby-Joe Breitkreutz, Jennifer Rust, Lorrie Boucher, Christie Chang, Nadine Kolas, Lara O'Donnell, Genie Leung, Rochelle McAdam, Frederick Zhang, Sonam Dolma, Andrew Willems, Jasmin Coulombe-Huntington, Andrew Chatr-aryamontri, Kara Dolinski, and Mike Tyers. The BioGRID interaction database: 2019 update. *Nucleic acids research*, 47(D1):D529–D541, November 2018.

[171] Michael Ovelgönne and Andreas Geyer-Schulz. An ensemble learning strategy for graph clustering. *Graph Partitioning and Graph Clustering*, 588:187, 2012.

[172] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. November 1999.

[173] Siddharth Pal, Feng Yu, Yitzchak Novick, Ananthram Swami, and Amotz Bar-Noy. A study on the friendship paradox – quantitative analysis and relationship with assortative mixing. *Applied Network Science*, 4(1):1–26, December 2019.

[174] A Roxana Pamfil, Sam D Howison, Renaud Lambiotte, and Mason A Porter. Relating Modularity Maximization and Stochastic Block Models in Multilayer Networks. *SIAM Journal on Mathematics of Data Science*, 1(4):667–698, 2019.

[175] Alexandre Papadopoulos, François Pachet, Pierre Roy, and Jason Sakellariou. Exact Sampling for Regular and Markov Constraints with Belief Propagation. In *Research in Computational Molecular Biology*, pages 341–350. Springer International Publishing, Cham, August 2015.

[176] Lia Papadopoulos, James G Puckett, Karen E Daniels, and Danielle S Bassett. Evolution of network architecture in a granular material under compression. *Physical Review E*, 94(3): 164, September 2016.

[177] Joel S Parker, Michael Mullins, Maggie C U Cheang, Samuel Leung, David Voduc, Tammi Vickery, Sherri Davies, Christiane Fauron, Xiaping He, Zhiyuan Hu, John F Quackenbush, Inge J Stijleman, Juan Palazzo, J S Marron, Andrew B Nobel, Elaine Mardis, Torsten O Nielsen, Matthew J Ellis, Charles M Perou, and Philip S Bernard. Supervised Risk Predictor of Breast Cancer Based on Intrinsic Subtypes. *Journal of Clinical Oncology*, 27 (8):1160–1167, February 2009.

[178] Judea Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. 1988.

[179] Leto Peel, Daniel B Larremore, and Aaron Clauset. The ground truth about metadata and community detection in networks. *Science Advances*, 3(5):e1602548, 2017.

[180] Tiago P Peixoto. Eigenvalue Spectra of Modular Networks. *Physical Review Letters*, 111 (9):098701, August 2013.

[181] Tiago P Peixoto. The graph-tool python library. *figshare*, 2014.

[182] Tiago P. Peixoto. The graph-tool python library, 2014. http://figshare.com/articles/graph_tool/1164194.

[183] Tiago P Peixoto. Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Physical Review E*, 92(4):042807, October 2015.

[184] Tiago P Peixoto. Nonparametric Bayesian inference of the microcanonical stochastic block model. *Physical Review E*, 95(1):012317, January 2017.

[185] Charles M Perou, Therese Sorlie, Michael B Eisen, Matt van de Rijn, and et al. Molecular portraits of human breast tumours. *Nature*, 406(6797):747–52, August 2000.

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

[187] Thomas Powles, Joseph Paul Eder, Gregg D Fine, Fadi S Braiteh, Yohann Loriot, Cristina Cruz, Joaquim Bellmunt, Howard A Burris, Daniel P Petrylak, Siew-leng Teng, Xiaodong Shen, Zachary Boyd, Priti S Hegde, Daniel S Chen, and Nicholas J Vogelzang. MPDL3280A (anti-PD-L1) treatment leads to clinical activity in metastatic bladder cancer. *Nature*, 515(7528):558–562, November 2014.

[188] Filippo Radicchi. Detectability of communities in heterogeneous networks. *Physical Review E*, 88(1):010801, July 2013.

[189] William M Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the american statistical association*, 66(336):846–850, 1971.

[190] Benjamin J Raphael, Jason R Dobson, Layla Oesper, and Fabio Vandin. Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine. *Genome medicine*, 6(1):5, 2014.

[191] J Reichardt and S Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 2006.

[192] J Reichardt and S Bornholdt. When are networks truly modular? *Physica D: Nonlinear Phenomena*, 2006.

[193] Sungmin Rhee, Seokjun Seo, and Sun Kim. Hybrid Approach of Relation Network and Localized Graph Convolutional Filtering for Breast Cancer Subtype Classification. *arXiv.org*, June 2018.

[194] Thomas Richardson, Peter J Mucha, and Mason A Porter. Spectral tripartitioning of networks. *Physical Review E*, 80(3):036111, September 2009.

[195] Naiyer A Rizvi, Matthew D Hellmann, Alexandra Snyder, Pia Kvistborg, Vladimir Makarov, Jonathan J Havel, William Lee, Jianda Yuan, Phillip Wong, Teresa S Ho, Martin L Miller, Natasha Rekhtman, Andre L Moreira, Fawzia Ibrahim, Cameron Bruggeman, Billel Gasmi, Roberta Zappasodi, Yuka Maeda, Chris Sander, Edward B Garon, Taha Merghoub, Jedd D Wolchok, Ton N Schumacher, and Timothy A Chan. Cancer immunology. Mutational landscape determines sensitivity to PD-1 blockade in non-small cell lung cancer. *Science*, 348(6230):124–128, April 2015.

[196] M Puck Rombach, Mason A Porter, James H Fowler, and Peter J Mucha. Core-Periphery Structure in Networks. *SIAM J. Appl. Math.*, 74(1):167–190, January 2014.

[197] Jonathan E Rosenberg, Jean Hoffman-Censits, Tom Powles, Michiel S van der Heijden, Arjun V Balar, Andrea Necchi, Nancy Dawson, Peter H O'Donnell, Ani Balmanoukian, Yohann Loriot, Sandy Srinivas, Margitta M Retz, Petros Grivas, Richard W Joseph, Matthew D Galsky, Mark T Fleming, Daniel P Petrylak, Jose Luis Perez-Gracia, Howard A Burris, Daniel Castellano, Christina Canil, Joaquim Bellmunt, Dean Bajorin, Dorothee Nickles, Richard Bourgon, Garrett M Frampton, Na Cui, Sanjeev Mariathasan, Oyewale Abidoye, Gregg D Fine, and Robert Dreicer. Atezolizumab in patients with locally advanced and metastatic urothelial carcinoma who have progressed following treatment with platinum-based chemotherapy: a single-arm, multicentre, phase 2 trial. *The Lancet*, 387(10031):1909–1920, May 2016.

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

[199] Martin Rosvall and Carl T Bergstrom. Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems. *PLOS ONE*, 6 (4):e18209, April 2011.

[200] Jianhua Ruan and Weixiong Zhang. Identifying network communities with a high resolution. *Physical Review E*, 77(1):016104, January 2008.

[201] Robert M Samstein, Chung-Han Lee, Alexander N Shoushtari, Matthew D Hellmann, Ronglai Shen, Yelena Y Janjigian, David A Barron, Ahmet Zehir, Emmet J Jordan, Antonio Omuro, Thomas J Kaley, Sviatoslav M Kendall, Robert J Motzer, A Ari Hakimi, Martin H Voss, Paul Russo, Jonathan Rosenberg, Gopa Iyer, Bernard H Bochner, Dean F Bajorin, Hikmat A Al-Ahmadie, Jamie E Chaft, Charles M Rudin, Gregory J Riely, Shrujal Baxi,

Alan L Ho, Richard J Wong, David G Pfister, Jedd D Wolchok, Christopher A Barker, Philip H Gutin, Cameron W Brennan, Viviane Tabar, Ingo K Mellinghoff, Lisa M DeAngelis, Charlotte E Ariyan, Nancy Lee, William D Tap, Mrinal M Gounder, Sandra P D'Angelo, Leonard Saltz, Zsofia K Stadler, Howard I Scher, Jose Baselga, Pedram Razavi, Christopher A Klebanoff, Rona Yaeger, Neil H Segal, Geoffrey Y Ku, Ronald P DeMatteo, Marc Ladanyi, Naiyer A Rizvi, Michael F Berger, Nadeem Riaz, David B Solit, Timothy A Chan, and Luc G T Morris. Tumor mutational load predicts survival after immunotherapy across multiple cancer types. *Nature genetics*, 51(2):202–206, 2019.

[202] Yardena Samuels, Zhenghe Wang, Alberto Bardelli, Natalie Silliman, Janine Ptak, Steve Szabo, Hai Yan, Adi Gazdar, Steven M Powell, Gregory J Riggins, James K V Willson, Sanford Markowitz, Kenneth W Kinzler, Bert Vogelstein, and Victor E Velculescu. High Frequency of Mutations of the PIK3CA Gene in Human Cancers. *Science*, 304(5670): 554–554, 2004.

[203] Fabio Saracco, Riccardo Di Clemente, Andrea Gabrielli, and Tiziano Squartini. Randomizing bipartite networks: the case of the World Trade Web. *Scientific reports*, 5(1): 10595–18, 2015.

[204] Somwrita Sarkar, James A Henderson, and Peter A Robinson. Spectral Characterization of Hierarchical Network Modularity and Limits of Modularity Detection. *PLOS ONE*, 8(1), January 2013.

[205] A B Schrock, D Fabrizio, Y He, J Chung, M Resnick, P J Stephens, J S Ross, V A Miller, S Ramkissoon, J A Elvin, S M Ali, M Fakih, and S J Klempner. 1170PAnalysis of POLE mutation and tumor mutational burden (TMB) across 80,853 tumors: Implications for immune checkpoint inhibitors (ICPIs). *Annals of Oncology*, 28(suppl - 5), September 2017.

[206] Christophe Schülke and Federico Ricci-Tersenghi. Multiple phases in modularity-based community detection. *Physical Review E*, 92(4):042804, October 2015.

[207] Saray Shai, Natalie Stanley, Clara Granell, Dane Taylor, and Peter J Mucha. Case studies in network community detection. *arXiv*, May 2017.

[208] Cheng Shi, Yanchen Liu, and Pan Zhang. Weighted community detection and data clustering using message passing. *Journal of Statistical Mechanics: Theory and Experiment*, 2018(3):033405, March 2018.

[209] E H Simpson. The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(2):238–241, July 1951.

[210] Alexandra Snyder, Vladimir Makarov, Taha Merghoub, Jianda Yuan, Jesse M Zaretsky, Alexis Desrichard, Logan A Walsh, Michael A Postow, Phillip Wong, Teresa S Ho, Travis J Hollmann, Cameron Bruggeman, Kasthuri Kannan, Yanyun Li, Ceyhan Elipenahli, Cailian Liu, Christopher T Harbison, Lisu Wang, Antoni Ribas, Jedd D Wolchok, and Timothy A Chan. Genetic basis for clinical response to CTLA-4 blockade in melanoma. *The New England journal of medicine*, 371(23):2189–2199, December 2014.

[211] Therese Sorlie, Robert Tibshirani, Joel Parker, Trevor Hastie, J S Marron, Andrew Nobel, Shibing Deng, Hilde Johnsen, Robert Pesich, Stephanie Geisler, Janos Demeter, Charles M

Perou, Per E Lønning, Patrick O Brown, Anne-Lise Børresen-Dale, and David Botstein. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the National Academy of Sciences*, 100(14):8418–8423, July 2003.

[212] Natalie Stanley, Saray Shai, Dane Taylor, and Peter J Mucha. Clustering network layers with the strata multilayer stochastic block model. *IEEE transactions on network science and engineering*, 3(2):95–105, April 2016.

[213] Natalie Stanley, Thomas Bonacci, Roland Kwitt, Marc Niethammer, and Peter J Mucha. Stochastic block models with multiple continuous attributes. *Applied Network Science*, 4 (1):54–22, 2019.

[214] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, and Jill P Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci USA*, 102(43): 15545–15550, October 2005.

[215] T Sugimura, M Terada, J Yokota, S Hirohashi, and K Wakabayashi. Multiple genetic alterations in human carcinogenesis. *Environmental Health Perspectives*, 98:5–12, November 1992.

[216] Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P Tsafou, Michael Kuhn, Peer Bork, Lars J Jensen, and Christian von Mering. STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research*, 43(Database issue):D447–D452, January 2015.

[217] David Tamborero, Abel Gonzalez-Perez, Christian Perez-Llamas, Jordi Deu-Pons, Cyriac Kandoth, Jüri Reimand, Michael S Lawrence, Gad Getz, Gary D Bader, Li Ding, and Nuria Lopez-Bigas. Comprehensive identification of mutational cancer driver genes across 12 tumor types. *Scientific reports*, 3(1):2650 EP –, October 2013.

[218] Dane Taylor, Saray Shai, Natalie Stanley, and Peter J Mucha. Enhanced Detectability of Community Structure in Multilayer Networks through Layer Aggregation. *Physical Review Letters*, 116(22):228301, June 2016.

[219] Min Yuen Teo, Kenneth Seier, Irina Ostrovnaya, Ashley M Regazzi, Brooke E Kania, Meredith M Moran, Catharine K Cipolla, Mark J Bluth, Joshua Chaim, Hikmat Al-Ahmadie, Alexandra Snyder, Maria I Carlo, David B Solit, Michael F Berger, Samuel Funt, Jedd D Wolchok, Gopa Iyer, Dean F Bajorin, Margaret K Callahan, and Jonathan E Rosenberg. Alterations in DNA Damage Response and Repair Genes as Potential Marker of Clinical Benefit From PD-1/PD-L1 Blockade in Advanced Urothelial Cancers. *Journal of Clinical Oncology*, pages JCO.2017.75.774–11, February 2018.

[220] Ali Torkamani and Nicholas J Schork. Identification of rare cancer driver mutations by network reconstruction. *Genome research*, 19(9):1570–1578, September 2009.

[221] V A Traag and Jeroen Bruggeman. Community detection in networks with positive and negative links. *Physical Review E*, 80(3):036115, September 2009.

[222] V A Traag and Jeroen Bruggeman. Community detection in networks with positive and negative links. *Physical Review E*, 80(3):036115, September 2009.

[223] V A Traag, P Van Dooren, and Y Nesterov. Narrow scope for resolution-limit-free community detection. *Physical Review E*, 84(1):016114, July 2011.

[224] V A Traag, G Krings, and P Van Dooren. Significant scales in community structure. *Scientific reports*, 3(1):75, 2013.

[225] V A Traag, L Waltman, and N J van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

[226] Vincent Traag. Louvain igraph. http://github.com/vtraag/louvain-igraph.

[227] Amanda L Traud, Eric D Kelsic, Peter J Mucha, and Mason A Porter. Comparing Community Structure to Characteristics in Online Collegiate Social Networks. *SIAM Review*, 53(3):526–543, January 2011.

[228] Justin G Trogdon, William H. Weir, S Shai, Peter J Mucha, T M Kuo, A M Meyer, and K B Stitzenberg. Comparing Shared Patient Networks Across Payers. *Journal of General Internal Medicine*, 359:1200–7, April 2019.

[229] Toni Vallès-Català, Francesco A Massucci, Roger Guimerà, and Marta Sales-Pardo. Multilayer Stochastic Block Models Reveal the Multilayer Structure of Complex Networks. *Physical Review X*, 6(1):011036, March 2016.

[230] Eliezer M Van Allen, Diana Miao, Bastian Schilling, Sachet A Shukla, Christian Blank, Lisa Zimmer, Antje Sucker, Uwe Hillen, Marnix H Geukes Foppen, Simone M Goldinger, Jochen Utikal, Jessica C Hassel, Benjamin Weide, Katharina C Kaehler, Carmen Loquai, Peter Mohr, Ralf Gutzmer, Reinhard Dummer, Stacey Gabriel, Catherine J Wu, Dirk Schadendorf, and Levi A Garraway. Genomic correlates of response to CTLA-4 blockade in metastatic melanoma. *Science*, 350(6257):207–211, October 2015.

[231] Fabio Vandin, PATRICK CLAY, Eli Upfal, and Benjamin J Raphael. Discovery of mutated subnetworks associated with clinical data in cancer. *Pacific Symposium on Biocomputing*. *Pacific Symposium on Biocomputing*, pages 55–66, 2012.

[232] O Vanunu, O Magger, E Ruppin, and T Shlomi. Associating genes and protein complexes with disease via network propagation. *PLoS Comput ...*, 2010.

[233] Roel G W Verhaak, Katherine A Hoadley, Elizabeth Purdom, Victoria Wang, Yuan Qi, Matthew D Wilkerson, C Ryan Miller, Li Ding, Todd Golub, Jill P Mesirov, Gabriele Alexe, Michael Lawrence, Michael O'Kelly, Pablo Tamayo, Barbara A Weir, Stacey Gabriel, Wendy Winckler, Supriya Gupta, Lakshmi Jakkula, Heidi S Feiler, J Graeme Hodgson, C David James, Jann N Sarkaria, Cameron Brennan, Ari Kahn, Paul T Spellman, Richard K Wilson, Terence P Speed, Joe W Gray, Matthew Meyerson, Gad Getz, Charles M Perou, and D Neil Hayes. Integrated Genomic Analysis Identifies Clinically Relevant Subtypes of Glioblastoma Characterized by Abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer cell*, 17(1):98–110, January 2010.

[234] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? *the 26th Annual International Conference*, pages 1073–1080, 2009.

[235] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[236] Bert Vogelstein, Nickolas Papadopoulos, Victor E Velculescu, Shibin Zhou, Luis A Diaz, and Kenneth W Kinzler. Cancer Genome Landscapes. *Science*, 339(6127):1546–1558, 2013.

[237] Benjamin Walker and William H. Weir. Modbp package: multilayer modularity belief propagation in python, 2019-2020. https://github.com/bwalker1/ModularityBP_Cpp.

[238] Chunyu Wang, Junling Guo, Ning Zhao, Yang Liu, Xiaoyan Liu, Guojun Liu, and Maozu Guo. A Cancer Survival Prediction Method Based on Graph Convolutional Network. *IEEE transactions on nanobioscience*, 19(1):117–126, 2019.

[239] Haiying Wang, Huiru Zheng, Jianxin Wang, Chaoyang Wang, and Fang-Xiang Wu. Integrating omics data with a multiplex network-based approach for the identification of cancer subtypes. *IEEE transactions on nanobioscience*, 15(4):335–342, 2016.

[240] Zhijie Wang, Jing Zhao, Guoqiang Wang, Zemin Zhang, Fan Zhang, Yuzi Zhang, Hua Dong, Xiaochen Zhao, Jianchun Duan, Hua Bai, Yanhua Tian, Rui Wan, Miao Han, Yan Cao, Lei Xiong, Li Liu, Shuhang Wang, Shangli Cai, Tony S K Mok, and Jie Wang. Comutations in DNA Damage Response Pathways Serve as Potential Biomarkers for Immune Checkpoint Blockade. *Cancer research*, 78(22):6486–6496, November 2018.

[241] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature Publishing Group*, 393(6684):440–442, 1998.

[242] Andrew Scott Waugh, Liuyi Pei, James H Fowler, Peter J Mucha, and Mason Alexander Porter. Party polarization in congress: A network science approach. *arXiv.org*, physics.soc-ph, 2009.

[243] Nils Weinhold, Anders Jacobsen, Nikolaus Schultz, Chris Sander, and William Lee. Genome-wide analysis of noncoding regulatory mutations in cancer. *Nature genetics*, 46 (11):1160–1165, 2014.

[244] William H. Weir, Scott Emmons, Ryan Gibson, Dane Taylor, and Peter Mucha. Post-Processing Partitions to Identify Domains of Modularity Optimization. *Algorithms*, 10(3):93, September 2017.

[245] William H. Weir, Ryan Gibson, and Peter J Mucha. CHAMP package: Convex Hull of Admissible Modularity Partitions in Python and MATLAB, 2017. URL https://github.com/wweir827/CHAMP.

[246] Michael C Wendl, John W Wallis, Ling Lin, Cyriac Kandoth, Elaine R Mardis, Richard K Wilson, and Li Ding. PathScan: a tool for discerning mutational significance in groups of putative cancer genes. *Bioinformatics*, 27(12):1595–1602, April 2011.

[247] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Virality Prediction and Community Structure in Social Networks. *Scientific reports*, 3(1):2522, 2013.

[248] David S Wishart, Yannick Djoumbou Feunang, Ana Marcu, An Chi Guo, Kevin Liang, Rosa Vázquez-Fresno, Tanvir Sajed, Daniel Johnson, Carin Li, Naama Karu, Zinat Sayeeda, Elvis Lo, Nazanin Assempour, Mark Berjanskii, Sandeep Singhal, David Arndt, Yonjie Liang, Hasan Badran, Jason Grant, Arnau Serra-Cayuela, Yifeng Liu, Rupa Mandal, Vanessa Neveu, Allison Pon, Craig Knox, Michael Wilson, Claudine Manach, and Augustin Scalbert. HMDB 4.0: the human metabolome database for 2018. *Nucleic acids research*, 46(D1):D608–D617, November 2017.

[249] Jedd D Wolchok, Harriet Kluger, Margaret K Callahan, Michael A Postow, Naiyer A Rizvi, Alexander M Lesokhin, Neil H Segal, Charlotte E Ariyan, Ruth-Ann Gordon, Kathleen Reed, Matthew M Burke, Anne Caldwell, Stephanie A Kronenberg, Blessing U Agunwamba, Xiaoling Zhang, Israel Lowy, Hector David Inzunza, William Feely, Christine E Horak, Quan Hong, Alan J Korman, Jon M Wigginton, Ashok Gupta, and Mario Sznol. Nivolumab plus Ipilimumab in Advanced Melanoma. *New England Journal of Medicine*, 369(2):122–133, June 2013.

[250] Jianing Xi, Minghui Wang, and Ao Li. Discovering mutated driver genes through a robust and sparse co-regularized matrix factorization framework with prior information from mRNA expression patterns and interaction network. *BMC bioinformatics*, 19(1):214–14, 2018.

[251] Zi Yang and George Michailidis. A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics*, 32(1):1–8, September 2015.

[252] Mark Yarchoan, Alexander Hopkins, and Elizabeth M Jaffee. Tumor Mutational Burden and Response Rate to PD-1 Inhibition. *New England Journal of Medicine*, 377(25): 2500–2501, December 2017.

[253] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium*, pages 236–239. 2003.

[254] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Information Theory*, 51(7):2282–2312, 2005.

[255] Wayne W Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4):452–473, December 1977.

[256] Lenka Zdeborová and Florent Krzakala. Statistical physics of inference: thresholds and algorithms. *Advances in Physics*, 65(5):453–552, September 2016.

[257] Jiongmin Zhang, Ke Jia, Jinmeng Jia, and Ying Qian. An improved approach to infer protein-protein interaction based on a hierarchical vector space model. *BMC bioinformatics*, 19(1):161–14, 2018.

[258] Junhua Zhang, Ling-Yun Wu, Xiang-Sun Zhang, and Shihua Zhang. Discovery of co-occurring driver pathways in cancer. *BMC bioinformatics*, 15(1):271, August 2014.

[259] Pan Zhang and Cristopher Moore. Scalable detection of statistically significant communities and hierarchies, using message passing for modularity. *Proceedings of the National Academy of Sciences*, 111(51):18144–18149, 2014.

[260] Pan Zhang, Florent Krzakala, Jörg Reichardt, and Lenka Zdeborová. Comparative study for inference of hidden classes in stochastic block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(12):P12021, December 2012.

[261] Pan Zhang, Florent Krzakala, Jörg Reichardt, and Lenka Zdeborová. Comparative study for inference of hidden classes in stochastic block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(12):P12021, 2012.

[262] Qiangfeng Cliff Zhang, Donald Petrey, Jose Ignacio Garzon, Lei Deng, and Barry Honig. PrePPI: a structure-informed database of protein–protein interactions. *Nucleic acids research*, 41(D1):D828–D833, November 2012.

[263] Shihua Zhang, Rui-Sheng Wang, and Xiang-Sun Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490, 2007.

[264] Wei Zhang, Jeremy Chien, Jeongsik Yong, and Rui Kuang. Network-based machine learning and graph theory algorithms for precision oncology. *npj Precision Oncology*, 1(1): 25–15, 2017.

[265] Jing Zhao, Sang Hoon Lee, Mikael Huss, and Petter Holme. The Network Organization of Cancer-associated Protein Complexes in Human Tissues. *Scientific reports*, 3(1):1583–12, 2013.

[266] Wu Zhuang, Junxun Ma, Xudong Chen, Guoqiang Wang, Jing Lu, Yanan Chen, Hua Dong, Shangli Cai, Yuzi Zhang, Xiaochen Zhao, Youcai Zhu, Chunwei Xu, Yunjian Huang, Zhangzhou Huang, Xiaofeng Zhu, Hong Jiang, and Zhijie Wang. The Tumor Mutational Burden of Chinese Advanced Cancer Patients Estimated by a 381-cancer-gene Panel. *Journal of Cancer*, 9(13):2302–2307, June 2018.