On Biological Network Visualization: Understanding Challenges, Measuring the Status Quo, and Estimating Saliency of Visual Attributes

Nikhil Gopal

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2017

Reading Committee:

John H. Gennari, Co-chair

Neil F. Abernethy, Co-chair

Jeffrey Heer

Program Authorized to Offer Degree:

Biomedical and Health Informatics

University of Washington

Abstract

On Biological Network Visualization: Understanding Challenges, Measuring the Status Quo, and Estimating Saliency of Visual Attributes

Nikhil Gopal

Co-chairs of Supervisory Committee:
John H. Gennari & Neil F. Abernethy
Department of Biomedical Informatics and Medical Education

Biomedical research increasingly relies on the analysis and visualization of a wide range of collected data. However, for certain research questions, such as those investigating the interconnectedness of biological elements, the sheer quantity and variety of data results in rather uninterpretable—this is especially true for network visualization, as a large and dense biological network is often compared to spaghetti or a hairball. The contents of this dissertation detail three major studies and a number of associated analysis studies that extend those studies. First, the challenges faced by researchers who analyze and visualize biological networks are elucidated, followed by a systematic review that analyzes and characterizes network figures from peer-reviewed bioinformatics literature. The systematic review dataset is further supplemented with an analysis of task completability, and the combination of the two are analyzed via Random Forest to provide insight into the varying importance of visual encodings in context of graph-based tasks. Next, a small theoretical framework that is valuable for framing network visualization research questions is detailed, followed by a description of visual encoding exploration software built on the framework. The final study included in this dissertation details the design and execution of a task-based perception study, where several visual encodings are estimated as functions of the measured task. Through these studies, I contribute to the understanding of network-related visualization challenges encountered by researchers, a measure of the status quo of network visualization, a conceptualization of a method to usefully frame research questions related to network visualization, visual encoding software that affords systematic and reproducible explorations of the visual encoding set space, and finally a set of functional estimates describing how numerous visual encodings are a related to one's ability to visually scan a network.

Acknowledgements

There are a great many people I would like to thank, for both directly and indirectly, knowingly or perhaps unknowingly, helping with this dissertation...

A Brief Reflection on my Personal Journey

When I entered the program in 2012, I arrived in Seattle on the heels of what I considered to be a very successful corporate career. I entered with what I considered to be a great number of strengths, and only a handful of weaknesses, which of course could be remedied through experience and hard work under the right mentors. However, over time, I learned that much of my own success might have been largely circumstantial. Admittedly, this thought was not an easy one to entertain, as I once prided myself greatly on what I viewed as high enthusiasm, vetted ethos, seasoned research skills, a keen eye, and a critical mind. Although individuals who had characteristics of equal caliber, if not greater may have always surrounded me, I unfortunately may have been too egotistical or self-indulgent to notice. In fact, I remember a time when I would reflect on my own work, the state of things in my research, and realizing that what I had initially considered to be my greatest strengths prior to entering the program, were fast becoming my greatest weaknesses. All the while, adding to these realizations was the feeling that the knowledge and skill in biology and bioinformatics I had entered the program with felt increasingly inert and outdated.

I long resisted any type of change (in perspective) as I initially viewed this as a change to my character, rather than as any form of growth. As many of my family members had shared tales from their own lives, and the lives of past generations, I found that hearing those stories instilled a set of values I had been using to define myself. I was reluctant to even entertain the idea of changing, as it risked defacing years of lessons learned and many family stories. Along the way, however, I decided that this was an irrational fear, and that all of those stories I recount so fondly were merely a starting point for a longer story that was continuing on with me, and my own experiences.

In terms of my development as a scientist, initially, I stubbornly held the perspective that anything that was not quantitative was not reliable, reproducible, or scientific. Of course, my view on this matter changed drastically over time as I learned more about the process of design and qualitative research. I recall having an extensive debate over the differences between quantitative versus qualitative methods with several professors during my first year in the program. It took many classes and a saintly amount of patience on their part, for me to finally understand that qualitative methods are not lesser. That moment marked for me, the first time in many years I felt baffled, challenged, and conscious of my own intellectual growth. Many other Biomedical and Health Informatics faculty members also contributed to reframing some of the views I argued were absolutely certain or true under all circumstances.

I also formerly held the perspective that any research that was not published in a top-tier journal was not top-tier science. However, after learning much more about how the world of research operates, the various stakeholders involved, the incentives and changes in the "system", and how much of a contributing factor luck is in the success of a scientists' career, my view on this matter also changed considerably.

From a certain standpoint, although what I refer to as my ego initially provided an emotional shelter, perhaps established on my past experiences, it was simultaneously encumbering. At this very moment, I feel I have shed whatever sense of self-satisfaction I may have had, and oddly enough, as a result, I do not think I could not feel more intellectually free. I no longer expect everything I touch to turn to gold, and I think that ultimately, this means that I am being challenged. At the very least, I think it means that I am living in the moment.

I will end this dissertation on the following note—I am grateful for the resources, lessons, opportunities, and many friends I have made along this journey; I am proud of the copious amounts of careful thought and hard work I have dedicated to this corpus of academic work; I am excited for the future of my personal career, and the future of scientific findings in biomedical informatics as a whole.

I was certainly a romantic when I began my research training. However, I have also learned to better balance the passion and sense of urgency that may be characteristic of me, with a respect for circumstances, reality and patience—at least more so than ever before. Furthermore, in context of research, I have learned that it is crucial to setup expectations, boundaries, and resource limitations prior to embarking on an experiment. Setting up these constraints serve as a reminder that every experiment has its limitations.

Finally, I would like to conclude by thanking my family and friends for their support. I do not believe I would have been able to conduct the research detailed in this dissertation without their patience and moral support. Although, in some sense, this dissertation marks the end of my formal training as a researcher, I anticipate that research will be a lifelong learning endeavor. I view this dissertation as one of many milestones to come.

A Retrospective View on Risks of this Research

When I began my line of research in network visualizations, one of the first cautionary words I received was that network visualization is an exceedingly difficult research area. One of the largest contributing factors in making network visualization a difficult area to research is that many research projects require long-term commitment. Long-term commitment to a research area can understandably deter many budding researchers due to a number of foreseeable risks, some of which may be (at least in my case): (1) investing years of research only to result in minimally useful, or irrelevant findings, (2) super-specialization in a domain area that may be quickly obsoleted (or never receive recognition), (3) leaving leadership wanting for progress, which is typically measured in the short-term.

The first risk is likely an underlying fear that many researchers have experienced. I actually do not anticipate being relieved of this latent fear anytime soon. One may interpret that statement as somewhat defeatist, but I consider this to be a healthy (and necessary) dosage of concern. Without this minimal level of forward thinking, I am afraid a great number of scenarios I am unprepared for may come to pass. On the other hand, I think that for many research projects, there are numerous unrecognized applications and opportunities (and if not in the present, then possibly in the future). To be clear, I am not dismissing this fear, but conceding that this fear is influenced by unforeseen and uncontrollable factors. For these reasons, it is worth contemplating, but perhaps not distressing over.

The second risk can be summarized as the fear of committing to a research agenda in a perpetually changing, fast-paced world. What if I decide develop a novel method, or conduct

what is intended to be a seminal study, and shortly after I complete it, new findings, tools, or politico-economic events obsolete it? Can I afford that sort of "failure?" As it might turn out, specialization is not a negative thing. We live in a world where specialization exists because we work in teams, rather than in isolation, and can ultimately be more productive working in a team where everyone is potentially a specialist.

Although the first two risks may be common among researchers, I underestimated the influence of the third. The third risk seems to be prevalent not only in the world of research, but also in industry. Many publicly owned companies are beholden to stakeholders that are more interested in short-term capital gains than long-term. In response, these publicly owned companies must demonstrate short-term gains (typically fiscal) every quarter, until they eventually (and possibly inevitably) fall short of expectations. The stakeholders then move their resources to another publicly owned company that promises better returns. In this regard, the publicly owned company is unapologetically used as an instrument for generating fiscal gains, and is disregarded in favor of a competitor as soon as it begins to bear fruit at a slower rate. Perhaps due to the current political administration, or the public understanding about how government-funded research tax dollars are utilized, academic research may indirectly be emulating this very model. This mismatch of incentives is certainly not a new observation. Many public companies in the science or technology sectors invest billions of dollars to research a novel technology, or pharmaceutical drug, and are granted permission to have exclusive selling rights for a number of years. However, this burden is then shifted onto the consumer, which is certainly not ideal for yet another set of reasons (that I will not recount here).

Obtaining an intuition for the mechanism responsible for this state of affairs was actually one of the driving reasons (for me) to take on the research covered in this dissertation. The likelihood of being presented with an opportunity to dedicate myself to a long-term line of research may dwindle—and the results from a long-term project such as this, evermore valuable and rare. I speculate that this passively acquired skill will be one of the most useful to my own independent development as a researcher, and to my career. If countless professionals are trained in leading short-term projects, only a fraction of those may also be capable of illustrating experience on long-term projects (and success).

Another analogy (of numerous others not covered in this conclusion) from industry and economics I am using to frame this research and my career is the Innovator's Dilemma. The Innovator's Dilemma is a phenomenon whereby an organization is held hostage by its own success [91]. Innovation is an inherently risky activity, and if the organization is already successful, there is understandably a sizeable amount of reluctance to abandon what is already successful for what will very likely amount to a string of failures and struggles in an effort to recreate success (without guarantee) in a new arena.

The other contributing factor that makes studying network visualization challenging, is that "success" is difficult to define and measure. This warning did not have an underlying logistical- or scarcity-related root like the prior warnings. Rather, the underlying basis of this advice was that there might not be a means by which to demonstrate an improvement over the status quo. Although I managed to find a way to demonstrate success, in retrospect, I should have heeded this advice more seriously. It was not until after completing the work in Chapter 5 that I hit concrete wall. At this juncture, the outcome of my research seemed bungled, and no satisfactory end was in sight. Only after earnestly addressing the question of

measuring "success" did I design the Information Triad; and only after learning to adequately frame my research questions, did the experimental designs and their potential outcomes begin to seem less opaque.

It was only after balancing and acknowledging these risks that a research project expounded on in this dissertation was capable of being completed. I acknowledge these forces because they seem to be seldom recognized or in scope for a number of research projects.

A Brief Reflection on Biomedical Informatics

Anecdotally, I have noticed that visualization has been garnering more attention from biomedical researchers. I find it wonderful that many ideas, techniques, research strategies, and novel approaches are being translated from the field of visualization. I maintain the view that, since visualization is a prominent form of communicating research findings, and also a method by which researchers explore datasets to generate novel hypotheses, visualization is well worth formally studying. I anticipate that formal study of visualization would especially benefit those focusing on applied research, or some manner of analytics.

Dedication

For my family, my mentors, my teachers, and future scholars.

Table of Contents

Table of Figures

Tables of Tables

Table of Equations

1. Introduction and Roadmap

The contents of this dissertation reflect approximately four years of full-time study and dedication in an attempt to find a solution to what some researchers affectionately refer to as, "the hairball problem". That is, when a network visualization is adequately dense and large, it becomes uninterpretable and unwieldy. However, due to technological and scientific trends, this problem is not expected to lessen anytime in the near future.

More specifically, the contents of this dissertation elaborate on three studies I carried out between 2012 and 2016. Among a number of other contributions, three primary contributions from my completed research projects are the following:

1. An understanding of the range of challenges associated with visualizing large biological networks
2. A characterization (in terms of visual properties) of how network visualizations are presently depicted in biomedical literature
3. Estimates of how the various visual attributes contained in network visualizations support or hinder task completability (presented and described in mathematical equations)

A complete list of contributions may be found at the end of this dissertation in the concluding chapter (Chapter 8).

In this chapter, I will first broadly explain why biological network visualization is an important problem to solve, followed by why finding a satisfactory solution has proven to be difficult. Next, I provide a brief roadmap to the contents of this dissertation, and what to expect in each of the following chapters.

1.1. The Problem with Biological Networks and Visualizations

Networks are useful data structures that store entities, relationships between those entities, and associated properties; they are promising data structures for representing relational complexity in a computable form. Interestingly, over past decades, several disconnected fields of research have used networks to model research problems, and provided findings and contributions that are only recently being translated from one field to another. Network analysis uses entity and relationship information in numerous computations, and has proven useful for identifying terrorists, clarifying interdependence of financial institutions, organizing the world wide web, and analyzing social networks [1]–[3]. Networks are also useful in biology for a variety of applications, modeling everything from protein-protein interactions to biological pathways [4]–[6]. Network analysis of biological networks can identify key genes and functionally related gene communities, infer relationships between entities, and show how large numbers of entities are related [7]–[9]. However, given the size, complexity, and richness of biological networks, when visualized as node-link diagrams they often appear convoluted. Although aesthetic, biological network visualizations are often static, cluttered, obscured, uninformative, and readers cannot decode everything an author encodes. These challenges need to be addressed since it is well known that visualization can lend clarity and resolution where statistics and computation alone cannot [10].

Biomedical science research increasingly relies on data analysis, and research questions are shifting towards lines of research that search for interconnectedness, rather than

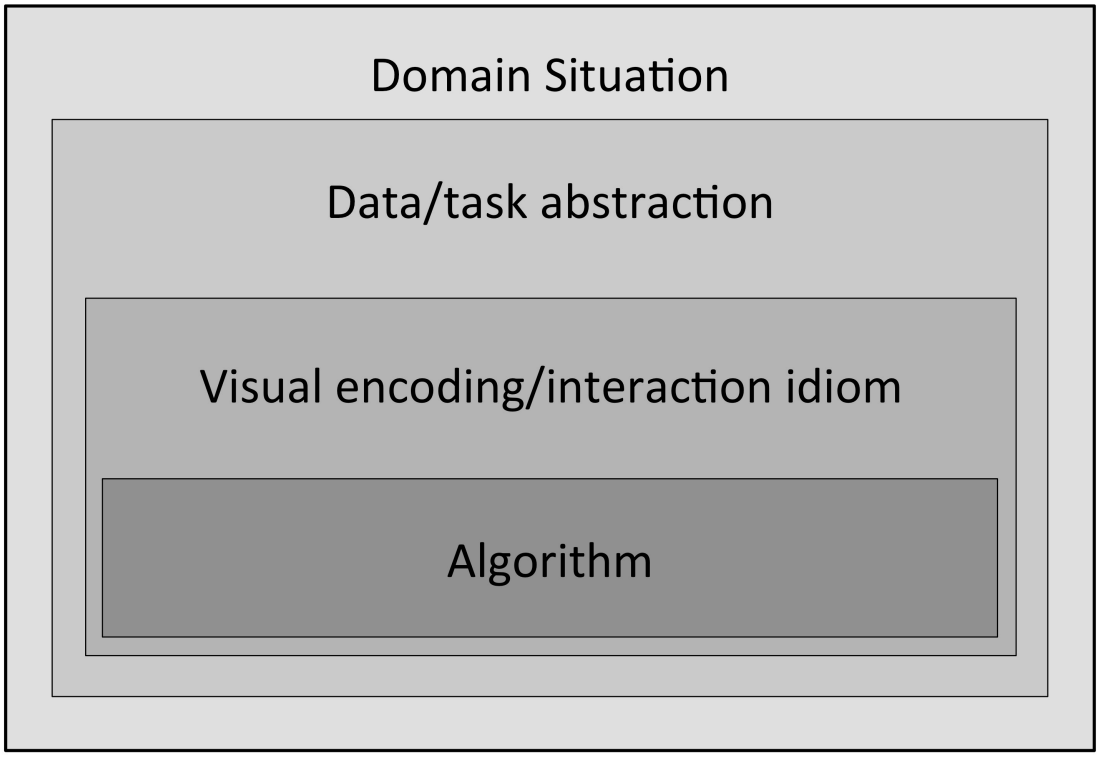
quantification or mere presence. Among a large number of staple tools used by biomedical researchers, network analysis and visualization tools are now common in many arsenals.

Recent novel approaches to network visualization are concerned with viewing data from different perspectives rather than understanding and contextualizing the data [11]–[14]. Data-driven methods can be used to support clarity and contextualization of results, without obfuscating or misinterpreting information (further defined in approach) [15]–[18]. Although a relatively new research area, researchers have developed frameworks to understand visualization structures, tactics, and information layers used in data visualizations [19]–[22]. Although statistics is sometimes criticized for its potential to be misleading or misused, visualizations have the same potential. The biomedical community may study visualizations less frequently than it studies statistical methods, but visualizations are commonly used in scientific communications. Although Data Storytelling may be predicted to be the next major iteration in visualization research, a handful of fundamental questions remain unanswered [23]. In this dissertation, I clarify these basic research questions and detail the findings of the studies used to investigate those questions.

1.2. Biological Network Visualization is Difficult

There are many contributing factors to why finding a definitive solution to improving biological network visualizations is so elusively difficult. In a general sense, the way we (as humans) process data visualizations is highly complex, and interacts at number of levels that are heterogeneously interconnected. The textbook, "Visualization Analysis & Design," describes some of these levels of information processing associated with data visualizations [24]:

Figure 1 - A recreation of "the four nested levels of vis [sic] design" from the book, "Visualization Analysis & Design"



As Figure 1 illustrates, there are multiple levels of processing a data visualization, and inefficiencies or errors may be lurking at any one of these levels; usually unbeknownst to neither the reader or author. Furthermore, these levels of visualizations are known to interact with each other, complicating the process of determining generalized solutions [24]. This dissertation touches on all four levels depicted in the figure. This is one reason why studying network visualization is so challenging—the complexity of the interactions between the various levels of information processing and design must be simultaneously tracked.

1.3. The Roadmap and major findings

The following subsection provides an overview of the contents of the chapters of this dissertation. Chapter 1 (the chapter you are currently reading) has been omitted.

Chapter 2 describes an interview study designed to understand the range of challenges experienced by researchers analyzing and visualizing large biological networks. Three major challenge areas were found, and each of them explained and supported with quotes from interview participants.

Chapter 3 details a systematic review of figures intended to obtain a quantified view of the visual properties and encodings of biological network visualizations, a qualitative analysis of selected figures, and to obtain an understanding of the status quo of biological network visualization. One of the findings of this study was that there are actually several sub-types of biological network figures, and two of the major sub-types were characterized and explained in this chapter.

Chapter 4 extends the study described in Chapter 3 through re-evaluating the same figures in context of a graph task taxonomy. The re-analysis of the data from Chapter 3 is followed with descriptive statistics. A major finding from this research was that the ability to complete a given task on a network visualization is correlated with the visual encoding choices, size, and density of a network.

Chapter 5 extends the material detailed in Chapters 3 and 4 with data analysis via Random Forest. Through Random Forest, importance scores are calculated, providing a guideline for which visual encodings (from Chapter 3) are most important in context of certain tasks (from Chapter 4). A major finding from this analysis was that certain groups of tasks may be associated with very different visual encodings, and that size and density are essential factors for task completability.

Chapter 6 describes a small conceptual framework which is useful for framing and understanding visualization research problems. In addition, this chapter also elaborates on a software application, which is built on the premise of the conceptual framework, and is designed to systematically and reproducibly describe and compute visual encodings in network visualizations. Aside from a description of the conceptual framework, a major finding included in this chapter is benchmark results for the software application.

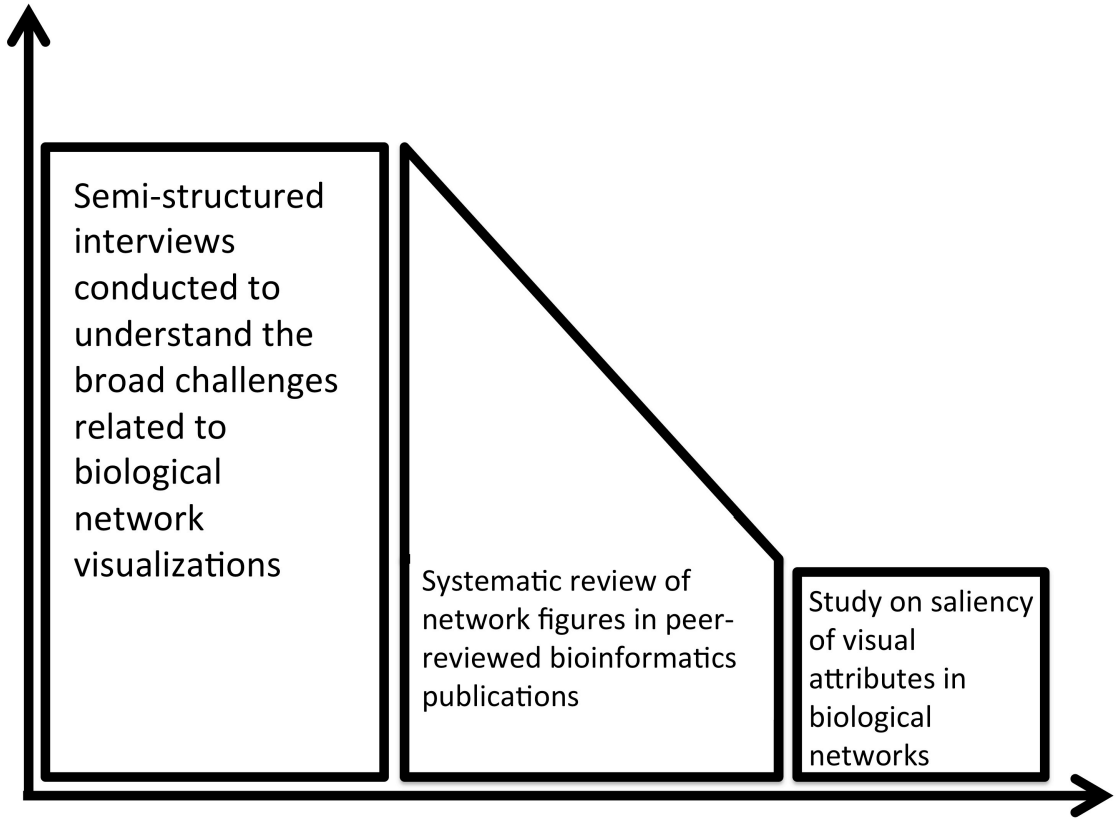
Chapter 7 explains the design, administration, and results of a task-centered perception study using Dynamo (described in Chapter 6). Random Forest is used to analyze the data collected from the experiment and the major findings include function estimates for how the completability of a task (visually scanning a network) is bolstered or hindered through the various parameterizations of visual encodings in networks.

Chapter 8 concludes this dissertation with a brief recap of contributions accounted for in each chapter, a description of my vision for the future of this line of research, and a brief personal reflection on my growth and development as a scientist.

1.4. Scope and structure of this dissertation

As one may surmise, the set space of research possibilities within the umbrella of biological network visualization is near infinite. Although the type of study I initially began was broad in scope, subsequent studies required exchanging breadth for greater depth of focus. The first two studies investigating problems with biological pathways and network visualization led to the discovery of visual encodings as a gap that could benefit from an algorithmic approach. Thus, the structure of this dissertation obeys to a similar pattern. The structure of this dissertation may be conceptualized as having a funnel-like shape. Figure 2 illustrates this funnel-shape by depicting relative breadth versus depth. As Figure 2 shows, the semi-structured interview study (Chapter 2) is broad in scope, and consequently has less specificity and depth. The study founded on the systematic review of network figures is more focused relative to the semi-structured; it has less breadth, and more specificity and depth. The final study included in this dissertation is an investigation of the saliency of visual attributes, which is much more focused and in-depth compared to the other two studies.

Figure 2 - Schema of the funnel-like shape of this dissertation


Semi-structured interviews conducted to understand the broad challenges related to biological network visualizations
Systematic review of network figures in peer-reviewed bioinformatics publications
Study on saliency of visual attributes in biological networks

2. Empirical Assessment of Network Visualization Challenges in Biomedicine: An Interview Study

2.1. Overview

Over the last 20 years, the biological community has been collecting and sharing experimental data on the web. More recently, researchers are exploring the relationships or connections among these data. Network analysis and visualization is a natural fit for this type of research. However, network visualization has a number of well-known challenges associated with it. Networks increasing in size and density eventually produce diminishing returns on comprehension and perception—vertices become occluded, edges cross, and the end result is a incomprehensible network that typically provides little value aside from communicating that one is viewing a complex data structure. Researchers have also shown that for a number of tasks (with the exception of path following), diagrams of networks (containing between 20 and 100 vertices) may be less efficient for task completion than matrix representations [25]. There is, however, a significant learning curve involved with reading matrix visualizations for the uninitiated, whereas reading graphs is more intuitive. This learning curve and the domain knowledge required to be able to comprehend matrix visualizations renders it prohibitive for communication to a wider audience.

Due to these limitations, many researchers who use network visualization and are consequently frustrated. However, this sense of frustration is mostly anecdotal, as there are limited studies about the users of network visualization. To better meet these needs, we need better designs for biological network visualization tools. However, for new designs to be effective in this domain, we must better understand the needs of these users and the challenges they may have experienced while visualizing with biological networks. In addition, we can and should leverage biology-specific aspects of the domain when designing network visualization systems. In this chapter, we present the results of an interview study we conducted aimed at understanding the challenges involved with network visualization in biology.

Through the findings and design implications detailed in this chapter, we provide suggestions for areas of clarification, and suggestions for additional information that is pertinent to improved biological network design. In particular, we highlight visual representation challenges related to unsupported biological constructs and perceptual challenges.

2.2. Related Work

There have been a number of valuable contributions to biological network visualization over the recent years, many of which have received attention in the bioinformatics community. In this section, we discuss current tools and techniques, as well as the remaining knowledge gaps. We first review the evolution of network layout algorithms. Gibson et al conducted a survey on two-dimensional graph layout techniques and showed that even when inclusion criteria narrows layout algorithms to only the force-directed family, there are 19 published algorithmic approaches to laying out a graph, each with its own novel approach [26]. The Gibson survey demonstrates that there are a wide variety of visually distinct representations for the same network, but that for some networks, none of these choices are universally satisfactory. For instance, various force-directed layouts techniques may facilitate tasks such

as counting node degrees or identifying adjacent nodes, but knowledge about the relationships represented in the graph may be unclear.

Other algorithms aim to organize vertices and minimize edge crossings to prevent overly cluttered layouts that can occur in visualizations of dense networks. Two such vertex-centered algorithms are Circos and Hive plots [11], [12]. Circos organizes data in a polar plane, akin to a polar bar chart, and adds edges in the middle of the polar bar chart to show which elements are connected to which other elements. The Hive plot linearizes vertices for a perceptually uniform representation, and then adds edges to show relationships between those vertices [11]. Although these layouts enforce some organization on the positions and classes of vertices, and alleviate some of the strain encountered when visualizing a large quantity of vertices, there are situations where edges may still cross and tangle.

Another approach to solving the problem of overly cluttered network visualizations is to organize the edges rather than the vertices. BioFabric is a layout algorithm that arranges a network in a manner reminiscent of an adjacency matrix [27]. Biofabric is useful for clarifying which vertices in a network are connected to which other vertices, and in a sense, "untangles" a standard hairball. Another edge-focused algorithm is edge compression, which consolidates edges into bundles. Edge-focused algorithms provide utility when working with dense biological networks [14]. Although edge crossing can be avoided through reducing the number of presented edges and organizing densely connected vertices into groups, vertices are sometimes lost or duplicated in order to visualize networks in this manner.

Constraint-based layout approaches have also been proposed as a useful algorithmic approach, particularly for biological applications [28]. Through setting up constraints, vertices and edges may both be organized into meaningful patterns. Constraint-based algorithms have been used to enforce hierarchical relationships in a network layout [29]. Constraint-based approaches can be used to develop specialized layout conventions for specific types of biological networks, or can be parameterized to serve a broader range of biological networks. Constraint-based approaches are versatile enough to not only enforce constraints of visual properties (e.g. ensuring a vertex remains in a certain range), but also constraints based on the data (e.g. ensuring a vertex representing protein is not placed in the center of the screen unless it is enriched past some calculated threshold).

All of these techniques afford a novel perspective on biological network data, and are arguably clearer than standard network representations for certain tasks. However, graph drawing algorithms are often evaluated based on metrics reflecting aesthetics, which is one reason why even though a network visualization may be laid out well, it still may not be insightful [30].

Among prepackaged network visualization tools, Cytoscape is the most widely used biological network visualization software [31]. Cytoscape software significantly eases the burden of obtaining biological network data from various resources and visualizing it. Through the Cytoscape App store, many researchers have been able to develop and share their own plugins that integrate into Cytoscape software. However, since these plugins are typically developed independently of each other, it is atypical to see a plugin building on, or depending on, another plugin. More recently, Cytoscape.js has become further developed and opportunities are growing for development of Javascript-based web applications. Another toolset, BioconductoR (used in the R programming environment), provides a

programming interface to a number of resources, techniques, and libraries commonly used in bioinformatics [32]. A number of packages that support network visualization and pathway visualization are easily downloaded and used via BioconductoR. In addition, there are also general-purpose R packages that support visualization, such as ggplot2, and network analysis, such as igraph, both of which are also widely used [33], [34]. There are a number of other widely used biological resources that are capable of delivering biological information via web visualizations, such as Ingenuity Pathway Analysis, KEGG, EcoCyc, MetaCyc, and HumanCyc [35]–[39]. However, these tools are actually geared more towards analysis than visualization, and focus more on properties of the data portrayed in familiar pathway layouts than on enabling arbitrary layout functions.

Although novel network layouts have received attention in the bioinformatics community, reorganizing existing information in networks may not meet the needs of biologists. Optimizing layout algorithms for clarity of visual arrangement without the context of data and tasks can be inadequate. The purpose of this study is to identify challenges encountered by real users in the process of understanding biological network visualizations. Although there have been studies assessing the utility of a graph layout in context of specific tasks, the graphs used in these studies tend to only contain a handful of nodes. Furthermore, the usability studies that have been conducted (on force-directed layouts) seem to be primarily focused on evaluating aesthetic graph drawing principles [26]. A literature search (conducted in Autumn of 2014) for studies assessing user needs on biological network visualization did not yield any results.

2.3. Method

In this section, I cover details about the administered semi-structured interviews: interview participants, interview questions, and thematic coding (by three coders). This study was approved by the University of Washington Institutional Review Board.

Scientists who have worked with, and visualized, biological graphs, were eligible to participate in the study. In total, 21 researchers (17 male / 4 female) completed in-person interviews. Interview subjects were recruited via snowball sampling (requesting referrals to other eligible researchers). Participants were recruited from 9 local research or research-oriented health care organizations. Interview participants held a number of job titles, including “principal investigator”, “research scientist”, “graduate student”, “data scientist”, “software engineer”, “director”, and “assistant professor”. Interview participants held a variety of academic credentials (14 PhD / 4 Masters / 2 Bachelors / 1 Medical) and ranged from less than 5 years of experience in biomedical research, to as many as 25 years. Interview participants included a wide variety of training backgrounds, self-identified as: “bioengineering”, “bioinformatics”, “visualization”, “biology”, “biostatistics”, “clinical”, “computer science”, “mathematics”, and “systems biologist”. The interviews were conducted either in-person or through video call. Although there were 10 interview questions in the template, many of these questions were followed up with additional questions intended to elicit more detail. A subset of the interview questions are:

- How do you currently visualize networks?
- What resources do you tend to use in your work?
- Do you find any biological relationships difficult to model?
- What do you find frustrating about pathway analysis?

- What do you wish you could do with pathway analysis that you currently cannot? Is there anything specific to visualization?
- How often do you compare networks to each other?

The interviews were audio recorded, unless the interview participant opted out, at which point notes were taken during the interview instead. The audio recordings were transcribed to text prior to any processing in the following steps. In cases where interview participants opted out of audio recordings, notes were taken instead. The interview transcripts were initially coded through the process of open-ended coding (using NVivo software), characterized by reading transcripts and tagging any notable or recurring themes. Two other researchers repeated this open-ended coding independently, and all thematic codes were analyzed to develop a consensus codebook. This consensus codebook was then used to thematically code the interview transcripts.

2.4. Challenges of Biological Network Visualizations

In this section, I characterize the challenge areas described by interview participants. Identified challenges were organized into three groups: (1) Data and analysis, (2) visual representation and interpretation, and (3) limitations of models. The first subsection enumerates challenges associated with standards and annotations, the high quantity and density of network data, and validation issues. The second subsection addresses challenges associated with unsupported biology, perceptual issues, and interpretation challenges. The third and final subsection covers challenges associated with the difference between exploration, explanation, and philosophies of science. Please refer to table 1 for an overview of the organization of content.

Table 1 - An overview of challenges from participants

Section	Subsection
Data and Analysis	Challenges associated with data in standardized formats
	Challenges related to high quantity of data, and high density of data
	Challenges validating, and ultimately trusting, information contained in pathway resources
Visual Representation and Interpretation	Challenges representing common biological constructs in networks
	Challenges associated with perception, and reading a network visualization
	Challenges associated with interpretation, and comprehending a network visualization
Limitations of Models	Challenges consolidating varying philosophies of science
	Acknowledging limitations of understanding

For the purpose of this paper, the terms "network", "graph", "node-link diagram", and "pathway" interchangeably. Although these terms refer to specific concepts in their respective domains, the interview participants had a variety of domain backgrounds. Hence, not all participants would use the same terms to describe the same concepts. For instance, a biologist may use the terms "network" and "graph" interchangeably, or a computer scientist may use the terms "network" and "pathway" interchangeably.

2.4.1. Data and Analysis

As shown in Table 1, three challenges were grouped into this "data and analysis" category: (1) problems with standards and annotations, (2) high density and large amounts of data, and (3) challenges with validation.

2.4.1.1. Standards and Annotations

Biological data are often communicated and shared under recognized standards. However, there are too many standards, and too many versions of these standards. Depending on the research question, the standard used to communicate data can be a vital component of the network visualization workflow. Standards vary in the information they contain, the representation of that information, and the definition of that information. One researcher explained that any downstream analysis and visualization of biological data is difficult, primarily due to inconsistencies between standards:

"The most difficult part about running statistical tests on networks is making sure definitions are consistent between models" –Participant 2

In addition to challenges associated with comparisons across different standards, there are also inconsistencies even within a single standard, due to updates and version changes. As another researcher explained:

"Standards are annoying because they change over time. You write a custom app that uses one version of a standard and a new version of the standard is released and you are angry because your code is broken." –Participant 5

Thus, comparing data collected more recently with historical data may be inappropriate depending on the modifications to the standard. Inferring relevance through comparing datasets to one another is a critical and common task—without being able to compare ideas, models, and results to that of other's, it is difficult to evaluate scientific results. Additional representation challenges stem from annotation, which often contains key information about the relevance, significance, and context.

2.4.1.2. High Quantity and Density of Data

Some interview participants find the nature of the data they are working with to be a challenge, for a number of reasons. Aside from the large size and high density of biological networks, a major challenge is representing all of the important features contained within the data. Although the data is rich with information contained across many rows and columns, due to visual encoding constraints, only a subset of that information can ultimately be used (without resorting to alternative approaches, such as dimensionality reduction). An interview participant explained:

"There are multiple aspects to why it's hard to look at networks. One is that they are big. However, that's not really the problem, there are others: one of them might be that you have only a few key nodes, but all of them are connected, so if you drew it with node and link it would be spaghetti. Another problem is some nodes belong to multiple groups or sets of things, and you can't draw a nice Venn diagram." –Participant 14

As interview participants explained, the topological and multifaceted properties of data affect the interpretability and readability of the network visualization. A number of interview participants grapple with the challenge of knowing whether or not visualizing biological data in a network will be informative in the first place—it's difficult to tell whether or not biological data will reveal anything interesting or insightful in node-link form.

2.4.1.3. Validation of Data

Interview participants are also concerned with the quality and trustworthiness of the biological data they are visualizing. Although many biological resources exist, the type of data they contain, the source of the data, and the context the data was captured in, may vary significantly from resource to resource. In a number of cases, resources also link information existing in another resource, creating a conglomeration of information that may not be ideal. A researcher working with human cell lines explained:

"These pathway databases are useful, but they often combine data from several organisms and tissue types. If I run an experiment using heart tissues from mice, then the pathway data available might just be misleading." –Participant 6

To further explain the quote from above, the pathways that are active in a specific species and tissue may be very different from the more generic pathway information in the resources. Although the aggregation of this biological information is useful in situations where experimental data on a specific organism, tissue type, or cell type is sparse, some interview participants exhibited uneasiness with using this information:

"If you are comparing pathways from data from other cell types, I perceive it as useless. I have no idea how annotated pathways hold up against cell types. If you have a cohesive, hodgepodge of pathways, and you take your cell type and overlay it, I don't know how powerful that is. The issue is that it's not getting to the root of the pathology. The reason animal models break down is that in a mouse, there is just enough evolutionary divergence such that the phenotype is due to differences in evolution. Muscle is similar between mice and humans, but how the nerves, vasculature, immune surveillance, hormonal implications all interact, cause more and more divergence from how humans do it. What if your phenotype in mouse is a consequence of having a different immune surveillance and you don't see it in human disease?" –Participant 6

As another interview participant explained, many challenges arise downstream during analysis and visualization primarily due to limitations imposed by available data and experimental designs.

"The number one challenge with biological networks: Getting good data! BioconductoR has helped a lot, but it is still somewhat of a challenge. I was relying on KEGG for a while, but then they went private…so, finding good (pathway) models is a bit of a challenge. Networks

are not clean, and experimental designs have issues we have to address, data have issues we have to address. When data are taken care of, everything becomes remarkably easier." – Participant 2

Although tools and resources exist to download data from various resources under various standards in programming environments, this accessibility and computation is considered a secondary challenge.

2.4.1.4. Visual Representation and Interpretation

As shown in Table 1, three challenges were grouped into this "visual representation and interpretation" category: (1) unsupported biological constructs, (2) perceptual issues of networks, and (3) challenges with interpretation of network metrics.

2.4.1.5. Unsupported Biology

Although a number of network visualization tools exist, interview participants still exhibited discontent with existing tools and techniques. There are a number of fundamental biological concepts that are either unsupported in existing tools, or difficult to represent. For instance, a researcher commented on the difficulty of accounting for consumable elements in a biological system:

"It's difficult to represent ADP and ATP, NAD and NADH pools in pathways. I don't know if they are the same pool or different pools...I can't do this in Cytoscape! A metabolite may only appear once in graphs." –Participant 5

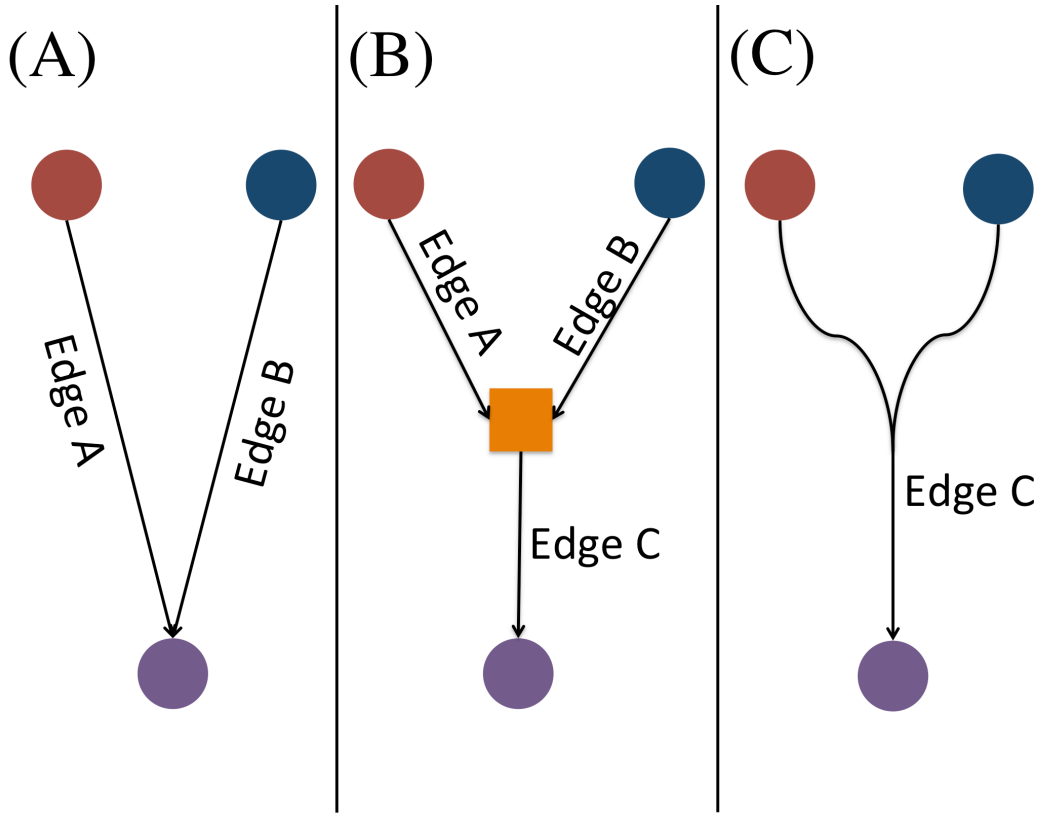
Since representation of consumable elements is not universally supported in tools, interview participants must keep track of these relationships and interactions mentally, as one interview participant explained:

"In science, especially biology, there is this vast sea of info biologists store in their minds and they put in context what they see in visualization frameworks and try to interpret the patterns they observe in context of all the principles of biology they are aware of. They look for interesting links and say, they think it's something new what I'm observing here, no one has seen this protein controlling these 2 genes of completely different functions in this environmental context--this is really important! The software has no notion of what the genes are, context, or surprise that they correlate--this is the biologists interpretation." – Participant 15

Aside from representation of consumables and currently accepted hypotheses in biology, certain types of interactions are also difficult to represent in biological networks. For instance, one interview participant mentioned having to omit fundamental biological information due to a lack of support for that type of information:

"We decided not to bother with small details (e.g. type of reaction), but there were situations where we were creating a directed network and didn't know if a reaction was reversible. We had lots of information we couldn't put into a network…I like hypergraphs, but there isn't support for that in network visualization tools. KEGG has some hypergraph type data, but it's still difficult to show this visually. We handled this by translating a hypergraph to basic set operations, but it's still not clear if the logic is 'OR' or 'AND'." –Participant 18

Figure 3 – Depiction of a hypergraph.



A hypergraph is a network where edges are not restricted to have only one source vertex and one sink vertex. For an illustration and explanation of what a hypergraph is and how it poses a challenge, please see Figure 3. Figure 3-A Illustrates a standard directed network where the red vertex and blue vertex link to the purple vertex. Figure 3-B Illustrates the workaround to representing a hyperedge via an additional "logic" vertex. Not only does this approach increase the number of vertices and edges in a network, but the logic is ambiguous. The red and blue vertices may both need to be present to link to the purple vertex (i.e. "AND" logic), or perhaps only one of the red and blue vertices need to be present to link to the purple vertex (i.e. "OR" logic). Figure 3-C Illustrates a hyperedge, where the red vertex and blue vertex simultaneously link to the purple vertex. Hyperedge relationships are common representations for biochemical reactions and biological events and are necessary to represent molecular complexes or metabolism involving more than two participants. Although they are commonplace in biological schematics, existing tools do not universally support hypergraph relations.

Some interview participants also reported a lack of support for temporal data in tools and techniques used to visualize biological networks. Researchers explained that any analysis and visualization related to temporal information was conducted through makeshift tools or using currently existing tools in ways they were not designed for. One researcher summarized the challenge in using currently existing tools and explained the consequences:

"The problem is that none of the tools really work or if they do they don't provide--they are either toys or are not that useful because the visuals are not what they really want, or it's hard to map what they want to visualize (transcript change, metabolite change, protein abundance change), mapping it to a metabolic network is tough because of scaling issues or metabolite pool data is relative and not absolute, etc. So they end up focusing on a very small portion--because it's tractable, they can draw it by hand and color it by hand." –Participant 5

Another challenge mentioned by interview participants was in reference to difficulty representing space and volume in a network. When networks are laid out for aesthetic optimization (e.g. layouts in the vein of force-directed layouts), there is no inherent meaning in the position of the vertices. However, an interview participant commented on how this lack of meaning in vertex position is challenging in context of cellular localization:

"Cellular localizations: whether a protein goes to the nucleus, membrane, peroxisome, etc. It's hard to look at the dynamics of that." –Participant 15

Where in the cell a certain reaction takes place is pertinent information. The same elements of the same pathway may behave very differently depending on cellular localization.

2.4.1.6. Perceptual Challenges

It is well known that comprehension and readability of network visualizations diminish past a certain size and density. Although this limitation is acknowledged, network visualizations of biological data remain prevalent in a number of situations. An interview participant explains:

"Respect the limits of my visual acuity and visual capacity to digest things with my eyes, please show me stuff I can parse. I can't parse a network. Too much data occlusion, too much stuff going on. If you rotate a network you can't even tell if it's the same thing! It's like the emperor's new clothes. We agree to not care because there doesn't seem to be a good alternative." –Participant 12

Much of the frustration with network visualizations stems from the visual strain of examining the information contained in networks. Among other tasks, participants explained that networks are visually examined to identify recurring patterns or signature, or to identify the location and interactions of a specific vertex in a network.

"It comes down to graph isomorphism, what structures am I looking for visually, what structures can I detect, and how can I detect them computationally, so I don't have to draw them and go hunting..." –Participant 12

Not only did research participants acknowledge their own perceptual limitations in this context, but they also brought up the point that due to a lack of biological visualization tools that compensate for perceptual limitations, the utility of new technology used to capture and study biology is indirectly limited:

"Now we can do whole genome RNA-Seq, but we still don't have the data visualization tools. The technology is better, the data better reflects what is happening in the cell at the time, but looking at it and comprehending what we are looking at hasn't changed much in 15 years. People still generate a huge dataset, a huge heat map, and then end up cherry picking,

and say we are going to look at a very small subset of the genes. You've almost thrown all of that technology away again." –Participant 6

Since human perceptual bandwidth is limited, researchers may be forced to focus down into a specific subsystem. As an interview participant stated:

"The problem with networks is that they tell 100 stories at once" –Participant 12

Thus, network visualizations are not only data-dense, but also information-dense. As a consequence, interpretation of the information contained in topological structures may not be straightforward. It is clear, however, that visualization plays an important role in interpreting a network, for a depiction can encode many informational components in a comprehensible manner.

2.4.1.7. Interpretation Challenges

Interpretation of the information contained in topological structures within networks is not straightforward. Since representation of data and visual encoding of properties can vary significantly between network visualizations, it is also difficult to compare interpretation of data between networks. In order to discuss structures within network visualization in a systematic and reproducible manner, a number of network metrics are calculated. However, a major challenge is that the metrics used to characterize networks, such as centrality, community detection, or network model, depend heavily on the construction of the network:

"I'm not quite convinced knowing scale-free or small-world is useful or interesting. To me it's more of an overview thing. If I have a scale-free network, and its scale free because I happen to filter out points, is that really interesting? There are so many ways to filter and manipulate networks" –Participant 14

The use and interpretation of these metrics is also somewhat inconsistent. Depending on domain knowledge, research question, and experimental design, the same metric may ascribe different meaning or significance. The consensus among interview participants is that biological networks are data structures, and should serve merely as starting points for exploration:

"Networks are intermediate results. You aren't finished yet. You need to show the insight! It doesn't matter what data structure you use…" –Participant 12

Since communication is an important part of science, and since the message of a network visualization can be obscured, many interview participants have expressed great interest in improving this aspect.

2.4.2. Limitations of Models

As shown in Table 1, two challenges were grouped into this "limitations of models" category: (1) challenges associated with exploration and explanation, and (2) differing philosophies of science.

2.4.2.1. Varying Philosophies of Science

One of the uses of network visualization is hypothesis-free research. However, an interview participant commented on how exploration is not entirely possible in the way it is generally imagined:

"Before trying to visualize anything, you want to know what would be interesting to see--which is counterintuitive to exploration!" –Participant 14

Although it is possible to use networks for hypothesis generation, there are numerous upstream decisions, ranging from data source to construction of the network to visual encoding, which can greatly affect exploration. Although it is possible to overcome this challenge through iteration of upstream components of the visualization workflow, there remains the notion of identifying a pattern that is scientifically interesting, which often means comparing the data contained in networks to currently existing hypotheses or scientific beliefs. In addition, interview participants commented on a philosophical struggle between those who work "bottom-up" versus those who work "top-down". An interview participant described, "bottom-up" as building something and hoping for a use, and "top-down" as finding a driving biological problem and then searching for solutions. Another interview participant elaborated on perspectives derived from statistical approaches may sometimes be different than a perspective obtained from a purely biological approach. As the participant described, statistical approaches suppose that data should "speak for themselves", and that one should be learning from the data rather than explaining it. However, the challenge is that, if working with pathways, there can be thousands of different priors, and many different types of priors, perhaps in weighted distribution form.

2.4.2.2. Limitations of Understanding

Among interview participants, there was wide acknowledgement of the limits of the modelling capabilities of tools, and the knowledge one may be able to obtain from experimental designs. For instance, one interview participant elaborated on the limitations of two-dimensional network models:

"Everything interacts with everything else in biology! The node is always near-by no matter where you are in a representation. It's like what you have in a cell, it doesn't make sense to position specific elements to represent a cell because the same element is everywhere in the cell in real life." –Participant 3

At times, biological network visualizations seem to represent information that is too abstracted and far-removed to be meaningful, at least without risk of over-interpreting. Although much of abstractness comes from the design of the network, some limitations of network models are a result of the limitations of experimental designs. An interview participant commented on the perplexing results of a study:

"A person I worked with at ISB studied Cystic Fibrosis in a mouse model—one of the best known single-gene diseases. He established that if mice had the standard mutation, 70% of them die as you expect, but 30% are fine. And all have same genome and mutation, and that can't be explained by the mutation. There are confounding factors we don't understand—so things aren't that simple." –Participant 16

Although there is significant understanding of specific biological subsystems, there is little understanding of how these biological subsystems are connected to each other. There are a number of well-studied canonical pathways:

"The clinicians like to look at pathways--they have concept of cascades and gene relationships, flow of metabolic intermediaries, etc. But they view it like one pathway at a time, but all of these pathways are connected to each other and happening at the same time, in the same place. This idea that there is one canonical pathway is not right and we can forget that. We want to simplify them because reality is complicated, but if we simplify them too much we have to remember the limitations of our conclusions due to those simplifications." –Participant 12

Historically, much biology was organized in hierarchical knowledge structures, such as clades. Since this serves as the starting point for understanding biology, it is natural to want to view biological network data in the same view.

"I think hierarchy is largely a human construct to simplify a complex set of knowledge. It makes things easy to talk about and communicate, but in reality things aren't ever that clean. But the fact that it has been used for thousands of years, says something. Now we know that the tree of life is more like a web of life. Trees can represent time, spectrums that have been quantified and categorized, but it's never really that simple." –Participant 10

Some interview participants argue that reasoning by induction and simplifying complex systems through abstraction is the key:

"A lot of how we understand things goes from lower level to higher level, processing by induction. I don't see a lot of this. It's nice to see data at a higher level and then drill down--maybe I'll find something in there that I can use at a higher level. There is this notion of moving around in a network" –Participant 17

Even when one is exploring a particular biological subsystem, there is a desire to be able to connect a specific observation with a larger context. However, when working within a biological subsystem, this is not always possible. Facilitating this process of inductive thinking may help compensate for a lack of tools that connect these subsystems.

2.4.3. Future Trends

In this section, I discuss the significance of future trends in science, specifically in context of standards, resources, and networks visualization tools and techniques.

As detailed in the findings, standards and resources are an important, yet underappreciated factor in network visualization. At the moment, there are 4 recognized, biologically specific standards: SBML, CellML, BioPax, and PSI-MI [40]. SBML and CellML are both standard for representing biosimulation models (where rate constants are known), BioPax is a standard for pathways, and PSI-MI is a standard for protein-protein interaction data. Each of these standards defines biological elements differently. As a consequence, merging biological information stored under different standard formats often results in biological networks that are difficult to interpret—computationally and visually. The availability of data across various biological resources is also an important factor, especially in context of validating the data contained in biological networks. As of November 2016, there are 547

pathway resources [40]. For certain lines of biomedical research, a major challenge of using biological networks to visualize canonical pathway data is that the pathway information contained in biological resources are an aggregate, which can be a form of contamination depending on the research question.

In general, the scientist using the network visualization determines whether or not the visualization is useful. A disadvantage of standardized depiction of biological networks is that the very same representation may contain too little or too much information depending on the reader. For instance, to a domain expert in Alzheimer disease (AD) there may be little interest in using a community representation of AD as it may be based off of a competing hypothesis, or contain annotation that are irrelevant or redundant. Simultaneously, that very same representation of AD may be enormously useful for a cancer researcher exploring the intersection between cancer and AD. For this reason, open, community-driven models of biological network representation, such as Wikipathways, has seen slow adoption from the cohort of interview participants [41].

Looking forward, biological resources should also contain cell-, tissue-, and condition-specific information [42]. As the amount of biological data the community collects, stores, and shares continues to increase, the requirements and specifications for these items will continue to be defined with increased specificity. However, since standards in biology are ordinarily developed with downstream objectives in mind, there seems to be a "chicken and egg" problem. For the time being, it is reasonable to assume the limitations imposed by the structure of the information contained in standards and resources.

One might anticipate more Cytoscape plugins will be developed over time. There are already Cytoscape plugins that address some of the challenges detailed in this paper. For example, Cerebral is a plugin facilitating layout informed by cellular localization [43]. However, with an application like Cytoscape, users must rely of developers to support modifications to formats and standards. Since other researchers have also argued that temporal information is difficult to portray in biological networks, I am optimistic increased support for temporal data will emerge in the future [6]. Dynamic network visualization is a growing research area, although there are no Cytoscape or Bioconductor extensions yet. Furthermore, I also anticipate that biological network visualization web applications will become more prevalent due to the number of Javascript-based network visualization libraries, including Cytoscape.js. A number of web services such as Pathway Commons and GeneMANIA already visualize biological networks in the browser [44], [45].

2.5. Design Implication and Recommendations

This section covers the design implications of the interview findings. In particular, this section contains recommendations regarding clarification of a number of network-related information, and recommendations for additional constraints in constraint-based layouts.

2.5.1. Clarifying Data, Network Components, and Interpretation of Topological Structures

Providing clarity and detail about what information is actually being represented in network visualization would improve network visualization significantly. However, for the time being, network visualization can be improved through clarifying the use and provenance of secondary data, clarifying the exact meaning and relevance of visual components (e.g. vertices and edges), and clarifying the relevance of the identified topological features.

Ensuring that biological data collected from experiments is openly available is critical for validating findings, and enabling others to build off of prior research results. One the many benefits of easy accessibility to biological data is widespread secondary use of this data (i.e. using data that was collected for another purpose). Secondary data offers enormous benefits for exploratory research, data aggregations, and statistical model development (e.g. machine learning and cross-validation). However, as some interview participants described, the resulting information may contain errors, inappropriate groupings, or vague definitions that limit their usefulness. Moreover, secondary data may be used or analysed in a manner that is inappropriate given the sample collection and purpose of the dataset.

To improve clarity and appropriate interpretation of networks, one must be more clear about exactly what a vertex or edge represents. Standards and resources play a pivotal role in enabling rigorous definition of this nature. Since standards committees depend on feedback from the community, those involved with the design of biological network visualization must communicate with standards committees about the downstream challenges for visualization —especially for challenges related to ambiguous definitions of controlled terms. There is also an onus on the part of the researcher constructing the network visualization to properly clarify and define the details and relevance of network visualization. In terms of design and construction, this may mean creating separate network visualization for each class of vertex, or edge, while preserving context by using an identical layout for each visual. However, if one were to make generic recommendations excluding any context of task, there are would be too many classes of vertices and edges to show. In this situation, a more abstracted, higher-level visualization may be more appropriate.

Since topological structures in network depictions are not easily recognizable in large, dense networks, visualizations should highlight topological structures and interpret their significance in context of the data. In section 3.2.2, delineating perceptual challenges, interview participants specified difficulty recognizing whether one structure is identical to another structure, given that structures may be somehow altered in representation (e.g. transformed, translated, or rotated). Also, a number of foundational biological concepts are either difficult to, or impossible to, represent with current tools and techniques. As a result, analysis, computation, representation, and communication of biological networks are hindered. For example, hyperedge representations (see Figure 3) are commonplace in biochemical schematics, but are not possible to represent or compute over without resorting to workarounds. Although workarounds are clever, researchers may risk inappropriately using a tool or technique, or inappropriately interpreting the results. In summary, by clarifying the provenance, limitations, representation, and interpretation of network visualization, readers are less likely to be confused and can make less assumptive inferences.

2.5.2. Layout Constraints via Biological Knowledge, Data, Tasks, and Experimental Design

Enforcing constraints informed by the attributes of vertices and edges in biological networks can reveal structure within the data, as many interview participants expect. Although constraint-based layout will not solve downstream perceptual challenges (such as vertex occlusions or edge crossings), there will be an expected structure. Network visualization with a sense of order can enable useful operations, as readers can compare the structure and pattern of the data with the structure and pattern they expect to see. For instance, hierarchical layouts can be implemented via constraints, and depending on the vertex/edge attributes used to inform the constraints, readers would typically expect a tree

structure. Schreiber et al present a general constraint-based approach that applies to a variety of biological networks [28]. An advantage of a constraint-based approach is that the algorithm can systematically emphasize relationships of interest, or be customized in a variety of ways that are reproducible and extensible. Depending on the intended use of the visualization, constraint-based approaches can also be used to setup layout rules based on currently accepted biological knowledge. This can be especially useful in layouts depicting pathways or networks containing specific types of biological data (e.g. protein-protein interaction, gene regulatory, etc.).

However, I argue that although enforcing constraints based on data, prior biological knowledge, and tasks is useful, designing layouts in context of experimental design would provide additional precision. Biological data is visualized with the inherent purpose of understanding how the collected data compares to currently accepted hypotheses and models. Many biological experimental designs are a systematic approach to answering a specific research question—providing a guideline for what is relevant to visualize. This approach is particularly suitable for scientific designs characterized as the "top-down". At first, it may seem as though "bottom-up" scientific designs require a different visualization methodology, searching for larger patterns in an agglomerative fashion. However, regardless of whether the "bottom-up" approach is being used for confirmatory or exploratory purposes, readers need to be able to compare the information contained in the visualization with their current understanding. Although a truly "bottom-up" approach that is "data-driven" would be ideal for certain research questions, there are also a number of limitations and biases that are not clear in the resulting visualization.

Another approach, similar in spirit but leading to more generalizable techniques, is to optimize network visualization for a single task, or set of tasks. Lee et al published a taxonomy of tasks completed in graph visualizations—this list can be used as a rubric to develop task-specific network visualizations that are designed to support the task the author of the visualization expects the reader of the visualization to complete [46]. Although tasks provide a specific context through which one can evaluate the efficiency of network visualization, using tasks alone as a basis for design and evaluation may result in tools that are inappropriate for the type of data being visualized. Thus, in summary, I would like to emphasize that it is necessary to understand the ramifications of the entire range of components used to develop a network visualization—from the experimental design and hypothesis, properties of the data, the anticipated tasks readers are expected to complete, and the assumed prior biological knowledge.

2.6. Conclusion

This chapter presents the results of an interview study with 21 researchers who have visualized biological networks. I detailed challenges in reference to data and analysis, visual representation and interpretation, and limitations of models. As biomedical visualization community moves forward, there is opportunity to improve networks visualizations through closely mirroring experimental designs, clarifying use of secondary data, clarifying meaning of visualization components, and highlighting meaning of network figures.

3. Systematic Review of Biological Network Figures Part I: Visual Encodings

3.1. Introduction

The goal of the interview study contained in the previous chapter was to understand the range of challenges that researchers encounter when working with biological network visualizations. Although the previous chapter specified a number of formidable challenges, the remainder of this dissertation will focus on visual representation and interpretation of biological networks. In this chapter I begin to feature quantitative research, documenting examples of how biological network visualizations are currently depicted, and later connecting those findings with what was discovered during the course of the interview study.

As the name implies, this chapter is part one of two. Chapter 3 details the protocol used to perform this systematic review of figures, and rudimentary results describing, among other things, frequency of visual encodings in bioinformatics literature. Chapter 4 builds on the results of Chapter 3 through re-analysis of the same dataset in context of graph task taxonomy published by Lee et al [46]. These two chapters, although presented separately, were part of the same research project. They have been separated in order to keep the contents of this dissertation organized. At the end of part two, I will connect the results of both chapters and illustrate how the ability to complete certain graph tasks are related to visual encodings.

3.1.1. A Small Roadmap

The background section sets the stage for the rest of this chapter through introducing related research and defining visual encodings. The method section explains the protocol(s) used in this study. The analysis section contains the results from the study, as well as an explanation of the significance of those results. More specifically, there are three subsections: (I) descriptive statistics, (II) statistical analysis of frequency distributions, and (III) a qualitative assessment of selected figures. The discussion section is used to add additional commentary and detail that are worth mentioning, but not necessarily pertinent to understanding the analysis results.

I will clarify the following terms, as they will be used frequently: data attributes, visual attributes, and visual encodings. Data attributes refer to data associated with a node or edge entity in a graph. Visual attributes refer to the visual properties that may be parameterized to alter its physical appearance. Visual encoding refers to the process of transducing data attributes to visual attributes.

3.2. Background

Even while controlling for the data contained in a biological network, and controlling for the layout used to organize the network, attributes attached to nodes and edges may be conveyed in myriad ways. For instance, the range of values of a node attribute may be depicted through different levels of a linked visual attribute. Some commonly supported visual attributes are color, size, shape, position, pattern, length, and volume. The various ways in which the attributes attached to nodes and edges may be represented are referred to as *visual encodings*. One of the seminal works on interpretation of visual encodings was published by Cleveland et al, which showed that some visual encodings are visually translated more accurately than others [47]. A relevant implication of this work is that even while controlling

for the data contained in a network, and the layout of the network, the visual encodings alone may have a significant impact on the interpretation of the network. Research on pre-attentive processing supports this premise, as there is reliable evidence that certain visual properties may be more or less salient depending on the distractors surrounding it [48]. Furthermore, research on Gestalt effects also support the general premise that the expressibility and effectiveness of network visualization may be affected by the combinations of visual encodings present in a network.

3.2.1. Defining Visual Encoding

In terms of a function, visual encodings use data attributes as the domain, and visual attribute(s) as the range. For example, if one has the following dimension of data (the domain of the function) [1, 1, 2, 3, 6, 10, 100], and would like to visually encode those values as some visual attribute (e.g. the area of a node), the following example function may describe this encoding:

Equation 1 – Example function illustrating range values as a function of domain values.

The input to the function above would be the values listed in the dimension of data [1, 1, 2, 3, 6, 10, 100], and the output would be a corresponding radius size that would transduce the domain values to corresponding radius values that would represent the area of a circular node, which would be [0.56, 0.56, 0.80, 0.98, 1.38, 1.78, 5.64]. Visual encoding may be used to transform data values into a variety of visual attributes. To clarify, the encoding is the function itself, mapping the data attributes (domain) to the visual attributes (range). Furthermore, the function mapping data attributes to visual attributes could be an identity function, or the input to another function of the input. To provide a brief, non-exhaustive list of node attribute examples, data may be attached to node size, node color, or node shape. A list of the visual attributes and encodings used in this study are detailed in Figure 5.

Given that the same data values may be transduced, or visually encoded, into a variety of visual representations, it is natural to hypothesize that not all visual representations are equally effective or appropriate. This research question is thoroughly examined in Chapter 7. However, in the following study detailed in this chapter, the objective is to obtain an understanding about how visual encodings are currently used in bioinformatics network figures.

3.2.2. Types of Visual Encoding

Visual attributes may be encoded in an assortment of ways. Similar to the concept of data types in statistics (e.g. numerical, interval, ratio, nominal, etc.), visual encodings may also be embodied in varying visual encoding data types. The function relating area (domain) and radius (range) in Equation 1 is a quantitative encoding. However, if the output of the function had been hue (color), for example, the encoding would be categorical (i.e. nominal). The distinction between types of visual encodings is essential, as I will demonstrate later in this chapter. Furthermore, the mappings binding node or edge attributes to visual attributes are presumed to maintain the property of bijection (i.e. a one-to-one mapping of elements between data attributes and visual attributes). This is covered further in Section 6.1. However, in practice, visual encodings are not necessarily assigned in a systematic manner.

3.3. Method

The method consists of five steps, with the former steps resembling that of a systematic review. Figure 4 provides an overview of the figure selection and data collection process. The protocol is described below:

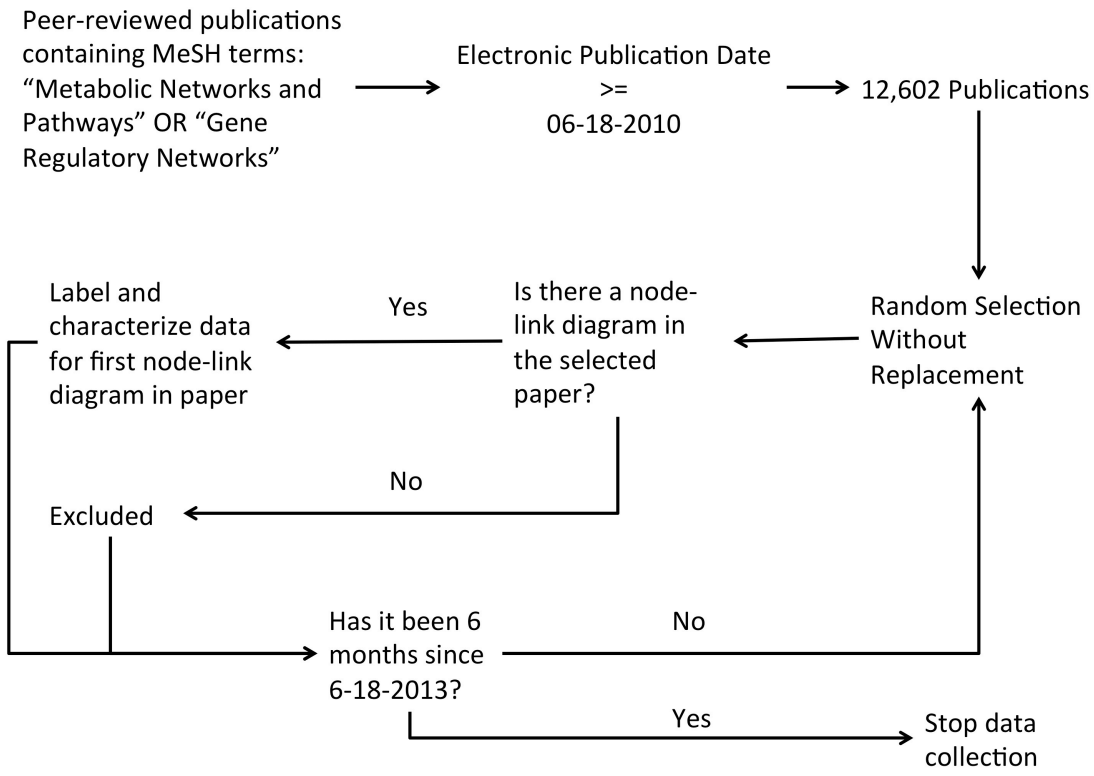
The following inclusion criteria were used to assess the validity of selected publications:

- The selected publication must be associated with either (or both) of the following MeSH terms: "Metabolic Networks and Pathways", "Gene Regulatory Networks"
- The selected publication must have been published on or after June 18, 2010.

There was a single exclusion criterion: the selected publication must contain a node-link diagram. If a node-link diagram was not present in the publication, the paper was removed from the pool of selectable papers.

Using the protocol described above, a PubMed Central query was conducted using manually selected Medical Subject Heading (MeSH) terms. MeSH is a controlled vocabulary used for indexing PubMed articles by the National Library of Medicine. These results were additionally filtered to only include peer-reviewed journal articles published on or after June 18, 2010 – yielding 12,602 scientific papers. From these 12,602 papers, 246 papers were randomly selected without replacement for analysis (although only a portion of these met both the inclusion and exclusion criteria). Due to the labor-intensive process of reading and recording data from selected papers, this selection process was carried out for a period of 6 months. The first node-link diagram figure in the randomly selected paper was characterized – if the paper does not contain a node-link diagram, sampling is performed again (until a paper with a node-link figure is found).

Figure 4 – Overview of selection of figures and data collection.



Step 1: Figure Selection

Using the NCBI MeSH terms, scientific papers classified under either of the two terms "Metabolic Networks and Pathways" or "Gene Regulatory Networks", and also published within the past three years, were identified. From this pool of eligible papers, papers were randomly selected without replacement. When a paper was selected, the first instance of a node-link diagram (if any) was used for measurement.

Step 2: Labeling Figures

Figures were associated with a label describing "type". The term used in the figure caption to describe the figure was the same term I used to label the "type." If more than one term was used, I chose the term that appeared most often in the paper. There were no situations where a term was not associated with a figure. However, after data collection, it came apparent that several terms might have been used in various publications to explain the same concept.

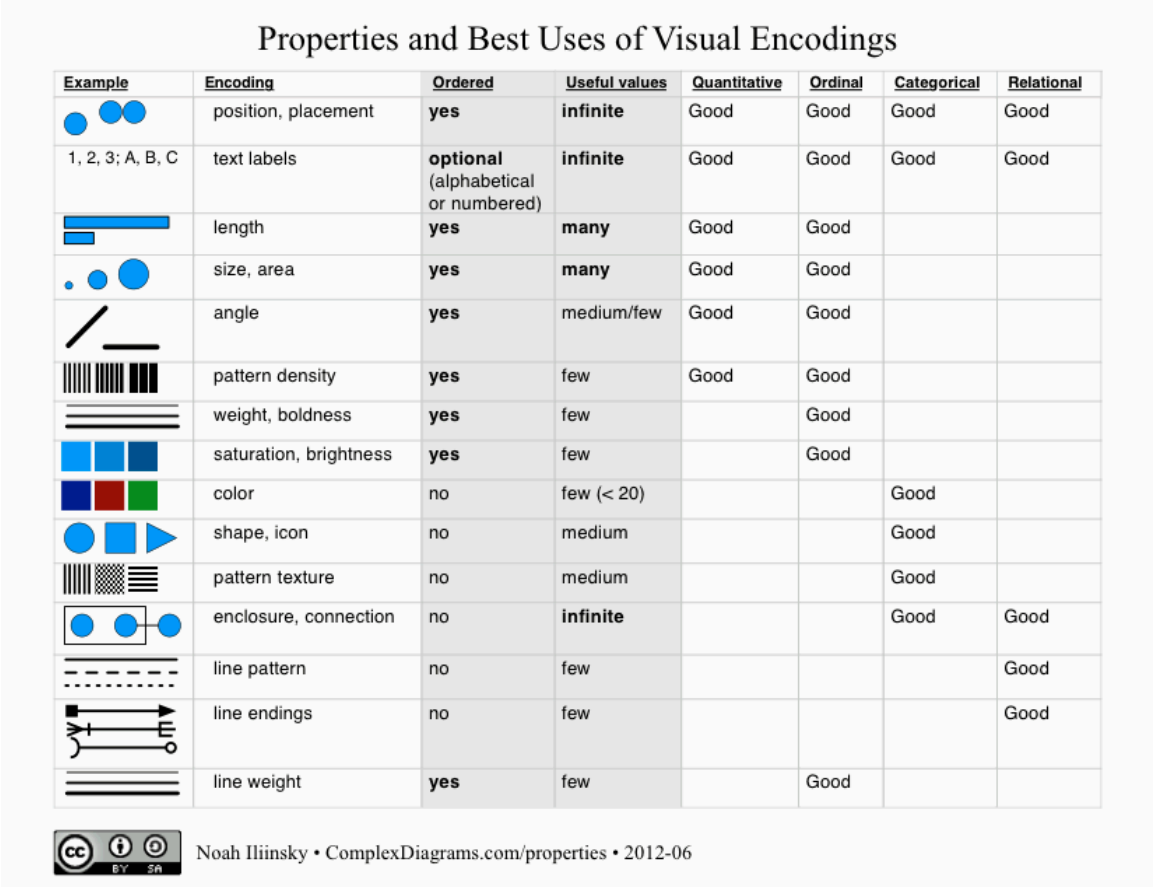
Of the 246 papers that were selected, only 96 (39%) of them contained a node-link figure. Of the 96 node-link diagrams, the figure captions associated with the figures described 34 of them as "networks", 27 of them as "pathways", 20 of them as "schematics", 14 of them as "models", and only 1 of them as a "map". Although some of the results from the previous chapter suggest that researchers who work with biological networks sometimes use the listed terms interchangeably, these listed terms were used as class labels to categorize figures for reasons of practicality and reproducibility. In the case where a figure caption did not explicitly state the class of the figure, the entire paper was reviewed to find the term used to describe the content of the figure.

The class labels for schematics, models, and map are collectively referred to as conceptual diagrams. Although the frequency count of conceptual diagrams is quite high, only analysis of networks and pathways will be detailed. The reason for this is that conceptual diagrams afford the license to present information in a wide variety of ways (according to desired emphasis, or taste). Conceptual diagrams may be designed to present an overview of the paper, or provide information about some workflow, and thus is not the focus of this chapter.

Step 3: Characterizing Figures

The visual encodings and properties listed in Figure 5 were used to characterize selected figures. To be systematic, nodes and edges were independently evaluated for each encoding in Figure 5. As a result, the raw data contains a number of encodings that have "zero" values (e.g. counting the number of instances of "line endings" appearing on nodes). These nonsensical values were later omitted during analysis. Visual encodings were labeled by type: quantitative, ordinal, categorical, or relational [49].

Figure 5 - A table representing various visual encodings and their associated data type properties (compiled by Noah Ilinsky) [50].



Properties and Best Uses of Visual Encodings

Example	Encoding	Ordered	Useful values	Quantitative	Ordinal	Categorical	Relational
	position, placement	**yes**	**infinite**	Good	Good	Good	Good
1, 2, 3; A, B, C	text labels	**optional** (alphabetical or numbered)	**infinite**	Good	Good	Good	Good
	length	**yes**	**many**	Good	Good		
	size, area	**yes**	**many**	Good	Good		
	angle	**yes**	medium/few	Good	Good		
	pattern density	**yes**	few	Good	Good		
	weight, boldness	**yes**	few		Good		
	saturation, brightness	**yes**	few		Good		
	color	no	few (< 20)			Good	
	shape, icon	no	medium			Good	
	pattern texture	no	medium			Good	
	enclosure, connection	no	**infinite**			Good	Good
	line pattern	no	few				Good
	line endings	no	few				Good
	line weight	**yes**	few		Good		

Noah Iliinsky • ComplexDiagrams.com/properties • 2012-06

3.4. Analysis

This section presents the results of the analyses that were performed on this data. First, I will present some of the count data (section I), followed by a brief statistical analysis comparing the distributions of those data (section II). After an explanation of the statistical results, I continue on to view exemplary figures found across various bioinformatics publications (section III).

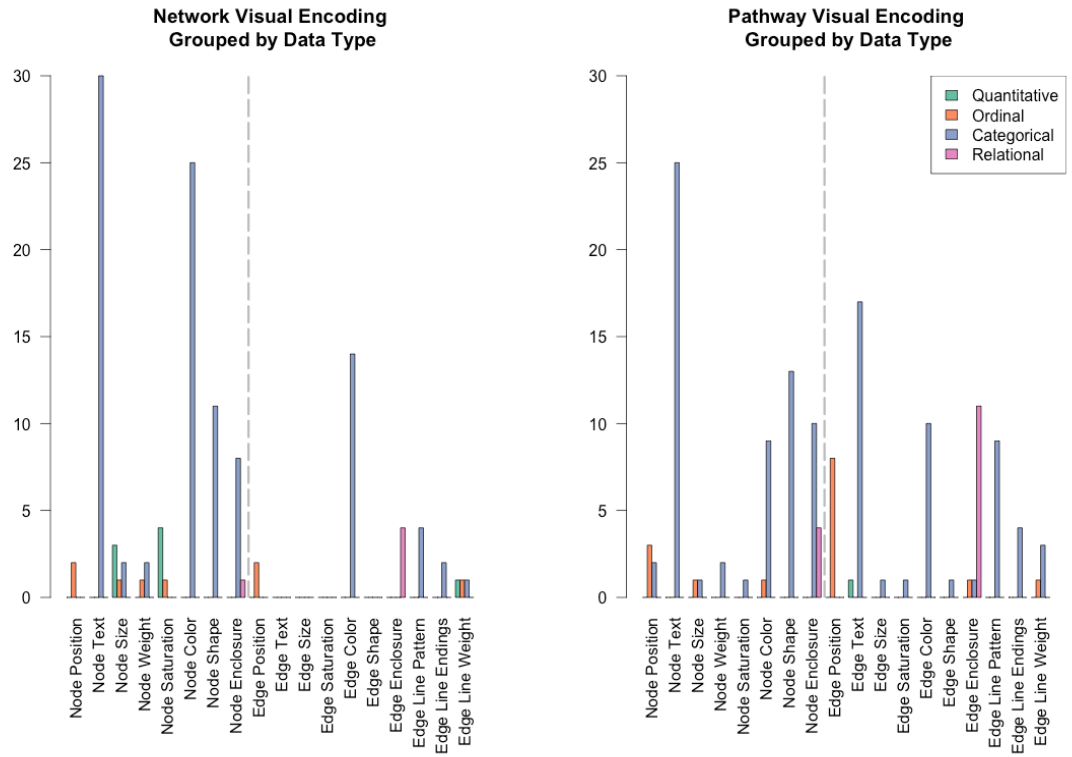
3.4.1. I. Descriptive Statistics and Counts

The measured visual encoding attributes for networks and pathways are presented in Table 2. The same information in Table 2 is presented in a slightly different perspective in Figure 6, which facilitates a visual comparison between encoding frequency counts between networks and pathways. Examination of Figure 6 illustrates clear differences in frequency between networks and pathways, itemized by nodes and edges. For instance, node color is notably higher in networks, and edge text is notably higher in pathways. Another observation is that edge position is encoded ordinally in pathways more frequently than in networks.

Table 2 - provides an overview of node versus edge encodings, separated by pathway versus network. Encodings that had a frequency of zero across nodes/edges and pathway/network have been removed from the table.

Encoding	**Nodes (Pathway/Network)**	**Edges (Pathway/Network)**
Position	5/2	8/2
Text	25/30	18/0
Size, Area	3/8	1/0
Weight, Boldness	2/3	0/0
Saturation, Brightness	1/5	1/0
Color	11/27	11/14
Shape, Icon	15/12	2/0
Enclosure, Connection	14/9	13/4
Line Pattern	0/0	9/4
Line Endings	0/0	4/2
Line Weight	0/0	5/3

Figure 6 - A side-by-side comparison of the frequency of visual encodings between networks and pathways, and the data type by which the encoding appears (quantitative, ordinal, relational, or categorical). In each bar plot, the former half of the plot represents node encodings, and the latter half represents edge encodings (as denoted by the vertical dashed line).



3.4.2. II. Comparing distributions of node and edge encoding counts between networks and pathways

To determine whether Networks and Pathways are detectably different types in figures from bioinformatics literature, I perform a fisher exact test to compare the distributions of the counts of the network and pathway types. Fisher's Exact test is ideal for this research question as it is designed to use binary count data, and can accommodate a relatively small sample size (whereas the common alternative, Chi-Square test is better suited for datasets with a larger sample size). As a note, there are additional statistical results for a test of unequal variance and ANOVA available in Appendix B and C, respectively.

Table 3 and Table 4 present frequency data used to perform a Fisher's Exact test. Table 3 compares counts from the node attributes between the network and pathway classes, and Table 4 compares counts from the edge attributes between the network and pathway classes. Using an alpha level of 0.05, the null hypothesis was accepted (although marginally) for a difference between node attributes, but rejected for a difference between edge attributes.

Table 3 - A Fisher's Exact test was run on the following table of counts for nodes:

Node Encoding	**Network**	**Pathway**
Position	2	5
Text	30	25
Size, Area	8	3
Weight, Boldness	3	2
Saturation, Brightness	5	1
Color	27	11
Shape, Icon	12	15
Enclosure, Connection	9	14

H_0: The visual encoding frequencies between nodes in networks and pathways = 0.
H_a: The visual encoding frequencies between nodes in networks and pathways 0.

The null hypothesis was that there is no difference between the network and pathway types in context of the collected node encoding count data. The alternative hypothesis is that there is a difference between the network and pathway types (two sided). The test was run with the standard alpha-level of 0.05. After running the Fisher's Exact test, the resulting p-value was determined to be 0.07. When strictly interpreted, this means the null hypothesis that the networks and pathways are not encoded differently was accepted. However, considering that the sample size is small, and that the p-value is close to 0.05, the results are insufficient to truly rule out that there is not difference. Although it may seem odd to include the results of a negative statistical test, the reason for doing so will become more clear at the end of this section.

Table 4 - A Fisher's Exact test was run on the following table of counts for edges:

Edge Encoding	**Network**	**Pathway**
Position	2	8
Text	0	18
Color	14	11
Enclosure, Connection	4	13
Line Pattern	4	9
Line Endings	2	4
Line Weight	3	5

H_0: The visual encoding frequencies between edges in networks and pathways = 0.
H_a: The visual encoding frequencies between edges in networks and pathways 0.

The null hypothesis was that there is no difference between the network and pathway types in context of the collected edge encoding count data. The alternative hypothesis is that there is a difference between the network and pathway types (two sided). The test was run with the standard alpha-level of 0.05. After running the Fisher's Exact test, the resulting p-value was determined to be 0.003. When strictly interpreted, this means the null hypothesis that the distribution of visual encodings between networks and pathways are different, was rejected. Again however, we must recall that the sample size is small when interpreting this result.

When interpreting these two statistical tests in context of each other, the results suggest that there may not necessarily be visual encoding differences among nodes between networks and pathways, but that there are certainly visual encoding differences among edges between networks and pathways. This finding allows the generalization that pathways are more "edge-focused" than networks, but it does not necessarily follow that networks are more "node-focused". This is a subtle, but important, distinction.

3.4.3. III. Visual Assessment of Exemplar Figures

To make the implications from the results of the statistical tests (explained in section II) more concrete, this section provides a handful of model figures that visually depict the differences between networks and pathways. Let us closely examine some of the figures that were visually analyzed during the course of this study.

3.4.3.1. The pathways emphasize edges, whereas networks emphasize nodes:

Pathway figures are much more likely to encode information using edge attributes, whereas network figures tend to be less "edge-focused". Edge positions are encoded more often in pathways than in networks, suggesting that there may be some sense of order or temporal property to the pathway. On the other hand, networks are more frequently encoded with node information, suggesting de-emphasis of the information contained in edge elements in networks. Node color, although used in pathway depictions, are used more frequently in networks than in pathways. Unlike pathways, networks sometimes visually encode node size and node saturation quantitatively. A surprising observation is that visual encodings are most often encoded in a categorical (i.e. nominal) manner, rather than quantitative, ordinal, or relational (please refer to Figure 6). This observation is surprising since it implies that networks and pathways may be designed to highlight a number of groups or categories. It also implies that information may be lost in situations where quantitative, ordinal, or relational data are reduced to categorical encodings.

As Figure 6 shows, node-link diagrams in the network group emphasize the nodes in a network, and the visualizations in the pathway group emphasize the edges in a network. This claim is not only supported by the observation that frequency of edge text is minimal for networks and markedly higher for pathways in Figure 6, but also with Figure 7 through Figure 13. When a pathway is being depicted, the entities involved are important, but the focus of the figure is on the nature of the relationship between entities rather than the entities themselves (please see Figure 10 - Figure 13). On the other hand, when a network is being depicted, the focus of the figure is around the entities and the relationship between entities provide supporting context, rather than serve as the primary point of emphasis (please see Figure 7 - Figure 9). The distinctions included in the figure captions below have design implications for graph algorithms and visualizations, as elaborated on in the discussion section of this chapter.

3.4.4. Exemplar Network Figures

The general properties of Figure 7 - Figure 9 are that nodes are always labeled, color encoding seems to be used to denote groups (when used), and edge encodings are not only minimally used, but are used in support of the information conveyed through node encodings.

Figure 7 shows connections between nodes, and only provides (gene name) labels for the node elements. Figure 7 does not focus on the connections themselves, rather that there are several clusters of genes that are more connected than the rest. Figure 8 also focuses on connections between nodes and only provides labels for node elements. Although this figure also highlights interconnectivity and clustering between nodes, color and edge thickness are used to emphasize certain groups, and the connections therein. The edges that are stressed in this figure seem to be intended to support node groupings, rather than act in their own service. Figure 9 contains many more nodes and edges than Figure 7 and Figure 8. The visual properties of the edges that connect the many nodes in this figure are uniform in color and thickness. Although nodes are uniform with respect to size, each node is labeled and assigned a color that assigns it into one of several mutually exclusive groups.

3.4.5. Exemplar Pathway Figures

The general properties of Figure 10 - Figure 13 are that nodes are sometimes colored and labeled, and are typically encoded in the service of highlighting edge properties. Edge encodings are rich and complex, although they leave something to be desired—this will be further covered in the discussion section.

Similarly to Figure 7 and Figure 8, there are a number of labeled nodes in Figure 10. However, Figure 10 also contains a variety of edge patterns, embeds direction and order, and even depicts some physical properties of a cell, such as the cell membrane. Relative to Figure 7 and Figure 8, this figure conveys more information about the connections between nodes. Figure 11 clearly emphasizes the properties of the edges connecting various nodes in the image. The red color is used to highlight relevant edges, and thickness is used to convey magnitude of the highlighted property. Figure 12 contains a minimal representation of nodes, edges, and cellular location. The gray background in which the entire graph sits is an abstract representation of the internals of a cell, and the smaller box within the cellular abstraction is another group representing a mitochondrial sub-reaction. Nodes, edges, and cellular location are also depicted in Figure 13. However, this figure contains more explicit annotations about reactions, cellular components, and it uses color to highlight notable sections of the overall reaction.

A noteworthy observation to point out is stark difference in emphasis between Figure 9 and Figure 11. Edges in a network figure (Figure 9) are colored gray to better emphasize the nodes, and in contrast, nodes in a pathway figures (Figure 11) are colored gray to better emphasize the edges.

Figure 7 - A figure example of the "network" type from Clark et al [51]. Notice the labeled nodes and plain edges.

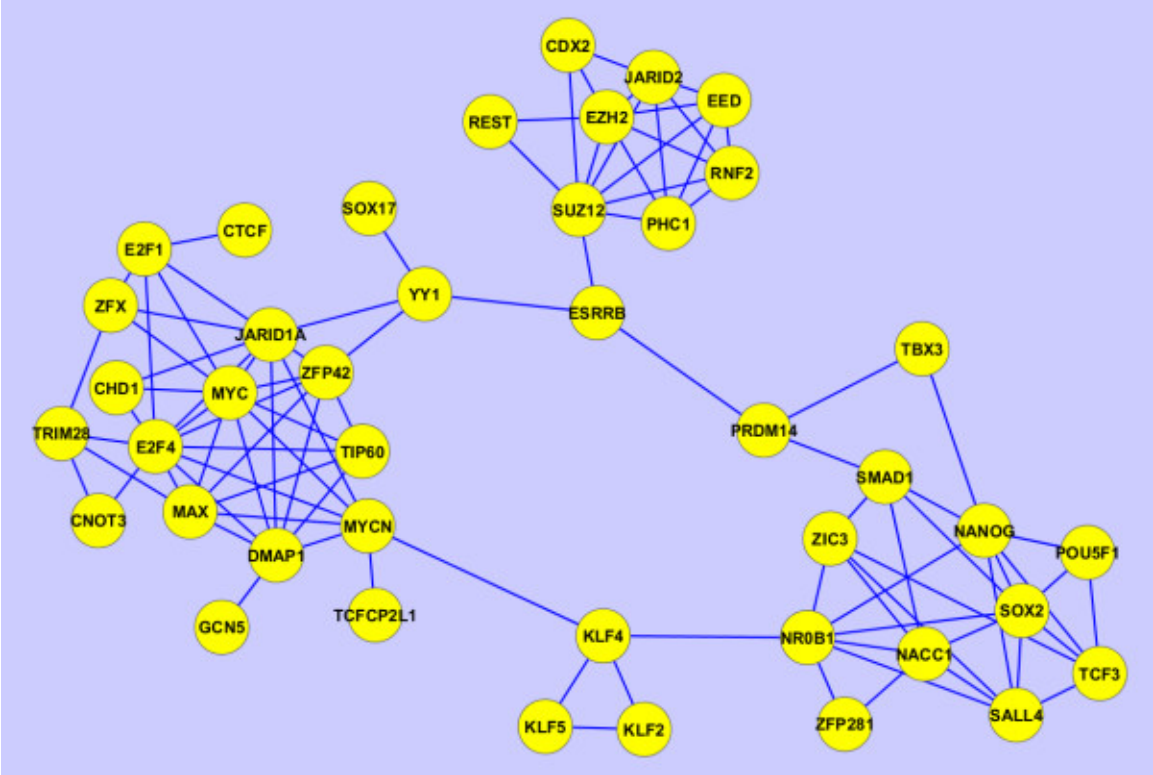


Figure 8 - Another figure of the type "network", published in Lei et al [52]. Observe the colors (indicating groups) and labels on nodes, and minimal use of edge encodings. Edge width is used merely to reinforce node groups signified via color.

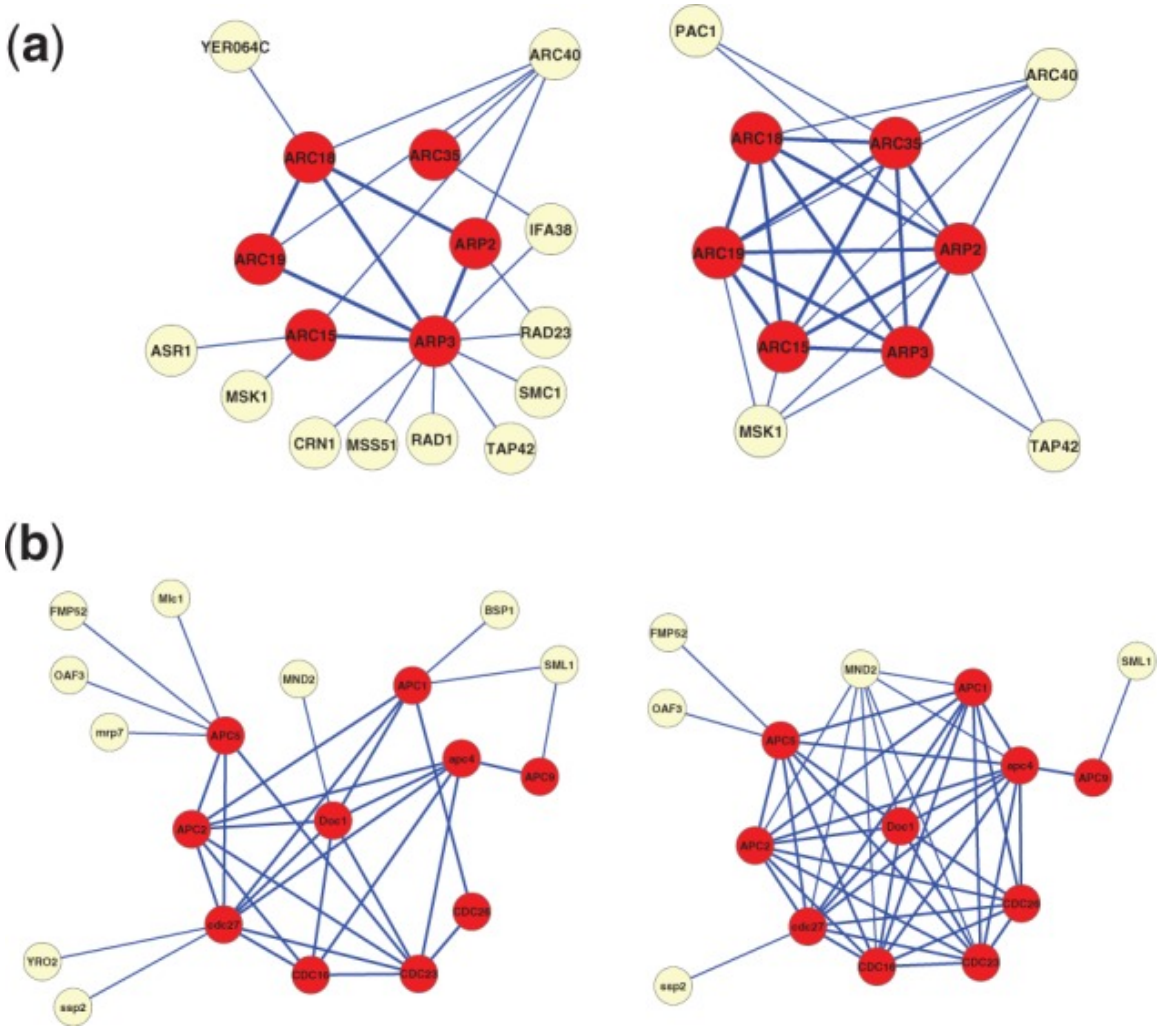


Figure 9 - Another example of a "network" figure, from Finka et al [53]. This figure contains the largest network among those presented in this set. Color is used to signify groups, and nodes are labeled. However, edges are simply used as a subdued backdrop for interconnectivity.

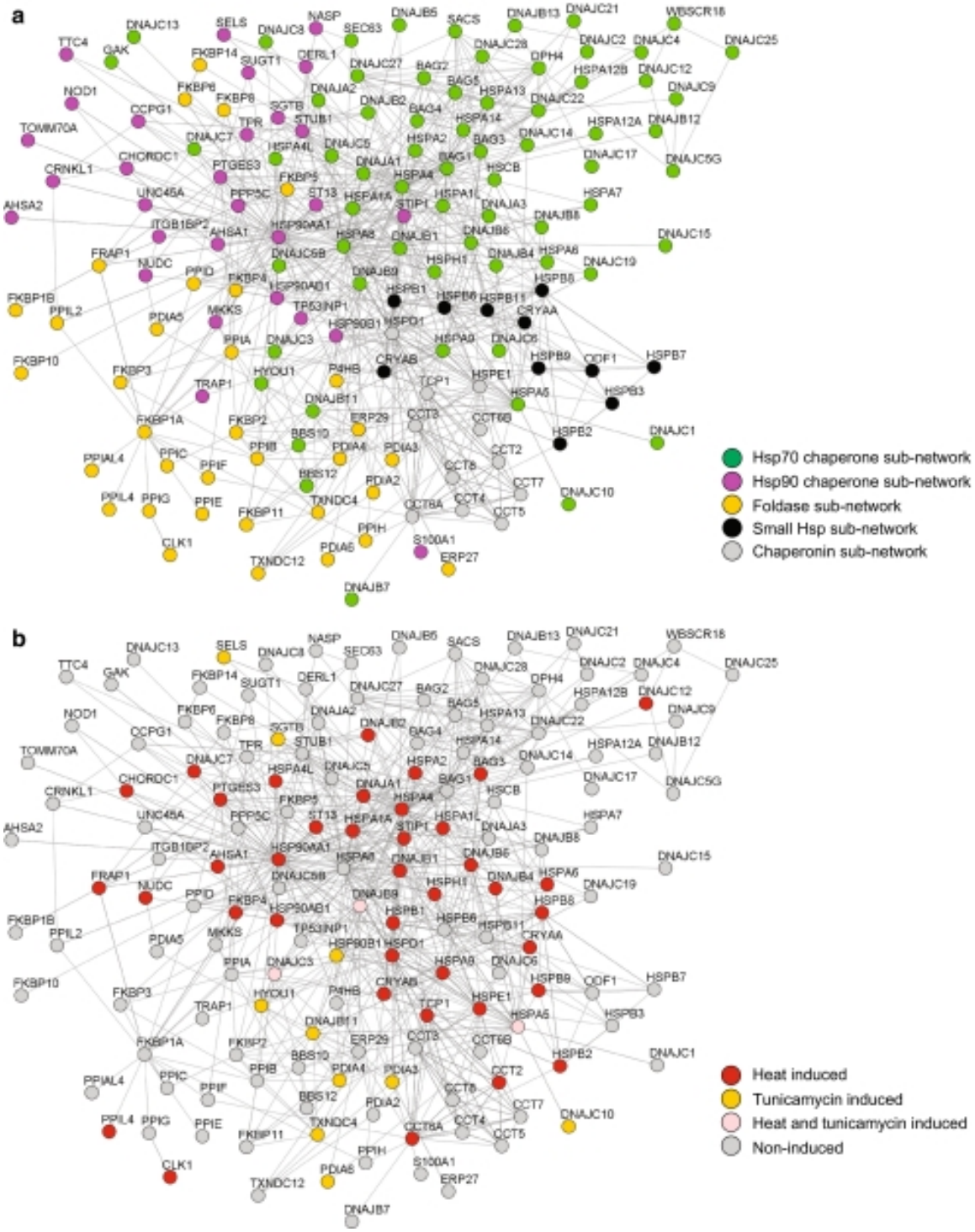


Figure 10 - This figure is an example of a "pathway" figure from Bakir-Gungor et al [54]. The depiction represents components of a cell and use directed edges with multiple patterns. Although nodes are color encoded and labeled, they are depicted in support of the information presented in the edges (e.g. landmarks in a pathway). This is not to say that the nodes are unimportant, but rather that the information the figure seems to convey is primarily contained within the edges, and operate in conjunction with the information provided by the nodes.

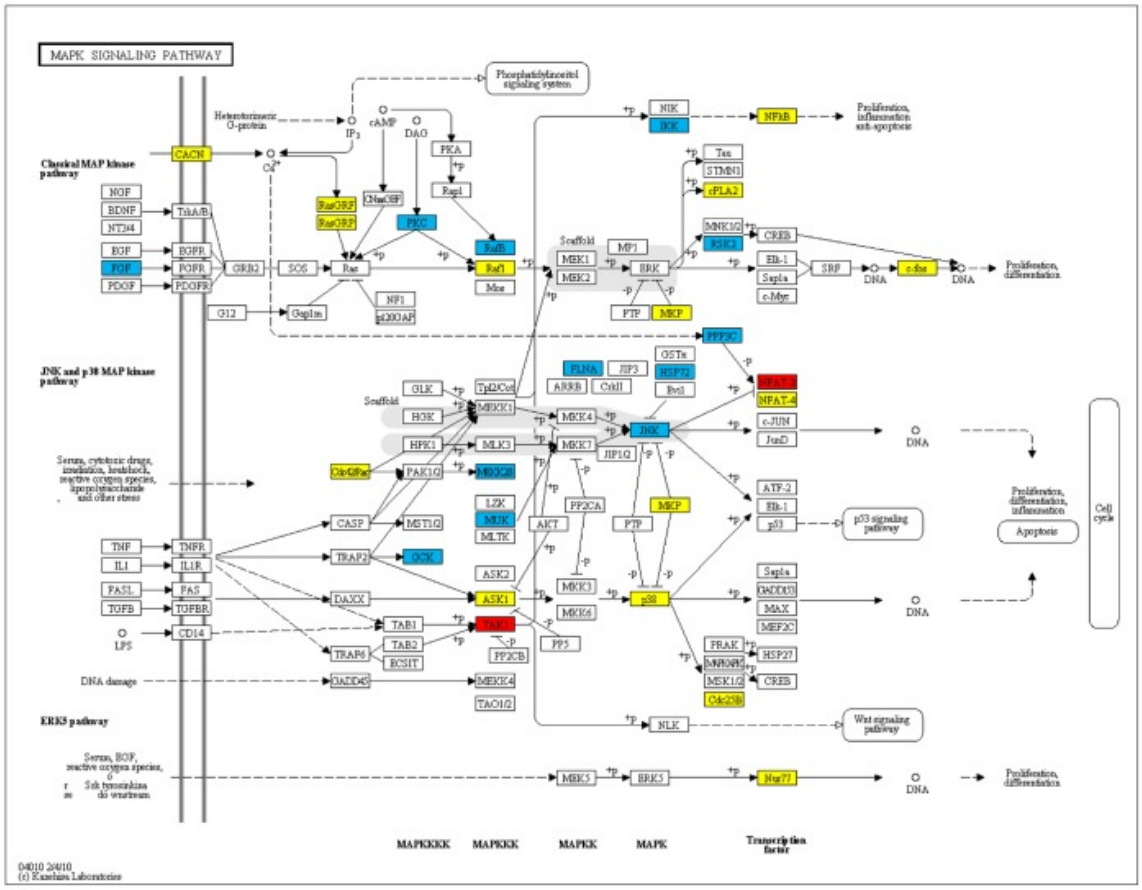


Figure 11 - Another example of a "pathway" figure, from Yamada et al [55]. Network figures typically use minimally encoded edges, usually in a gray hue to de-emphasize them in a figure. In this figure, nodes are now de-emphasized in a gray hue, and edges are encoded with color and thickness.

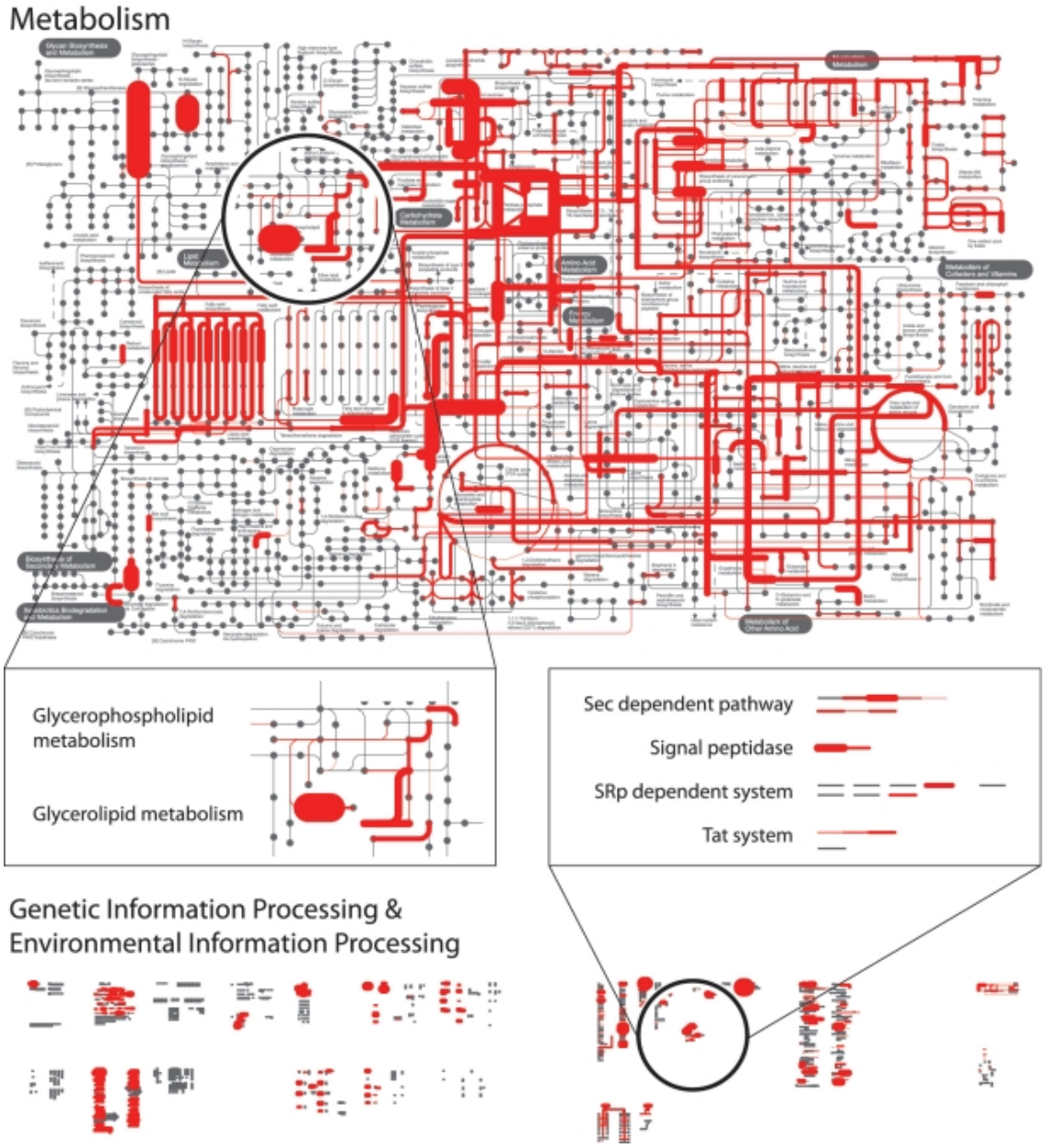


Figure 12 - Another example of a "pathway" figure from Kailavasan et al, although less intricately detailed relative to the rest of the pathway figures included in this chapter [56]. The nodes in this graphic depict the steps in the glycolysis pathway.

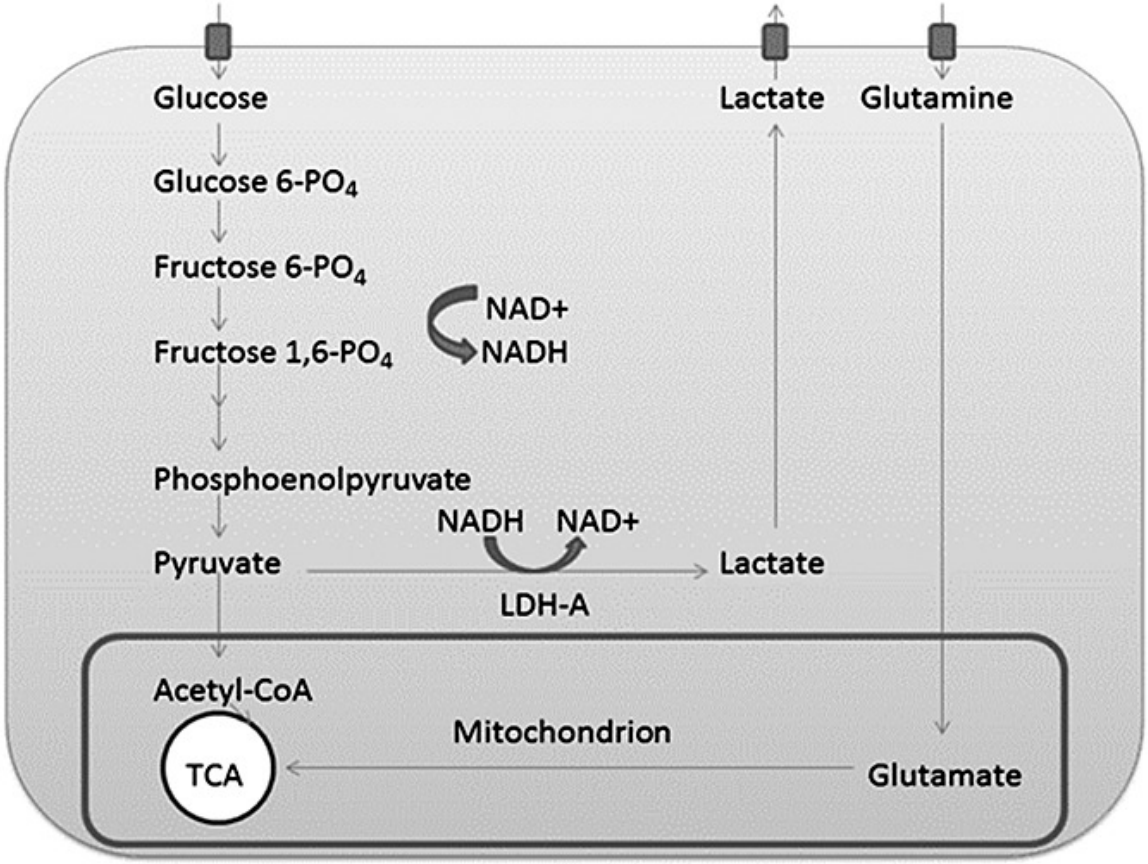
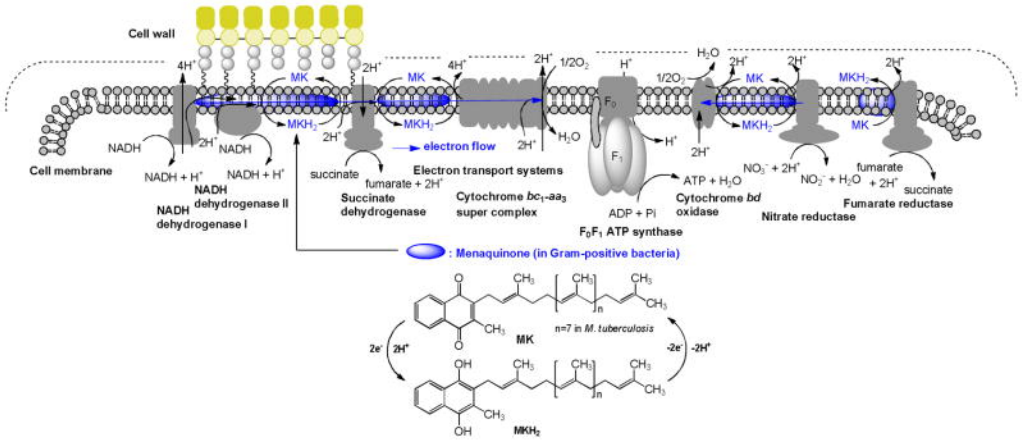


Figure 13 - This is another example of a "pathway" figure from Debnath et al, and in contrast to Figure 12, more complexly detailed [57]. Note the well-defined portrayal of a cell membrane and the intricate edge relationships supporting the contextualization of chemical reaction information, tending from the left side of the image to the right side, while also stressing the cyclical nature of the process.

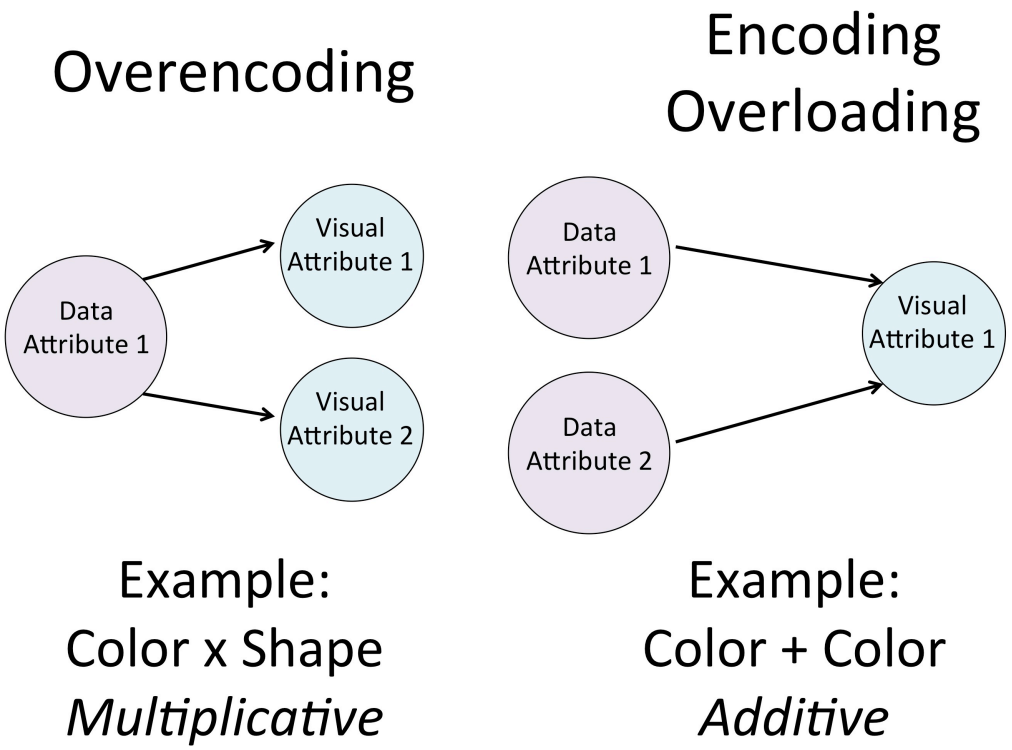


Depending on the level of depth one may be interested in, edge representation techniques can also exemplify ambiguity—for example, when a reaction step is mediated by yet another, independent reaction step. An instance of this is found clearly in Figure 12. In Figure 12, the step that connects "pyruvate" to "lactate" is mediated by another step that oxidizes "NADH" to "NAD+". Presumably the "pyruvate" to "lactate" step takes place simultaneously as "pyruvate" is also yielding "Acetyl-CoA", but that is unclear without further reading about Glycolysis. The same representation may be used to depict reactions where one step either precedes or follows another (rather than transpire concurrently). Some participants in the interview study detailed in Chapter 2 also expressed frustration about the inherent ambiguity of concurrently depicted edges. A handful of edges are also depicted as "hyper-edges", meaning an edge that originates as a singular edge eventually forks into two or more edges downstream—thus, a single source node may connect to several target nodes via a hyperedge (a clear example of a hyperedge may be found in Figure 18, presented in a later section of this chapter). Although some may use hyperedges as a drawing convenience (or optimization), findings from the interview study detailed in Chapter 2 revealed that in hyperedges also characterize simultaneous decomposition of a molecule, or denote a chemical by-product.

Overencoding and Overloading Visual Attributes:

In some visualization scenarios, there are not enough visual attributes to encode all of the desired (and sometimes required) information. Although there is too much information in a network, there are two forms by which a network may contain too much information to display. For clarity, let us define two terms: *overencoding* and *encoding overloading* (also depicted in Figure 14).

Figure 14 – A visual depiction of overencoding and encoding overloading.



I define *encoding overloading* as when the same visual attribute is used to encode information for multiple dimensions of data in the network. For instance, one might imagine using color nominally to define various edge groups, while simultaneously using a divergent color scale to convey correlation strength of edges in the same network—in other words, the same visualization has color used in two different ways. In this sense, the term "overloading" is used in the same manner that the term would be used in computer science, to denote a scenario where multiple copies of a computational method are referenced under the same name and co-exist in the same scope. In contrast, I define *overencoding* as when two distinct encodings are combined, in the equivalent of a cross product operation, to increase the set space of depictions for classes contained in a network. For example, if one wants to depict 20 genes in a network, then the cross product of 10 color hues and 2 shapes would yield 20 visually discrete elements. Since no verified examples were found in the corpus of sampled figures, encoding overloading is defined merely to clarify and distinguish it as distinct from encoding overloading.

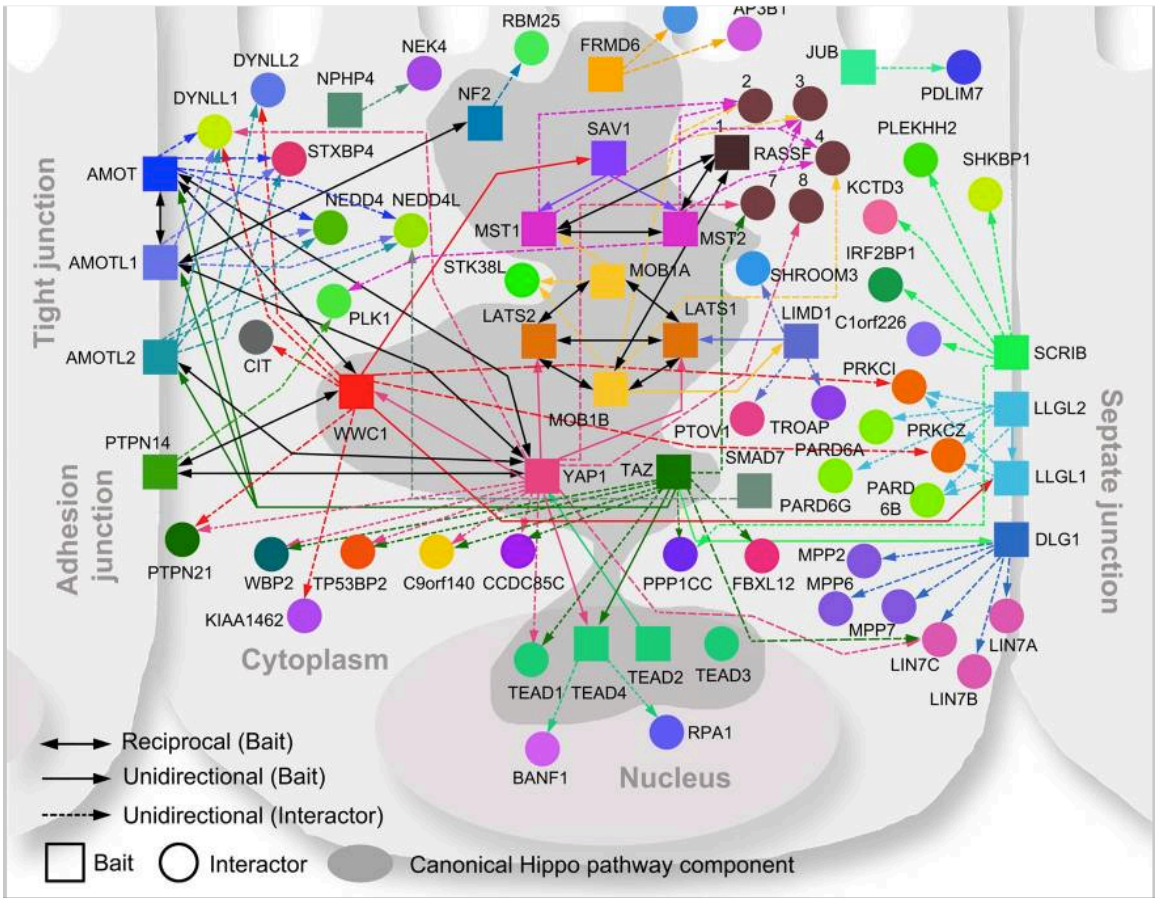
When overencoding is present, there is risk of reader misinterpretation whereby the reader may be unable to visually discern discrete entities, confounding the meaning of each employed visual attribute, or possibly even the combination of visual attributes if the resulting icon is reminiscent of another that is commonly used (e.g. when overencoding color and shape, one may misinterpret a red octagon to signify stopping). Research in visualization and psychology has shown that overencoding burdens the readers' cognitive

and perceptual capabilities [49]. This practice of overencoding forces the reader to perform "conjunction search" when cognitively decoding the visualization. "Conjunction search" refers to the process of visually searching for visual entities that satisfy criteria across multiple visual channels—for instance, searching for red circles in a depiction of circles, squares, and triangles, each of which can be red, green, or blue [58]. This process is recognized in visualization literature and has been shown to be ineffective at scale because users cannot store the interpretation of the visual set space in their iconic (i.e. working) memory. In fact, Figure 15 below directly violates many of the best practices for graph readability [30]. In Figure 15, thirteen colors (hues) are used to illustrate various nodes and edges. Nodes are further categorized by shape, denoting whether the node is an "interactor" or "bait". Furthermore, edges are independently encoded with three patterns that could emit from any of the nodes.

However, the misinterpretation risks associated with encoding overloading are distinct from the type of misinterpretation that may occur due to overencoding, which may be ascribed to confusion due to comingling or overlapping ranges values used for visual attributes.

In contrast to encoding overloading, certain biological graph images found in literature overencoding visual attributes by assigning multiple sets, ranges, or discretized values to a single visual attribute—this can be seen in Figure 15.

Figure 15 - Example of overencoding using categorical visual encodings from Wang et al [59].



3.5. Networks and Pathways in Biological Network Visualization Tools or Resources

In order to further support the claims in this chapter, snapshots of results from two popular biological network visualization tools or resources are provided in this section. Figure 16 and Figure 17 are examples of tools that depict networks, and Figure 18 and Figure 19 are examples of tools that depict pathways. To provide some context, there are a total of 547 recognized pathway resources, and 202 recognized pathway visualization tools (as of September 13, 2016) [40].

In reference to Figure 16 (GeneMANIA), the results show BRCA2 at the center of the layout, along with other genes that are presumed to be relevant. It is unclear whether the sizes of the nodes reflect properties. Typically, in depictions such as this, one might expect the node size to be proportional to degree centrality (the number of edges the node connects to). Additionally, the edges in this figure are encoded with categorical information about the "type" of edge (via hue). The network in Figure 17 (STRING) is accompanied by a report (not shown) that explains the function of BRCA2, and also provides a line of reasoning for why the other (ostensibly relevant) nodes are presented along with BRCA2. Both the nodes and edges use color (hue) to visually encode categorical information. In Figure 18 (Reactome), selecting an edge or node element provides the user with additional information, presented in the inspector box below the pathway diagram. For instance, Reactome states the following for the highlighted hyperedge in the figure: "PPP5C-mediated dephosphorylation of TP53BP1 serine residues S25 and S1778 contributes to dissociation of TP53BP1 from DNA doubles strand break (DSB) sites and termination of DSB repair (Kang et al. 2009)". When node elements are selected, the inspector box provides cross-links (i.e. linking to a URI or external identifier in separate databases) to biological resources where entries for entities are stored, such as UniProt, or Protein Data Bank. Similar to the other tools in this section, the information conveyed through the Reactome tool seems to be primarily categorical information, although it is possible to obtain more detailed quantitative information through the inspector window. Figure 19 (KEGG) provides some notion of orientation as certain steps of the reaction lead into other pathways (e.g. "Mismatch Repair"). Edges also possess labels, "+p" or "+u", and come in varying patterns to denote differing classes of edges. Once again, the information attached to the edge elements is categorical information, and are expressed through a categorical visual encoding (pattern).

Figure 16 - A snapshot of the query result for the human breast cancer associated gene BRCA2 from GeneMANIA [45].

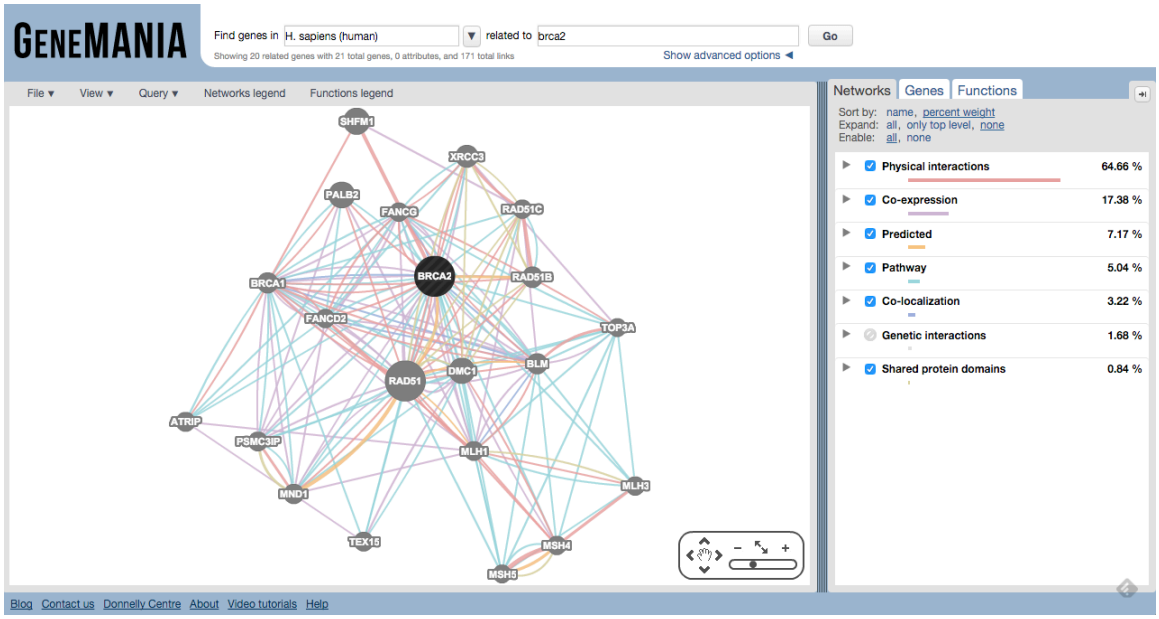


Figure 17 - A snapshot of the same query for the BRCA2 gene in the STRING tool [60].

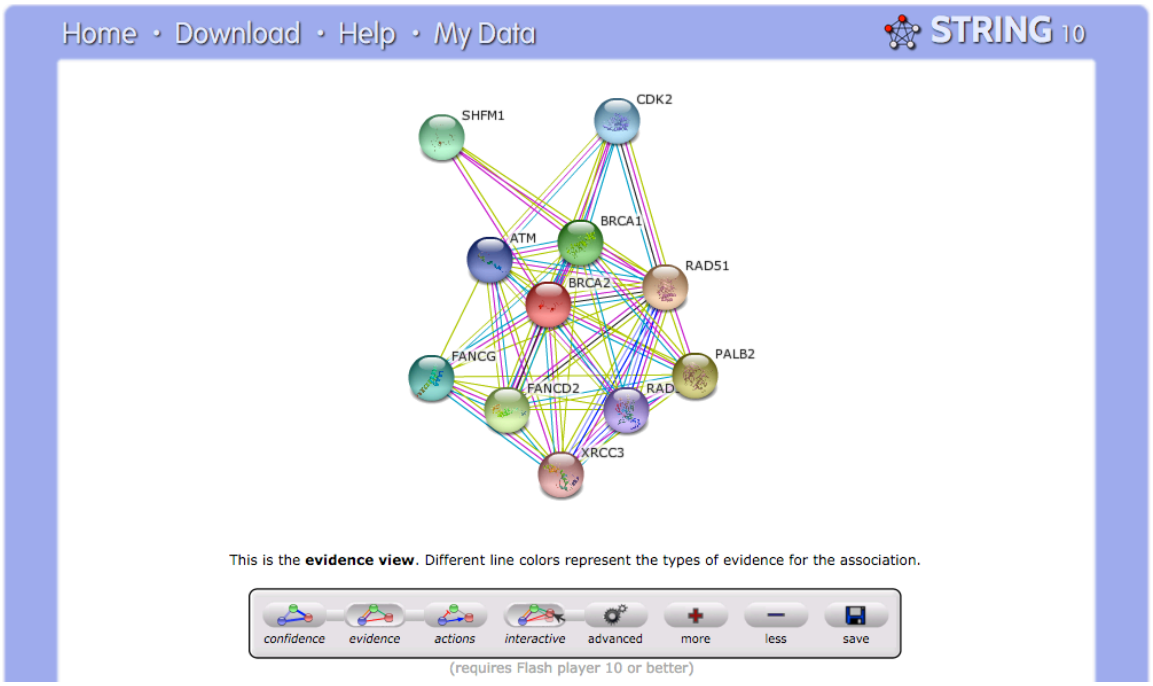


Figure 18 - A snapshot of a component of a pathway that contains BRCA1 from Reactome [61].

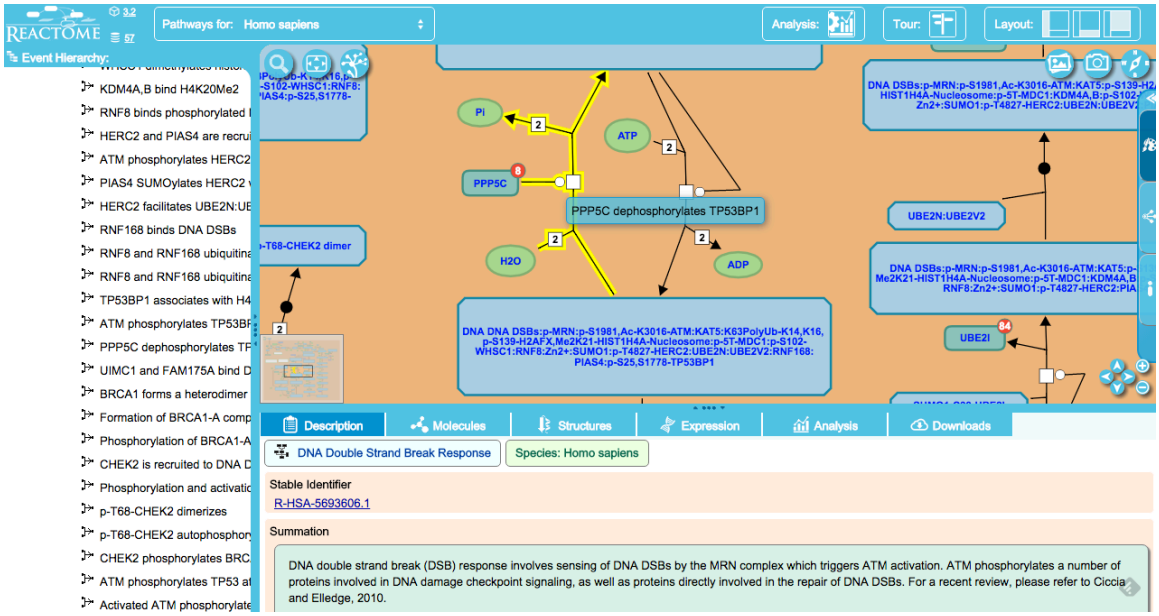
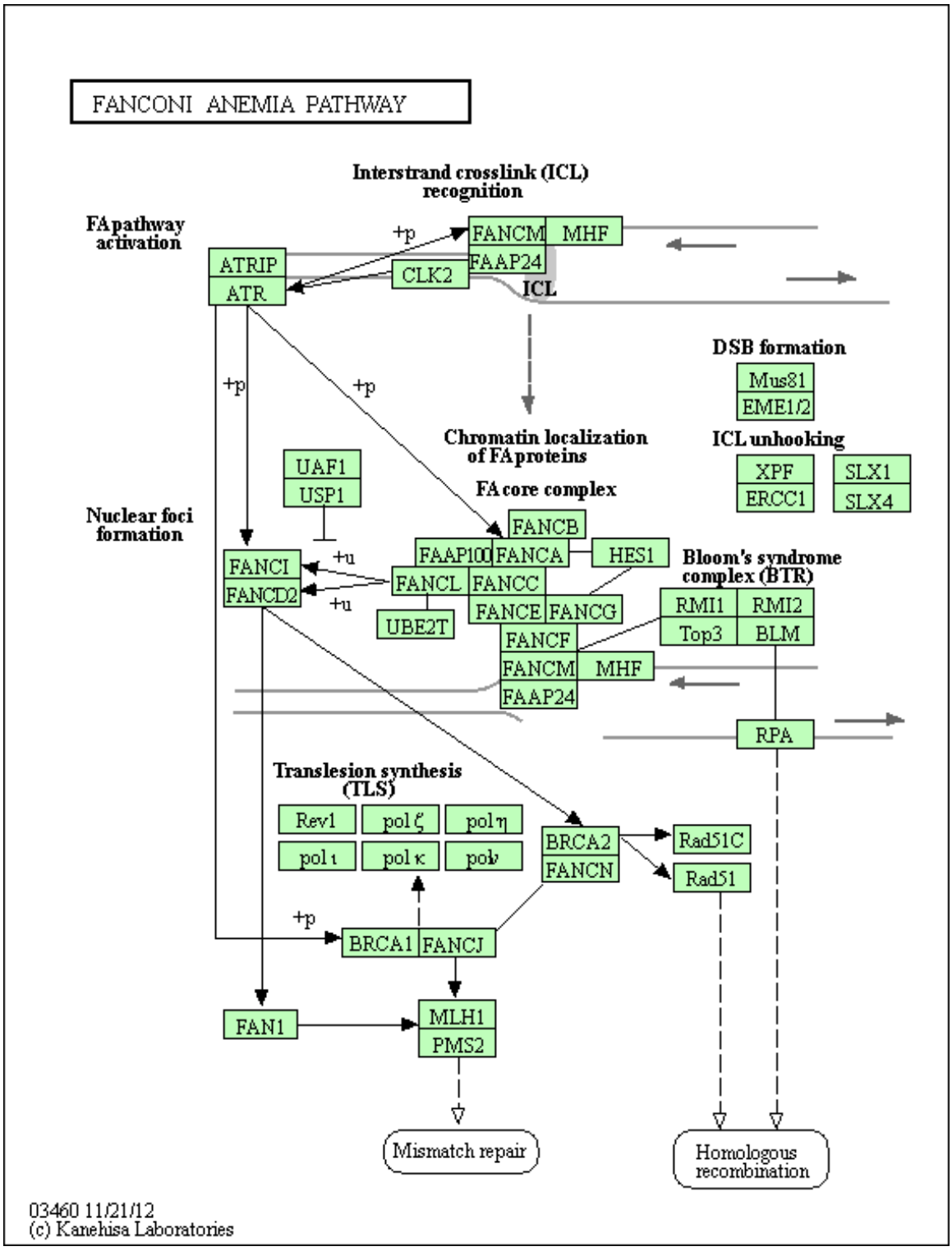


Figure 19 - This figure is an illustration of the "Fanconi Anemia Pathway" from KEGG, which also contains BRCA1 [36].



Even amongst tools, the "network" and "pathway" characteristics previously described seem to be present. It may be the case, however, that figures in bioinformatics literature are

generated using a common set of tools, and are as a result, limited by the tools available to create depictions of pathways and networks.

3.6. Discussion

This section of the paper expands on the findings presented in the previous sections. Figures from various literature sources are presented to help illustrate some claims on commonalities and distinctions.

The finding that majority of visual encodings are of the categorical data type is interesting because it highlights the potential need for a visualization that can depict a large number of categorical entities and relationships—preferably under an organization that lends itself to predictability and interpretability. Both overencoding and overloading have the similar root cause of attempting to depict more information than a visualization is either capable of containing, or reader is capable of deciphering. To resolve this challenge, there are two practical strategies. I do not claim the following list is comprehensive, merely suggested approaches. (1) Present only as much information in a graph visualization as it can contain, or a reader can accurately comprehend. This may mean ruthless omitting unnecessary information completely, or may manifest in the use of a technique such as *small multiples* [62]. An example of a tool that takes this approach is the Cerebral plugin for Cytoscape [43]. (2) Present a higher-level, further abstracted view of the information to fit all of the desired information into visual encodings that respect perceptual limitations.

The fact that the results from the *analysis of biological visualization tools* corroborated the findings of the *visual assessment of exemplar figures* and *descriptive analysis of encoding features* is certainly an exciting finding. However, there may be a sizable amount of confounding. This doubt stems from the fact that there are only a handful of tools that were used to generate the figures obtained from the literature search. If the assessed figures were generated using the same tools, there will obviously be agreement among the results. Moreover, the visual encodings that were empirically captured are merely a quantitative representation of the same information (i.e. figures generated from the same tools). I interpret this triangulation of results to mean that, in the scope of this study, these findings are quite strong. However, in the grand scheme, the applicability of these results outside of biological networks and pathways may be limited.

Given the rise of quantitative data output from next-generation sequencing instruments, one might expect that most entity and relation attributes in a graph are determined quantitatively rather than categorically [63]. However, from the findings of this systematic review of biological graph figures, the quantification of biological information seems to be a mere stepping-stone through which the author determines elements or relationships to be nominally interesting (or not). Even though this may seem counterintuitive, imagine a scenario where researchers are attempting to visualize a mass amount of data once applying one or more statistical tests or bioinformatics methods—a number of these approaches label data points as "significant" or "insignificant"; "enriched" or "not enriched". This is not to imply that statistical tests and bioinformatics methods are coarse tools, but to emphasize that current condition may be such that those methods are used as noise filters, so that we may focus on a manageable and relevant fraction of data—which are eventually visualized, post-analysis, to reflect that information. Although, conceivably, a categorical assignment is less burdensome to interpret than extracting quantitative values as an end-user, the sheer quantity of categorical possibilities is perceptually overwhelming.

Determining which combinations of visual encodings is most "effective" for biological node-link diagrams is a rational proposal for future work. However, further details about how effectiveness may be evaluated, and in what context, remains insufficiently defined. These parameters that are currently missing (i.e. tasks, encoding types, etc.) are further specified in the next chapter. For the sake of this discussion, let us define effectiveness as referring to comprehension of the information encoded in a network visualization (although it may also be defined to encompass saliency of visual attributes, attention span, aesthetics, etc.). Explained in a slightly different manner, let us define effectiveness as the ability for readers to decode the information encoded in a network visualization. Visualization literature has shown that determining which visual channels are more or less salient depends heavily on parameterization of that feature and context of other features (and this idea was strong driving force in determining the experimental design for the study covered in Chapter 7). Still, there are a few accepted generalizations that may be used to guide the development of future tools. For instance, explicit counting is necessary once the number of objects in a group exceeds four, or that there are about 10 colors that can be used (in a nominally encoded manner) before perceptual acuity is hindered. In addition, there is literature that shows that categorical attributes are best visually encoded (in terms of user perception accuracy) using spatial dimensions, color, motion, and then shape [64]. As shown in Figure 6, color is used in networks and pathways as node and edge encodings. Figure 16 shows a snapshot of the widely used tool GeneMANIA [45].

The following claims are made in reference to Table 2. Shape is used in networks and pathways as a node encoding, but much less frequently as an edge encoding (although a few instances were counted for pathways). Spatial position was seldom visually encoded. When spatial position was encoded (as node or edge positions), the encoding was ordinal, rather than categorical. Even still, order is difficult to convey with nodes and links since it is difficult to discern events that happen simultaneously from those that happen independently (but by the same path). Although there are instances where node-link depictions are manually drawn, there are many instances where node-link depictions are computationally generated. In these situations, spatial position is determined by choice of graph layout algorithm (unless there are any spatial constraints applying post-layout), so the spatial dimension is depleted before an author can deliberately encode the spatial dimension. Although it is possible to convey motion statically, this visual dimension was not investigated in this study.

This challenge of visually encoding large amounts of information may sound familiar to the seasoned cartographer. The field of cartography is rich with research about perception, accuracy, and saliency of visual encodings in the context of maps. This notion of comparing network and pathway visualization to cartographic maps is not as outlandish as it may initially seem—some biological resources are quite direct about being founded on this idea, such as BioCarta [65]. In Chapter 5, I cover how I apply and translate certain ideas and approaches from cartography to network visualization.

3.7. Conclusion

This chapter detailed a systematic review of figures, and three different analyses that were conducted on the data. Two different types of figures were discussed, "networks" and "pathways". The next chapter will expand on the results presented here by adding task information to the dataset, and additional analyses focused on tasks, rather than solely on visual encodings.

4. Systematic Review of Biological Network Figures Part II: Graph Tasks

4.1. Introduction

This chapter extends the results of the previous chapter by connecting those results with tasks. Tasks are one of the three necessary components (tasks, encodings, and data) of the Information Triad. The Information Triad will be more fully described in Chapter 6. However, suffice it to say that without the context of a task, it is not possible to objectively evaluate the performance of one biological network visualization against another.

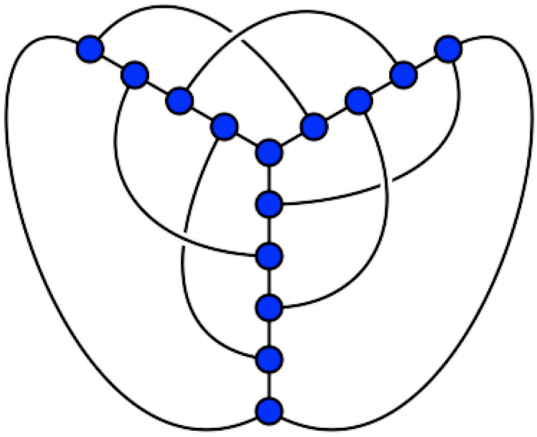
4.2. Background

In the past, rather than evaluating network visualizations in context of tasks, researchers would assess a number of visually aesthetic traits. A number of (planar) graph drawing algorithms have been designed to use these visual aesthetics as layout rules. Sometimes these rules are enforced as hard constraints (i.e. constraints that cannot be violated), and other times they are implemented as optimizations (i.e. maximizing or minimizing a designated score that represents how well a graph has been drawn) [30], [66]. Here below is a small sample of these graph drawing rules:

- Minimizing edge crossings,
- Minimizing node overlaps,
- Minimizing edge bends,
- Minimizing the sum of the lengths of the edges,
- Maximizing angles between edges connecting to a node,
- Maximizing symmetry

As mentioned above, these are general rules that apply to planar graphs. Other types of planar graphs, such as orthogonal graphs, tree graphs, polyline graphs, etc. have their own set of practical rules. There are a few strategies for applying these graph drawing rules. One strategy, as used by Ioannis et al, is to specify layout constraints depending on the topological and graph model attributes [66]. Another strategy is to approach the problem from a readability perspective, as used by Dunne et al, to specify layout depending on empirical findings from user studies [30]. Both of these approaches are founded on the idea that graph layout is the most impactful variable in determining the readability of a graph. Although the aesthetic traits defined by Ioannis et al are generally accepted as "best practice", there are a number of situations where it may be beneficial to revisit the importance of these "rules of thumb" in context of a desired task. For instance, BioFabric is a biological network visualization tool specifically designed to accommodate users interested in the task of path following. Path following entails starting at a source node and following along an edge until it reaches a target node [27]. Thus, some of the rules of aesthetics (e.g. minimize edge crossings, or sum of the lengths of edges) are deliberately disobeyed in order to better serve the desired task (i.e. path following). Another example is the Hive Plot, which essentially creates a Heawood graph, a type of mathematical graph known to minimize edge crossings when drawn (please see Figure 20 below) [11].

Figure 20 - A side-by-side comparison of a Heawood graph with a crossing number of 3 (left), and the similarly inspired Hive Plot (right).



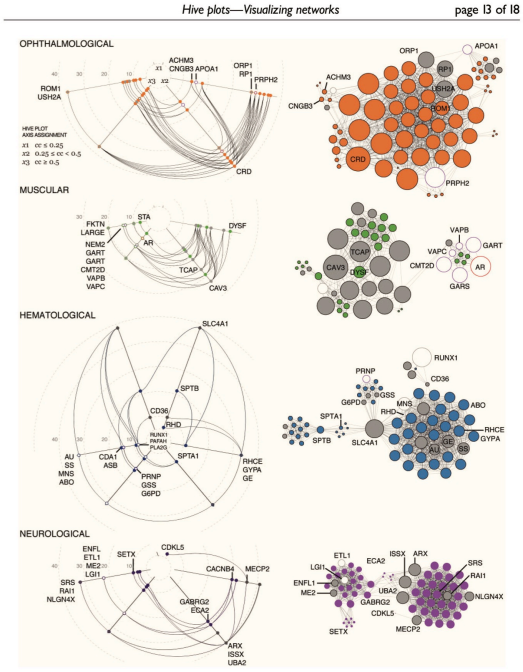


Figure 8: Comparison of largest connected components of ophthalmological, muscular, hematological and neurological subsystems in the DGN, drawn with the same layout rules as Figures 7A and 6B. Node size encodes the connectivity of the gene in the entire DGN. HP grid shows connectivity in steps of 10 edges.

A – Heawood Graph with a crossing score of 3

B – Figure from the Hive Plot paper

Task-based approaches to evaluating network visualizations, perhaps due to their inherent specificity, provide just enough clarity determine how well a visualization meets its intended purpose. Although researchers in visualization may have begun by evaluating figures and graphs using aesthetic measures, recently, it has become more common to use task-based metrics. Shneiderman et al published a "task by data" taxonomy that has been well-used since publication [67]. Morset et al published taxonomic guide for evaluating visualizations, which Lee et al added onto and further specified to publish a graph-specific task taxonomy [46], [68]. By associating tasks with visual encodings (determined in Chapter 3), visualization practices may be empirically measurable, explicitly stated, and better comprehended. Thus, there is a strong motive for understanding which tasks are possible to complete on each of the network visualizations evaluated in the previous chapter.

4.3. Method

In the protocol that follows, I use the 96 figures collected from the primary literature described in Chapter 3. This database of examined network figures from bioinformatics literature is available at the following URL: https://github.com/ngopal/systematic-review-network-figures

Obtaining Task Completion Data: Using the dataset collected in Chapter 3, every network and pathway was re-examined and assessed for task completability on 10 of the 13 graph tasks detailed in Lee et al [46]. Task completability refers to whether or not a task was classified as

"possible" or "not possible" after following the criteria described in Table 5. The protocols that were used to assess completability are specific examples of the general description of the tasks presented in Lee et al [46]. The protocols are available in Section 10.5.

Description of Task Completability: The presumption underlying task completability is that, if given adequate time and resources, a task may be performed to completion. Inversely, if a task is marked as not completable, then the task cannot be conducted to completion, even with adequate time and resources. This measure of completability is used, rather than accuracy, since fully completing a given task may require a substantial amount of time and effort for certain graphs. Further details about the criteria that must be met are available in Section 10.5.

The omitted tasks ("revisit" and "overview") were difficult or nonsensical to assess. Even though this leaves 11 tasks to be assessed, there are only 10 tasks listed in Table 5 because the "adjacency" task is considered a repetition of "accessibility" task; thus they are combined in the table and assessed only once. A table of assessed tasks and associated clarification questions is available in Table 5. If it was possible to complete the task, the task was marked with a "1", otherwise it was marked as "0". The criteria and questions that were employed to determine whether a task was completeable (adapted from Lee et al [46]) are provided in Table 5. Section 10.5 contains specific protocols describing the exact steps and operations used to determine whether the criteria listed in Table 5 were met.

Table 5 - This table contains an abbreviated list of tasks, descriptions of each task, and the criteria used to assess whether or not it was possible to complete that task.

Task	Description	Criteria (in question form)
Find Common Connection	The ability to determine if a set of nodes is directly connects two given nodes.	Can I visually find a node connected to X?
Find Articulation Points	The ability to identify nodes that, when removed, results in an unconnected graph	Can I visually find a node that when removed will disconnect the graph?
Find Bridges	The ability to identify edges that, when removed, results in an unconnected graph	Can I visually find an edge that when removed will disconnect the graph?
Find Shortest Path	The ability to find the shortest path between two nodes	Can I visually find the shortest path between two randomly selected nodes?
Find Clusters	The ability to distinguish groups of nodes within a graph	Can I visually find at least two (sufficiently distant) groups of nodes?
Find Connected Components	The ability to find connected components (two or more nodes connected by edges, with paths between the nodes)	Can I visually find disconnected groups of nodes (two or more nodes connected by edges)? Isolated singleton nodes are considered to be their own connected components.
Find Node Attributes	The ability to identify nodes defined by specific visual attributes	Can I visually find nodes based on any of the encoded attributes (e.g. color, shape, size, etc.)?
Find Edge Attributes	The ability to identify edges defined by specific visual attributes	Can I visually find edges based on any of the encoded attributes (e.g. color, shape, size, etc.)?
Follow a Path	The ability to follow a given path through a graph	Can I follow a path described in the caption (if there is one)? Can I visually follow a path between two randomly selected nodes?
Finding Adjacency and Accessibility	The ability to recognize that another node is connected to, or accessible from, a given node	Can I visually determine whether two randomly selected nodes are connected?

In order to systematically judge whether or not a given figure should receive a "1" or a "0" for each task, I applied the questions shown in the "Criteria" column of Table 5 to each of the 61 "network" or "pathway" figures in my collection.

In addition to tasks detailed in Table 5, this study also captured data on two characteristic attributes that were not captured in the previous study: number of nodes and number of edges. This information is important to record, as Ghoniem et al (among others), has shown that the ability to complete tasks is a function of the size and density of a network [69].

4.4. Analysis

This section provides three sets of analysis results: (a) descriptive statistics about task completion, (b) the findings of an exploratory analysis investigating the relationship between task completion and the number of nodes and edges contained in the network, and (c) a statistical test of independence investigating the relationship between the previously defined types of biological network (network and pathway) and tasks.

4.4.1. Descriptive Statistics

Table 6 provides an overview of the descriptive statistics resulting from the collected task data. Since the data is binary, the "Mean" column provides a measure of frequency, demonstrating how often it was possible to complete a given task.

Table 6 - A table of descriptive statistics for task completion (across both networks and pathways). The tasks are presented in order according to (ascending) mean value of the ratio of "possible" (1) to "not possible" (0) task completion statuses across the 96 figures.

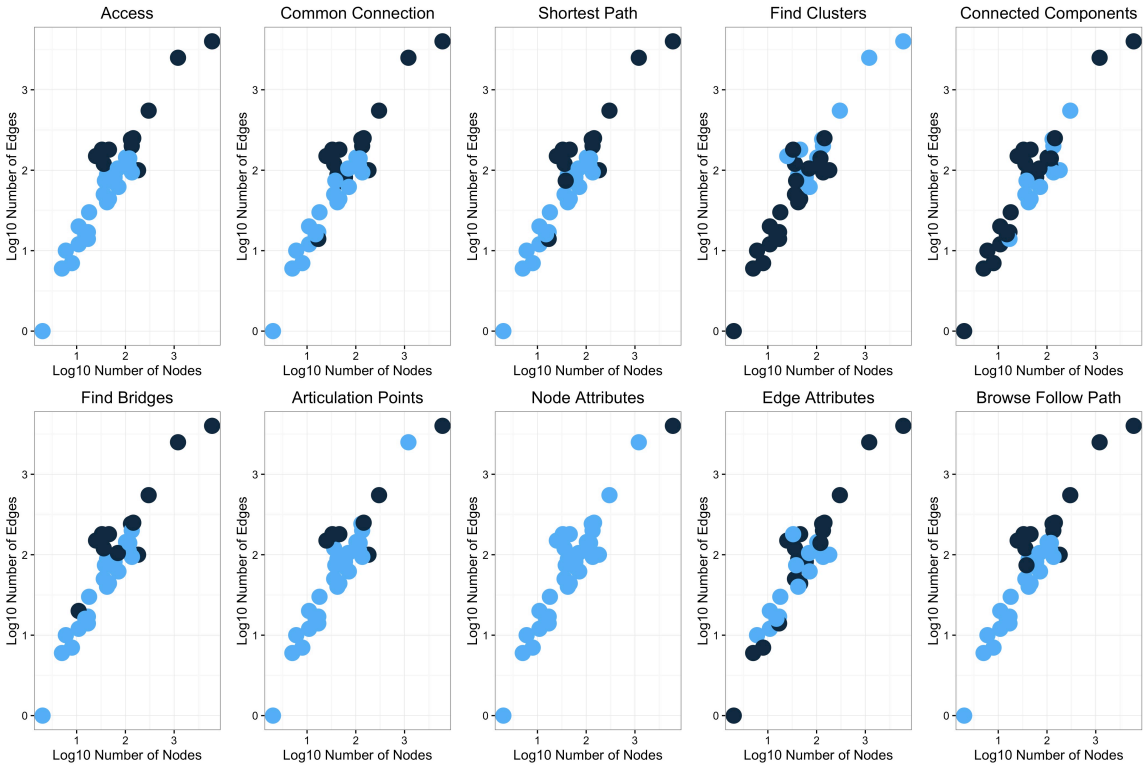
Task	Mean
Finding Clusters	0.3529
Finding Connected Components	0.3529
Finding Edge Attributes	0.4412
Finding Common Connection	0.5588
Finding Shortest Path	0.6176
Finding Bridges	0.6471
Following a Path	0.6471
Adjacency and Accessibility	0.6765
Finding Articulation Points	0.7941
Finding Node Attributes	0.9706

These descriptive statistics only convey a portion of the information obtained from this task-focused analysis. The next section explains how these values are affected by the number of nodes and number of edges in a network.

4.4.2. Tasks Interact with Visual Encodings

Figure 21 illustrates how the number of nodes and number of edges interacts with the ability (or inability) to complete each of the 10 tasks listed in Table 5. Each data point on the plot represents a network (pathways are not included in these plots). The light blue color denotes that a task was completable for a given network, whereas the dark blue color denotes that the task was not completable for a given network. The tasks of finding accessibility, finding common connections, finding shortest paths, finding bridges, finding articulation points, and following a given path seem to be difficult tasks to complete in large networks. For clarity, I will use the term "small-scale tasks" to refer to the tasks that are most often completable when conducted on a network with 100 nodes or edges, or less. Furthermore, I will use the term "large-scale tasks" to refer to the tasks that are most often completable when conducted on a network with more than 100 nodes or edges. Finding edge attributes and finding connected components seem to be completable in some interval between "small" and "large" networks. Finding node attributes is completable in even large networks. Notably, finding clusters is only completable in networks over a certain size, and is not completable in small networks.

Figure 21 - These plots depict the relationship between number of nodes, number of edges, and the ability to complete each of the 10 tasks listed in Table 5, for networks. In context of each task (i.e. panel), the dark blue dots signify "not completable," while light blue dots signify "completable."



4.4.3. Task distribution between networks and pathways are not significantly different

Chapter 3 differentiates between networks and pathways as different types of biological networks representations (based on frequency of use of visual encodings). Given the distinction, a question that naturally arises in context of this study is whether networks and pathways have significantly different distributions for the ability to complete each task.

Table 7 - This table provides frequency counts for task completions, itemized by graph type (network or pathway), and task.

Task to Complete	Frequency for Network	Frequency for Pathway
Adjacency and Accessibility	23	25
Finding Common Connection	19	25
Finding Shortest Path	21	26
Finding Clusters	12	0
Finding Connected Components	12	1
Finding Bridges	22	25
Finding Articulation Points	27	25
Finding Node Attributes	33	25
Finding Edge Attributes	15	24
Following a Path	22	27
Revisiting a Node or Edge	34	27
Determining the Underlying Graph Model	0	0
Finding Patterns	1	1
Finding Outliers	4	0
Scanning the Network	34	27
Performing Set Operations	0	0

A two-tailed t-test (paired and assuming homoscedasticity) was used to determine whether the distribution of task completion is different between networks and pathways. The ability to complete each task was an independent observation. Using the data contained in Table 7, a t-test yielded the following results:

H_0: The difference between the means of task completion frequency between networks and pathways = 0
H_a: The difference between the means of task completion frequency between networks and pathways 0

Table 8 - Descriptive statistics results for the data in Table 7.

VAR	Sample size	Mean	Standard Deviation	Variance
Network	16	17.4375	11.81507	139.59583
Pathway	16	16.125	12.66425	160.38333

Table 9 - Paired (two-tailed) two-sample t-test results from the data in Table 7.

Degrees of Freedom	15
Hypothesized Mean Difference	0
Pooled Variance	149.98958
Test Statistics	0.83049
Pearson R	0.86887

Due to a p-value of 0.42 (well above the standard alpha level of 0.05), the null hypothesis is not rejected. Thus, the distributions of task completion between network and pathway figures are not significantly different. There is no evidence to support the claim that the ability to complete the tasks provided in Table 5 depends on whether the figure was a network figure or pathway figure. Furthermore, this implies that findings from a task-based experiment using one type of figure may also be applicable to the other. The Pearson correlation value between the two groups is 0.86, which indicates a very strong relationship between the two groups. Had there been enough data, it may have been beneficial to conduct an analysis of variance analysis (ANOVA) between tasks and visual encodings (to identify interactions between variables).

4.5. Discussion

A general observation from Table 6 is that the tasks that had the lowest ratio of completable to non-completable values were those that required large networks (e.g. finding clusters, finding connected components, etc.). Observing this trend is interesting because it provides guidance on which tasks are generally difficult to complete, and which ones are not. For instance, when a network figure is depicting clusters in a large network, due to the algorithm, and possibly thanks to enclosures, highlights, and callouts, finding clusters may not be a difficult task. However, when a network is smaller in size, or a layout algorithm does not yield a configuration that does not facilitate perceiving groupings, finding clusters may become a much more difficult task to accomplish.

The second least frequently completed tasks were those that were operations on edges (finding edge attributes, following a path, etc.). As a network scales in size (and proportionally in density), it can quickly transform from readable to unreadable. Moreover, edges are plenty, and difficult to visual encode, so a reader may not easily be able to perceive what is important versus unimportant at a glance. Edge-related tasks may be supported through appropriate communication and emphasis techniques, but one can expect the ability to comprehend the contained information to degrade as networks scale. This frustrating

reality is commonly cited as a driving factor for the development of novel network visualization techniques.

The tasks that were completed most frequently were those that were operations on node attributes. This suggests that nodes are the most frequently parameterized visual attributes. It also suggests that tasks related to node attributes may generally be completed more easily than tasks related to edge attributes. However, it is not clear exactly why this may be the case. Perhaps, visual attributes related to nodes are simply the most obvious to parameterize, or perhaps nodes are easier to parameterize than edges, or perhaps nodes are easier to encode and decode than edges.

Another observation from Figure 21 is the log-linear relationship between the number of nodes and number of edges in a network. The log-log relationship suggests that the number of nodes and edges in biological networks may scale according to a power-law. This is a practically useful observation, and one that has been corroborated by prior research [70], [71]. There seems to be a relationship between the log10-corrected number of nodes and edges in a network, and the ability to complete each of the tasks in the graph task taxonomy. In particular, certain tasks such as *clustering* seem more likely to be completed on networks with a higher number of nodes and edges, and other tasks such as *path following* seem more likely to be completed on networks with a lower number of nodes and edges. Yet, other tasks, such as *finding edge attributes*, seem to be completed most often in networks of some size between "small" and "large".

4.6. Conclusion

This chapter is the task-centered extension of the study performed in Chapter 3. From the collection and analysis of task-based data, we have seen that there is a clear relationship between the size (number of nodes and edges) of biological networks and the ability to complete tasks on that network. Although ANOVA between visual encodings and tasks is not possible with this dataset, it is possible to obtain an understanding of the relationship between visual encodings and tasks through the Random Forest technique—this will be further detailed in the next chapter.

5. Determining Relations Between Visual Encodings and Tasks Via Random Forest

5.1. Introduction

This chapter further extends the results of the previous chapter, which covered collection and analysis of task-related data from a systematic review of figures from peer-reviewed bioinformatics publications. The contents of this chapter could have been included as part of the analysis section for the previous chapter. However, the length and detail of the contents of this chapter would have rendered the previous chapter disproportionately large —thus I present the information here, in its own chapter.

In this chapter, I use the random forest algorithm to analyze the data that were collected during the data collection step detailed in the previous. The analysis section in the previous chapter compared frequency distributions of visual encodings between networks and pathways. The dataset contains the following information on figures: visual encodings that were used, tasks that could be completed, the number of nodes and edges in the figure, and associated meta-data (e.g. Pubmed ID, etc.)—the visual encoding characteristics and task completion characteristics were recorded as binary data (i.e. a particular encoding was used or not used, and a particular task was completable or not completable). The random forest algorithm was used to determine the relationship between task completion ability and visual encodings. The dataset contains binary and categorical variables, but there are very few analytical techniques that can appropriately model such a dataset. The typical choice when working with a binary dependent variable is logistic regression. However, because of the categorical independent variables, and also because of the relatively small number of samples versus predictors, logistic regression is not an ideal choice. For the research question and dataset at hand, Random Forest is a reasonable, convenient, and appropriate choice (with several supporting reasons listed in the background section).

5.2. Background

Background on visual encodings and tasks have been covered amply in the previous chapters. Thus, in this particular background section, I will provide background for the Random Forest algorithm. In short, the Random Forest algorithm creates a large number of decision trees (typically on the scale of thousands), and ultimately provides an "average" of the results from those decision trees. Random Forest is one of the more recently developed machine learning algorithms, published in 2001 [72]. One of the appealing characteristics of this algorithm is that it is quite versatile; it can be used for regression, classification, clustering, and even survival analysis.

One of the advantages of Random forest over linear regression (and its applicable variants) is that Random forest does not assume a linear relationship between predictor and output variables. In fact, Random forest does not assume that predictor variables are independent of each other either. However, allowing for non-linear modeling may become a detriment if one wants to reason about how theoretical values (or new values) may affect results. Another advantage of random forest is that it does not require cross-validation as many other machine learning techniques would. By nature of the algorithm, 60% of the data is used to create a decision tree and the remaining 30% of the data is used to assess the accuracy of the tree. When using Random forest, referring to "out of bag error" (OOB) is considered equivalent to "area under the curve" (AUC) from a receiver-operator curve (ROC) plot.

However, there are a number of necessary considerations about the data input into Random forest. Despite the benefits listed above, there are a number of caveats to account for when using random forest. The first caveat is that random forest is very sensitive to class balance in a dataset. That is, if there is a rare class (i.e. the occurrences of that class are 15% or less of the dataset) one is expecting to model using random forest, then one of a number of techniques must be used to balance the classes when bootstrap sampling. One approach is to use *downsampling*, which utilizes all of the rare instances, and randomly samples an appropriate number of the majority class instances to create a balanced dataset. Another approach is *supersampling*, also known SMOTE, which disproportionately samples (with replacement) the rare class data to achieve a balanced dataset [73]. Another limitation is that random forest is incapable of tolerating missing data. The common methods of handling missing data are to throw them out completely, or to use an imputation method that estimates a reasonable value for the missing data. Random forest, like regression, is also sensitive to multicollinearity (when two or more variables are highly correlated and linearly predictive of one another).

Among others, we will be calculating two useful values from Random forest output. The first is an "importance" measure (by way of Gini score), which quantifies how sensitive the value of a variable is to determining classification. However, Random forest tends to systematically assign excessive importance to categorical variables with a large number of levels. Thus, like many other machine learning algorithms, random forest requires careful consideration of included variables and parameter tuning to obtain peak performance. The other item we will be calculating is referred to as a "prototype", which lists the predictor variable values that occur most frequently for a given class.

5.3. Method

This method is an extension of the results of the previous chapter (visual encodings part II).

The data that is input into the Random forest algorithm was generated using the following protocol. The tasks that were assessed in this study were obtained from a scientific publication containing a taxonomy of graph tasks [46].

In addition to tasks previously detailed in Table 5 of Chapter 4, this study also captured data on two characteristic attributes that were not captured in the previous study: number of nodes and number of edges. This information is important to record, as Ghoniem et al (among others), has shown that the ability to complete tasks is a function of the size and density of a network [69].

Random Forest calculates "out-of-bag error" (OOB), which is considered to be virtually equivalent to area under the curve (AUC) of a receiver operator curve (ROC). Thus OOB may be used to evaluate classification performance by the model (using following estimation of 1 - OOB).

5.4. Analysis

The analysis in this chapter consists of two major sections: descriptive statistics (with supporting figures), and results from random forest regression. The section with descriptive statistics provides are overview of how tasks may be associated various visual encodings.

5.4.1. Random Forest Results

Random Forest provides output that is very information rich. Thus, I will explain the structure and relevancy of these reported results. The first sub-section provides an overview of model performance and parameterization, which provides information on how reliable or trustworthy the generated Random Forest models are. Next, in the second sub-section, I provide an explanation of the metric "Variable Importance", which indicates which visual encodings are most "important" in determining the ability or inability to complete a given task. Finally, in the third and last sub-section, I provide an overview of the "prototypes" generated from applicable Random Forest models—which reflect the most likely predictor variable parameterization that would produce a desired outcome variable.

An independent random forest model was constructed for each response variable (i.e. each task). This means that a total of 10 random forest models were created, each using the 16 predictor variables (i.e. visual encodings). Although it is customary to provide descriptive statistics prior to presenting the results of a more complex analysis that section will be omitted here as descriptive statistics about visual encodings and tasks are available in the analysis sections of chapters 3 and 4, respectively.

5.4.2. Evaluating model performance and parameterization

The obtained Random Forest models had satisfactory performance, as shown in Table 10. The random forest models were tuned to use an "mtry" value (i.e. the number of variables in each decision tree) of 16. This value varied for every model, but 16 was the most common "mtry" value. The parameter "m" (in reference to "mtry") denotes the number of predictor variables to include in each decision tree—in this case, it means most decision trees used all 16 predictor variables. The minimum error rate across all models was 14.71%, and the maximum error rate across all models was 29.41%. Given the number of samples provided from which to generate and test decision trees, the classifiers performed surprisingly well. The task of "finding node attributes" yielded a 0% error rate due to a NAN class error. Simply put, this means that there were so few "0" values in the data used to create the decision trees for that particular random forest classifier that error rate could not be accurately evaluated.

Table 10 - This table provides an overview of the performance of all of the random forest prediction models that were generated during this analysis.

RF Model for Task	OOB Error Rate
Adjacency and Accessibility	17.65%
Finding Common Connections	26.47%
Finding Shortest Path	20.59%
Finding Clusters	29.41%
Finding Connected Components	29.41%
Finding Bridges	26.47%
Finding Articulation Points	23.53%
Finding Node Attributes	0% (NAN)
Finding Edge Attributes	14.71%
Following A Path	14.71%

5.4.3. Importance of Variables

In this model, the visual encodings were the predictor variables, and the ability or inability to complete a task was the response variable. As Table 11 below shows some example importance output from a selected random forest model. Importance is calculated through a perturbation process, where the value of a decision node in a decision tree is set to be a higher or lower value, and then the outcome of the classifier is assessed to see if the classifier call is different from its unperturbed state. If altering that particular variable changes the classifier call, then it receives a higher importance value. If altering that particular variable does not change the classifier call, then that variable receives a lower importance value. It would follow that variables with higher importance scores affect the outcome of the classifier more heavily than those variables with lower importance scores. Explained another way, variable importance scores may be viewed as an analog to variable sensitivity.

Importance values of the variables in the model were calculated using the Gini impurity index, and the resulting importance values are presented and explained in Figure 22. Gini impurity maximizes the average purity of children nodes in a decision tree, and thus selects splits that that decrease the Gini index the most [74]. In context of Random Forest, the Gini impurity index is a measure of how often a randomly chosen response class from a decision tree would be incorrectly classified if it were randomly classified according to the distribution of classifications in the subset. Alternatively, other measures of impurity may be used, such as entropy or classification error, although in most cases those metrics would yield very similar results. However, for this study, Gini impurity index is the choice that best meets the needs of the collected data, and is most straightforward to interpret.

Table 11 - Output for variable importance from the random forest model predicting the ability to complete the task, "Finding Clusters". Similar tables may be produced for the other 9 random forest models.

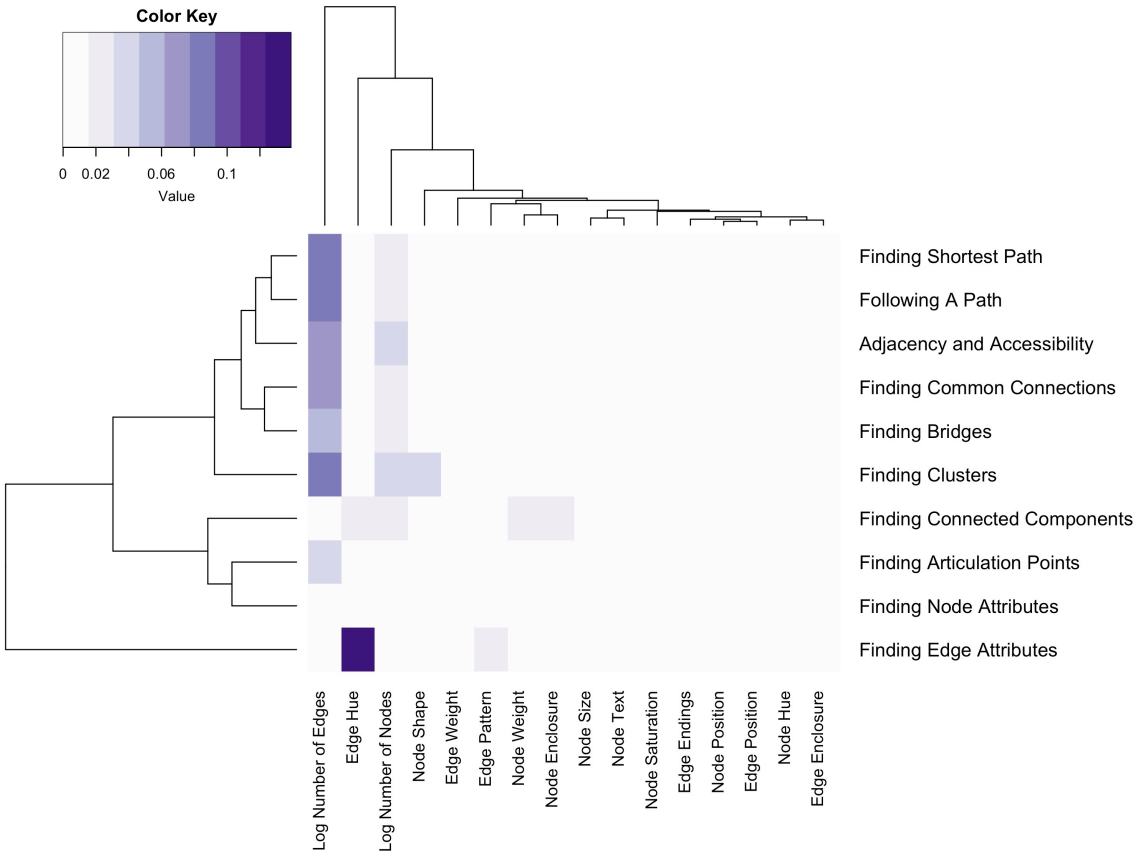
Visual Encodings	**0**	**1**	**Mean Decrease in Accuracy**	**Mean Decrease in Gini Index**
Node Position	-0.000952	0.00109	-0.000231	0.126
Edge Position	2.00E-04	0.00269	0.000629	0.140
Node Text	0.00489	0.0129	0.00779	0.464
Node Size	-0.000702	-0.002	-0.00178	0.278
Node Weight	0.00301	0.00677	0.00427	0.346
Node Saturation	-0.001	-0.00805	-0.00366	0.353
Node Hue	-0.000721	-0.00983	-0.00283	0.227
Edge Hue	0.0126	0.0215	0.0139	0.779
Node Shape	0.00787	0.037	0.0166	0.840
Node Enclosure	0.00396	0.00975	0.00627	0.510
Edge Enclosure	-0.00372	-0.00286	-0.00335	0.123
Edge Pattern	-0.00221	-0.00923	-0.00428	0.183
Edge Endings	0.000133	0.00257	0.000716	0.098
Edge Weight	0.0036	0.0044	0.00355	0.328
Log Number of Nodes	0.0285	0.037	0.0301	2.55
Log Number of Edges	0.0389	0.0711	0.0478	2.82

Since the each of the 10 random forest models contains 14 variables, the data can most concisely be represented as a heat map, as in Figure 22. Figure 22 is rich with useful and interesting information, which I will now cover. The first observation is the clear impact of the log number of nodes and log number of edges across the majority of models. The second observation is that the log number of edges seems to have a more substantial impact on the ability to complete tasks (in general) than the log number of nodes. The third observation is that "edge hue" is clearly an important variable for the task of "finding edge attributes". The fourth observation is that the task "finding node attributes" does not seem to be affected by any of the included predictor variables.

Furthermore, there seem to be certain tasks that may be affected by combinations of visual encodings, such as "finding clusters", "finding shortest path", "finding common connections", "finding connected components", and "finding edge attributes". In contrast, there also seem to be tasks that are only affected by the number of nodes and edges in a networks, which are "following a path", "adjacency and accessibility", "finding bridges", and

"finding articulation points." This suggests that the latter group of tasks may not be as affected by visual encodings as the former group.

Figure 22 - A heat map of variable importance values organized by task. The tasks are hierarchically clustered to show similarity of variable importance values across random forest models.



5.4.4. Prototypes resulting from the random forest model

From this Random Forest model, one may also obtain prototypical examples. For instance, if one may pose the question, "what variable parameterization would yield the value of X?" Obtaining the prototype from a Random Forest model would help answer that question. In terms of the model, this is effectively obtaining the parameterization that would make the chosen response variable most likely. Since the response for ten tasks are predicted, the prototypes are presented below in Table 12. For instance, if one were to render a network visualization where the objective task was "finding node attributes", then one might consider using the visual encodings of node hue, edge hue, and ensure that the number of nodes is less than ~5,900, and that the number of edges is less than ~4000. If one were to render a network visualization where the objective task was "finding node attributes", one might consider using node hue and edge hue. Although edge hue may not be directly encoded with information for this task in particular, the prototype of the model conveys that the ability to successfully complete the task of "finding node attributes" is associated with the use of the visual attribute of edge hue (color).

Estimating values for predictor variables cannot be accomplished in certain cases. For this dataset, only 3 of the 10 Random Forest models were able to output a prototype. The rest of the models did not have enough information in the data to be able to determine a prototype. This is not unexpected, as many of the tasks and visual encodings have near identical values and importance (notice the overall similarity denoted by the "white" color in Figure 22).

Table 12 - Prototypes generated from random forest models. The three tasks provided in this table are the only tasks for which prototypes could be estimated.

	Finding Common Connections	**Finding Node Attributes**	**Finding Edge Attributes**
Node Position	0	0	0
Edge Position	0	0	0
Node Text	1	0	1
Node Size	0	0	0
Node Weight	0	0	0
Node Saturation	0	0	0
Node Hue	1	1	1
Edge Hue	0	1	0
Node Shape	0	0	0
Node Enclosure	0	0	0
Edge Enclosure	0	0	0
Edge Pattern	0	0	0
Edge Endings	0	0	0
Edge Weight	0	0	0
Log Number of Nodes	2.11	3.77	1.78
Log Number of Edges	2.30	3.60	2.17

5.5. Discussion

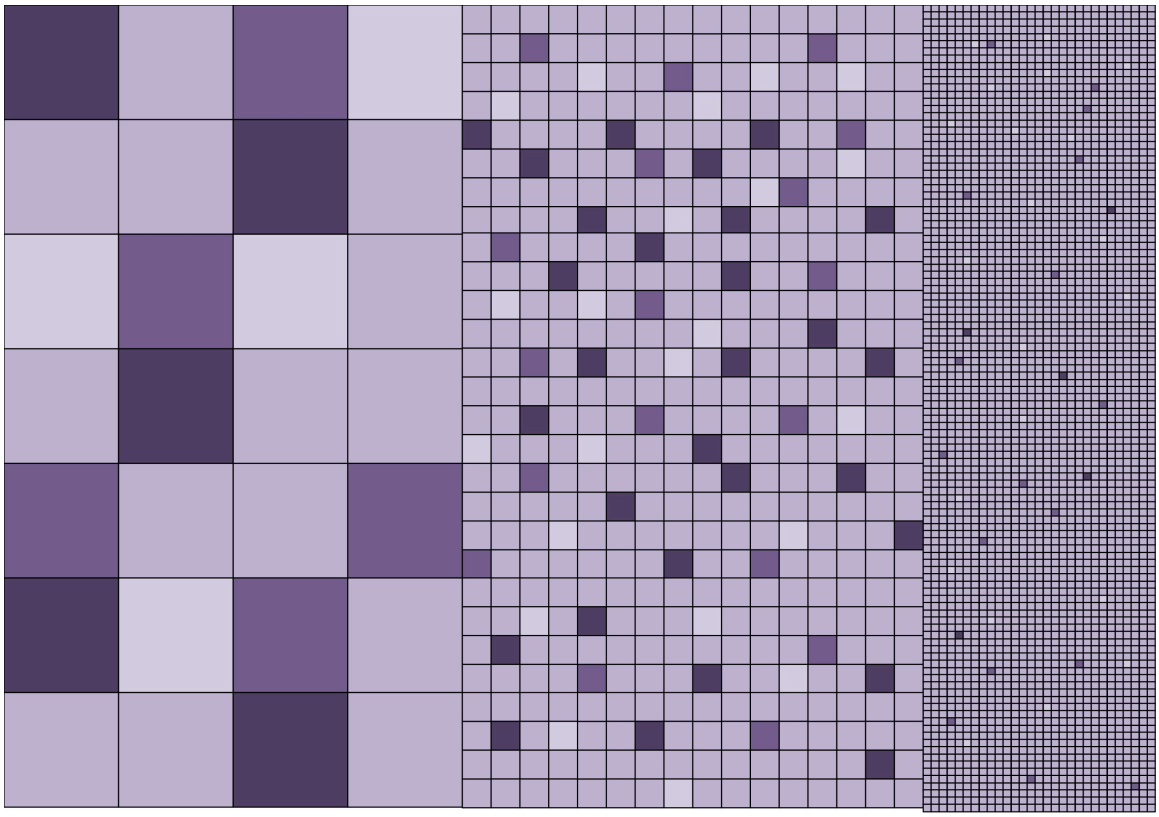
Prior task-based research has shown that even networks with nodes exceeding 30 nodes may pose significant hurdles for the completion of certain tasks [69]. The findings of this chapter show a somewhat conflicting result, that many of the selected tasks may be accomplished in networks with higher orders magnitude.

However, the networks used in previous studies could have had different underlying graph models than the graph models generally thought to underlie biological networks. Biological networks follow a number of graph models, most often a scale-free hierarchical graph [70], [75], [76]. In fact, this observation was echoed with my own findings in the previous chapter.

However, this often depends on the information used to construct the graph, and ultimately, the information included in the graph. The line of reasoning is that by understanding the distributions and behaviors of various graph models, researchers can predict the distributions and behaviors of graph models found in real-world datasets (and detect outliers). However, to date, there are no publications detailing the relationship between biological graphs and the ability complete graph tasks—the study described in this chapter is the first.

It may be the case that certain tasks in the task taxonomy are more and more difficult to complete as networks scale in size and density. An instance of this may be found through briefly examining a visual limitation of hierarchically clustered heat maps, as illustrated in Figure 23. A hierarchically clustered heat map is an interesting visualization technique, as it provides information at multiple levels of abstraction. The actual data values themselves are reflected in the matrix (typically through color saturation, sometimes even with numerical values), while the higher-level, more abstract relationships (such as similarity) are reflected in the hierarchically clustered dendrogram(s). Although having the ability to see both levels of information is quite useful, these heat maps provide diminishing returns (in context of readability) as the matrix increases in size. Since the cells become smaller, the colors are more difficult to distinguish, and eventually both the dendrogram(s) and matrix cannot fit on the same plane as they would both be too large. Furthermore, the color of the cells becomes increasingly difficult to read and interpret, as depicted in Figure 23. In other words, the tasks that were possible to complete when a hierarchically clustered heat map was a smaller size are no longer completable as the heat map exceeds a certain size threshold. The study detailed in this chapter shows that, in this same manner, network visualizations may also have a size range for which certain tasks are more easily completed than others.

Figure 23 – An example illustrating that reading various shades of color becomes more difficult in heat maps as the number of cells in the heat map increases.



Although this chapter focuses very heavily on visual encodings and tasks, I believe it is essential to acknowledge that graph layouts play a significant role in the completability of each of the 10 tasks that were assessed. In general however, it seemed that small networks and pathways may have been laid out by hand, and that large networks and pathways were laid out computationally. However, the layout technique employed on a given network or pathway figure was rarely reported or otherwise impossible to determine with even a minimal level of confidence. Add to this uncertainty the ability to fine-tune layout parameters, and the ability to lay out these figures by hand, and consequently measuring and interpreting layout information becomes remarkably difficult. I mention these items to emphasize that the role of graph layouts is not underestimated, and to also remind the reader that graph layout can be an informative variable to include in any prediction models similar to the ones presented in this chapter (if possible).

5.6. Conclusion

This chapter completes the analysis of the data originally collected for the systematic review of figures (Chapter 3), and supplemented by the data collected for task analysis (Chapter 4). The combined dataset resulting from both studies was successfully analyzed in this chapter. The next chapter will detail the development of a network visualization system built on the findings of the research contained in Chapters 2-5.

6. Design and Architecture of Dynamo

Dynamo is a system that affords the ability to depict the set space of possible visual encodings used to represent information in a network, while controlling for layout and topology. A system such as Dynamo is useful for investigating the effect of various combinations of encodings on user perception and interpretation. In order to implement a system that operates on task and encoding information, I must first define what I call the "Information Triad". The following concepts are in alignment with the definitions presented in the previous chapters, but are, however, further abstracted to add robustness.

6.1. The Information Triad

While designing and developing network visualization utilities, it sometimes becomes apparent that there are many unforeseen assumptions that must be made in order to actually depict any network. The "information triad" (IT) is composed of three elements: (1) dimension, (2) task, and (3) encoding, all of which are interconnected. First, a definition for each of these three individual elements:

- Dimension – refers to a class of data attached to a node or edge entity. For instance, nodes may have certain data attached to them,
- Task – refers to an objective that the user would like to accomplish
- Encoding – refers to visual attributes of nodes and edges that may be used to represent information

Specific instances of dimensions, tasks and encodings must be defined in conjunction to ensure clarity when conveying information through network visualization. When one fails to define even one of these three items, there is enough ambiguity to make it difficult to assess whether or not the visualization is working as intended. For instance, if one imagines that edges in a biological network have a "correlation" value attached to them, and if one decides to visually encode that data using an "edge width" attribute, it remains unclear whether attaching "correlation" to "edge width" is a reasonable choice. However, if a task, such as "clustering", were defined in conjunction with dimension and encoding, then one has the minimal required information to assess effectiveness of network visualization design choices. Although I present this triad as the minimal information required to evaluate the design of network visualization, there are certainly other factors, such as size, density, topology, model, content (i.e. the underlying data in the network), and experimental design that will contribute to defining the overall success or failure. The focus of Dynamo is enable exploration of the information triad while controlling for these other factors, which, for practical reasons, are out of scope for this tool.

Another way to describe IT is that it is a tool to help frame research questions about data visualizations. Evaluation of data visualizations is a notoriously tricky endeavor, and IT can help provide framework for reasoning about the design and interpretation of an evaluation. Specifically, one may simply hold any two of the three elements in IT constant, and compare the differences among variations of the element that was not held constant. For instance, if one were interested in understanding how a certain visualization performs in context of two different tasks, this could be tested by holding "dimension" and "encoding" constant, while varying the measurement used for "task." Another example could be if one were interested in understanding how a certain visualization performs in context of two different encodings, this could be tested by holding "task" and "dimension" constant, while varying the

measurement used for "encoding." Finally, if one were interested in understanding how a certain visualization performs in context of two different data dimension, this could be tested by holding "task" and "encoding" constant, while varying the measurement used for "dimension." Although the research questions that would be answered through this process would all be different research questions, the IT framework supports all of them.

6.2. Assumptions of the Information Triad

In the simple, three-member conceptual framework presented here, visual encoding is a mapping between data attributes (data attached to nodes or edges) and visual attributes (size, color, shape, etc.) via a function. This was briefly covered in Chapter 3. However, there are certain underlying assumptions that are clarified in this section.

- The mapping between data attributes and visual attributes are bijective. That is, the mapping between data attributes and visual attributes are one-to-one.
- The mapping between data attributes and visual attributes is monotonic functions (i.e. the order of the data attribute values is preserved by the visual attribute being used)
- Identity functions are possible

Although the Dynamo application described later in this chapter is built on the Information Triad, Dynamo also supports surjective mapping (i.e. many-to-one relationship between domain and range) and injective mapping (i.e. every element in the domain must be mapped to a unique range element). These terms and their significance are clarified in Figure 26 and Section 6.9.4.

6.3. Formulating the problem: matching visual encodings to dimension-tasks

In case the problem may not already be clear, I will formally state the problem in this section. One way to think about the core function of Dynamo is that it solves a bipartite matching problem—matching visual encodings to dimension-tasks. There are a few formal definitions of the various flavors of bipartite matching problems:

- Maximal Bipartite Matching – A matching where all available edges have been assigned to the nodes.
- Maximum Bipartite Matching – A matching where the largest possible number of edges has been used.
- Perfect Bipartite Matching – A matching where edges connect every node in the graph

Graph flow algorithms (of which bipartite matching is a specialized case) are well studied and can be applied to a very wide range of problems. Although the model being used is a graph, the implementation of a graph flow problem can take many forms. Given the requirements of this tool, and my preference for clarity over efficiency in this matter, I have opted to use linear programming. One of the major advantages of linear programming is that constraints are easy to interpret and define (relative to deciphering constraints implemented in a specific computer language, such as Java or C). More details about linear programming are provided in the following sub-section.

6.4. A brief background on operations research

The optimization method employed in Dynamo is from the field of operations research (OR). Operations research was originally given its title during World War II, when military forces developed and employed techniques that optimized the use of resources under

varying constraints. Some of the fundamental methods and concepts that underpin the field of biomedical informatics were originally translated from the field of operations research. For instance, receiver operating characteristic (ROC) plots were developed during World War II to analyze radar signals—however, ROC plots are now conventional in machine learning and biomedical informatics as way to evaluation two-class classification model performance (e.g. logistic regression). Graph flow algorithms were used to determine maximum flow (also known as minimum cut) for a variety of war-related purposes. One of the most famous applications of maximum flow minimum cut was finding the "bottleneck" regions of the Soviet railroad network [77]. The network contained data on capacities, distances, and estimated value of payloads carried by trains on the network, and the optimal attack points of the railroad network were determined using a constraint-based method.

The same constraint-based technique used in Dynamo has been used to solve optimization problems concerning transportation, racial balance in schools, various forms of network analysis, and even game theory [78]. Constraint satisfaction techniques are well studied, and are still used in many application areas today. One notable example would be American Airlines, which uses a constraint-based approach to schedule flights, hotels, staff, and refueling.

As Dynamo is built using linear programming (LP) as its underlying method of optimization, it is subject to the same advantages and limitations. These are covered below in the next sub-section. There are a handful of LP algorithms, but the two I will briefly acknowledge here are the Simplex algorithm and Karmarkar's algorithm [79], [80]. The Simplex method is solvable by hand (which I will not provide an example of), and theoretically may take a very long time to find a valid solution. Luckily, in practice, Simplex typically converges on a solution within 2-3 iterations for most LP formulations. Karmarkar's algorithm is a computational improvement over Simplex as it runs in polynomial time, rather than exponential time. The improvement is possible due to omitting certain computational steps through making approximations of the optimal solution, and in practice the approximation is sufficiently close to the optimal solution. Since the input to the linear program in Dynamo is not expected to be large, and since execution speed is not a top priority, this software uses Simplex. However, using an optimized linear solver could easily boost performance, if desired or necessary.

6.5. An Overview of Dynamo

Dynamo is an experimental tool designed to facilitate the exploration of visual encoding properties of network visualizations. The architecture of the tool is illustrated in Figure 25 below. To summarize, Dynamo presents a web visualization to a user, who may enter visual encoding prioritizations (e.g. rankings) into an input table (along with a few other constraint parameters), and provides the user with an optimal encoding assignment. This entire process is done through the web browser, but the application relies on a lightweight web server (written in Javascript) and a constraint-based assignment tool (written in R).

The following sections will cover in detail, the input table, design assumptions, architecture, and example configurations.

6.6. The Input Table

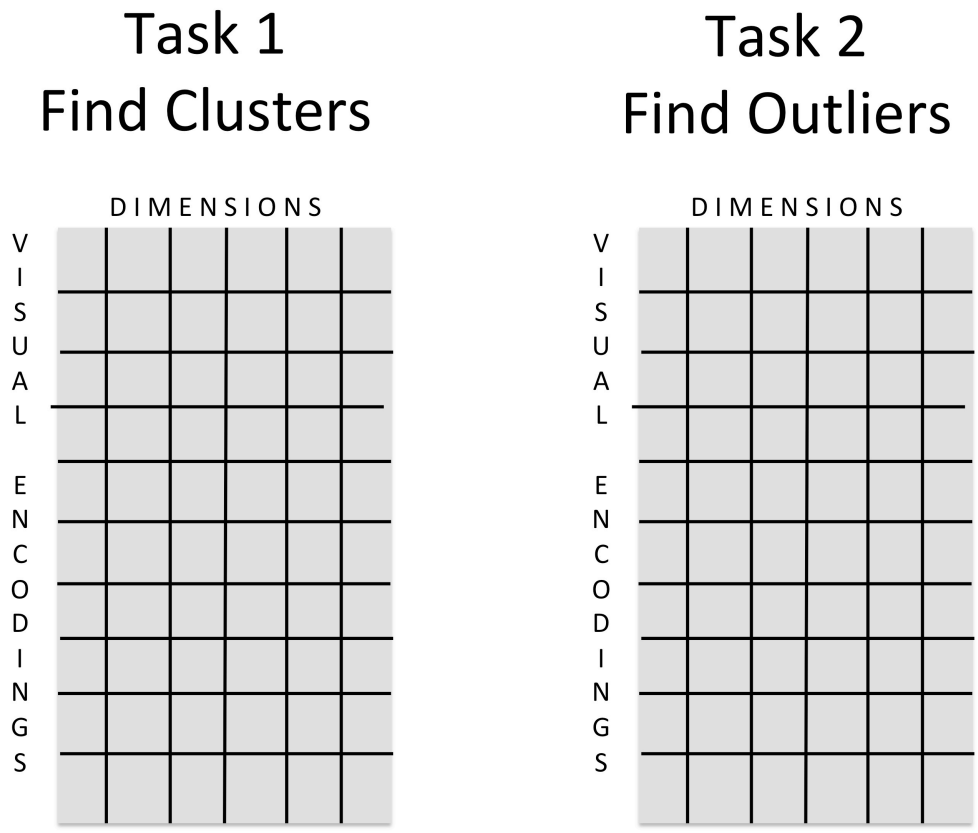
Table 13 below provides an example of what the input table to Dynamo may look like. The rows are visual encoding values, while the columns are a combination of dimension and task. The cells of the input table are intended to store statistical ranks.

Table 13 - An example of the structure of the input table used by Dynamo. The contents of this table have intentionally been left empty.

	Dimension1*Task1	Dimension2*Task1	Dimension1*Task2	Dimension2*Task2
Encoding1				
Encoding2				
Encoding3				

Although this combination of dimension and task may seem (intuitively) difficult to grasp at first, designing the input table in this manner allows us to represent the Information Triad in only two dimensions. In reality, it would also have been possible to combine encoding with task, or even encoding with dimension. However, dimension and task were combined because the resulting data are more intuitive than the other combinations, and more straightforward to collect baseline data on. When combining dimensions and tasks, it is natural to think that there are certain tasks one might want to complete using certain dimensions of data. For instance, if nodes in a network have a weight dimension, one may be interested in using that value to find clusters, or perhaps alternatively, find outliers. In this instance, node weight is quantitative and thus may be used with the tasks of finding clusters or outliers. However, if the dimension of data in question was categorical, such as "node id" or "node name", then there are clearly certain tasks that are inappropriate or nonsensical to perform (e.g. mean, median, and associated summary statistics). Although it is possible to combine encodings and tasks, or even dimensions and encodings, the result would imply they are more interrelated than we currently know them to be. Yet another way to conceptualize this is that for each task, there is one input table with dimensions as columns and visual encodings as rows (please see the schema in Figure 24).

Figure 24 - A schema that represents another way to conceptualize the Information Triad.



From the contents of the Figure 24, one might initially think that task qualifies visual encodings in the same manner that it qualifies dimensions. However, all of the visual encodings are static in the visualization, and are independent of task.

6.7. Design decisions and constraints of the Information Triad

Although the Information Triad is designed to shed light on assumptions would otherwise be implicit, there are a handful of axioms that must be defined in order to permit this property. These axioms are listed below:

1. Only one encoding may be assigned to one dimension*task.
2. The same dimension of data may be assigned to another encoding if appropriate for the task objective.
3. Dynamo searches for the optimal solution to a formulation, rather than simply a feasible solution.

In addition to the assumptions set forth by the design of Dynamo, there are a number of assumptions inherent in the method of linear programming itself. These assumptions are covered here at a cursory level, and were originally explained in the book, "Operations Research" [78]:

1. Proportionality – the assumption that quantities are directly proportional

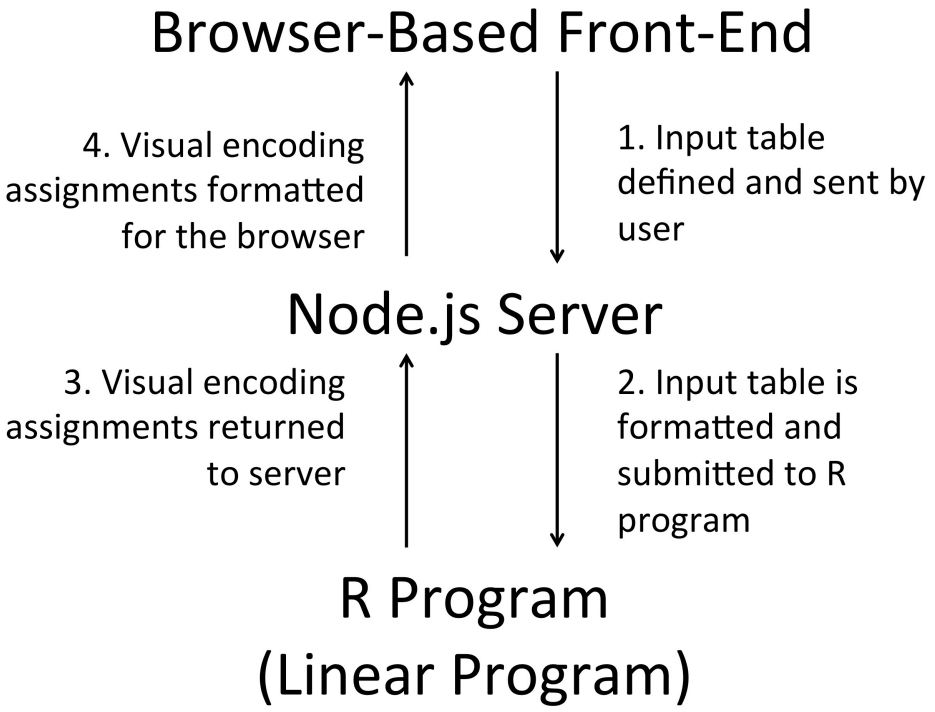
2. Additivity – the assumption that there are no non-linear interactions among variables in the model. That is, there are no situations where variables may interact to produce a value that is not linearly predictable.
3. Divisibility – the assumption that variable units can be subdivided into fractional levels (i.e. the solution to a LP is not guaranteed to be an integer).
4. Deterministic – the assumption that all of the parameters input into the model are known constants. Practically, this assumption is violated in a number of application areas, especially when LP is being used for prediction. In order to mitigate the risk of violating this assumption, sensitivity analysis is often used to understand the range of possible predictions.

Some of the assumptions above are limitations of the approach, whereas others are merely simplifying assumptions. In a number of cases, limitations imposed by certain assumptions may be overcome through proper reformulation of the problem (although this is out of scope and will not be covered in this dissertation). These assumptions just listed above are part of the design of Dynamo, and are implemented in the architecture.

6.8. Architecture

Dynamo is a web application written in Javascript and R. The code is available at http://www.github.com/ngopal/dynamo. Although the application has been implemented in Javascript and R, it is possible to implement the same application using another language or platform. Figure 25 below presents an overview of how the application architecture is designed:

Figure 25 - A schema of the various sub-systems that compose Dynamo.



The program obtains input from an input table in the browser, and the contents of the input table are sent to the Node.js server through a POST command. The communication that occurs between the Node.js server and the R program are essentially translation steps that format the input and output in appropriate forms. Once the input table is processed through the R code and results are returned, the encoding assignments are formatted into JSON and returned to the web browser for display processing in Javascript. In this case, the names of the visual encodings are mapped to style attributes defined in Cytoscape.js. Since the application calls R code on the backend, powerful graph analysis and bioinformatics libraries are readily available for use.

Within the R code, the "Rsymphony" library is used to interface R with command-line linear solver tools [81]. This library may be substituted for another as long as the output is guaranteed to be the same (so as to not violate the assumptions of the rest of the code downstream of that operation).

The network visualizations that are served in this version of Dynamo are implemented using a combination of D3.js and Cytoscape.js [82], [83]. However, the visualization libraries have been modularized from the visual encoding assignments, so in addition to being easily adaptable to major software design changes, new visualizations libraries may also be used in the future.

6.9. Example Configurations

In the Chapters 3 and 4 (systematic review of figures), a number of ranks were determined for visual encodings (on a per task basis). Those results are not reproduced here in this chapter, but can be referenced in Chapter 4, if desired. The reason those rankings are mentioned is because Dynamo is designed to take an input table populated with such rankings. In this subsection, I will proceed to explain and illustrate the method behind Dynamo.

6.9.1. Parameters

The application accepts three parameters: (1) an arbitrary number of encodings, (2) an arbitrary number of dimension-tasks, and (3) desired assignment prioritizations in the form of ranks. These ranks are the reverse of traditional statistical rank—thus, in the input table, the higher the number the higher its rank. In practice, the number of visual attributes any particular network visualization software supports limits the number of encodings. For instance, if a particular software package only allows node color and size to be specified, then that would clearly limit the number of encodings to a two possibilities.

6.9.2. Input Table Constraints

The application allows for constraints to be customized, although at the moment this must be accomplished by editing the code base. The "Future Work" section later in this chapter describes how constraints may be stored in their own format, and eventually into to shareable files.

Table 14 - A list of the constraints applied to the rankings contained in the input table. Please reference Table 13 and Table 15 for an example structure.

Constraint Number / Name	Description
1. Parity ("Balance Constraint")	This constraint ensures that assignments are "balanced" and that all of the assignments as equally weighted as possible.
2. Task Sum ("Assignment Rows")	This constraint ensures that the sum of the rows is less than or equal to 1.
3. Weight Sum ("Weightage Rows")	This constraint weights the assignments and are a direct reflection of the weights input into the Dynamo system.

Table 15 - Example 3x3 Input Table

	Dim-Task 1	Dim-Task 2	Dim-Task 3
Visual Encoding 1	3	3	2
Visual Encoding 2	1	1	3
Visual Encoding 3	2	2	3

Table 16 - Example Linear Programming Matrix

	VE1	VE1	VE1	VE2	VE2	VE2	VE3	VE3	VE3	P1	P2	Constraint
DT1	1	0	0	1	0	0	**1**	0	0	0	0	
DT2	0	**1**	0	0	1	0	0	1	0	0	0	
DT3	0	0	1	0	0	**1**	0	0	1	0	0	
W1	3	0	0	1	0	0	**2**	0	0	-1	0	
W2	0	**3**	0	0	1	0	0	2	0	-1	0	
W3	0	0	2	0	0	**3**	0	0	3	-1	0	
W4	3	0	0	1	0	0	**2**	0	0	0	-1	
W5	0	**3**	0	0	1	0	0	2	0	0	-1	
W6	0	0	2	0	0	**3**	0	0	3	0	-1	

6.9.3. An Example Computation

Table 15 represents an example input table. An input table such as the one found in Table 15 would be parameterized and submitted by a user through the web browser. Once the R program receives the input table (as depicted in Figure 25), it converts the information contained in Table 15 into what is seen in Table 16. The linear program actually runs on the information contained in Table 16. The columns in Table 16 represent visual encodings (VE) and the rows represent dimension-tasks and weights (DT and W). The 0 values are de-emphasized in gray. The only underlying constraint operations are that the selected items in each row must sum to 1—since there are three possible 1's in each of the DT rows, this ensures that only one of the 1's are selected. As Table 16 shows, the weights (W) are assigned directly by the user, as the contents of the input table are what populate the W rows. The objective function is to minimize the sum of the weights (sum of DT and W values).

Table 15 shows 9 variables (each cell of the table is a variable, that value of which is either 0 or 1 after the linear program has completed running). The first constraint listed in Table 14 defined a balance constraint (the P columns in Table 16), which ensures that encoding assignments are distributed as evenly as possible. Without this constraint, the linear program may select an assignment configuration that is both valid and optimal, but heavily unbalanced (e.g. certain encodings or tasks may receive all of the encoding assignments while others receive none). This balance constraint is enforced through the addition of two more variables to the linear program, which are set to be the minimum and maximum ranks per dimension-task. The formulation is setup such that the difference between the maximum and minimum value is minimized.

Table 17 - A solution to the LP devised in Table 16

Variable	V1	V2	V3	V4	V5	V6	V7	V8	V9	P1	P2
Input Table Values	3	1	2	3	1	2	2	3	3	-1	1
Assignment	0	0	1	1	0	0	0	1	0	3	2

Table 18 - Example Input Table with Assignments highlighted

	Dim-Task 1	Dim-Task 2	Dim-Task 3
Visual Encoding 1	3	**3**	2
Visual Encoding 2	1	1	**3**
Visual Encoding 3	**2**	2	3

The objective function resulting from solving the linear program is 7. This number is calculated by finding the sum product of the input table values and assignment values from Table 17, followed by subtracting 3 (-1 * 3) and adding 2 (1 * 2).

Equation 2 – Objective Function

The resulting visual encoding assignments are shown in Table 17 and Table 18, with the assignments highlighted in bold text and presented in the original input table format in Table 18. The example computation just shown is a scenario where there is exactly the same number of visual encodings as there are dimension-tasks. Dynamo actually supports a few more scenarios, which are detailed in the next sub-section.

6.9.4. Input Table Scenarios

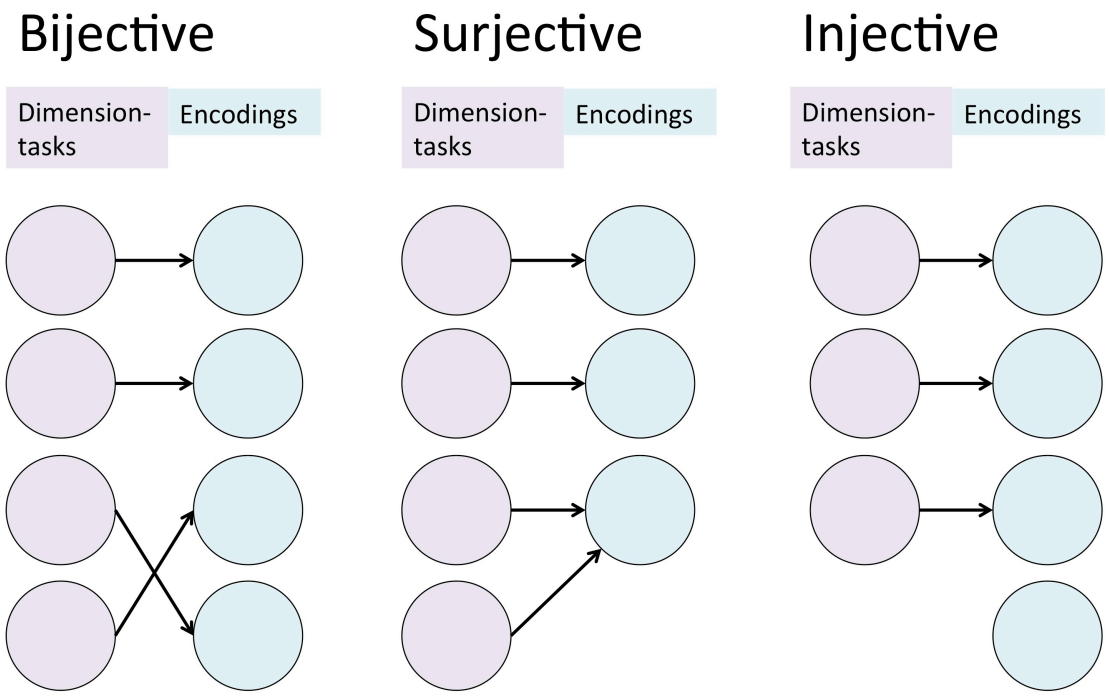
Dynamo has been designed to handle a variety of input tables of various dimension conformations. The conformation expected by the R code is one where there are more dimension-tasks (columns) than visual encodings (rows). However, the software is also designed to handle scenarios where there are less dimension-tasks (columns) than visual encodings (rows). The former scenario describes the typical scenario of a visualization author having to determine an optimal encoding assignment decision among a number of visual encoding options. The latter scenario describes the non-prototypical scenario of a visualization author having to determine an optimal assignment of data to visual encodings when there are more dimension-tasks than visual encodings—this results in the phenomenon of *encoding overloading*, which was described in Chapter 3. *Encoding overloading* is when a certain visual encoding is used more than once in the same visualization to represent distinct dimensions of data.

If the user does not want to perform the same task on different dimensions of data (e.g. "dimension 1 task 1", "dimension 2 task 1", "dimension 3 task 1"), then the corresponding

cells need simply be set to zero (or an appropriately low rank value, depending on the rank parameterization entered by the user).

There are three scenarios where the computational processing of input tables is affected. The first case is when there are an equal number of dimension-tasks and visual encodings. The second case is when there are more dimension-tasks than visual encodings. The third case is when there are more visual encodings than dimension-tasks. I will elaborate on these three of these scenarios below. The terms bijective, surjective, and injective are clarified in Figure 26 below.

Figure 26 – A visual depiction of dimension-tasks and encodings in context of bijective, surjective, and injective relationships. Bijective relationships are also injective and surjective.



In the scenario where there are an equal number of dimension-tasks and visual encodings, the underlying linear program is processing a square matrix. Ultimately, a square matrix would result in a bijective one-to-one mapping between dimension-tasks and visual encodings. The previous sub-section (example computation) illustrates the steps involved in processing a square matrix.

In the scenario where there are more dimension-tasks than visual encodings, the underlying linear program is processing a matrix that is larger in width than it is by height. Due to the unequal number of rows and columns in the matrix, the mapping would be injective or surjective depending on which (number of rows or columns) is fewer. If there are more

dimension-tasks than there are visual attributes available to map, then the mapping would be surjective (multiple dimension-tasks may be encoded to a single visual attribute). On the other hand, if there are fewer dimension-tasks than there are visual attributes, then the mapping would be injective (every dimension-task would be uniquely mapped to an available visual attribute). The smaller of the two dimensions becomes the limiting factor, so in practice, the input table is simply transposed for computation. The constraints are defined broadly enough that they do not need to be modified. Multiple research papers have shown that visualizations intended to satisfy the requirements of multiple tasks risk becoming cluttered, unmanageable, and less interpretable visualizations [43], [62], [84], [85]. However, this sort of parameterization is still valuable to support due to requirements imposed by certain mediums, such as print publications.

In the scenario where there are more visual encodings than dimension-tasks, computation proceeds on the supplied matrix without any modification. Although Dynamo supports the two scenarios list above, this particular scenario is what Dynamo was intended for. Given the contents of the input table, the linear program will select as many visual encodings as there are dimension-tasks.

6.9.5. Visual Encodings in Dynamo

The following section presents a table containing the visual encodings currently available in Dynamo. Since Dynamo is currently built using Cytoscape.js and D3.js, the collection of visual attributes available for use are subject to the bounds stemming from a combination of the abilities of these software libraries and suitable standardized web technologies (e.g. Scalable Vector Graphics, Canvas, etc.). D3.js and Cytoscape.js provide convenient data binding, analysis, and visualization capabilities, enabling highly expressive and customizable web visualizations. In reference to the list presented below, visual encoding "types" are followed by text in parentheses, which is an abbreviation for the same term. These are essential descriptors, and their meaning is explained below:

- Sequential (Seq) – Sequential refers to a sequential color scale (e.g. 10 color steps starting from white ranging up to red)
- Diverging (Div) – Diverging refers to a diverging color scale (e.g. 10 color steps starting from blue, turning to white, followed by another transition from white to red).
- Categorical (Cat) – Categorical refers to nominal data. This could be in the form of a categorical color scale (e.g. a color scale where hue is used to represent various categories) or in reference to a number of discrete shapes (e.g. circle, triangle, square, star, rhomboid, etc.)
- Quantitative (Quant) – Quantitative refers to numerical data that may be mapped directly to a visual attribute. For instance, the numbers 15-30 may be mapped directly to node radius. Admittedly, the term "quantitative" has purposely been loosely defined to accommodate non-linear functions.
- Binned (Bin) – Binned refers to the result of transforming quantitative data into discrete categories. For instance, the numbers 15-30 may be transformed into 3 bins: 15-20, 21-25, and 26-30.

In general, visual attributes that may be encoded with quantitative input data may visually encode quantitative output, or binned output. On the other hand, visual attributes that may be encoded only with nominal input data may only visually encode categorical output. The validity of visual encodings was covered in greater depth in Chapters 3-5. The following

visual encodings are currently available for use in Dynamo (supplemented with the encoding "type" explained above):

1. Node Color (Seq)
2. Node Color (Div)
3. Node Color (Cat)
4. Node Shape (Cat)
5. Node Border (Quant)
6. Node Border (Bin)
7. Node Size (Quant)
8. Node Size (Bin)
9. Edge Width (Quant)
10. Edge Width (Bin)
11. Edge Color (Seq)
12. Edge Color (Div)
13. Edge Color (Cat)
14. Edge Pattern (Cat)

In the source code, compatibility is defined through a hash data structure. That is, visual encodings (e.g. "Node Border (quant)") are connected to a specific function (e.g. a function that directly maps attached node attribute data to node border thickness). In addition to the function, a style attribute (e.g. "border-width") is required, as the defined attribute is passed as an argument to the previously mentioned function. The hash structure also contains data on whether it specifies a node or edge encoding, the range of valid output values, and an array that contains a list of other visual encodings the visual encoding at hand is incompatible with. D3.js is used to define and implement the functions that map input data to output data, and Cytoscape.js is used to attach the output data to visual attributes in the network. The functions only accept numerical input, but the output type may be sequential, diverging, categorical, binned, or quantitative.

As a small clarifying note, one may wonder about the topology of the graph itself. Until this point there has been no mention of the structure of the graph. How does one detail the connectivity between nodes, or the attributes attached to nodes and edges? Currently, Dynamo requires that nodes and edges be specified using the format in the Cytoscape.js documentation. The format essentially describes an edge-list, but is implemented as a Javascript object, so it also has a few additional properties that need to be included in order for Cytscape.js to accept it as valid input.

6.10. Benchmarks

Although the name Dynamo refers to the entire application from web page to R code, the assignment optimization step (in the R code) is the component that is most likely to produce a substantial bottleneck if the input table is too large. In order to assess the practicality of Dynamo, this assignment code was tested with random input tables of varying sizes and dimensions.

These input tables were generated by a process where every column in a NxN table was populated with random integer values ranging from 1 to N. Duplicate rankings were permitted, so any number between 1 and N may appear more than once in a column—this could represent visual encodings that are "tied" in prioritization or rank. The input tables were also designed to be square. Overall, the results of this benchmark evaluation more than

accounts for computational feasibility, as there theoretically should not be a scenario where one would define 40 visual encodings and 40 dimension-tasks. Nevertheless, the benchmark for this input table (and several other more sensible input tables) is provided in Table 19. To generate the data in Table 19 and Figure 27, six input tables of the same NxN size were processed one after the other on my 13-inch 2012 Macbook Pro (2.5 GHz Intel Core i5, 8GB 1600 MHz DDR3 RAM, SSD, OSX 10.11.3).

As Figure 27 illustrates, the time to compute scales up quite quickly past a 30x30 input table. Prior to that size, worst-case computational processing takes less than 10 seconds. The input tables given to Dynamo in the study detailed in Chapter 7 used input tables that were roughly 16 (visual encodings) x 4 (dimension-tasks), which are estimated to take less than 3 seconds each to complete according to Figure 27. Figure 28 and Figure 29 provide screenshots of the interface of Dynamo, and a close-up of a network visually encoded through Dynamo (respectively). The network used in Figure 28 was obtained by performing a random walk starting from a randomly selected node from the larger GeneMANIA protein-protein interaction network [45].

Table 19 - Benchmarks for Dynamo as determined on a 13-inch 2012 Macbook Pro (2.5 GHz Intel Core i5, 8GB 1600 MHz DDR3 RAM, SSD, OSX 10.11.3). The columns define the size of the input tables, the rows represent replicates, and the cell values show elapsed time in seconds.

	5x5	**10x10**	**15x15**	**20x20**	**25x25**	**30x30**	**35x35**	**40x40**
1	0.331	0.743	1.975	4.034	4.739	7.312	6.948	19.645
2	0.326	0.819	1.984	4.674	4.659	7.591	6.753	18.472
3	0.232	0.703	1.934	4.552	9.095	8.933	8.514	21.612
4	0.369	0.843	2.24	5.422	5.55	5.083	9.548	12.54
5	0.207	0.972	1.947	5.3	5.007	7.581	11.257	12.853
6	0.325	0.947	1.942	6.035	6.819	6.786	28.629	33.958
Min	*0.207*	*0.703*	*1.934*	*4.034*	*4.659*	*5.083*	*6.753*	*12.54*
Mean	*0.298*	*0.837*	*2.003*	*5.002*	*5.978*	*7.214*	*11.941*	*19.846*
Max	*0.369*	*0.972*	*2.24*	*6.035*	*9.095*	*8.933*	*28.629*	*33.958*

Figure 27 – Performance of Dynamo for square matrices containing random ranks (as matrices scale in size).

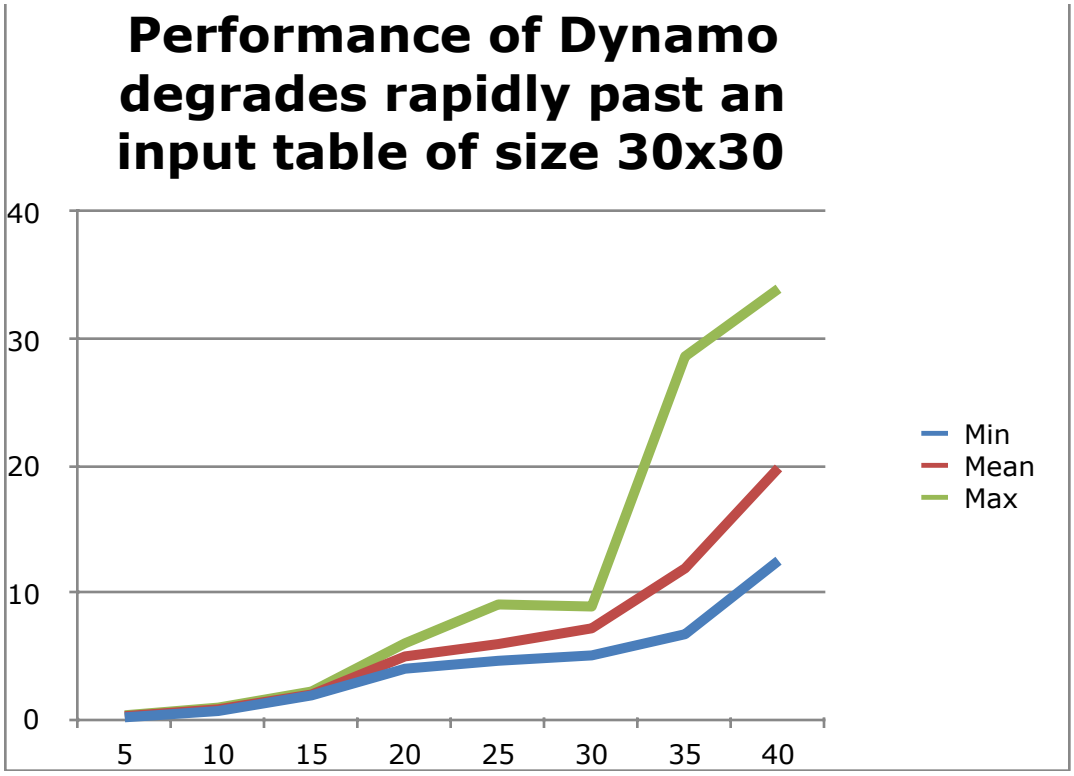
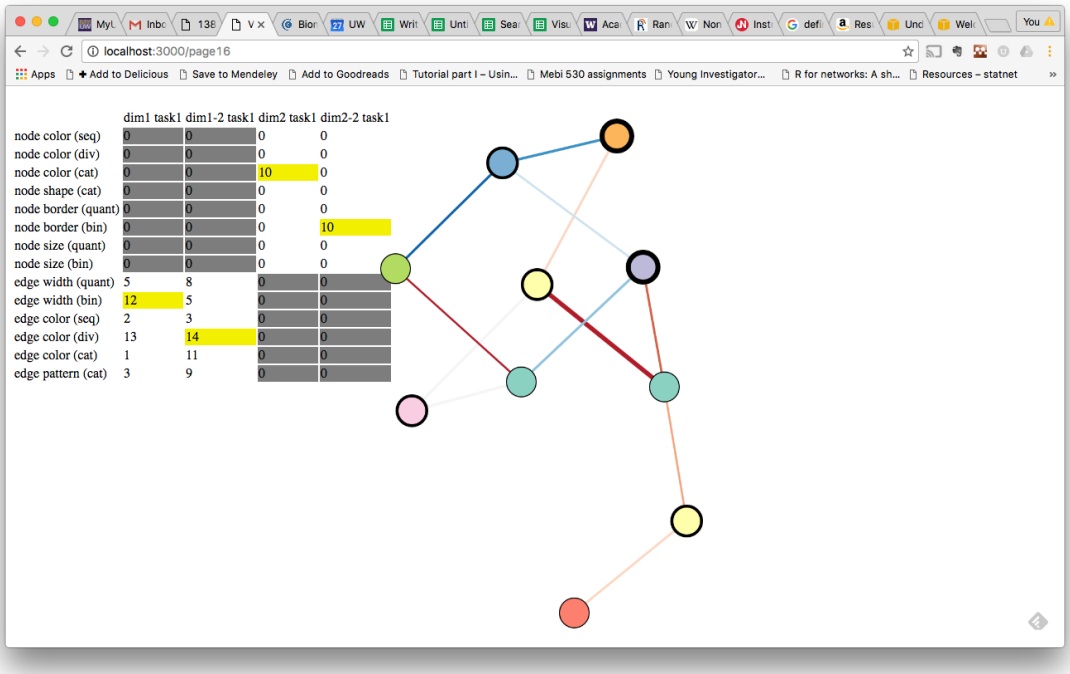
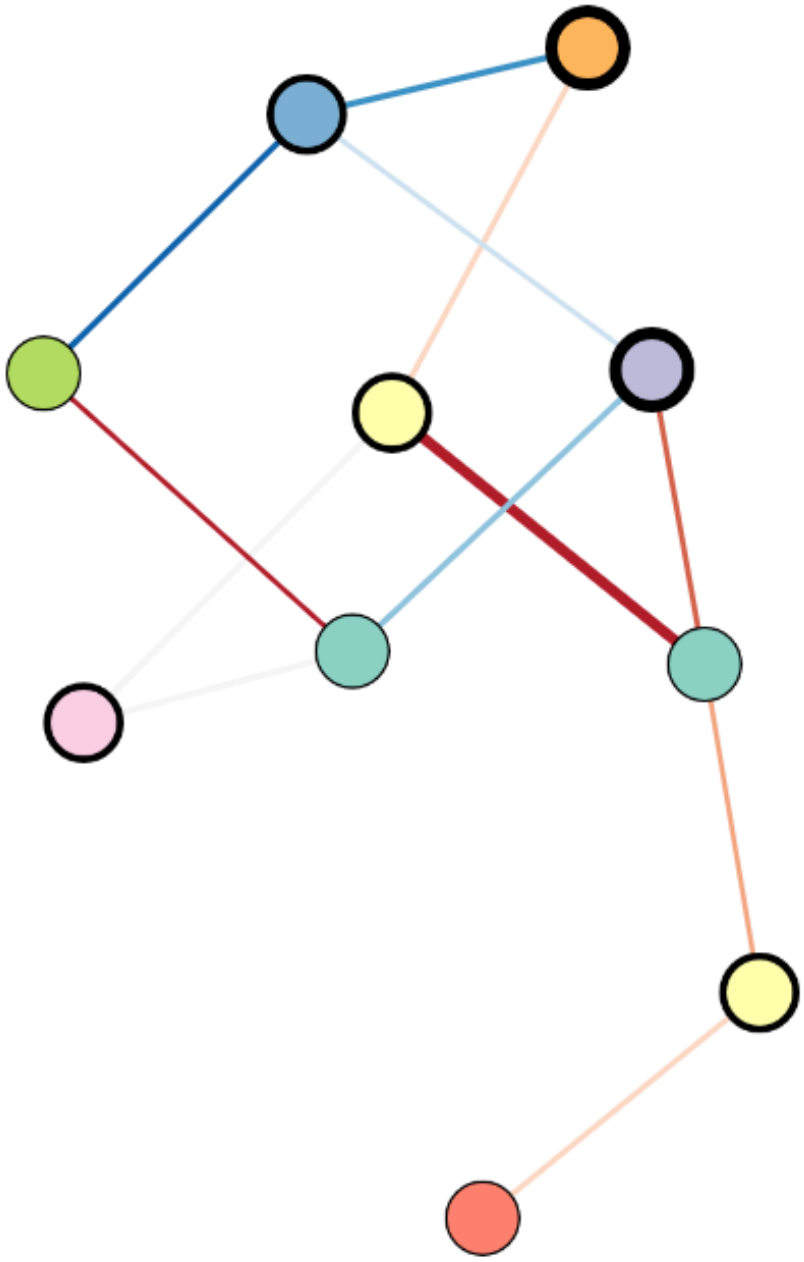


Figure 28 - A screenshot of the input table, resulting visual encoding assignments, and depiction of the visual encodings on a biological sub-graph.



	dim1 task1	dim1-2 task1	dim2 task1	dim2-2 task1
node color (seq)	0	0	0	0
node color (div)	0	0	0	0
node color (cat)	0	0	10	0
node shape (cat)	0	0	0	0
node border (quant)	0	0	0	0
node border (bin)	0	0	0	10
node size (quant)	0	0	0	0
node size (bin)	0	0	0	0
edge width (quant)	5	8	0	0
edge width (bin)	12	5	0	0
edge color (seq)	2	3	0	0
edge color (div)	13	14	0	0
edge color (cat)	1	11	0	0
edge pattern (cat)	3	9	0	0

Figure 29 - A closer view of the biological sub-graph from Figure 28.



6.11. Discussion

One of the useful benefits of Dynamo is that is affords the ability to rapidly iterate over a large number of visual encoding configurations in a systematic and reproducible manner. Another benefit of using Dynamo is that it supports an explanation of the design decisions that contributed to creating a network visualization. Although Dynamo currently does not include any presets, generating a number of preset rankings based on tasks would be a useful extension of the tool. As you will see, the study conducted in Chapter 7 is a step towards

this vision. Lee et al provides a solid foundation for research on tasks performed on graphs, although it would also be beneficial to conduct a thorough, more detailed assessment of tasks related to biological network visualizations [46]. "Tasks" has been defined broadly enough that virtually anything could be defined as a "task" and it would still be valid in context of the Information Triad. However, the end-user would see limited returns if this were the case. Specifying "task" with a greater level of precision will produce superior visual encoding assignments. Through the Information Triad, data, visual encodings, and tasks need to be carefully defined in order to obtain useful results. Furthermore, as long as at least two of the three components are defined, the third component may be determined through an experiment (more on this in Chapter 7). Moreover, the Information Triad may conceivably be extended to work on interactive network visualizations, although in that scenario, visual encodings would also be conditional on task (rather than only dimension). Using Dynamo to obtain preset rank prioritization can be valuable; saving time, supporting reproducibility, and outlining an open reasoning process for design decisions about representation.

Although 14 visual encodings are currently supported by Dynamo (presented in a list in the *Visual Encodings in Dynamo* sub-section), this may be extended in the source code. As mentioned earlier, the software libraries that are used to implement Dynamo limit the variety of possible visual encodings. In the grand scheme, visual perception limits the use of available visual encodings may impose more limitations than the variety of visual encodings available for use. If Dynamo were re-implemented in another language with an associated software package that offered more granular parameterization of visual attributes, more visual encodings could theoretically be supported.

6.12. Future Work

Cytoscape (written in Java) is a fantastic candidate platform for an extension of Dynamo. Cytoscape has a community of users that create and share open-source plugins—reimplementation of Dynamo as a Cytoscape plugin would be useful future work, primarily because it would make the application available to a large community of researchers that use biological networks. Although Cytoscape.js was the fastest way to implement a working version of Dynamo, and is a good choice for the objective, future visualization libraries should be considered as features are added to Dynamo.

The topology of the graph is contained in a separate file from the visual encoding assignments. Although Dynamo currently accepts graphs in the format required by Cytoscape.js, the application may be extended to accept any standardized (or custom) graph format—interconversion between graph formats would be possible thanks to the built-in functions of the iGraph library in R [33]. By using the conversion functions in iGraph, any graph format supported by iGraph can be converted into the format accepted by Cytoscape.js.

Dynamo can be extended to visualize the results of various network visualizations generated from a number of bioinformatics methods. Since many biological network algorithms are implemented and readily useable in R through BioConductoR, the assignment algorithm can connect to methods in BioConductoR (working in conjunction or unison) [32]. This would be a useful extension, as using Dynamo in combination with the Information Triad would allow researchers to identify the "best" visual representation (in context of tasks, data dimensions, and visual encodings) for the results they obtain from popular BioConductoR packages.

Furthermore, D3.js has recently been updated to support plugins, which allows developers to modularize visualizations and the functions that support them. Dynamo could potentially be implemented as a D3.js plugin as Dynamo continues to mature.

As seen in the benchmarks subsection of this chapter, the performance of visual encoding assignments decays quite rapidly since linear programming can be completed in polynomial time in the best case. However, as one might expect, larger input tables require an increased amount of compute time and resources. Although it is possible to obtain a valid solution for many networks (even large ones), the application is not optimized or tuned for performance in any way.

No auxiliary constraints are defined or implemented in the R program at the moment. However, there are a number of constraints that could be added to the program in order to facilitate specialized visual encoding assignments. For instance, Jacques Bertin defines a visual hierarchy in Semiology of Graphics, and this hierarchy could be translated into a set of constraints that set bounds on visual encoding assignments [85]. Although these may be valuable constraints to enforce in context of visual encoding assignment, the principles being enforced must be validated before being included as a default option. Visual encoding constraints and rules from the field of cartography may have received more careful study, and could prove to be, at the very least, a promising source of inspiration for future constraints [86]. Future versions of Dynamo will support an external constraints file that will contain all of the constraints that are to be applied during visual encoding assignment. An externally defined constraints file would further promote the sharing of constraint files, support the reproducibility of assignment results, and would organize the constraints in such a way that they would be organized (and perhaps even indexed for record-keeping purposes). Furthermore, externally defining constraints also provide the foundation for LPs that include constraints that are higher complexity (e.g. combinations and interactions).

6.13. Conclusion

This chapter detailed Dynamo, a system founded on the ideas of the Information Triad. Using Dynamo, one can explore the set space of visual encodings for any particular graph in any (of the supported) graph layouts. As the following chapter will show, Dynamo can also be used to administer studies that are aimed at solving for a missing component of the Information Triad.

7. Evaluating Prominence of Visual Encodings using Dynamo

7.1. Introduction

The previous chapter explained Dynamo, a system to optimally assign visual encodings based on an input table containing prioritizations of visual encodings for tasks. In context of the click-based perception study detailed in this chapter, Dynamo acts as the underlying engine for a network visualization tool designed to serve visual encodings in a reproducible, systematic manner. This chapter presents the results from a click-based perception study designed to measure the effect of visual encoding parameterizations on saliency of nodes and edges. In particular, one of the analysis goals of this study is to develop a model that can predict which node or edge a user will select for the scan task (as defined by Lee et al) [46].

7.2. Background: about the design of this study

There is support for the hypothesis that visual encodings affect interpretation of content in visualization literature [47]. Accordingly, there is also reason to believe that graph layouts may be improved by better understanding visual encodings.

One constraint influencing the design of this experiment is the need to obtain a large sample size. Although random forest is an analysis method robust for situations where N (sample size) < P (number of predictors), it is still better to obtain a large N for improved performance and the ability to use another analysis method. Since I do not have the resources to compensate participants for their time, I must keep the experiment as short as possible to maximize potential participation. I will not expand on the details of the Random Forest algorithm in this chapter. For additional information about the Random Forest, please reference the background section of the previous chapter, or the publication by Breiman et al [72].

7.2.1. Description of the study

Study participants were presented with one of 34 randomly selected networks (the process by which a network was randomly selected is covered in Section 7.3.1). 34 networks were presented to participants because 34 different pairs of visual attributes were chosen for encoding. Participants were prompted with a network and asked to select, "the most noticeable node," or "the most noticeable edge," in the network. Figure 30 and Figure 31 show an example of how the topology of a network is separated from the visual encodings —in these figures, for each of the rows, the text on the far left signifies the visual attributes being used for encoding, the networks in the middle depict the topological structure of the network, and the networks on the far right show the network presented in the middle with data encoded by the designated pair of visual attributes. Every visual encoding combination is compared across every network used in the study. Furthermore, Figure 32 shows four networks that may have been served to participants during the course of the study.

Random forest regression was used to model selectivity based on a number of variables representing visual encodings, and other properties of networks (e.g. topology). Put another way, the goal is to estimate how likely a participant is to select a node or edge based on its visual attributes.

Figure 30 – An example figure showing different networks where the same data attribute is visual encoded through a pair of visual attributes.

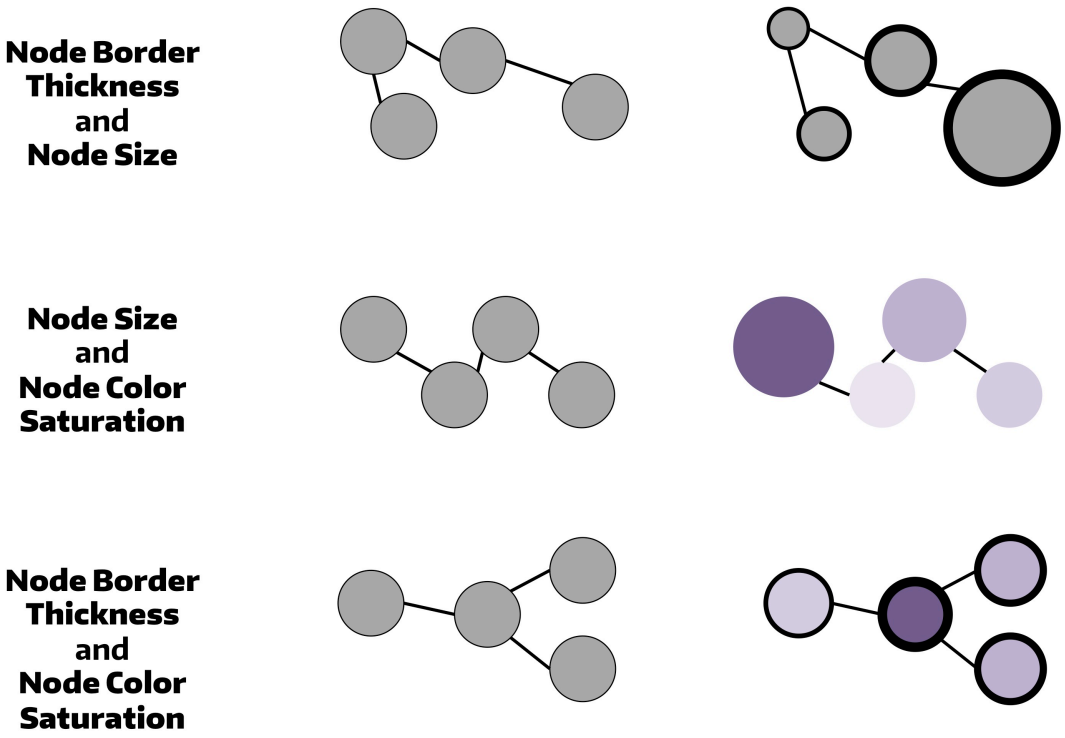


Figure 31 - An example figure showing the same network (i.e. the same data attributes) visually encoded through a different pairs of visual attributes.

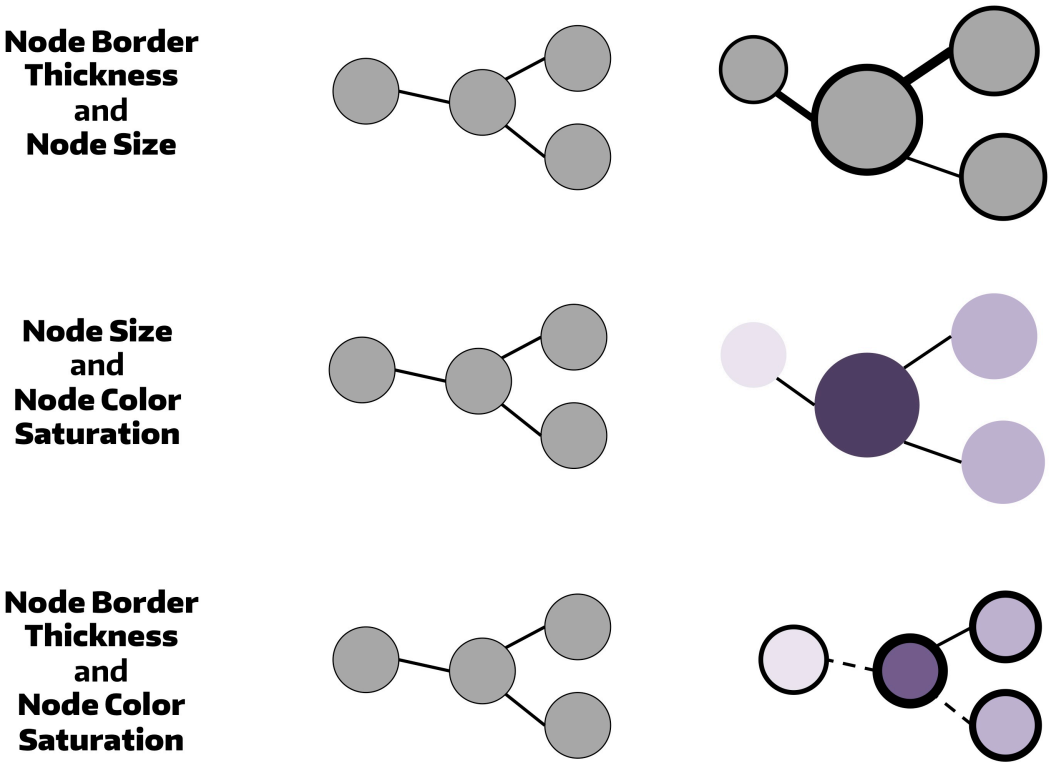
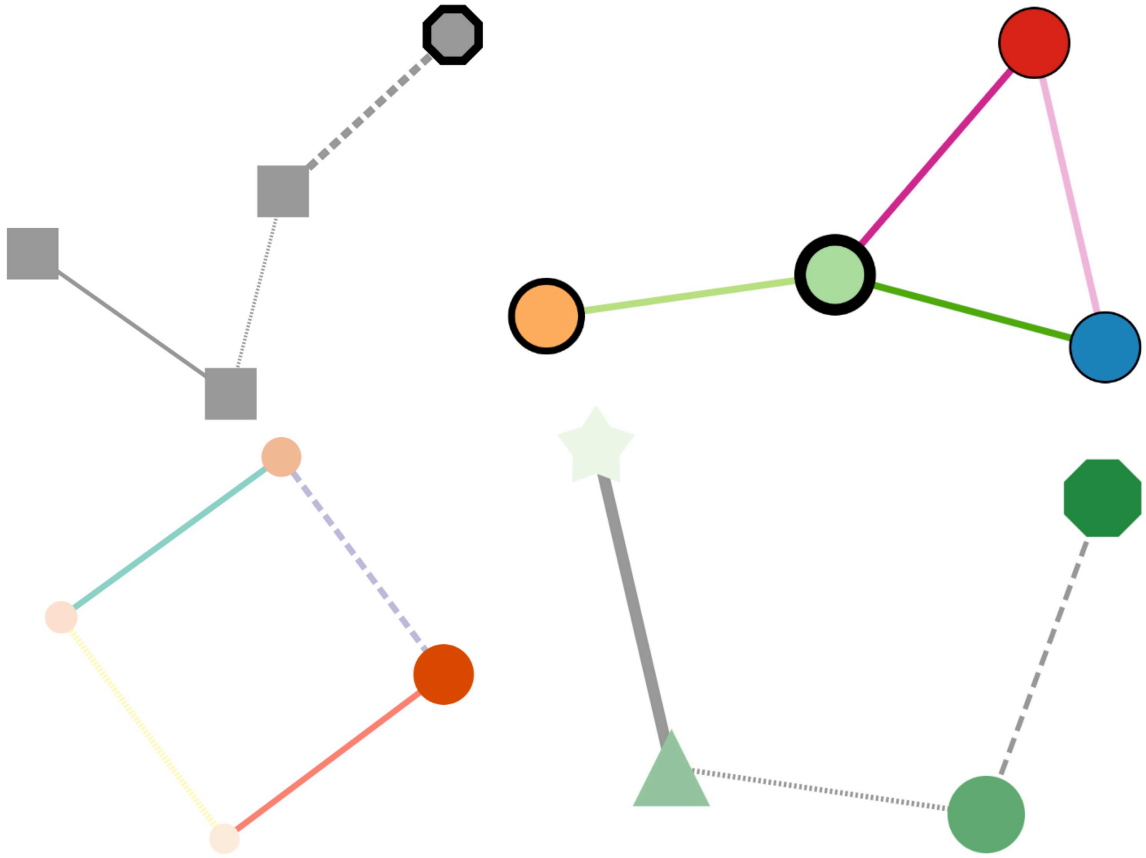


Figure 32 – A figure illustrating four different networks and encoding pairs that were served to participants during the course of the study.



From a machine learning perspective, the performance of an algorithm can be generalized through what is known as the bias-variance trade-off. Bias is a description of how systematically erroneous the prediction of an algorithm is (independent of input, and typically a result of assumptions contained in parametric techniques)—those algorithms that are systematically "off-target" are said to have a high bias. Variance is a description of how much the prediction of an algorithm fluctuates depending on the values of the predictor variables—those algorithms that fluctuate more than others are said to have high variance. The bias-variance trade states that methods with high bias also have low variance, that methods with low bias also have high variance, and that this trade-off is ever-present and largely unavoidable. The general approach to handling this trade-off is to minimize extremes, although this heavily depends on the problem that is being modeled, as minimizing the trade-off may not be necessary if the problem is framed in a convenient manner. Regression has a high bias, but a low variance (which has a different meaning within the context of a regression model). Decision trees have a low bias, but a high variance. Random Forest is bootstrapped, and consequently generalized over hundreds or thousands of decision trees, so the high variance in the bias-variance trade-off is curbed. Regression is not an ideal choice to use for analysis of this data for a number of reasons (and high bias being one of them). Although the collected data may also be analyzed through logistic regression, the resulting model would be quite complex. First, it is already clear that the variables that would be in the logistic regression model are not independent. Second, the model would need to include

mixed effects factors, as participants add random effects, and the presented networks add fixed effects. Third, the underlying distribution of the data was unclear (until the end of this study), so there was uncertainty around the parameterization of the model—should one select OLS, weighted OLS, or maximum likelihood as the objective function? Should one select a linear or non-linear model? Proper parameterization would obviously reduce the bias in the bias-variance trade-off, but the risk of presuming incorrect or false parameterizations is too high (as the research question posed in this dissertation is largely understudied). Thus, my preference is for a non-parametric technique.

The experiment below is founded on the hypothesis of the "Information Triad" (IT) defined in the previous chapter. To frame this experiment in context of the IT hypothesis:

- *Task*: Scanning
- *Data*: GeneMANIA subgraphs generated via random walks
- *Visual Encoding*: Determined via Dynamo

Only one *task* was evaluated in this experiment, which was *scanning,* referring to the ability to visually scan a network [46]. The *data* that was visually encoded was derived using random walks on a GeneMANIA network (further explained in the Methods section) and cross-referencing Pubmed. The visual encodings used to represent the data were delivered using the Dynamo tool (also further explained in the Methods section).

Furthermore, the study was designed such that the same data would be double encoded to create competition between the saliency of the visual attributes used to encode the data. Administering this experiment through a computer is preferable as it is how biological researchers seem to create and explore network visualizations (as inferred from the findings in Chapter 2) – be it through an application deployed via a web browser, or a domain-specific tool such as Cytoscape [31].

7.3. Method

This study uses a fractional factorial design and presents a number of networks with carefully chosen visual encoding combinations. Those participating in this study are presented with a number of networks, varying in size, shape, color, etc.—they must visually scan the network and click on the element that is most noticeable. In context of the graph tasks covered in Chapter 4, the task evaluated in this study is "overview," which involves scanning an entire graph. This study was approved by the University of Washington Institutional Review Board.

7.3.1. Generation of networks

The characteristics and underlying graph model of a presented network greatly affect visual analysis and interpretation. Through some preliminary work exploring the set space of potential parameter values, I estimated that a network with 4 nodes and 3-5 edges would be ideal for this experiment. One may consider 4 nodes to be too few—however, for a study that is to be conducted completely online, likely even on mobile devices, and attempting to measure saliency of visual encodings (rather than topological properties), this is an ideal size. Although there are estimated specifications for the number of nodes and edges in a network presented to a participant, the method used to obtain representative connectivity is described next. In order to obtain an accurate representation of the interconnectedness of biological networks, I downloaded the network used in GeneMANIA, and used random walks to obtain subgraphs (from randomly selecting starting nodes in the GeneMANIA network) that

meet the aforementioned specifications. This protocol yielded a number of representative small biological networks.

7.3.2. Recruiting and Sampling of Participants

The inclusion criteria defined for participants in this study are: (1) participants must be able to complete the experiment from a compatible, web-connected device. The exclusion criteria defined for participants in this study are: (1) participants must be at or over the legal age of majority where they are participating, and (2) participants must be fluent in English. Calls for participant were posted on Reddit and a number of University of Washington community mailing lists. Due to the stipulations of the IRB, I am unable to estimate the number of participants that joined the study from each source.

7.3.3. Measured Variables

The presented networks also have a number of attached style attributes. The states of these style attributes are computationally determined using the Dynamo system (detailed in the previous chapter). These style attributes, along with information about clicks, element positions, etc., are captured in JSON format. The full list of captured attributes is provided below in Table 20 through Table 22. Although Table 20 and Table 22 both contain captured variables, a number of other variables may be derived from these data points. Additional variables obtained in this manner are exemplified in Section 10.7.

Table 20 - A list of collected data variables that describe node encodings. The * denotes HSV values that were converted from hexadecimal color values.

Variable	Meaning	Input Data Type	Input Parameterization	Output Data Type
Node Border Width (nodeborderwidth)	Refers to the size of the border around a node (if any)	Numeric	1.5px to 4.5px	Quantitative or Binned
Node Size (nodeheight)	Refers to the size of a node	Numeric	15px to 30px	Quantitative or Binned
Node Shape (nodeshape)	Refers to the shape of a node	Categorical	Randomly select 1 of 12	Categorical
Node Hue* (nodeHue)	Refers to the (color) hue of a node	Numeric	0-1 units	Categorical
Node Value* (nodeValue)	Refers to the (color) value of a node	Numeric	0%-100%	Categorical
Node Saturation* (nodeSaturation)	Refers to the (color) saturation of a node	Numeric	0%-100%	Categorical

Table 21 - A list of collected data variables that describe edge encodings. The * denotes HSV values that were converted from hexadecimal color values.

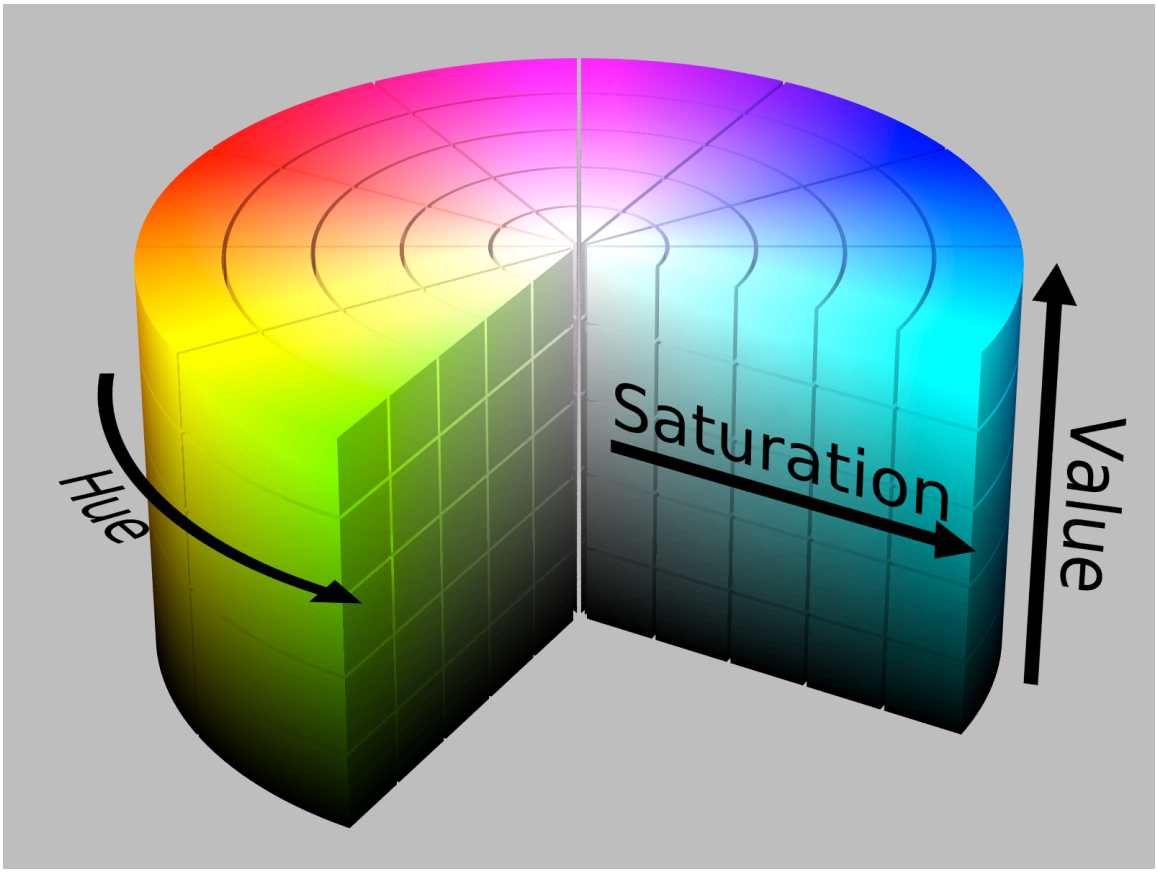
Variable	Meaning	Input Data Type	Input Parameterization	Output Data Type
Edge Width (edgewidth)	Refers to the size (thickness) of an edge around a node.	Numeric	2px to 8px	Quantitative or Binned
Edge Pattern (linepattern)	Refers to the pattern used to represent an edge	Categorical	Randomly select 1 of 3	Categorical
Edge Hue (edgeHue)	Refers to the (color) hue of an edge	Numeric	0-1 units	Categorical
Edge Value (edgeValue)	Refers to the (color) value of an edge	Numeric	0%-100%	Categorical
Edge Saturation (edgeSaturation)	Refers to the (color) saturation of an edge	Numeric	0%-100%	Categorical

Table 22 - A list of collected data variables that describe the randomly sampled networks.

Variable	Meaning	Input Data Type	Output Data Type
Number of Edges (numEdges)	Refers to the total number of edges in the visualization	Numeric	Quantitative
Network (network)	Refers to 1 of 34 randomly sampled, 4-node biological networks.	Categorical	Categorical
Node Degree (numConnected)	Refers to the number of edges that connect to a node	Numeric	Quantitative

Although hue is generally captured and processed as a categorical variable, by converting hexadecimal colors to the HSV color scale, hue is scaled between 0 and 1. In the HSV color space, hue is actually circular (See Figure 33). Random Forest is capable of gracefully handling hue in this manner, although it would be troublesome for a regression model, where one may have to estimate and label hues by their closest ROYGBIV color (and treat those labels nominally, rather than numerically). Saturation and value are also on 0 to 1 scales, running from 0% saturation to 100% saturation, and 0% value (black) to 100% value (white), respectively.

Figure 33 - A depiction of the HSV color space [87].



The dependent variable that is being measured is referred to as "selectivity." Selectivity is the variable name describing how salient a particular visual encoding of a node or edge may be, and ranges from 0, not salient, to 1, extremely salient.

7.3.4. Sampling Networks for Analysis

Random Forest is very sensitive to class balance for the response variable (in this study, it would be "selected" and "unselected"). If the minority class is less than 15% of the data, then the results will surely be skewed. As the number of nodes and number of edges are scaled up, the minority class becomes less prevalent. Since the number of nodes is held constant in this study, this is not a problem. However, in order to obtain results that are well balanced, the following sampling method was used:

For each user
For each network
1. Identify selected node or edge
2. Randomly select one (of several possible) unselected nodes or edges

This process of sampling ensures that the dataset provided as input into Random Forest will have satisfactory class balance (50% majority class and 50% minority class).

7.4. Results

This results section is very long, so I will briefly provide an overview of the contents of this section. This section begins by providing demographic information about those who participated in the study. Following the demographic information is a sub-section about evaluating the selectivity of visual encodings in network visualizations.

7.4.1. Demographic Information

The participants in this experiment may generally be described as non-colorblind 25-34 year olds who are in- or have completed graduate school, and does not have prior experience with network. All of this information is summarized in Figure 34 through Figure 37.

Figure 34 - An overview of the age range of participants.

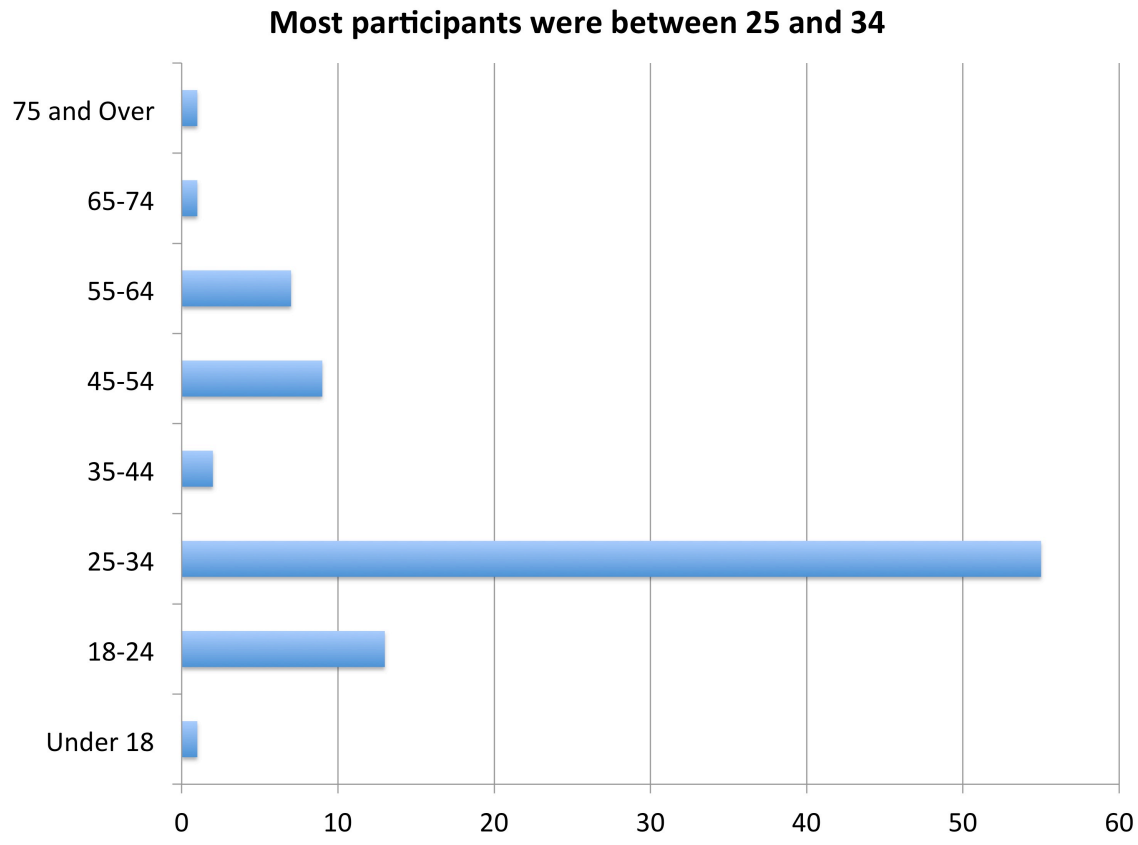


Figure 35 - An overview of the range of educational background of participants.

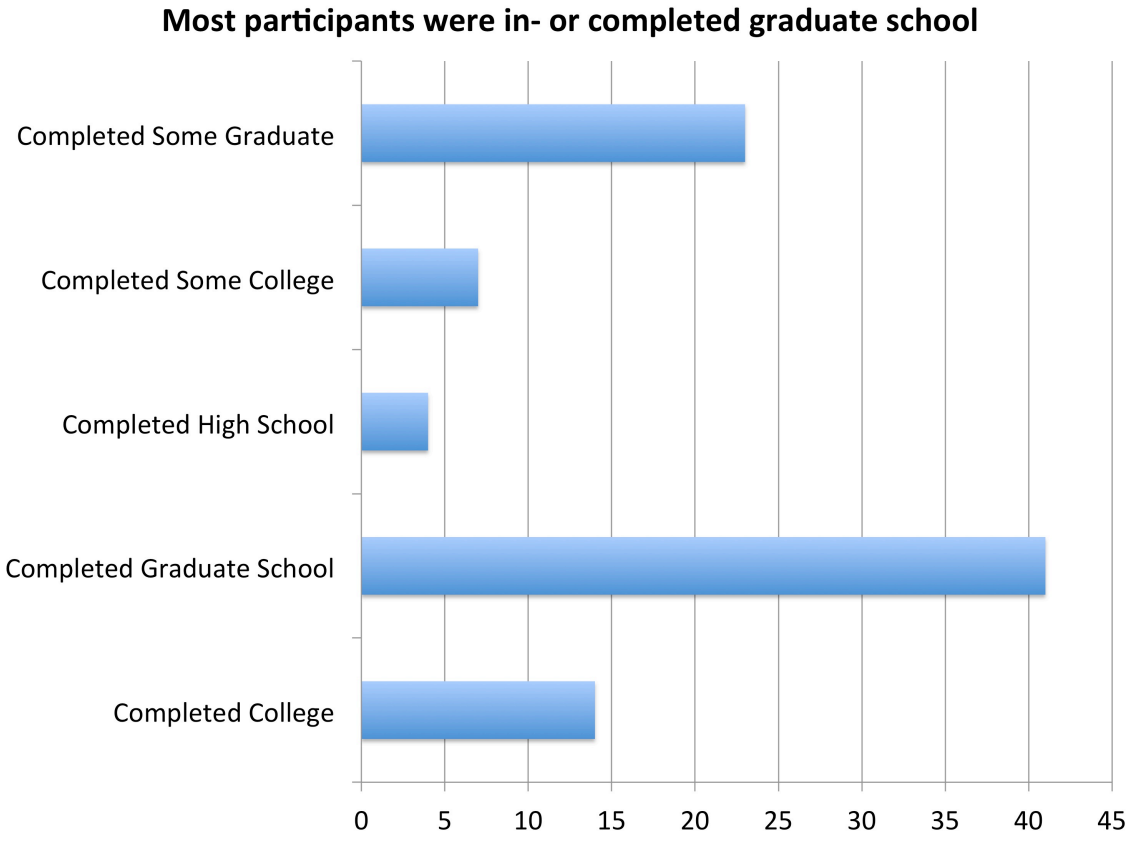


Figure 36 - An overview of prior experience participants' had with networks.

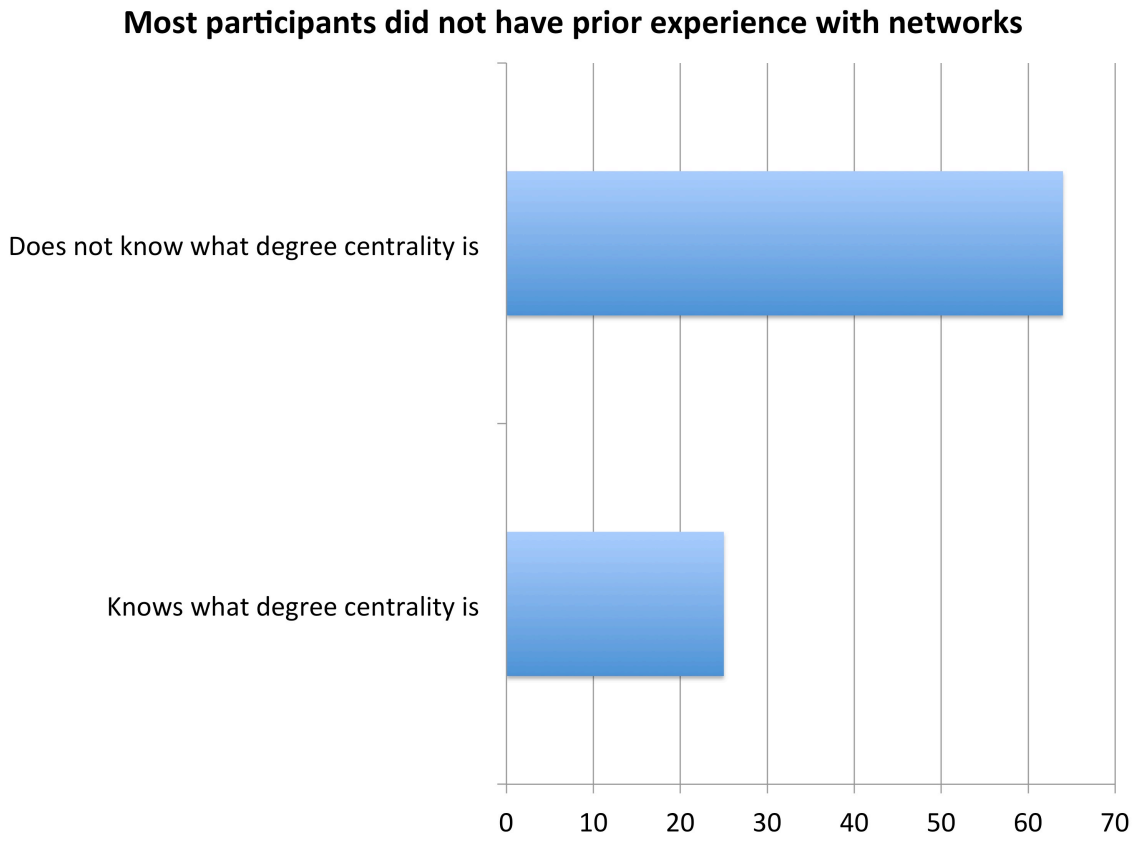
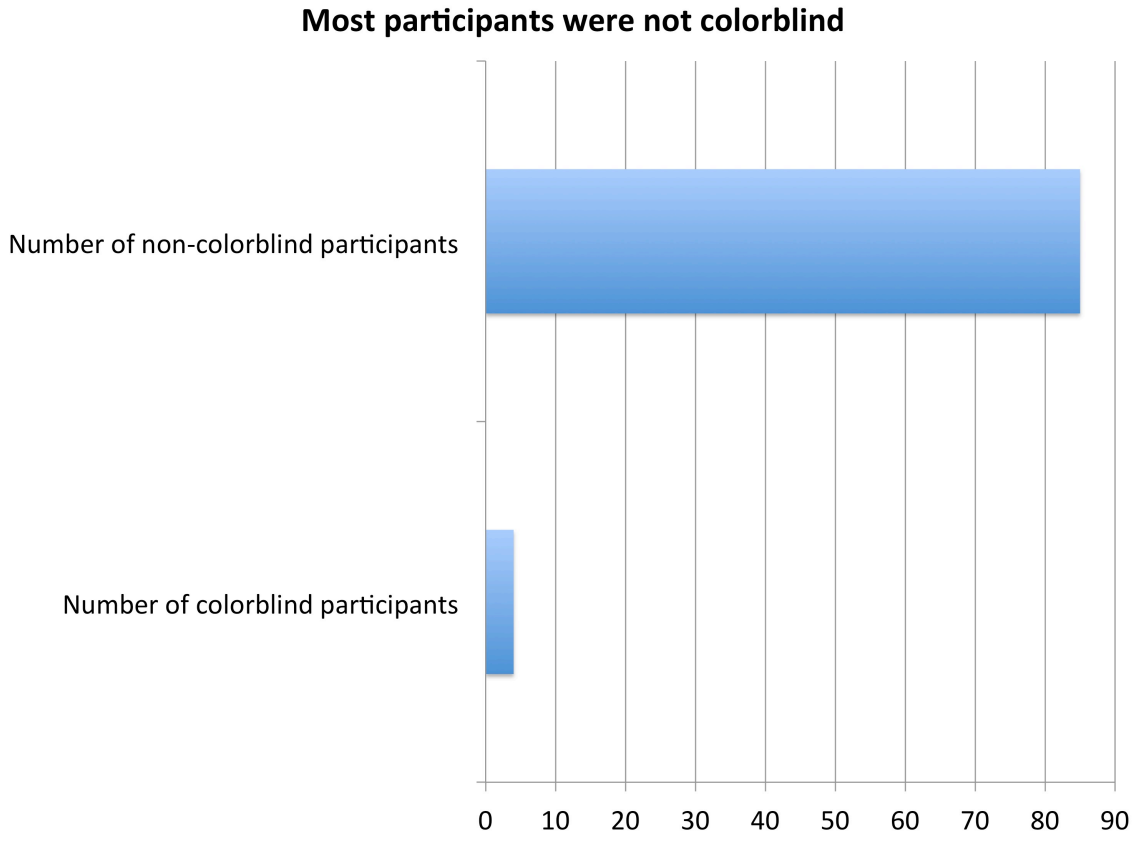


Figure 37 - An overview of the proportion of participants that were colorblind.



7.4.2. Evaluating Saliency of Visual Encodings in Networks

The resulting data were analyzed using Random Forest Regression. Random Forest Regression produces a handful of metrics that allow one to understand the performance of the algorithm. In addition to this, I provide partial dependency plots, which shows the partial contribution of each variable input into the Random Forest algorithm, the distribution of data (selected and unselected), and estimates a mathematical function that fits that data. Data from participants who were colorblind were removed from the data used for this analysis.

7.4.3. Performance of Random Forest Model for Nodes

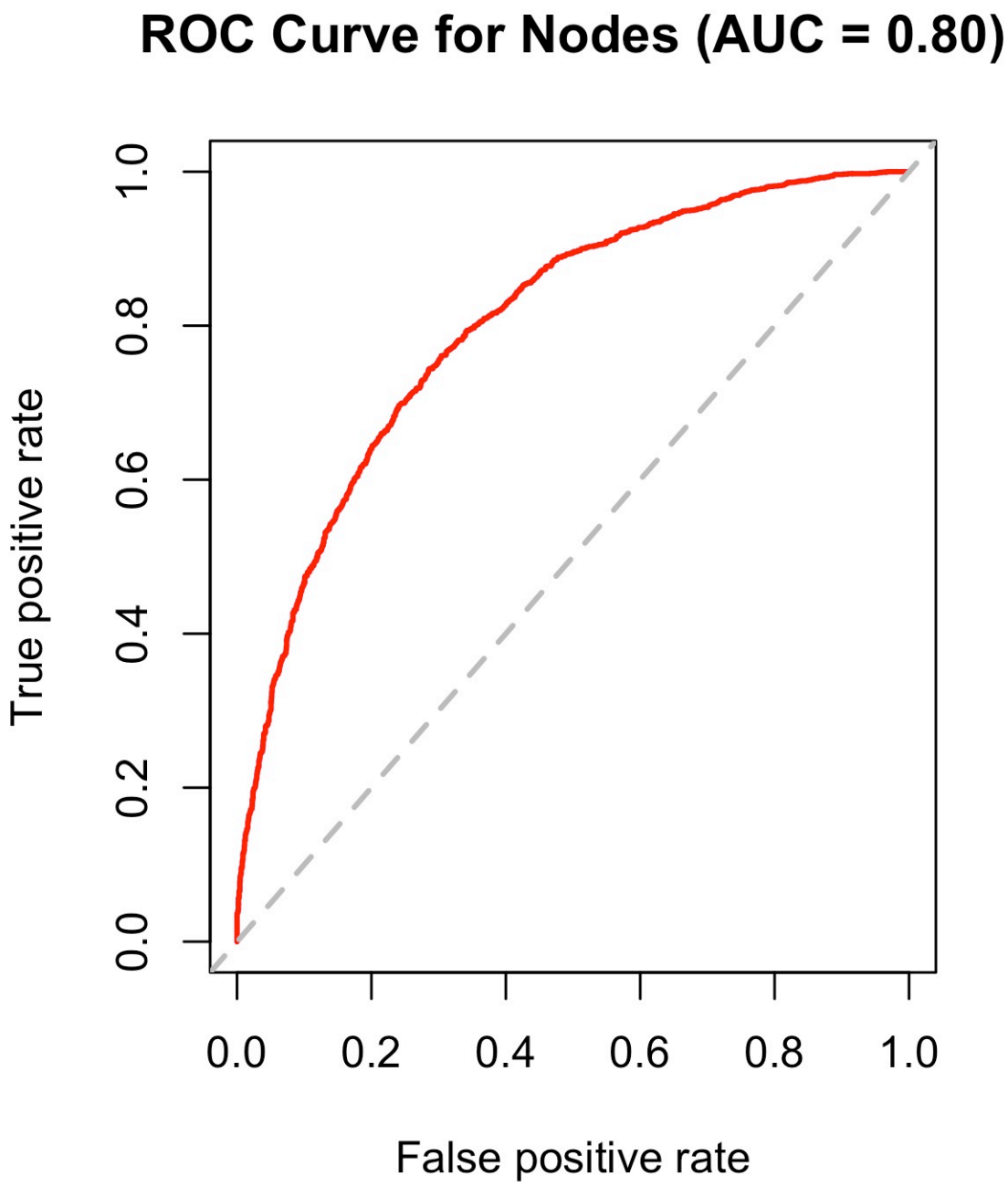
A Random Forest regression model was created using the following variables: Node Shape, Network, Node Size, Node Degree, Node Border Width, Node Hue, Node Saturation, Node Value, X Coordinate Position, Y Coordinate Position, and Eigenvector Centrality. The model explained roughly 26.82% of the variance in the data. 500 regression trees were created and 2 variables (mtry = 2) were tried at each split in each decision tree.

Table 23 - Random Forest model performance for both node and edge models. The values shown in the table are values at the Youden Index [88].

Model	**Error**	**Sensitivity**	**Specificity**	**PPV**	**NPV**	**AUC**
Nodes	0.27	0.74	0.71	0.72	0.73	0.80
Edges	0.21	0.80	0.75	0.76	0.79	0.86

A ROC plot is available in Figure 38 and the AUC for the model was 0.80. No multicolinearity (i.e. linear relationship between predictor variables) was detected.

Figure 38 - ROC plot for Random Forest regression run on nodes.



The cut-off for sensitivity and specificity was determined assuming that they are equally important.

7.4.4. Variable Importance for Nodes

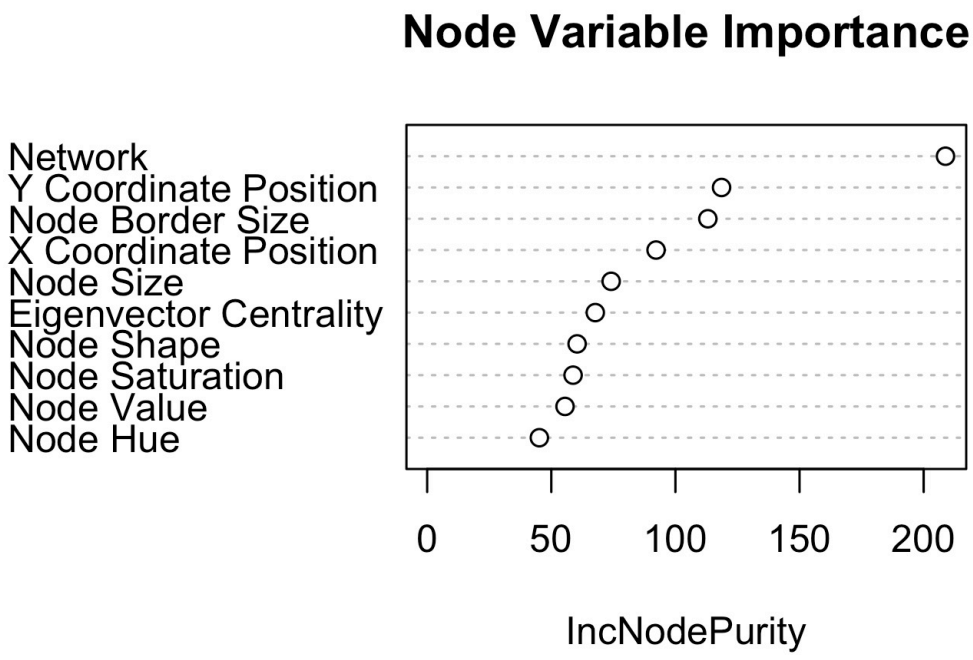
Variable importance values were used to determine the effect of node variables on selectivity. Figure 39 provides a visual overview of the importance values in the Random

Forest model (as determined through mean squared error), and Table 24 provides a more detailed view.

Table 24 - List of Node Importance values. The "%IncMSE" column is the mean decrease in accuracy and the "IncNodePurity" column is the mean decrease in MSE. *Node hue is listed as a numerical variable, although hue is typically categorical (further explained in the text).

	%IncMSE	**IncNodePurity**	**Variable**
Network	0.0090	208.79	Categorical
Y Coordinate Position	0.0155	118.57	Numerical
Node Border Width	0.0499	113.05	Numerical
X Coordinate Position	0.0055	92.20	Numerical
Node Size	0.0350	74.11	Numerical
Eigenvector Centrality	0.0142	67.72	Numerical
Node Shape	0.0006	60.39	Categorical
Node Color Saturation	0.0145	58.77	Numerical
Node Color Value	0.0133	55.52	Numerical
Node Color Hue	0.0072	45.18	Numerical*

Figure 39 - A plot of variable importance from the Random Forest model explaining node selectivity.



As Table 24 and Figure 39 show, the network variable seems to be the most important (although it is not a visual encoding measure). The second most importance node variable is Y Coordinate Position, while X Coordinate Position is ranked fourth, suggesting that node selectivity may be more sensitive to the height at which a node is positioned, relative to its lateral position. Eigenvector Centrality ranked behind Node Size, suggesting that node selectivity may be more sensitive to the size of a node than its "importance" –in context of eigenvector centrality, high "importance" denotes nodes that are connected to other essential nodes.

It is important to note that Random Forest models are systematically biased in favor of categorical variables (tending to give them higher importance scores) [89]. However, I do not think this fact casts doubt on the rank order of importance values.

As far as visual encodings, these results suggest that Node Border Width is most important. Nodes with a thick node border would be emphasized or highlighted in relation to others that have smaller Node Border Widths. However, node border colors were held static in this experiment, and always presented as the color black. This is an important note, as a small node with a large Node Border Width may have a substantial portion that is colored black, and may also make the node seem larger.

Node Size is the second most important visual encoding, suggesting that this would be the natural second option for visually encoding information in a small network. This is followed by the categorical variable of Node Shape, suggesting that certain conformations may be

interpreted to have unintended meaning (e.g. an octagon may be misinterpreted to be a "stop" symbol), or that some shapes may capture a participant's attention better than other shapes. Node Color Saturation is the next visual encoding option. In context of color, saturation has a higher importance than color value (i.e. the amount of black or white in a color), and color value has a higher importance than color hue (what is commonly referred to as "color"). All of these variables are further explored in the sub-section containing partial dependency plots.

Additionally, the network variable was further decomposed in an effort to better understand if any related attributes contribute to better explaining a higher percentage of the variability in the model. In short, the network variable may be decomposed to a variable(s) representing: topology, layout, special properties, and user-specific attributes. Section 10.7 contains detailed results on these models.

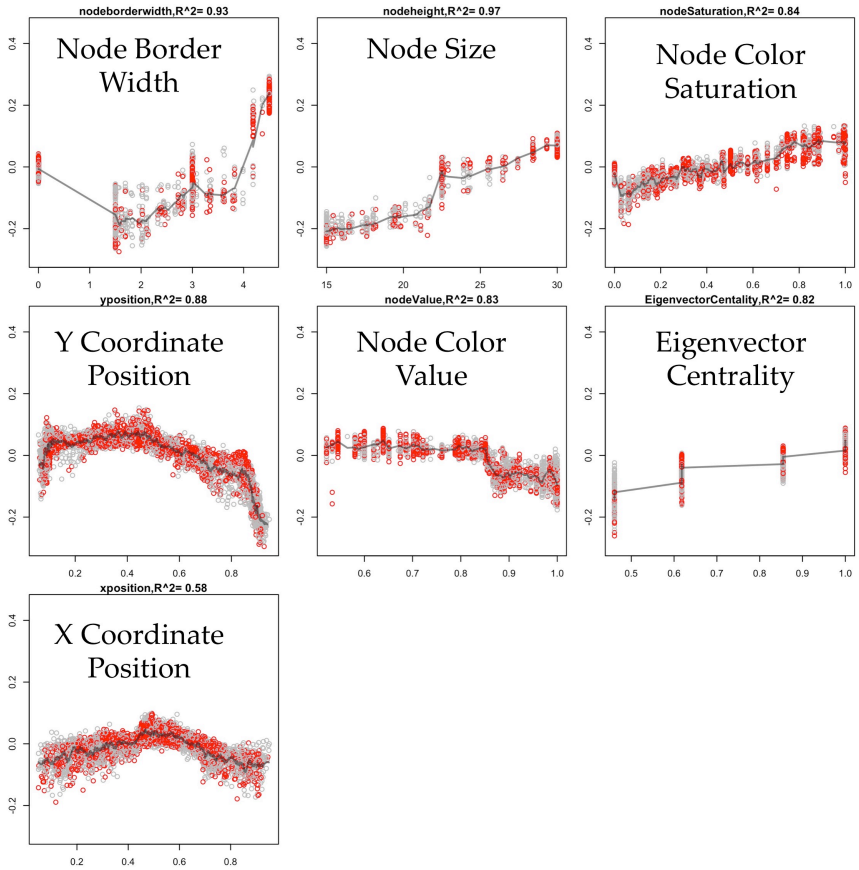
7.4.5. Prototype for Nodes

Although the prototype for a "selective" node could not be estimated, the prototype for a "non-selective" node can be described (in context of the parameterizations of this study) as circular, with a radius of 17px, node saturation value of 0.02 (very bland), and node value of 0.6, node hue most closely described as the color "gray" (which is the same for "non-selective"). The fact that "non-selective" and "selective" nodes have the same prototypical color is sensible because the data did not indicate any relationship between selectivity and color hue (see "nodeHue" in Figure 40).

7.4.6. Partial Dependency Plots for Nodes

Random Forest is sometimes criticized as being a "black box" algorithm, and that the line of reasoning that leads to data being labeled one class or another are obscured. However, this is not exactly true, and it is possible to obtain an understanding of the underlying model through using it. This is where partial dependency plots are useful. With partial dependency plots, one may view how the data behave in the model from the perspective of each variable. It is essential to note that each of perspectives is not orthogonal, independent, adjusted, or otherwise isolated from the rest of the data. Partial dependency plots show how data behaves from the perspective of a variable in context of all of the other variables. Another crucial point to is that the lines in the partial dependency plots are "curve fitted" to the data points, and are not generalized models. The r-squared values contained in these partial dependency plots are slightly different from the r-squared value one may obtain from a linear regression model. In this case, the r-squared value is actually a "pseudo r-squared" value, and it reflects the strength of the relationship between the predictor variable and the outcome variable.

Figure 40 - Partial dependency plot for nodes.



The functions in the partial dependency plot illustrate the complexity of underlying relationships between each attribute and selectivity. The input values (x-axis) for each of the partial dependency plots were ensured to be linear and spaced at even intervals. Parameterizing the input functions in this manner clarifies the underlying relationship and provides guidance on which type of predictive model one should use for future studies.

The "jump" between 0 and 1.5 in the Node Border Width ("nodeborderwidth" in Figure 40) plot is due to the parameterization of possible Node Border Width values. "0" means no border", and "1.5" is the smallest border that is possible. This was necessary by design. Otherwise, if Node Border Width were used categorically (e.g. to denote groups), then one of the groups would not have a border. Node Border Width seems to scale exponentially with respect to selectivity, although a linear model with a relatively steep slope may provide an adequate approximation.

Node Size ("nodeheight" in Figure 40) seems to follow a sigmoidal curve (such as in logistic regression). This suggests that encodings on Node Size may provide diminishing returns for values outside of a certain node size interval. Intriguingly, due to the shape of the curve, the distribution of selectivity arranged according to size seems to resemble a normal distribution.

Node Saturation seems to scale linearly. Selectivity approximately increases 0.1 units for every 0.5 units of increase in color saturation. If 50% Node Saturation corresponds to 0 units of selectivity, then 0% saturation corresponds to -0.1 units of selectivity and 100% corresponds to 0.1 units of selectivity.

Node Value also seems to be approximately linear, although the slope is negative. That is, if 50% Node Value corresponds to 0 units of selectivity, then 0% value corresponds to 0.1 units of selectivity, and 100% value corresponds to a -0.1 units of selectivity. A "0" value corresponds to the color "black" and a "1" value corresponds to the color "white". This suggests that "lighter" nodes are less selective than "darker" nodes.

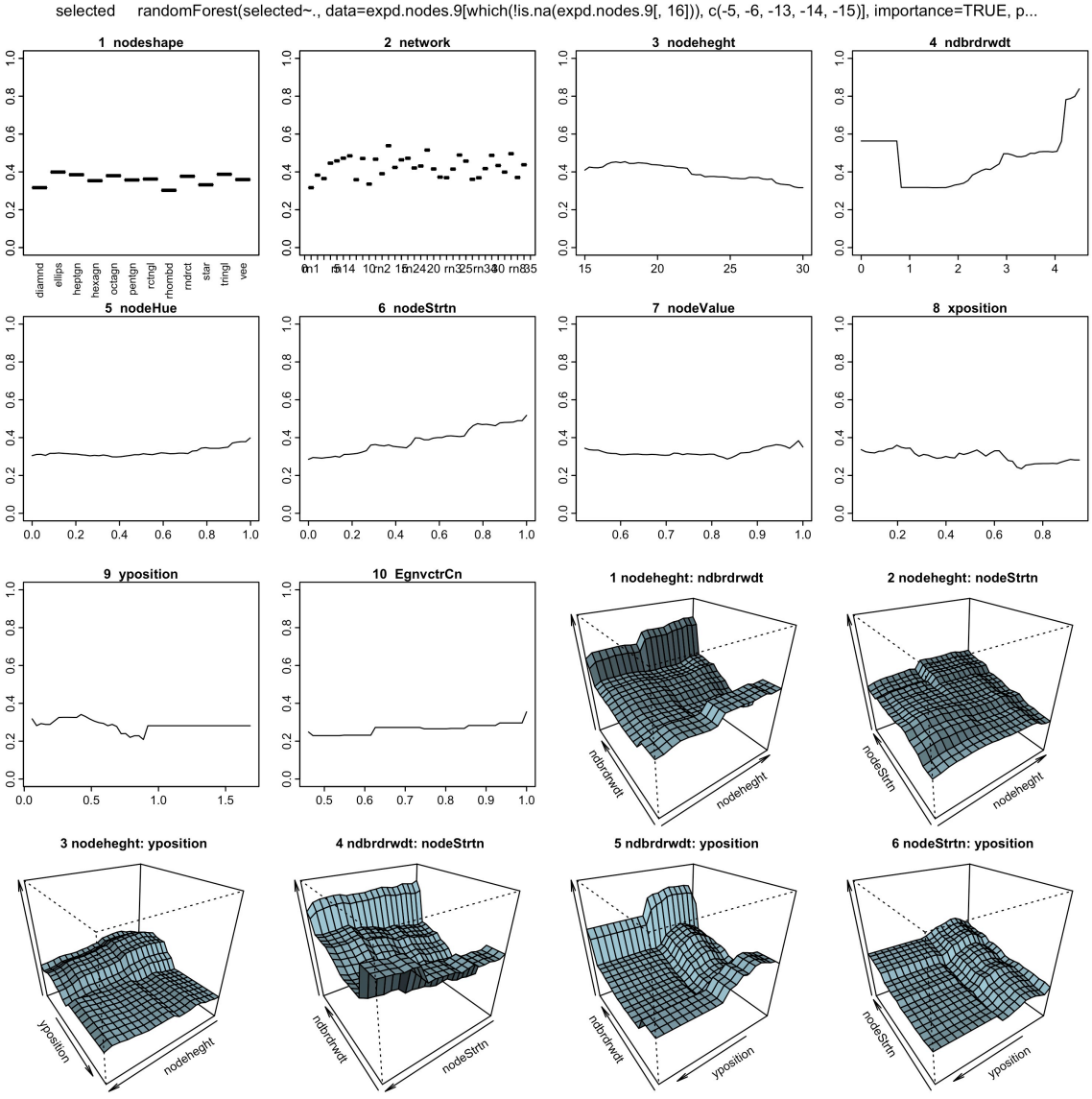
The combination of X Position and Y Position of the nodes seem to suggest that the most selective nodes are physically located in the middle of the screen. Although lateral position seems to decrease selectivity evenly regardless of whether the node is left or right, the partial plot for vertical position seems to suggest that nodes that are higher up on the screen are less selective.

Node Hue, Node Shape, and Network were not included in Figure 40 as they are categorical predictors and posit no meaningful interpretation from these plots.

7.4.7. Assessing Variable Interactions for Node Variables

Variable interactions can be assessed using the three-dimensional plots in Figure 41. The y-axes in all of the plots in Figure 41 represent selectivity. Most of the plots in Figure 41 show a gradual upward climb as values of node encoding parameters increase (linearly). The only exceptions are the plots containing Node Border Width—and the reason for this was explained in the previous sub-section. Overall, this implies that interactions among node variables may demonstrably affect selectivity.

Figure 41 - Variable interactions for Nodes.



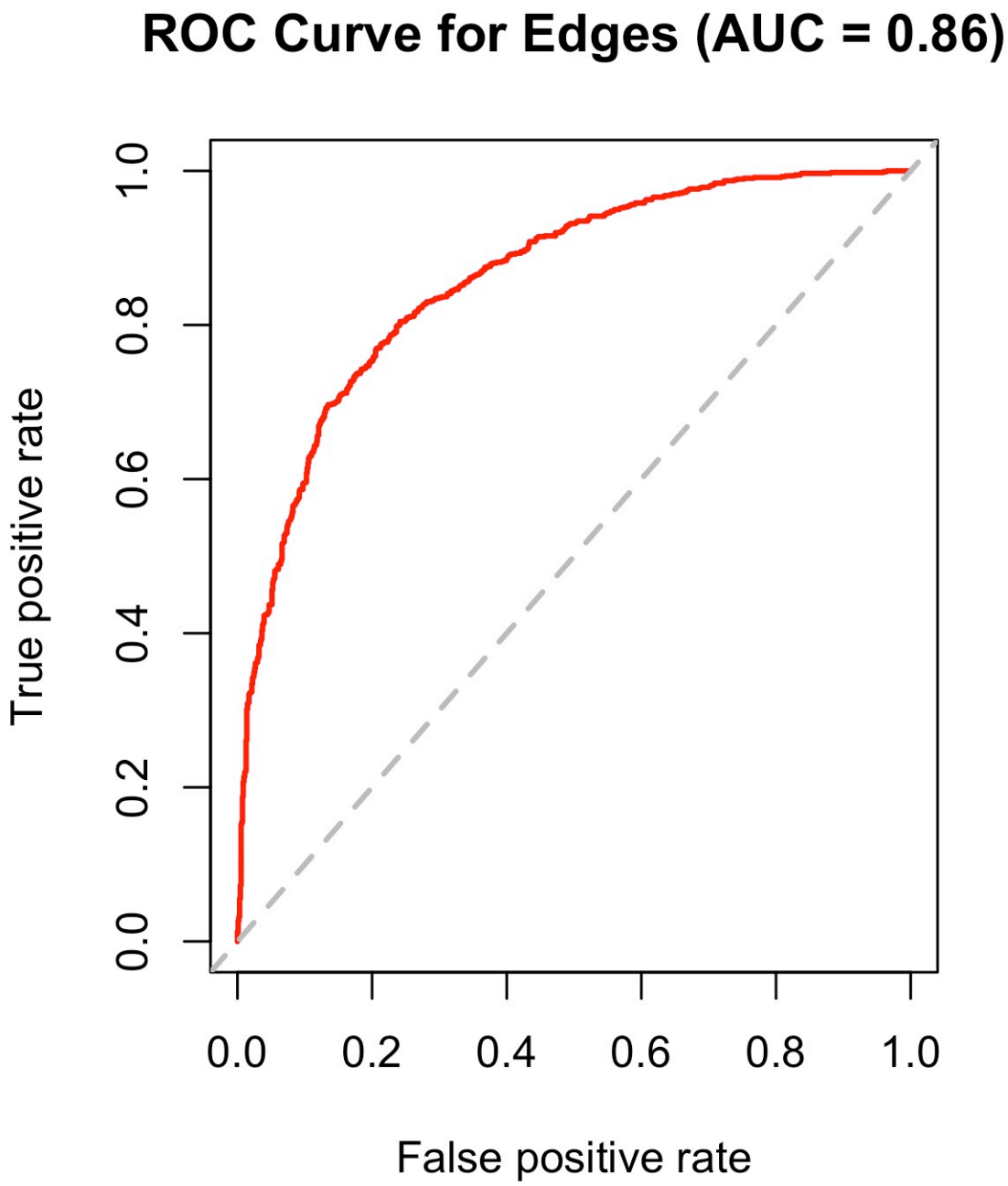
As the results presented thus far only pertain to nodes, the following results now pertain to edges.

7.4.8. Performance of Random Forest Model for Edges

A Random Forest regression model was created using the following variables: Edge Width, Network, Edge Saturation, Edge Value, Edge Hue, Edge Length, and Edge Pattern. The model explained roughly 38.67% of the variance in the data. 500 regression trees were created and 2 variables (mtry = 2) were tried at each split in each decision tree. Further details about the performance of the Random Forest model for edges are provided in Table 23.

A ROC plot is available in Figure 42 and the AUC for the model was 0.86. No multicolinearity was detected in the predictor variables.

Figure 42 - ROC plot for Random Forest regression run on edges.



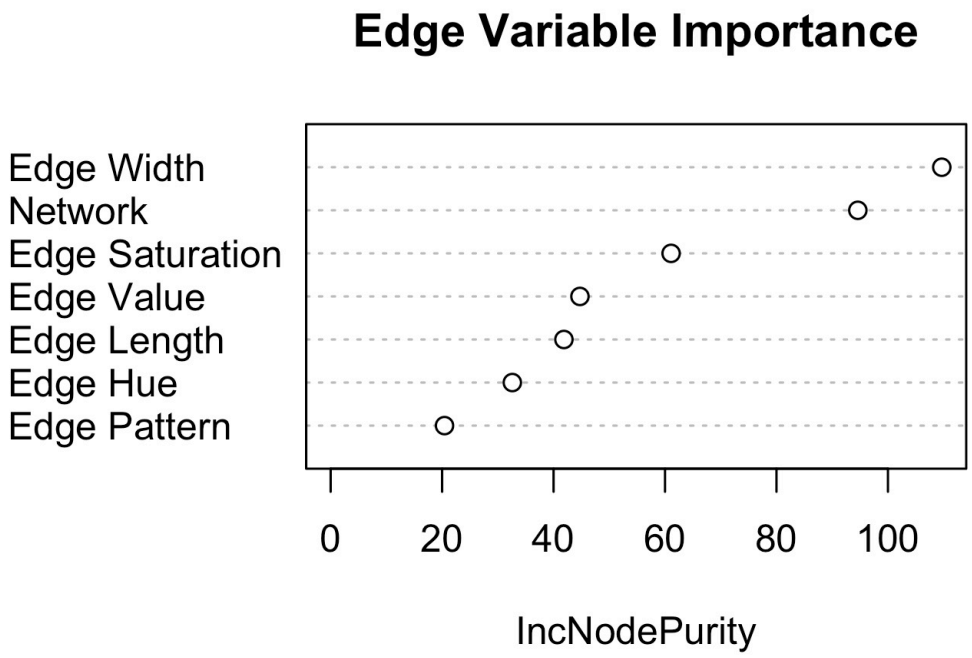
7.4.9. Variable Importance from Random Forest Model for Edges

Variable importance values were used to determine the effect of edge variables on selectivity. Figure 43 provides a visual overview of the importance values in the Random Forest model (as determined through residual sum of squares), and Table 25 provides a more detailed view. Residual sum of squares is also referred to as "increase in node purity."

Table 25 - List of edge Importance values. The "%IncMSE" column is the mean decrease in accuracy and the "IncNodePurity" column is the mean decrease in MSE. *Edge hue is listed as a numerical variable, although hue is typically categorical (further explained in the text).

	%IncMSE	**IncNodePurity**	**Variable**
Edge Width	0.1014	109.66	Numerical
Network	0.0121	94.59	Categorical
Edge Color Saturation	0.0504	61.08	Numerical
Edge Color Value	0.0368	44.73	Numerical
Edge Length	-0.0008	41.85	Numerical
Edge Color Hue	0.0197	32.60	Numerical
Edge Pattern	0.0265	20.43	Categorical

Figure 43 - Variable Importance for edges



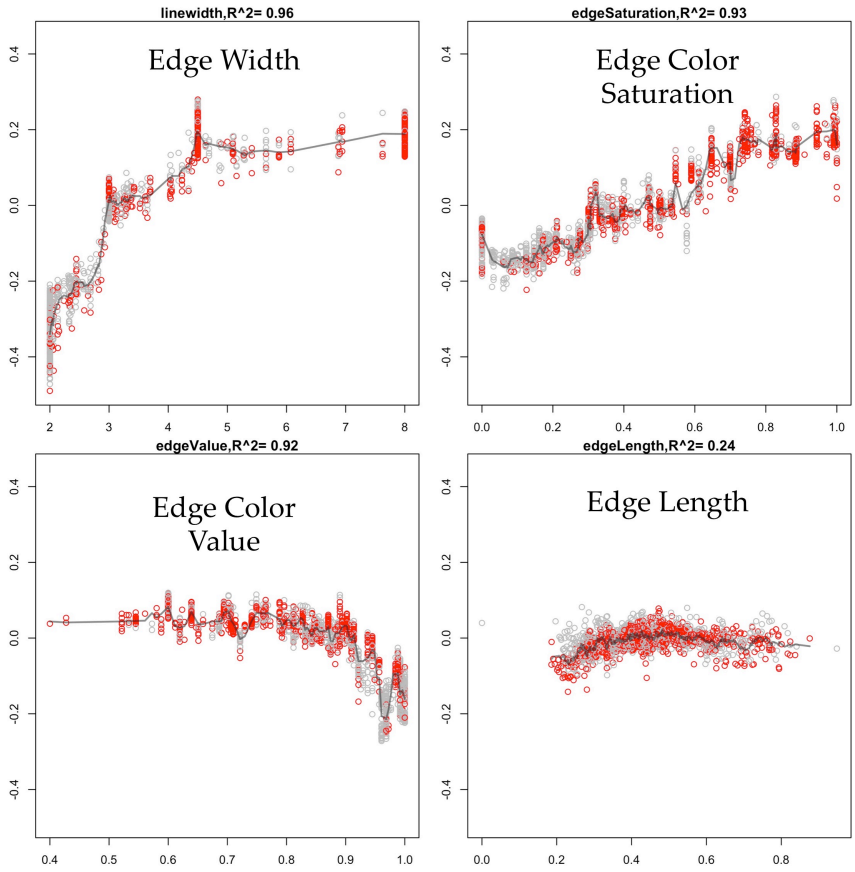
7.4.10. Prototypes from Random Forest Model for Edges

Although there the prototype for a "selective" edge could not be estimated, the prototype for a "non-selective" edge can be described (in context of the parameterizations of this study) as solid, with a thickness of 2.3px, edge saturation value of 0.17 (very bland), and edge value of 0.92, edge hue most closely described as the color "pink."

7.4.11. Partial Dependency Plots from Random Forest Model for Edges

The partial dependency plots for edges are contained in Figure 44. Again, it is crucial to emphasize that each of perspectives is not orthogonal, independent, adjusted, or otherwise isolated from the rest of the data. Partial dependency plots show how data behaves from the perspective of a variable in context of all of the other variables. Another imperative clarification is that the lines in the partial dependency plots are "curve fitted" to the data points, and are not generalized models. The r-squared values contained in these partial dependency plots are slightly different from the r-squared value one may obtain from a linear regression model. In this case, the r-squared value is actually a "pseudo r-squared" value, and it reflects the strength of the relationship between the predictor variable and the outcome variable.

Figure 44 - Partial dependency plots for edges.



As Figure 44 shows, many of the curves for edge variables are not linear. Edge Width ("linewidth" in Figure 44) follows what seems to be a logarithmic curve. This suggests diminishing returns on selectivity for edges that are particularly thick.

Edge Saturation ("edgeSaturation" in Figure 44) seems to follow a sigmoidal pattern. Increasing Edge Saturation seems to increase selectivity until selectivity maximizes at 0.2 units. Similarly, decreasing Edge Saturation seems to decrease selectivity until selectivity minimizes at -0.2 units.

Edge Value ("edgeValue" in Figure 44) seems to have a fascinating relationship with selectivity. As Edge Value increases past approximately 0.75, edge selectivity seems to rapidly decay, until it actually starts to negatively impact edge selectivity. Similarly to Node Value, "0" corresponds to the color "black" and "1" corresponds to the color "white." This means that edges with more than 75% color value are subject to a prompt drop in selectivity.

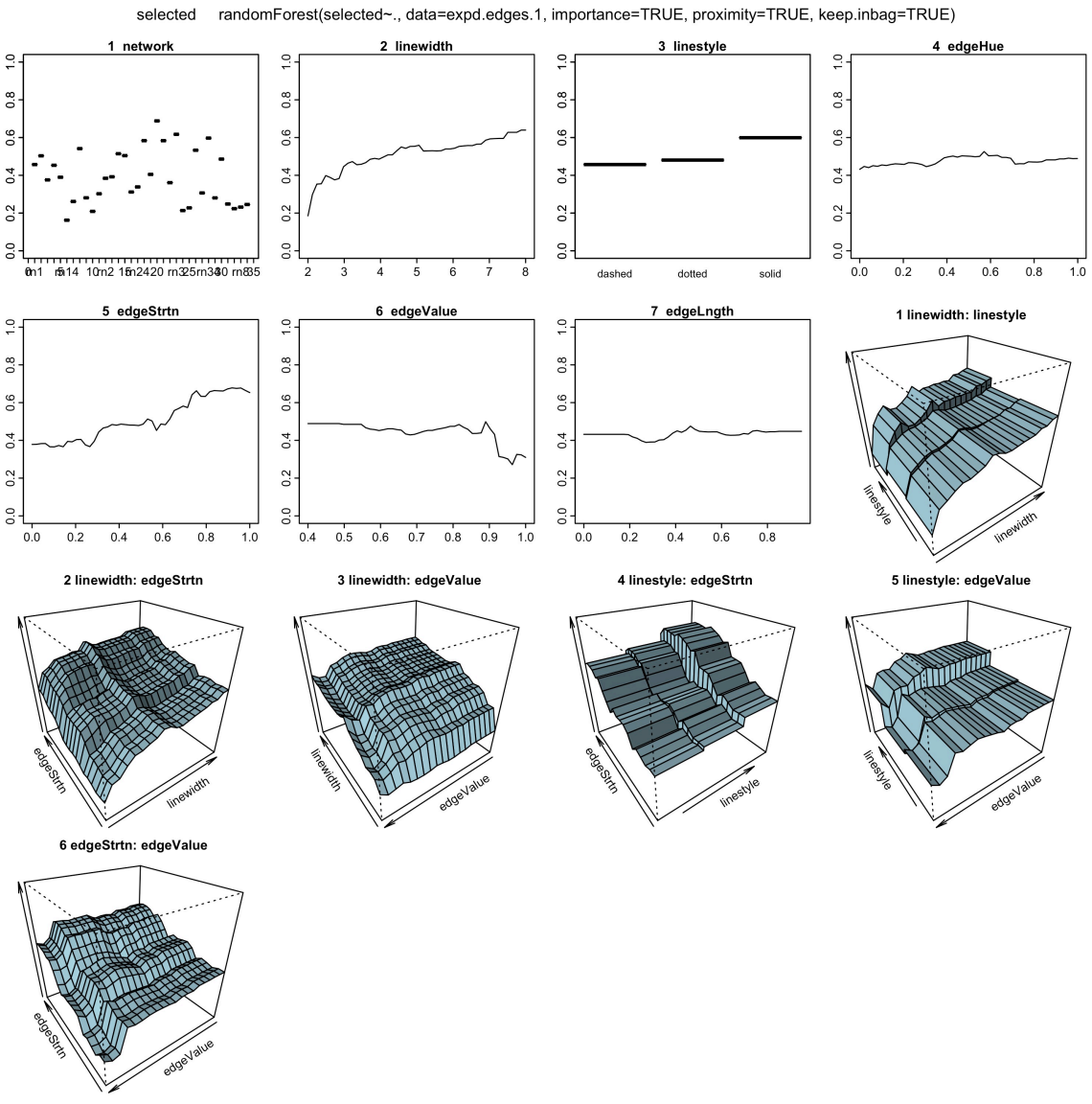
Edge Length ("edgeLength" in Figure 44) seems to have a relationship with edge selectivity where short edges are less selective, while mid- to long- length edges have a marginally positive effect on edge selectivity.

Edge Hue, Edge Pattern, and Network were omitted from Figure 44 as they are categorical predictors and posit no meaningful interpretation from these plots.

7.4.12. Assessing Variable Interactions for Edge Variables

Variable interactions can be assessed using the three-dimensional plots in Figure 45. The y-axes in all of the plots in Figure 44 represent selectivity. In contrast to the node variable interactions depicted in Figure 41, most of the plots in Figure 45 show a flat line, a flat line preceded by a rapid increase, or a flat line followed by a rapid decrease. Overall, Figure 45 implies that interactions between edge variables may not affect selectivity as much as the interactions found among node variables in Figure 41.

Figure 45 - Variable Interactions for Edges



7.4.13. Qualitative Results: Fill-in Responses

In the following section, I provide selected fill-in responses from participants explaining why they clicked on a certain node or edge. The purpose of the fill in response is to capture

information that may otherwise be missed if one were to solely evaluate the quantitative data.

7.4.13.1. Why Participants Selected Nodes

The fill-in responses for node encodings contained a large gamut of responses. The following two responses were selected to illustrate the range of descriptiveness of received input (Quote 1 and Quote 2).

> Quote 1: "The shape. Probably the external corner which is > 90 degrees."
> Quote 2: "Prominent"

The following quotes (Quotes 3-5) suggest that when not visual attributes were truly prominent, some selection decisions were made based on convenience:

> Quote 3: "closer to the curser" [sic]
> Quote 4: "I'm on my phone, and it was closest to my right thumb." [sic]
> Quote 5: "It's at eye level"

The majority of fill-in responses indicated the nodes that garnered the most attention were large in size, brightly colored, and had a thick border. Quotes 6 and 7 below is examples:

> Quote 6: "It's a combination of the node that's closest to the center of the image, the biggest-sized node, and has a thick outline."
> Quote 7: "thick rim. dark color pops out." [sic]

An interesting observation that came to light was how important certain color hues were. Specifically, the color red seemed to attract more attention than other colors. Given the population of participants, perhaps there is some prior association with the colors and their perceived meaning.

> Quote 8: "The red attracted my attention"
> Quote 9: "red, top, also I'm sitting a bit left of the screen"
> Quote 10: "biggest and the green colour is inviting" [sic]

However, there were also a few cases where participants selected an item because it was under-saturated:

> Quote 11: "least saturated color"
> Quote 12: "it is purple, not orange as the others"

This suggests that selectivity may be affected not specifically by color saturation, but by contrast. A handful of quotes support this idea, and Quotes 13 and 14 are examples of those:

> Quote 13: "The color stood out - it contrasted with the rest of the nodes."
> Quote 14: "the contrast is greater, there is a lighter fill color against the black line"

As mentioned in the earlier subsections, the topology of the network may have an affect on guiding participants' selections. The quotes below are examples of participants' explaining that they made certain selections as a result of topology, rather than visual encodings:

Quote 8: "connected to other nodes, in a central locaiton, prominent thick edge" [sic]
Quote 9: "it has 3 edges"
Quote 10: "Central location and 2 edges leading to it created the most focalization."

In summary, the fill-in answers for nodes suggest that participants generally selected nodes that were larger in size, darker in color, and had thicker borders. These results also suggest that color may be symbolic, or otherwise hold some meaning (e.g. danger, safety, etc.). Furthermore, the selectivity of a node is also affected by the visual encodings of surrounding nodes and edges, and also the topology of the network itself. Lastly, when no one node is salient over another, the deciding factor seems to be some form of convenience (e.g. the node that is most centered on the screen, closest node to the cursor, etc.).

7.4.13.2. Why Participants Selected Edges

I will now describe the fill-in answers for edges. The variety of patterns that were tested seemed to contain some underlying meaning to participants. The responses that were received ranged from comments explaining that a certain pattern represented uncertainty, to comments expressing frustration:

Quote 11: "dotted lines suggest uncertainty"
Quote 12: "It's in the middle and stands out from the distracting dashes"

Participants to invariably interpret thicker edges as an indication of a stronger relationship:

Quote 13: "It's the thickest line. It obviously implies a stronger relationship."
Quote 14: "It is an outlier....? It is also the thickest." [sic]

Another observation from the comments were that participants tended to select edges that seemed to be outliers, which is similar to the observation detailed earlier about nodes:

Quote 15: "it stands out. the color is different from the other nodes and edges"
Quote 16: "the color is different/brighter from the rest"

Furthermore, color and location were also contributing factors to edge selection. The color red also seemed to garner heightened attention for edges. However, an interesting comment from a participant was that a certain color was "diluted":

Quote 17: "It's solid. The green one might have been more noticeable if there weren't two of them diluting the prominence."
Quote 18: "The red and central location attracted my attention."

In summary, the fill-in answers for edges suggest that participants generally selected edges that were larger in size, were solid, and seemed to be outliers. These results also suggest that color may be symbolic, or otherwise hold some meaning (e.g. danger, safety, etc.), and that visual encodings like color may be "diluted" with increased presence. Additionally, the location of an edge also seems to contribute to selectivity.

7.5. Discussion

This study has revealed previously unknown relationships that may govern selectivity of nodes and edges in network visualizations intended for visual scanning. Table 26 below

organizes the visual encodings used in the Random Forest models according to the families of mathematical curves that estimate their relationship with selectivity. Table 26 shows 5 distinct types of curves: linear, sigmoidal, logarithmic, decay (reverse sigmoidal), and exponential.

Characterizing these curves is useful for obtaining an intuition for the behavior of these encodings, and understanding which visual encodings can be scaled upwards. For instance, if one were to choose between encoding Node Border Width and Node Size, the former may be generally be a better choice as Node Size begins to yield diminishing returns after a certain point. If one were to conduct future research on any of the visual encodings that were investigated as a part of this study, knowing which curve to use will be immensely useful for model selection and analysis. I do not claim that the visual encodings included in the table maintain these properties when used in other contexts.

Table 26 - A table organizing visual encodings by the mathematical function that characterizes their relationship with selectivity.

Estimated Curve	**Visual Encoding**	**Importance**	**R-squared**
Linear	Node Saturation	3	0.84
Linear	Node Degree	8	0.90
Sigmoidal	Edge Saturation	3	0.94
Sigmoidal	Node Size	5	0.96
Logarithmic	Edge Width	1	0.95
Decay (reverse sigmoid)	Edge Value	4	0.93
Decay (reverse sigmoidal)	Node Value	6	0.86
Exponential	Node Border Width	2	0.92

Although Random Forest has less variance (in context of the bias-variance tradeoff) than decision trees due to bootstrapping, it may be that the logistic function is accurate enough to create a useful prediction model. Although there are numerous reasons why logistic regression is unsuitable for this particular dataset, a future study employing Dynamo may be designed such that logistic regression may be used. Although a random forest model is able to capture interaction effects without having to explicitly define them in the model, a logistic regression model may still be able to provide similar performance (0.73 AUC for logistic regression versus 0.81 AUC for random forest, using the same variables). Further detail, as well as the performance of a number of logistic regression models is covered in Section 10.8. In short, it seems no one variable may be used to reasonably predict node or edge selectivity.

As mentioned earlier the "network" variable suggested that topology is quite an important factor in determining what is visually noticeable in network visualizations. However, the partial dependency plot showed that the r-squared value was 0.08 for nodes, and 0.07 for edges, suggesting no relationship between the network variable and selectivity. R-squared is a

measure of the amount of variance that is explained by that visual encoding, and importance is a measure of how sensitive selectivity is to value perturbations of a particular variable. A future study might classify network topology into a number of sub-classes, and further investigate the relationship between these topology-based sub-classes and selectivity.

Cleveland et al published a seminal paper about graphical perception in 1984 [47]. Before this study, there was no strong evidence that the rankings provided by Cleveland et al translated to network visualizations. However, the findings of this study seem to be in alignment with the findings from the Cleveland et al paper, which may also strengthens the link between visual encodings in networks and newer research founded on the findings of the Cleveland et al paper (cited by 1173 as of September 20, 2016 according to Google Scholar). Another notable observation came from the fill-in section. The fill-in comment about color dilution was interesting as it echoed some of the experiences published in, "Semiology of Graphics, " which raised the concern that the effectiveness and interpretability of color as a visual encoding dwindles as more data points are encoded with it [85].

Future work may include a reproduction of this study, with the addition of varying the number of nodes in networks presented to participants. The results from this study provide enough information about the variables used to provide confidence in the expected values and distributions underpinning the data. This proposed future study would need to obtain a significantly higher number of participants, but would be appropriate for analysis using a logistic regression model, which would yield a simply interpretable regression equation. Using Dynamo to more accurately identify the mathematical functions that underlie this phenomenon could also be a fruitful path for future work. Another idea for a future project, based on a publication about hand-drawn layouts by Kieffer et al, would be to use Dynamo to allow participants to visually encode and position nodes and edges in a network [90]. This would provide a novel perspective, and could reveal the range of ways in which the same task may be accomplished while varying layout and visual encoding.

Since the random forest models have been saved, the models may be used to help optimize visual encodings for those creating small biological networks. However, this particular study used representative networks with only four nodes. Thus, these results are not expected to generalize to large biological networks. Various configurations of visual encodings assignments may be input in the model, and "selectivity" scores may be output for every node and edge in the network. This would be immensely useful in guiding network visualization authors on whether the visual encodings they have chosen to use actually support or hinder their intended message.

7.6. Conclusion

This chapter detailed the design and execution of a study intended to estimate the importance of various visual encodings in network visualizations. The contributions of this chapter provide quantitatively derived insights into how node and edge variables affect selectivity, and qualitative findings on how participants' selected one set of encodings over another.

8. Conclusion

This is the concluding chapter of this dissertation. In this chapter, I cover contributions, summarize the long-term findings from my work, elaborate on potential future work, and finish with a reflection on the field of biomedical informatics, and on the whole process of earning a doctorate. Multiple sections may meld personal experiences and thoughts in this chapter (where appropriate).

8.1. Contributions

I will list, in order of the chapters, the contributions of the findings contained in each chapter. Chapter One is omitted, as it is the introductory chapter.

- Chapter 2: There are no other publications investigating the challenges of visualizing biological networks. The findings present an overview of a set of broadly defined challenges impacting biological network visualization.

- Chapter 3: The findings in chapter three feature an overview of how biological network visualizations are presently depicted. The results from this research essentially supply us with a collection of null hypotheses.

- Chapter 4: The findings in this chapter supplement the discoveries of the previous chapter through the addition of the dimension of tasks.

- Chapter 5: The results of this chapter are two-fold. First, I establish that the Random Forest algorithm is an ideal analytical tool to comprehend the dataset obtained from chapters three and four—this is no small feat, since nearly all of the collected data are nominal, which renders common quantitative analysis methods difficult to impossible to use. Second, the output of Random Forest showed that the quantity of nodes and edges in a biological network were implicated with the ability to complete tasks and visual encodings.

- Chapter 6: This chapter detailed the design, development, and testing of Dynamo, a tool intended to support the study of visual encodings in network visualizations. Aside from the contribution of the tool itself, this chapter also contains an explanation of the premise of "the information triad," stressing the intricate relationship between tasks, visual encodings, and data. With this theoretical framework, evaluations of visualizations can be properly framed, clarifying assessment of network visualizations and paving the way for rigorous, thorough, and useful evaluations that may be compared from the lens of tasks, visual encodings, or data.

- Chapter 7: This chapter describes the design and execution of an experiment devised to uncover the mathematical relationships between a certain task ("visual scanning") and visual encodings. The findings of this study provide evidence for the distribution of data and mathematical curves underlying varying visual encodings—this information is practically useful for future studies, and also for understanding how visual encodings interact with one another.

8.2. Research Vision

From a macroscopic perceptive, my research vision can be summarized as bringing human-computer interaction methodologies to bioinformatics. I have executed on this research vision through the subject of biological network visualization. A number of biological networks are created in a rather ad-hoc manner, and I believe the studies I have conducted demonstrate that it is possible and worthwhile to look systematically at the design decisions behind networks. Although I do not claim to have fully achieved this vision, I think much progress has been made.

Furthermore, many important areas of research are referred to, by some, as "soft science." In fact, Ernest Rutherford (allegedly) went so far as to state, "all science is either physics or stamp collecting." I believe that an overarching result of this dissertation demonstrates how computational methods and adequate study design can transform a so-called "soft science" into a "hard science" (although admittedly, there are still enhancements that can be made). Although the term "soft science" may not have been intended to offend, it is sometimes verbally stated in a dismissive tone, as if "soft science" research is universally inferior—I hope to change, or be involved in changing, the spirit underpinning this sentiment over the next few decades. Much of these "soft sciences" are either closely intertwined with, or live under the umbrella of what is academically recognized as the humanities. Many researchers working in the humanities have already begun to use quantitative and computational methods to pose and answer research questions. One of the major limiting factors of research questions posed in the humanities (and informatics) is that there is not always an adequate way to measure a necessary attribute. My perspective is that advancements in technology (and increased accessibility of that technology) afford researchers the ability reframe age-old research questions, or even pose new ones that are only now feasible due to new technology.

The study detailed in Chapter 7 was only possible as an artifact of the time and place we live in (on a historical scale). In that particular study, I was able to recruit over 100 participants for free over the Internet. The fact that I was able to reach 100 participants in less than a week, over the Internet, and that all of them had a compatible device that supported data collection over roughly 30 network visualizations in under 5 minutes, is a benefit of our time. That very same study design would not have been possible even just a few years ago. The software packages, web standards, and cloud services the experiment was built and deployed on did not exist in until at least 2009, and some of which did not exist until as recently as 2011.

The claim and vision that the "soft science" will eventually transform into "hard science," through the combination of new technology and computational techniques, may imply I am advocating for an increase in experimental complexity. However, in context of solutions, my bias is towards the humble solution, to new and age-old challenges alike. Highly complex experiments that offers multiple interpretations of the same findings is a defining limiting factor of "soft science"—this is not necessarily the "fault" of the one who designed the study, as some research questions may not have a rigorous statistical method that is conveniently applicable, resources to frame the research question in a highly specific way, or prior research to guide experimental design.

I had considered adding another section about my vision for my research career, but my impression is that the world is so fast-paced and rapidly changing that any long-term career

plan (even with a scope as short as 5 years) would be promptly outdated. Thus, when it comes to my research career, I will go wherever the wind may carry me. This is not to say I will resign to the impositions of the world, but it is to say that I will remain open to new possibilities, and will revisit my position on this often.

8.3. Potential Future Work

There are several lines of research I hope to pursue in the future. Although I realize I may only be able to choose one or two, I will list three of my future research ideas in this subsection.

1. Conduct a study to understand the tasks that are performed on graphs. The knowledge we currently have on the tasks that are performed on graph is useful enough to be actionable. However, biology-specific tasks could warrant additional research. As implied by the Information Triad, the higher the specificity of the task, the more likely a model designed to explain how visual encodings are a function of that task will produce satisfactory predictions.
2. Conduct a large number of studies similar to the study detailed in Chapter 7, in order to obtain a set of default visual encoding ranks that researchers may use to select visual encodings in biological networks. This would entail replicating the design detailed in Chapter 7, while modifying the task participants' are asked to perform. There would be one experiment for every task that is to be characterized.
3. Design a large-scale experiment that accounts for structure in addition to visual encodings, to obtain a "more complete" perspective on how visual encodings and graph topology work as function of task completability.
4. Extend the Information Triad to include interactive data visualizations. Data visualizations delivered on any computational device seem to have an interactive component more and more often.

Over the next few years, I plan to write grant proposal to fund one or more of the research synopses listed above in Section 8.3. I do not anticipate that the budgets need to be very large, since most of the groundwork has already been laid out.

9. References

[1] X. Qi, R. D. Duval, K. Christensen, E. Fuller, A. Spahiu, Q. Wu, Y. Wu, W. Tang, and C. Zhang, "Terrorist Networks , Network Energy and Node Removal : A New Measure of Centrality Based on Laplacian Energy," vol. 2013, no. January, pp. 19–31, 2013.

[2] J.-H. S. Yang, "Social network influence and market instability," *J. Math. Econ.*, vol. 45, no. 3–4, pp. 257–276, Mar. 2009.

[3] L. Page and S. Brin, "The PageRank Citation Ranking: Brining Order to the Web," 1998.

[4] C. J. Ryan, P. Cimermančič, Z. a Szpiech, A. Sali, R. D. Hernandez, and N. J. Krogan, "High-resolution network biology: connecting sequence with function.," *Nat. Rev. Genet.*, vol. 14, no. 12, pp. 865–79, Dec. 2013.

[5] H. Bolouri, "Modeling genomic regulatory networks with big data.," *Trends Genet.*, vol. 30, no. 5, pp. 182–91, May 2014.

[6] N. Gehlenborg, S. I. O'Donoghue, N. S. Baliga, A. Goesmann, M. a Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, and A.-C. Gavin, "Visualization of omics data for systems biology.," *Nat. Methods*, vol. 7, no. 3 Suppl, pp. S56–68, Mar. 2010.

[7] K.-J. Dietz, J.-P. Jacquot, and G. Harris, "Hubs and bottlenecks in plant molecular signalling network," *New Phytol.*, vol. 188, no. 4, pp. 919–938, 2014.

[8] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3–5, pp. 75–174, Feb. 2010.

[9] R. De Smet and K. Marchal, "Advantages and limitations of current network inference methods.," *Nat. Rev. Microbiol.*, vol. 8, no. 10, pp. 717–29, Oct. 2010.

[10] F. J. Anscombe, "Graphs in Statistical Analysis," *Am. Stat.*, vol. 27, no. 1, pp. 17–21, 1973.

[11] M. Krzywinski, I. Birol, S. J. M. Jones, and M. a Marra, "Hive plots--rational approach to visualizing networks.," *Brief. Bioinform.*, vol. 13, no. 5, pp. 627–44, Sep. 2012.

[12] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. a Marra, "Circos: an information aesthetic for comparative genomics.," *Genome Res.*, vol. 19, no. 9, pp. 1639–45, Sep. 2009.

[13] N. Henry and J. Fekete, "MatrixExplorer : a Dual-Representation System to Explore Social Networks," vol. 12, no. 5, 2006.

[14] T. Dwyer, N. Henry Riche, K. Marriott, and C. Mears, "Edge compression techniques for visualization of dense directed graphs.," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2596–605, Dec. 2013.

[15] S. Shapin, "Should scientists tell stories?," *Nat. Methods*, vol. 10, no. 11, 2013.

[16] M. Krzywinski and A. Cairo, "Correspondance: Data Storytelling," *Nat. Methods*, no. ii, 2013.

[17] M. Krzywinski and A. Cairo, "Storytelling," *Nat. Methods*, vol. 10, no. 8, 2013.

[18] Y. Katz, "Against storytelling of scientific results," *Nat. Methods*, vol. 10, no. 11, 2013.

[19] E. Segel and J. Heer, "Narrative Visualization : Telling Stories with Data," vol. 16, no. 6, pp. 1139–1148, 2010.

[20] J. Hullman, S. Member, and N. Diakopoulos, "Visualization Rhetoric : Framing Effects in Narrative Visualization," vol. 17, no. 12, pp. 2231–2240, 2011.

[21] K.-L. Ma, I. Liao, J. Frazier, H. Hauser, and H.-N. Kostis, "Visualization Viewpoints," in *Scientific Storytelling Using Visualization*, 2012.

[22] J. Hullman, S. Drucker, N. H. Riche, B. Lee, D. Fisher, and E. Adar, "A Deeper Understanding of Sequence in Narrative Visualization," vol. 19, no. 12, pp. 2406–2415, 2013.
[23] R. Kosara and J. Mackinlay, "Storytelling : The Next Step for Visualization."
[24] T. Munzner, *Visualization Analysis and Design*. A K Peters/CRC Press, 2014.
[25] M. Ghoniem, J. Fekete, and P. Castagliola, "A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations," *IEEE Symp. Inf. Vis.*, pp. 17–24, 2004.
[26] H. Gibson, J. Faith, and P. Vickers, "A survey of two-dimensional graph layout techniques for information visualisation," *Inf. Vis.*, vol. 12, no. 3–4, pp. 324–357, Sep. 2012.
[27] W. J. Longabaugh, "Combing the hairball with BioFabric: a new approach for visualization of large networks," *BMC bioinformatics*, 2012. [Online]. Available: http://www.biomedcentral.com/content/pdf/1471-2105-13-275.pdf. [Accessed: 26-Jun-2014].
[28] F. Schreiber, T. Dwyer, K. Marriott, and M. Wybrow, "A generic algorithm for layout of biological networks.," *BMC Bioinformatics*, vol. 10, p. 375, 2009.
[29] T. Dwyer, "Scalable , Versatile and Simple Constrained Graph Layout," vol. 28, no. 3, 2009.
[30] C. Dunne and B. Shneiderman, "Improving graph drawing readability by incorporating readability metrics: A software tool for network analysts," *Univ. Maryland, HCIL Tech Rep. HCIL-2009-13*, 2009.
[31] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome Res.*, vol. 13, pp. 2498–2504, 2003.
[32] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, G. Laurent, G. Yongchao, G. Jeff, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, L. Friedrich, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. Yang, and J. Zhang, "Bioconductor: open software development for computational biology and bioinformatics.," *Genome Biol.*, vol. 5, no. 10, p. R80, 2004.
[33] G. Csárdi and T. Nepusz, "The igraph software package for complex network research," *InterJournal Complex Syst.*, vol. 1695, p. 1695, 2006.
[34] H. Wickham, "ggplot2," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 3, no. 2, pp. 180–185, 2011.
[35] Ingenuity Systems, "IPA Network Generation Algorithm," 2005.
[36] M. Kanehisa and S. Goto, "KEGG: kyoto encyclopedia of genes and genomes.," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 27–30, Jan. 2000.
[37] I. M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I. T. Paulsen, M. Peralta-Gil, and P. D. Karp, "EcoCyc: A comprehensive database resource for Escherichia coli," *Nucleic Acids Res.*, vol. 33, no. DATABASE ISS., pp. 334–337, 2005.
[38] P. Romero, J. Wagg, M. L. Green, D. Kaiser, M. Krummenacker, and P. D. Karp, "Computational prediction of human metabolic pathways from the complete human genome," *Genome Biol*, vol. 6, no. 1, p. R2, 2005.
[39] R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C. a. Fulcher, T. a. Holland, I. M. Keseler, A. Kothari, A. Kubo, M. Krummenacker, M. Latendresse, L. a. Mueller, Q. Ong, S. Paley, P. Subhraveti, D. S. Weaver, D. Weerasinghe, P. Zhang, and P. D. Karp, "The MetaCyc database of metabolic pathways and enzymes and the BioCyc

collection of Pathway/Genome Databases," *Nucleic Acids Res.*, vol. 42, no. D1, pp. 459–471, 2014.
[40] G. D. Bader, M. P. Cary, and C. Sander, "Pathguide: a pathway resource list.," *Nucleic Acids Res.*, vol. 34, no. Database issue, pp. D504–D506, 2006.
[41] M. Kutmon, A. Riutta, N. Nunes, K. Hanspers, E. L. Willighagen, A. Bohler, J. Mélius, A. Waagmeester, S. R. Sinha, R. Miller, S. L. Coort, E. Cirillo, B. Smeets, C. T. Evelo, and A. R. Pico, "WikiPathways: capturing the full diversity of pathway knowledge," *Nucleic Acids Res.*, vol. 44, no. October 2015, 2015.
[42] P. Khatri, M. Sirota, and A. J. Butte, "Ten years of pathway analysis: current approaches and outstanding challenges.," *PLoS Comput. Biol.*, vol. 8, no. 2, p. e1002375, Jan. 2012.
[43] A. Barsky, J. L. Gardy, R. E. W. Hancock, and T. Munzner, "Cerebral: A Cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation," *Bioinformatics*, vol. 23, no. 8, pp. 1040–1042, 2007.
[44] E. G. Cerami, B. E. Gross, E. Demir, I. Rodchenkov, O. Babur, N. Anwar, N. Schultz, G. D. Bader, and C. Sander, "Pathway Commons, a web resource for biological pathway data.," *Nucleic Acids Res.*, vol. 39, no. Database issue, pp. D685–90, Jan. 2011.
[45] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris, "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function.," *Genome Biol.*, vol. 9 Suppl 1, p. S4, 2008.
[46] B. Lee, C. Plaisant, J. Fekete, N. Henry, I. Futurs, L. R. I. Bat, and U. Paris-sud, "Task Taxonomy for Graph Visualization Categories and Subject Descriptors."
[47] W. S. Cleveland, R. Mcgill, and S. Cleveland, "Graphical Perception : Theory , Experimentation , and Application to the Development of Graphical Methods," *J. Am. Stat. Assoc.*, vol. 79, no. 387, pp. 531–554, 1984.
[48] C. G. Healey, K. S. Booth, and J. T. Enns, "High-speed visual estimation using preattentive processing," *ACM Trans. Comput. Interact.*, vol. 3, no. 2, pp. 107–135, 1996.
[49] C. Ware, *Information Visualization*. .
[50] N. Iliinsky and J. Steele, *Designing Data Visualizations*. O'Reilly Media, 2011.
[51] N. R. Clark, R. Dannenfelser, C. M. Tan, M. E. Komosinski, and A. Ma'ayan, "Sets2Networks: network inference from repeated observations of sets.," *BMC Syst. Biol.*, vol. 6, no. 1, p. 89, 2012.
[52] C. Lei and J. Ruan, "A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity," *Bioinformatics*, vol. 29, no. 3, pp. 355–364, 2013.
[53] A. Finka, R. U. H. Mattoo, and P. Goloubinoff, "Meta-analysis of heat-and chemically upregulated chaperone genes in plant and human cells," *Cell Stress Chaperones*, vol. 16, no. 1, pp. 15–31, 2011.
[54] B. Bakir-Gungor and O. U. Sezerman, "The Identification of Pathway Markers in Intracranial Aneurysm Using Genome-Wide Association Data from Two Different Populations," *PLoS One*, vol. 8, no. 3, pp. 1–12, 2013.
[55] T. Yamada, I. Letunic, S. Okuda, M. Kanehisa, and P. Bork, "iPath2.0: interactive pathway explorer.," *Nucleic Acids Res.*, vol. 39, no. Web Server issue, pp. W412–5, Jul. 2011.
[56] M. Kailavasan, I. Rehman, S. Reynolds, A. Bucur, G. Tozer, and M. Paley, "NMR-based evaluation of the metabolic profile and response to dichloroacetate of human prostate cancer cells," *NMR Biomed.*, vol. 27, no. 5, pp. 610–616, 2014.
[57] J. Debnath, S. Siricilla, B. Wan, D. C. Crick, A. J. Lenaerts, S. G. Franzblau, and M.

Kurosu, "Discovery of Selective Menaquinone Biosynthesis Inhibitors against Mycobacterium tuberculosis," *J. Med. Chem.*, vol. 16, no. 3, pp. 387–393, 2013.
[58] A. M. Tresiman and G. Gelade, "A Feature-Integration," *Cogn. Psychol.*, vol. 136, pp. 97–136, 1980.
[59] W. Wang, X. Li, J. Huang, L. Feng, K. G. Dolinta, and J. Chen, "Defining the protein-protein interaction network of the human hippo pathway.," *Mol. Cell. Proteomics*, vol. 13, no. 1, pp. 119–31, 2014.
[60] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. Lin, P. Minguez, P. Bork, C. Von Mering, and L. J. Jensen, "STRING v9.1: Protein-protein interaction networks, with increased coverage and integration," *Nucleic Acids Res.*, vol. 41, no. D1, pp. 808–815, 2013.
[61] I. Vastrik, P. D'Eustachio, E. Schmidt, G. Joshi-Tope, G. Gopinath, D. Croft, B. de Bono, M. Gillespie, B. Jassal, S. Lewis, L. Matthews, G. Wu, E. Birney, and L. Stein, "Reactome: a knowledge base of biologic pathways and processes.," *Genome Biol.*, vol. 8, no. 3, p. R39, Jan. 2007.
[62] E. Tufte, *The Visual Display of Quantitative Information*. 1983.
[63] D. a Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, "GenBank.," *Nucleic Acids Res.*, vol. 41, no. D1, pp. D36–42, Jan. 2013.
[64] T. Muzner, *Visualization Analysis & Design*. CRC Press, 2015.
[65] D. Nishimura, "BioCarta," *Biotech Softw. Internet Rep.*, 2004.
[66] G. T. Ioannis, D. B. Giuseppe, P. Eades, and R. Tamassia, *Graph Drawing Algorithms for the Visualization of Graphs.pdf*. 1999.
[67] B. Shneiderman, "The Eyes Have It : A Task by Data Type Taxonomy The Eyes Have It : A Task by Data Type Taxonomy for Information Visualizations," 1996.
[68] E. Morset, M. Lewis, and K. A. Olsen, "Evaluating visualizations: using a taxonomic guide," *Int. J. Human-Computer Stud.*, vol. 53, pp. 637–662, 2000.
[69] M. Ghoniem, J.-D. Fekete, and P. Castagiola, "A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations," *IEEE Symp. Inf. Vis.*, pp. 17–24, 2004.
[70] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and a L. Barabási, "The large-scale organization of metabolic networks.," *Nature*, vol. 407, no. 6804, pp. 651–4, 2000.
[71] M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law," *Power laws, Pareto Distrib. Zipf's law. Contemp. Phys.*, vol. 46, no. 5, pp. 323–351, 2005.
[72] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[73] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
[74] E. a Madigan and O. L. Curet, "A data mining approach in home healthcare: outcomes and service use.," *BMC Health Serv. Res.*, vol. 6, p. 18, 2006.
[75] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis.," *Stat. Appl. Genet. Mol. Biol.*, vol. 4, no. 1, p. Article17, Jan. 2005.
[76] V. Lacroix, L. Cottret, P. Thébault, and M.-F. Sagot, "An introduction to metabolic networks and their structural analysis.," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 5, no. 4, pp. 594–617, 2008.
[77] T. E. Harris and F. S. Ross, "Fundamental of a method for evaluating rail net capacities," 1955.
[78] F. S. Hillier and G. J. Lieberman, *Operations Research*. Holden-Day, Inc., 1967.
[79] J. A. Nelder, R. Mead, B. J. a Nelder, and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.
[80] N. Karmarkar, "A New Polynomial-Time Algorithm for Linear Programming," 1984.

[81] R. Lougee-Heimer, "The common optimization INterface for operations research: Promoting open-source software in the operations research community," *IBM J. Res. Dev.*, vol. 47, pp. 57–66, 2013.
[82] M. Bostock, V. Ogievetsky, and J. Heer, "D^3: Data-Driven Documents.," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–9, Dec. 2011.
[83] M. Franz, C. T. Lopes, G. Huck, Y. Dong, O. Sumer, and G. D. Bader, "Cytoscape.js: A graph theory library for visualisation and analysis," *Bioinformatics*, vol. 32, no. 2, pp. 309–311, 2015.
[84] E. Tufte, *Envisioning Information*. 1990.
[85] J. Bertin, *Semiology of Graphics*. 1967.
[86] M.-J. Kraak and F. Ormeling, *Cartography: Visualization of spatial data*. Addison Wesley Longman Limited, 1996.
[87] G. H. Joblove and D. Greenberg, "Color Spaces for Computer Graphics."
[88] R. Fluss, D. Faraggi, and B. Reiser, "Estimation of the Youden Index and its associated cutoff point," *Biometrical J.*, vol. 47, no. 4, pp. 458–472, 2005.
[89] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: illustrations, sources and a solution.," *BMC Bioinformatics*, vol. 8, p. 25, 2007.
[90] S. Kieffer, T. Dwyer, K. Marriott, and M. Wybrow, "HOLA: Human-like Orthogonal Network Layout," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 349–358, 2016.
[91] C. M. Christensen, *The Innovator's Dilemma: The Revolutionary Book That Will Change the Way You Do Business*. Harper Business, 2011.
[92] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 74, no. 1, pp. 1–16, 2006.
[93] T. W. Valente, K. Coronges, C. Lakon, and E. Costenbader, "How Correlated Are Network Centrality Measures?," vol. 28, no. 1, pp. 16–26, 2008.
[94] E. Costenbader and T. W. Valente, "The stability of centrality measures when networks are sampled," *Soc. Networks*, vol. 25, no. 4, pp. 283–307, Oct. 2003.

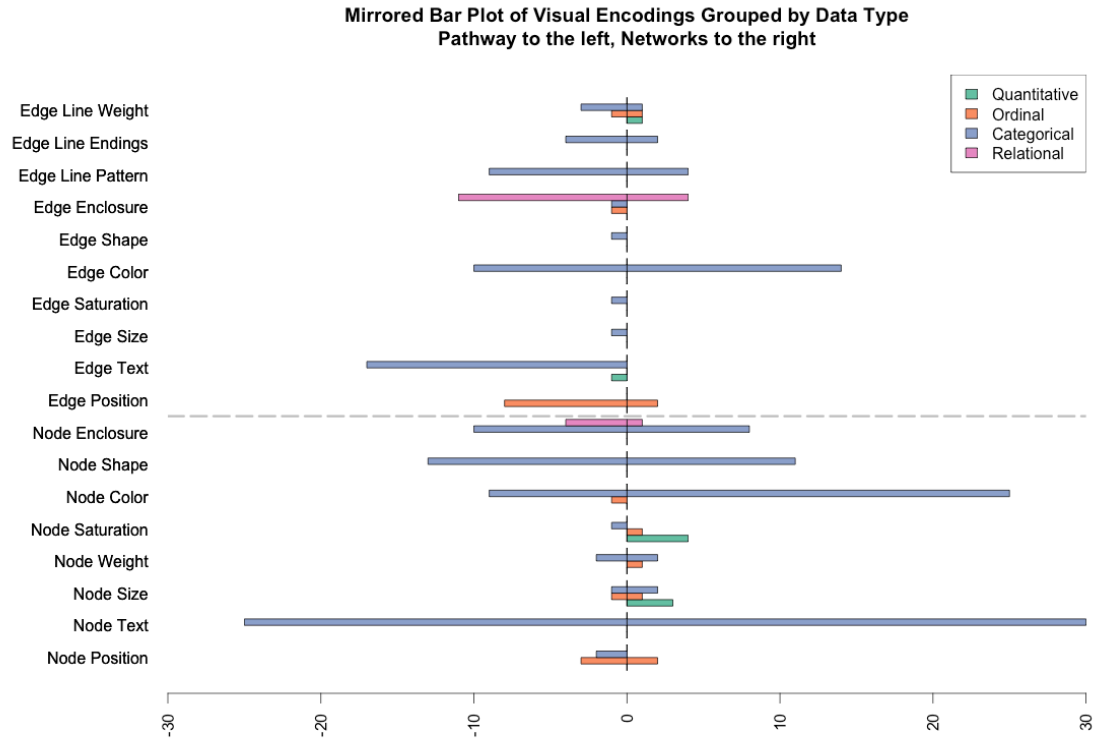
10. Appendix

This section contains supplementary figures, tables, and test results. The items contained in this section were not deemed pertinent to the content of the chapters, but are provided here for completeness.

10.1. Appendix A – Additional Bar Chart (Chapter 3)

The bar chart in Figure 46 illustrating frequency counts of visual encodings, divided by data type, presented side-by-side with pathway counts extending to the left, and network counts extending to the right. This mirrored presentation along with the breakdown of data type allows one to compare the relative proportions of visual encodings and data types without any normalizing operations (and remaining consistent in scale with Figure 6).

Figure 46 – The data in Figure 6 presented as mirrored bar charts.



10.2. Appendix B – Test for unequal variances (Chapter 3)

This section presents the results of a statistical test for unequal variances. Variances can sometimes be heterogeneous, meaning that the variance value varies across the entire range of the data. Recognizing unequal variance is important since many statistical tests assume equal variance. Although finding that one's data has unequal variance may not alter one's choice of statistical test, it is necessary information when interpreting statistical results.

Statistical tests for unequal variance were run on the data found in Table 3 and Table 4. In each table, the data in the network column was compared to the data in the pathway column. The null hypothesis for both statistical tests was that the true ratio of variances is equal to 1. Both statistical tests for unequal variance returned negative, leading us to accept the null hypothesis that the true ration of variances is equal to 1. However, since the degrees of freedom are 7 and 6, respectively, this interpretation is accompanied with some level of doubt. The data are provided below:

Statistical results for Table 3*:*
F = 1.6612, num df = 7, denom df = 7, p-value = 0.5192
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.332570 8.297329
sample estimates:
ratio of variances
1.661157

Statistical results for Table 4*:*
F = 0.89549, num df = 6, denom df = 6, p-value = 0.8968
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.153871 5.211544
sample estimates:
ratio of variances
0.8954918

10.3. Appendix C – Two-way ANOVA between encodings, networks versus pathways (Chapter 3)

Two-way ANOVA		
Summary		
Response	*Value*	
Factor #1	*Encoding*	*Random*
Factor #2	*Type*	*Random*

Descriptive Statistics					
Factor	*Group*	*Sample size*	*Mean*	*Variance*	*Standard Deviation*

Encoding	*Color*	4	15.75	58.25	7.63217
Encoding	*Enclosure, Connection*	4	10	20.66667	4.54606
Encoding	*Line Endings*	4	1.5	3.66667	1.91485
Encoding	*Line Pattern*	4	3.25	18.25	4.272
Encoding	*Line Weight*	4	2	6	2.44949
Encoding	*Position*	4	4.25	8.25	2.87228
Encoding	*Saturation, Brightness*	4	1.75	4.91667	2.21736
Encoding	*Shape, Icon*	4	7.25	54.25	7.36546
Encoding	*Size,Area*	4	3	12.66667	3.55903
Encoding	*Text*	4	18.25	172.25	13.1244
Encoding	*Weight, Boldness*	4	1.25	2.25	1.5
Type	*Network*	22	5.68182	70.98918	8.42551
Type	*Pathway*	22	6.72727	48.39827	6.95689
Encoding x Type	*Color x Network*	2	20.5	84.5	9.19239
Encoding x Type	*Color x Pathway*	2	11	0	0
Encoding x Type	*Enclosure, Connection x Network*	2	6.5	12.5	3.53553
Encoding x Type	*Enclosure, Connection x Pathway*	2	13.5	0.5	0.70711
Encoding x Type	*Line Endings x Network*	2	1	2	1.41421
Encoding x Type	*Line Endings x Pathway*	2	2	8	2.82843
Encoding x Type	*Line Pattern x Network*	2	2	8	2.82843
Encoding x Type	*Line Pattern x Pathway*	2	4.5	40.5	6.36396
Encoding x Type	*Line Weight x Network*	2	1.5	4.5	2.12132
Encoding x Type	*Line Weight x Pathway*	2	2.5	12.5	3.53553
Encoding x Type	*Position x Network*	2	2	0	0
Encoding x Type	*Position x Pathway*	2	6.5	4.5	2.12132
Encoding x Type	*Saturation, Brightness x Network*	2	2.5	12.5	3.53553

Encoding x Type	*Saturation, Brightness x Pathway*	2	1	0	0
Encoding x Type	*Shape, Icon x Network*	2	6	72	8.48528
Encoding x Type	*Shape, Icon x Pathway*	2	8.5	84.5	9.19239
Encoding x Type	*Size,Area x Network*	2	4	32	5.65685
Encoding x Type	*Size,Area x Pathway*	2	2	2	1.41421
Encoding x Type	*Text x Network*	2	15	450	21.2132
Encoding x Type	*Text x Pathway*	2	21.5	24.5	4.94975
Encoding x Type	*Weight, Boldness x Network*	2	1.5	4.5	2.12132
Encoding x Type	*Weight, Boldness x Pathway*	2	1	2	1.41421

ANOVA							
Source of Variation	*SS*	*d.f.*	*MS*	*F*	*p-level*	*F crit*	*Omega Sqr.*
Factor #1 (Encoding)	1,434.91	10	143.49091	6.80932	0.00277	2.97824	0.40781
Factor #2 (Type)	12.02273	1	12.02273	0.57053	0.46746	4.9646	0
Factor #1 + #2 (Encoding x Type)	210.72727	10	21.07273	0.53813	0	2.2967	0
Within Groups	861.5	22	39.15909				
Total	2,519.16	43	58.5851				
Omega squared for combined effect	0.32651						

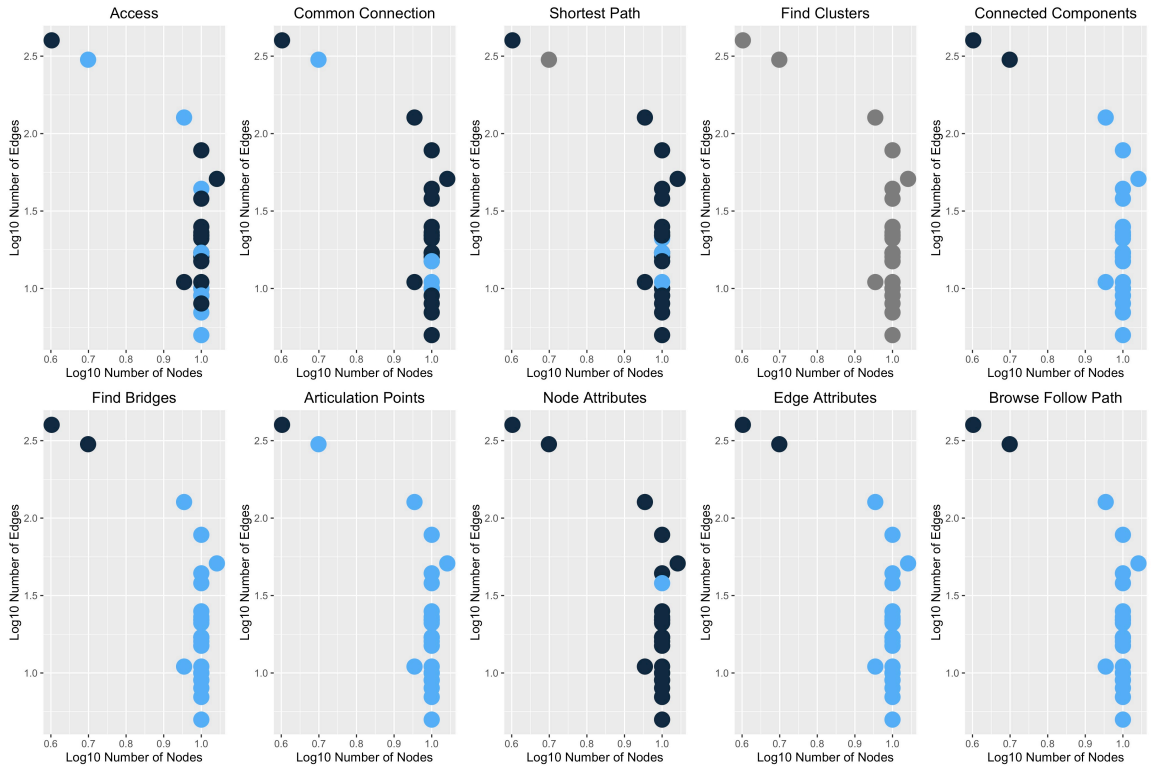
Appendix D – R Code for fisher's exact test comparing frequency distributions between networks and pathways (Chapter 3) (https://gist.github.com/ngopal/1c6aa0bc2de4860280cb921635a9d241)

```
aim2d <- matrix(rbind(
  c(2,    5),
  c(30,   25),
  c(8,    3),
  c(3,    2),
  c(5,    1),
  c(27,   11),
  c(12,   15),
  c(9,    14)
), 8, 2)
colnames(aim2d) <- c("Network",  "Pathway")
```

```
rownames(aim2d) <- c("Position",
                     "Text",
                     "Size,Area",
                     "Weight, Boldness",
                     "Saturation, Brightness",
                     "Color",
                     "Shape, Icon",
                     "Enclosure, Connection")

fisher.test(aim2d)
#prop.test(aim2d, conf.level = 0.95, correct = TRUE)

aim2de <- matrix(t(rbind(
c(2,0,0,0,14,0,4,4,2,3),
c(8,18,1,1,11,2,13,9,4,5))), 10, 2)
colnames(aim2de) <- c("Network",  "Pathway")
rownames(aim2de) <- c("Position",
                      "Text",
                      "Size,Area",
                      "Saturation, Brightness",
                      "Color",
                      "Shape, Icon",
                      "Enclosure, Connection",
                      "Line Pattern",
                      "Line Endings",
                      "Line Weight")
aim2de <- aim2de[c(1,2,5,7,8,9,10),] #removing inappropriate values (zero values and
encodings that don't make sense)

fisher.test(aim2de)
chisq.test(aim2de)

var.test(aim2d[,1],aim2d[,2])
var.test(aim2de[,1],aim2de[,2])
```

10.4. Appendix D: Task completability plot for pathways (Chapter 4)

Figure 47 – These plots depict the relationship between number of nodes, number of edges, and the ability to complete each of the 10 tasks listed in Table 5, for pathways. In context of each task (i.e. panel), the dark blue dots signify "not completable," while light blue dots signify "completable."



10.5. Appendix E: Protocols for conducting tasks from Chapter 4

Here below are operational protocols for assessing task completability. These tasks were originally defined in Lee et al [46]. It may be presumed for all task protocols that ambiguous results, or indeterminate results (perhaps due to definitions in the protocol, size and density of graph, etc.), are generally to be assessed as not completable. As mentioned in Chapter 4, the protocols in this document specify operational steps for specific and specialized versions of the tasks detailed in Lee et al [46].

Depending on the properties of a graph (e.g. layout, size, density, etc.), completability may be assessed quickly for many of these tasks. However, there may also be graphs for which following the given set of protocols may be time consuming or fatiguing—to account for this, the protocols have been defined to include practical constraints, such as a time limit.

This document contains three sections: definitions of terms used in protocols, an explanation on completability, and the protocols themselves.

Definitions of terms used in protocols

- Node: An entity in a graph. Nodes can be thought of as "nouns."

- Edge: A relationship in a graph, connecting two nodes. Edges can be thought of as "verbs."
- Graph: a set of nodes connected by edges, representing entities and the interrelations among them.
- Degree: A measure of the number of edges connecting to a node. A node with three edges connecting to it has a degree of 3. Degrees can also be used to convey how many "steps removed" two nodes may be from each other—for instance, if two nodes are connected only by a third, those two nodes would be 2 degrees away from each other.
- Adjacent Node: A node that directly connects to the node at hand. Adjacent nodes have a degree of 1 in relation to the node at hand.
- Connected Component: A subgraph in which any two nodes are connected to each other through paths. This nodes in the subgraph would be not be connected to any additional nodes in the larger graph.
- Dyad: a pair of nodes and the edge(s) connecting them.
- Encoded:

A Description of Task Completability

The presumption underlying task completability is that, if given adequate time and resources, a task may be conducted to completion. Inversely, if a task is marked as not completable, then the task cannot be conducted to completion, even with adequate time and resources. In a sense, completability is a "lower bound" on the whether or not a task is doable.

This measure of completability is used rather than accuracy since fully completing a given task may require a substantial amount of time and effort for certain graphs.

For a task to be completeable, the following criteria must be met:

1. The protocol for the associated task must be followed from beginning to end.
2. If it takes more than N seconds (defined in a protocol) to complete any protocol, it must be marked as not completable. This N second threshold was a practical constraint that was exercised while completing tasks. Such constraints were necessary, as following a defined protocol from beginning to end may take a substantial amount of time for certain graphs.

Due to variability within the graph itself, the same task may not be completable in the same graph in a different region.

Process for randomly selecting a point in a figure:

The R code below can be used to randomly select one or more points in a given image, which may be a necessary step for certain steps depending on the protocol being administered. Both the "generateRandomCoordinates" function and "selectRandomNPointsInImage" function must be loaded into the R environment. The "png" and "jpeg" packages may need to be installed using the install.packages() command, if not already installed by default.

```
generateRandomCoordinates <- function() {
  return( c(sample(1:100 / 100, 1), sample(1:100 / 100, 1)) )
}
```

```
selectRandomNPointsInImage <- function(imgURL, numberOfPoints = 2) {
 # Function handles pngs and jpegs
 library(png)
 library(jpeg)
 if (numberOfPoints < 1) {
  stop("Please enter a valid argument for number of points. It must be numeric and > 0")
 }
 ima = tryCatch({
  readPNG(imgURL)
 }, error = function(e) {
  # Going to try JPEG now
  readJPEG(imgURL)
 }, finally = {
  print("Please use a JPG or PNG")
 })
 plot(0:1, 0:1, type='n', main="'Two Random Points", xlab="x", ylab="y")
 lim <- par()
 rasterImage(ima, lim$usr[1], lim$usr[3], lim$usr[2], lim$usr[4])
 li = list()
 for (l in 1:numberOfPoints) {
  li[[l]] = t(as.matrix(generateRandomCoordinates()))
  points(li[[l]])
 }
 print(li)
}

# Example command in R environment
selectRandomNPointsInImage("/Users/nikhilgopal/Downloads/gkq482f1.jpg", 2)
```

Protocols:

Find Common Connection

Description of task: The ability to determine if a set of nodes that directly connects two given nodes
Additional Constraints: Mark as not completable if it takes more than 60 seconds to complete this protocol.
Protocol:

1. Randomly select a node in the graph
2. Randomly select another node in the graph
3. Identify all of the adjacent nodes for the first randomly selected node
    a. If unable to identify all adjacent nodes for the first randomly selected node, mark as not completable.
4. Identify all of the adjacent nodes for the second randomly selected node
    a. If unable to identify all adjacent nodes for the second randomly selected node, mark as not completable.
5. Determine if any of the nodes from the first set are also in the second set.
    a. If able to determine whether the randomly selected nodes have common connections, mark as completable. Otherwise, mark as not completable.

Find Articulation Points

Description of task: The ability to identify nodes that, when removed, results in an unconnected graph.
Additional Constraints: Mark as not completable if it takes more than 60 seconds to complete this protocol.
Protocol:

1. Review all of the nodes in the graph to identify at least one node that connects bodies of other nodes (such that it would disconnect the graph into multiple components or sub-graphs if removed).
    a. If after reviewing all of the nodes in the graph, no articulation points were found, mark the task as completable (as there may not have been any valid articulation points in the graph)
    b. If an articulation point was found, mark the task as completable.
    c. Otherwise, mark the task as not completable.

Find Bridges

Description of task: The ability to identify edges that, when removed, results in an unconnected graph
Additional Constraints: Mark as not completable if it takes more than 60 seconds to complete this protocol.
Protocol:

1. Review all of the edges in the graph to identify at least one edge that connects bodies of other nodes (such that it would disconnect the graph into multiple components or sub-graphs if removed).
    a. If after reviewing all of the edges in the graph, no bridges were found, mark the task as completable (as there may not have been any valid bridges in the graph)
    b. If a bridge was found, mark the task as completable.
    c. Otherwise, mark the task as not completable.

Find Shortest Path

Description of task: The ability to find the shortest path between two nodes
Additional Constraints: Mark as not completable if it takes more than 120 seconds to complete this protocol.
Protocol:

1. Identify the node in the graph that is furthest to the top and to the left. This will be the source node.
2. Randomly select a target node in the graph
    a. Please note that both the source and target node should be from the same connected component (i.e. sub-graph)
3. Starting from the source node, follow the edges through successive adjacent nodes until converging on the target node. Respect direction of edges, if applicable. This is essentially a visual graph search, and is akin to using your finger to travel from one end of a maze to another. Repeat this process until a path is found.
    a. If more than one path has been found, then mark the task as completeable
    b. If one path has been found, repeat this step
    c. If not paths have been found, but every possible path has been assessed, mark as completable
    d. Otherwise, mark as not completable

Find Clusters

This task is not necessary to complete, since it is difficult to agree on the definition of a cluster. However, for thoroughness, I will list how I determined if a cluster exists.

Description of task: The ability to distinguish groups of nodes within a graph
Additional Constraints: Mark as not completable if it takes more than 30 seconds to complete this protocol.
Protocol:

1. Review the nodes in the graph
2. Identify groups of nodes that are closer in proximity to each other than to other nodes in the graph
    a. If at least two different groups of nodes may be identified, mark as completable.
    b. If there is another indicator to denote a grouping (e.g. color, enclosure, etc.), then mark this task as completable.
        i. If unable to identify at least two different groups of nodes, mark as not completable.

Find Connected Components
Description of task: The ability to find connected components (two or more nodes connected by edges, such as a sub-graph)
Additional Constraints: Mark as not completable if it takes more than 60 seconds to complete this protocol.
Protocol:

1. Review all of the nodes in the graph
2. Identify disconnected groups of sub-graphs. Sub-graphs are considered distinct if they are unconnected. If sub-graph A is contains no node that is connected to a node in sub-graph B, then sub-graph A and B are considered to be two distinct connected components. In this particular case, a single isolated node is **not** considered its own connected component—the minimal representation of a connected component is a dyad.

Find Node Attributes
Description of task: The ability to identify nodes defined by specific visual attributes
Additional Constraints: Mark as not completable if it takes more than 60 seconds to complete this protocol.
Protocol:

1. Review all of the nodes in the graph
2. If nodes are visually uniform, or otherwise not encoded, mark this task as not completable. Otherwise:
    a. For each visual attribute encoding data in the graph (e.g. color, size, shape, etc.).
        i. Review all of the nodes in the graph encoded with the visual attribute at hand. This step may alternatively be described as a visual filtering step.
    b. If, for each attribute, one is able to review all of the nodes with those attributes, mark the task as completable. Otherwise, mark as not completable.

Find Edge Attributes
Description of task: The ability to identify edges defined by specific visual attributes

Additional Constraints: Mark as not completable if it takes more than 60 seconds to complete this protocol.
Protocol:

1. Review all of the edges in the graph
2. If edges are visually uniform, or otherwise not encoded, mark this task as not completable
    a. For each visual attribute encoding data in the graph (e.g. color, thickness, shape, etc.). Please note that length is not considered a visual attribute (due to variability resulting from layouts).
        i. Identify edges using on the visual attribute at hand.
    b. If it is possible to identify edges using at least one of the visual attributes encoded in the graph, mark as completable. Otherwise, mark as not completable.

Follow a Path
Description of task: The ability to follow a predetermined path through a graph
Additional Constraints: Mark as not completable if it takes more than 30 seconds to complete this protocol.
Protocol:

1. Randomly select a node in the graph
2. Starting from that node, conduct a random walk by hand. However, this random walk is modified such that one cannot visit the previous node, not the current node on the next step (although it is legal to revisit previously visited nodes as long as it is not the last node that was visited—this is to allow for cyclical paths). Stop after 4 random walk steps. Respect direction of edges, if applicable.
    a. If able to complete the modified random walk as defined, mark as completeable. Otherwise, mark as not completable.

Finding Adjacency and Accessibility
Description of task: The ability to recognize that another node is connected to or accessible from a given node.
Additional Constraints: Mark as not completable if it takes more than 60 seconds to complete this protocol.
Protocol:

1. Randomly select a node in the graph
2. Review all of the nodes in the graph and determine if it is possible to identify all of the adjacent nodes for the randomly selected node. Respect direction of edges, if applicable.
    a. If possible to identify all adjacent nodes, mark as completable. Otherwise, mark as not completable.

A Note On Labeling Figures
The process of selecting figures to characterize was detailed in Chapter 4. In short, the label for a figure was the term used to describe the figure in the caption. However, included below is a protocol for labeling figures as well.
Protocol:

1. If there is one term the author(s) use to refer to the figure, that term becomes the label
2. If there are multiple terms the author(s) use to refer to the figure, the latter term should be used for the label.

Note: Some terms used to describe node-link figures may be used more seldom than others. In Chapter 4, networks and pathways were not only two of the most frequently used labels, but were also used to describe node-link diagrams that were systematically or computationally encoded (as opposed to mental maps, diagrams, etc.)

10.6. Appendix F: Example configuration of Dynamo for the task-focused perception study detailed in Chapter 7

Figure 48 - A screenshot of a network visualization generated using Dynamo. The input table configuring the visual encodings is presented in Table 27.

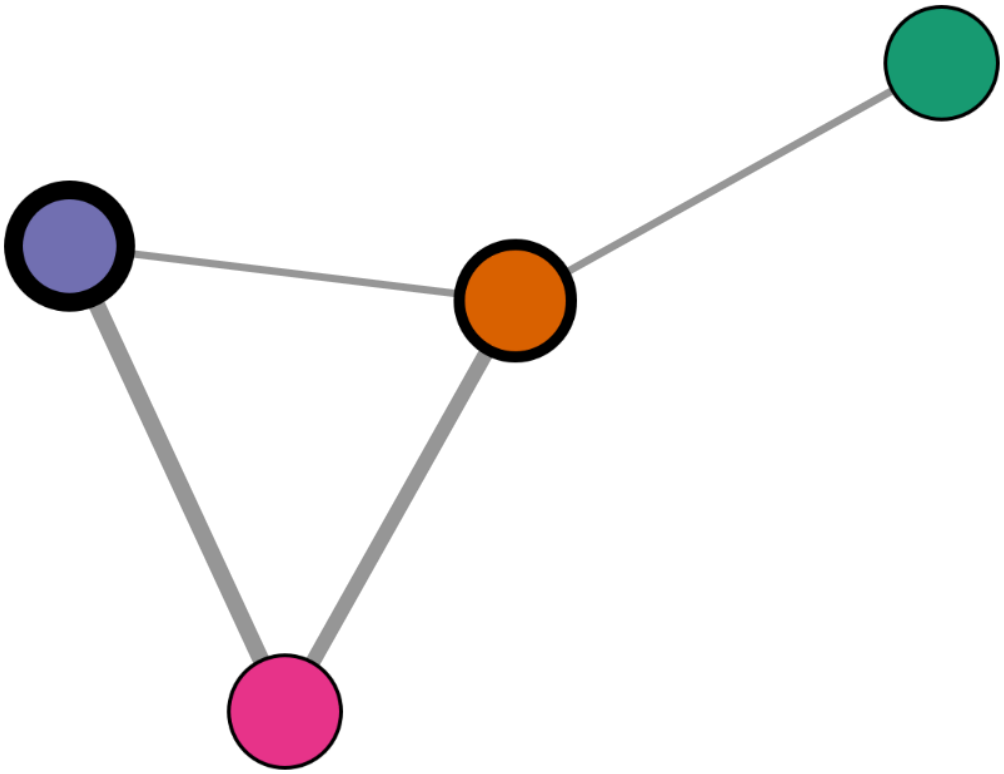


Table 27 - The visual encoding configuration table provided as input to Dynamo in order to generate the network presented in Figure 48.

	Dimension 1-1 Task 1	**Dimension 1-2 Task 1**	**Dimension 2-1 Task 1**	**Dimension 2-2 Task 1**
Node Color (Seq)	0	0	0	0
Node Color (Div)	0	0	0	0
Node Color (Cat)	0	0	10	0
Node Shape (Cat)	0	0	0	0
Node Border (Quant)	0	0	0	0
Node Border (Bin)	0	0	0	10
Node Size (Quant)	0	0	0	0
Node Size (Bin)	0	0	0	0
Edge Width (Quant)	1	12	0	0
Edge Width (Bin)	14	10	0	0
Edge Color (Seq)	6	2	0	0
Edge Color (Div)	4	9	0	0
Edge Color (Cat)	1	9	0	0
Edge Pattern (Cat)	2	2	0	0

In Table 27, the "N-K" notation in the columns describing the dimensions represent dimension N, and replicate K (e.g. "Dimension 1-2"). The assignment table represents the optimal assignment, given the visual encoding ranks for "Task 1". The table cells highlighted in yellow are assigned visual encodings, and the cells highlighted in gray represent invalid encoding assignments (e.g. assigning data attached to nodes to an edge encoding, such as Edge Color). Since visual attributes may be encoded in multiple ways, each manner of encoding is treated as its own encoding. That is, if node size may be encoded quantitatively (e.g. mapping a numerical node attribute directly to node size) and also categorically (e.g. binning numerical node attributes and mapping the bins to discrete sizes), then those are considered two separate encodings. Figure A depicts a network with the visual encoding assignments resulting from input Table 27. Nodes are double encoded with color (categorically) and node border (binned), and edges are double encoded with two types of edge width. Please note that Figure A is from a node-encoding example, and that the edge encodings are only present to provide "noise." When edge-encodings examples are being served, constraints are in place to prevent the same visual attribute from being encoded in multiple ways.

10.7. Appendix G: Decomposing the network variable

Decomposing the network variable: an additional analysis
This section provides additional results from further analysis of the network variable. The intention of this particular analysis was to identify variables (either collected in the data and unused, or derived from collected data) that would increase the percent of variance explained in the data used for modeling. The second goal was to identify variables that have a high variable importance, and the third goal was to assess the predictive capability of the variables through AUC.

Further investigation into decomposing the network variable in the model proposed in this chapter revealed a handful of insights. The primary goal of this investigation was to derive or find variables that would increase the percent of variance explained by the data. The secondary goal is to identify variables that have a high variable importance. The third goal, is to assess predictive capability through AUC.

Additional Model Parameters
Although the goal of this study was primarily to model the relationship between the visual attributes used in the study to predict which node or edge one might select, the model suggested that certain topological attributes of a network play a more important role than anticipated. Model X is still the primary model, due to its predictive performance (AUC of 0.86). However, a number of alternative models containing terms, which represent aspects of the network variable in greater detail, are included below.

Based on prior knowledge and study design, the network variable has been decomposed into the following four parts (as shown in the list below): topology, layout, special properties, and user error.

- Network
    - Topology
        - Centrality of Nodes (Node Degree, Node Betweenness Centrality, Closeness Centrality, Eigenvector Centrality)
    - Layout
        - XY Coordinates of Nodes (X Position and Y Position)
    - Special Property
        - Group (communityGroup)
    - Error
        - User ID as variable (unable to be included in model)

Furthermore, beneath each of the four parts, the list conveys variables that were either collected or derived. Performance of three of the four parts could be evaluated in new random forest models—the user variable was unable to be evaluated. The performances of the models before and after the inclusion of these variables below are presented in Table 29, Table 30, and Table 31.

Variables:

Although a large number of variables were collected during the study, the model with the highest predictive capability did not use all of them. Other variables could be derived through computation on collected data.

Table 28 – Variables used to represent various parts of the decomposed network variables.

Variable	**Source**
Node Degree	Collected
Node Betweenness	Derived
XPosition	Collected
YPosition	Collected
DistanceFromCenterOfScreen	Derived
ClusterCoefficient	Derived
CommunityGroup	Derived
UserID	Collected

Derivation of DistanceFromCenterOfScreen
The DistanceFromCenterOfScreen value was derived using the height and width of a participant's screen, as well as the X and Y coordinate positions of the nodes. The X and Y coordinates were simply normalized to fit into a 0 to 1 scale, and then Euclidean distance was calculated from each node position (XY coordinate pair) to the center of the screen (0.5, 0.5).

Derivation of CommunityGroup
The communityGroup variable was derived using the spinglass.community function in the igraph library (in the R environment). The spinglass algorithm is used to identify communities in a network (i.e. groups of nodes with many edges interconnecting them, while that same group has fewer edges connecting nodes part of that particular community to nodes outside of that community) [33], [92].

Derivation of ClusteringCoefficient
The clusteringCoefficient variable was derived using the transitivity function in the igraph library (in the R environment) [33]. Clustering coefficient is the probability that the adjacent nodes of a node are themselves connected.

Metrics:
% Variance Explained: % Variance Explained is a cross-validated measure of out of bag error. Concisely, it is a measure of the variability in the data that is accounted for by the model. As more variables are added to the model, and consequently as model complexity increases, the % variance explained is also expected to increase. However, % Variance Explained is distinct from AUC.

AUC: AUC is a measure of predictive power (typically obtained in context of an ROC curve). In short, it is a measure of how often the predicted outcome calculated from randomly drawn data is correct.

Under certain circumstances, such as when a model is overfit, the % Variance Explained may increase while AUC decreases. In terms of the single variable additions to a null model, this would be interpreted to mean that although the additional variable explains more of the variance in the data, it simultaneously reduces the predictive capability of the model.

Assessment of additional variables

Each of the three parts (resulting from decomposition of the network variable, and listed in the bullet list above) is presented in its own section. The first section covers the addition of centrality measures, followed by a section covering the addition of layout variables, followed by a section covering the addition of special properties.

Centralities

Table 29 includes performance results from the addition of a single centrality variable. The first two rows of the table depict models using only encodings, and encodings with the network variable. The remaining four rows detail performance and variable importance values when each of four different centrality measures was added to the model containing encodings and network variable.

Of the three centrality measures investigated, eigenvector centrality had the highest variance explained, while the variance explained increased by similar proportions. The similarity in the variance explained was expected, as a prior study has shown that centrality measures are correlated by approximately 80%. However, the AUC only increased by 0.02 when including any of the four types of centrality.

The variable importance measure for each of the included centrality variables ranked at the very bottom of the list of variable importance measure, except for eigenvector centrality, which ranked third—this suggests that eigenvector centrality may have some predictive ability in estimating which node a user may select [93], [94]. However, Valente et al found that degree centrality and eigenvector centrality have a correlation of 0.92, which implies that an algorithmic difference between eigenvector centrality and degree centrality may account for why eigenvector centrality has a much higher variable importance measure than degree centrality when included in the random forest model [93].

Table 29 – Performance of various centrality measures as predictor variables when added to a random forest model containing only visual encodings and a network variable.

% Variance Explained	AUC	Variables	Importance
21.91%	0.77	Encodings	1. Node Border Width 2. Node Size 3. Node Color Saturation 4. Node Color Value 5. Node Shape 6. Node Color Hue
18.25%	0.75	Encodings + Network	1. Network 2. Node Border Width 3. Node Size 4. Node Color Saturation 5. Node Color Value 6. Node Shape 7. Node Color Hue
22.25%	0.77	Encodings + Network + DegreeCentrality	1. Network 2. Node Border Width 3. Node Size 4. Node Color Saturation 5. Node Color Value 6. Node Shape 7. Node Color Hue 8. Degree Centrality
22.43%	0.77	Encodings + Network + BetweennessCentrality	1. Network 2. Node Border Width 3. Node Size 4. Node Color Saturation 5. Node Color Value 6. Node Shape 7. Node Color Hue 8. Node Betweenness Centrality
22.53%	0.77	Encodings + Network + ClosenessCentrality	1. Network 2. Node Border Width 3. Node Size 4. Node Color Saturation 5. Node Color Value 6. Node Shape 7. Node Color Hue 8. Closeness Centrality
22.21%	0.77	Encodings + Network + EigenvectorCentrality	1. Network 2. Node Border Width 3. Eigenvector Centrality 4. Node Size 5. Node Color Saturation 6. Node Color Value 7. Node Shape 8. Node Color Hue

Layout

Table 30 includes performance results from the addition of a variables representing layout. The first two rows of the table depict models using only encodings, and encodings with the network variable. The remaining two rows detail performance and variable importance ranks when X and Y coordinate positions were added to the model, and when a variable representing distance from the center of the screen was included.

Both including X-Y positions and distance from the center of the screen increased the amount of variance explained and AUC relative to the model that contained only encodings and the network variable. The variable representing Y-Position and distance from the center of the screen both had variable importance ranks of 2, implying that the vertical position of a node and a node's position relative to the center of the screen may contribute to a node's selectivity. The inclusion of either the XY coordinate positions or distance from the center of the screen increases the AUC, suggesting that the addition of either of those variables to the model can improve predictive capabilities.

Table 30 - Performance of variables representing physical location of nodes as predictor variables, when added to a random forest model containing only visual encodings and a network variable.

% Variance Explained	AUC	Variables	Importance
21.91%	0.77	Encodings	1. Node Border Width 2. Node Size 3. Node Color Saturation 4. Node Color Value 5. Node Shape 6. Node Color Hue
18.25%	0.75	Encodings + Network	1. Network 2. Node Border Width 3. Node Size 4. Node Color Saturation 5. Node Color Value 6. Node Shape 7. Node Color Hue
26.58%	0.80	Encodings + Network + XPosition + YPosition	1. Network 2. Y Position 3. Node Border Width 4. X Position 5. Node Size 6. Node Shape 7. Node Color Saturation 8. Node Color Value 9. Node Hue
24.21%	0.78	Encodings + Network + DistanceFromCenterOfScreen	1. Network 2. Distance from Center of Screen 3. Node Border Width 4. Node Size 5. Node Color Saturation 6. Node Color Value 7. Node Shape 8. Node Hue

Special Properties

Table 31 includes performance results from the addition of a variables representing special properties. The first two rows of the table depict models using only encodings, and encodings with the network variable. The remaining two rows detail performance and variable importance ranks when variables representing clustering coefficient and clustering community were included to the model originally containing only encodings and the network variable.

Neither variable, when added to the model containing only encodings and the network variable, increased the percentage of variance explained in the data, nor the AUC.

Furthermore, the variable importance ranks for both variables were at the bottom of the ranked list. Based on these three observations, it seems that neither clustering coefficient nor communities found in the network provide any additional explanatory power, or predictive power. However, it is vital to re-emphasize that the networks used in this particular study were four node networks, and that these variables may be more useful in a model representing networks with a larger number of nodes, or higher density.

Table 31 - Performance of variables representing clustering coefficient and community structure as predictor variables, when added to a random forest model containing only visual encodings and a network variable.

% Variance Explained	**AUC**	**Variables**	**Importance**
21.91%	0.77	Encodings	1. Node Border Width 2. Node Size 3. Node Color Saturation 4. Node Color Value 5. Node Shape 6. Node Color Hue
18.25%	0.75	Encodings + Network	1. Network 2. Node Border Width 3. Node Size 4. Node Color Saturation 5. Node Color Value 6. Node Shape 7. Node Color Hue
18.24%	0.75	Encodings + Network + communityGroup	1. Network 2. Node Border Width 3. Node Size 4. Node Color Saturation 5. Node Color Value 6. Node Shape 7. Node Hue 8. Community Group
18.40%	0.75	Encodings + Network + clusteringCoefficient	1. Network 2. Node Border Width 3. Node Size 4. Node Color Saturation 5. Node Color Value 6. Node Shape 7. Node Hue 8. Clustering Coefficient

From the results detailed above, a revised model including eigenvector centrality and XY position was created and evaluated, which resulted in a model that explained 26.82% of variance in the dataset, an AUC of 0.80, and the following variable importance ranks:

- Network
- Y Position
- Node Border Width
- X Position
- Node Size
- Eigenvector Centrality
- Node Shape
- Node Color Saturation
- Node Color Value
- Node Color Hue

This model described above is the same model that was described in Chapter 7.4.

10.8. Appendix H: Single Variable Logistic Regression Models (Chapter 7)

Performance of Single Variable Logistic Regression Models for Nodes
In order to contextualize the performance of the random forest model, this section provides an overview of the performance of several logistic regression models containing only a single variable. Although I have already established, in the sections of Chapter 7, that there are numerous reasons for choosing a random forest model over a regression model to represent the expected relationships among variables from this study, I provide a table containing results from regression models to serve as a baseline to aid in interpretation of random forest model performance. The performance of each regression model is detailed in Table 32 and Table 33 below. The full logistic regression model summaries are available below the tables.

As shown in Table 32 and Table 33, the predictive capability of any model with a single predictor variable (whether logistic regression or random forest regression) exhibited modest improvement in predictive capability.

Table 32 – AUC values from node models between logistic regression models using only one predictor variable, and random forest regression models using only one predictor variable.

Variable	Logistic Regression AUC	Random Forest AUC
Node Degree	0.57	0.48
Node Border	0.57	0.62
Node Size	0.63	0.53
Node Color Value	0.56	0.58
Node Color Saturation	0.56	0.60
Node Color Hue	0.51	0.60
Node Shape	0.52	0.42
Network	0.53	0.03
X Position	0.49	0.52
Y Position	0.57	0.52
Distance from center of screen	0.62	0.52
All Variables	0.73	0.81

Table 33 - AUC values from edge models between logistic regression models using only one predictor variable, and random forest regression models using only one predictor variable.

Variable	Logistic Regression AUC	Random Forest AUC
Edge Width	0.76	0.74
Edge Color Value	0.62	0.75
Edge Color Saturation	0.67	0.76
Edge Color Hue	0.52	0.75
Edge Pattern	0.58	0.43
Edge Length	0.51	0.48
Network	0.54	0.00
All Variables	0.73	0.85

Logistic Regression Summaries:
The following models were trained on ~66% of the dataset collected for the study detailed in Chapter 7. The models were evaluated using the remaining ~33% of the data.

Node Degree

Call: glm(formula = selected ~ eval(parse(text = singleVariable)),
family = binomial(link = "logit"), data = dataset[trainingRows,
])

Coefficients:

(Intercept)	eval(parse(text = singleVariable))
-0.6941	0.3873

Degrees of Freedom: 2581 Total (i.e. Null); 2580 Residual
Null Deviance: 3579
Residual Deviance: 3538 AIC: 3542
AUC: 0.5634069

Node Border Width
Call: glm(formula = selected ~ eval(parse(text = singleVariable)),
family = binomial(link = "logit"), data = dataset[trainingRows,
])

Coefficients:

(Intercept)	eval(parse(text = singleVariable))
-0.3306	0.2128

Degrees of Freedom: 2581 Total (i.e. Null); 2580 Residual
Null Deviance: 3579
Residual Deviance: 3493 AIC: 3497
AUC: 0.5650433

Node Size
Call: glm(formula = selected ~ eval(parse(text = singleVariable)),
family = binomial(link = "logit"), data = dataset[trainingRows,
])

Coefficients:

(Intercept)	eval(parse(text = singleVariable))
-2.61121	0.09957

Degrees of Freedom: 2581 Total (i.e. Null); 2580 Residual
Null Deviance: 3579
Residual Deviance: 3369 AIC: 3373
AUC: 0.6303886

Node Color Value
Call: glm(formula = selected ~ eval(parse(text = singleVariable)),
family = binomial(link = "logit"), data = dataset[trainingRows,
])

Coefficients:

(Intercept)	eval(parse(text = singleVariable))
0.898	-1.135

Degrees of Freedom: 2581 Total (i.e. Null); 2580 Residual
Null Deviance: 3578
Residual Deviance: 3559   AIC: 3563
AUC: 0.549312

Node Color Saturation

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                      -0.2614                              0.8346

Degrees of Freedom: 2581 Total (i.e. Null);  2580 Residual
Null Deviance:    3579
Residual Deviance: 3527         AIC: 3531
AUC: 0.561424
```

Node Color Hue

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                    -0.09067                             0.19113

Degrees of Freedom: 2581 Total (i.e. Null);  2580 Residual
Null Deviance:    3579
Residual Deviance: 3576         AIC: 3580
AUC: 0.5107136
```

Node Shape

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                                  (Intercept)
                                      -0.5108
      eval(parse(text = singleVariable))ellipse
                                       0.4914
     eval(parse(text = singleVariable))heptagon
                                       0.6779
      eval(parse(text = singleVariable))hexagon
                                       0.2595
      eval(parse(text = singleVariable))octagon
```

0.6931
eval(parse(text = singleVariable))pentagon
0.8473
eval(parse(text = singleVariable))rectangle
0.6751
eval(parse(text = singleVariable))rhomboid
0.1335
eval(parse(text = singleVariable))roundrectangle
0.7457
eval(parse(text = singleVariable))star
0.6931
eval(parse(text = singleVariable))triangle
0.3736
eval(parse(text = singleVariable))vee
0.3882

Degrees of Freedom: 2581 Total (i.e. Null); 2570 Residual
Null Deviance: 3579
Residual Deviance: 3566 AIC: 3590
AUC: 0.5305005

Network
Call: glm(formula = selected ~ eval(parse(text = singleVariable)),
family = binomial(link = "logit"), data = dataset[trainingRows,
])

Coefficients:

(Intercept)	eval(parse(text = singleVariable))rn10
-1.268e-01	2.850e-01
eval(parse(text = singleVariable))rn11	eval(parse(text = singleVariable))rn12
1.268e-01	2.047e-01
eval(parse(text = singleVariable))rn13	eval(parse(text = singleVariable))rn14
2.535e-01	-6.899e-02
eval(parse(text = singleVariable))rn15	eval(parse(text = singleVariable))rn16
4.290e-01	4.451e-15
eval(parse(text = singleVariable))rn17	eval(parse(text = singleVariable))rn18
2.574e-01	1.268e-01
eval(parse(text = singleVariable))rn19	eval(parse(text = singleVariable))rn2
8.969e-03	-5.557e-02
eval(parse(text = singleVariable))rn20	eval(parse(text = singleVariable))rn21
-6.780e-03	2.047e-01
eval(parse(text = singleVariable))rn22	eval(parse(text = singleVariable))rn23
5.077e-02	5.575e-01
eval(parse(text = singleVariable))rn24	eval(parse(text = singleVariable))rn25
3.616e-01	3.810e-01
eval(parse(text = singleVariable))rn26	eval(parse(text = singleVariable))rn27
-6.780e-03	-1.755e-01
eval(parse(text = singleVariable))rn28	eval(parse(text = singleVariable))rn29
2.850e-01	3.974e-02

```
 eval(parse(text = singleVariable))rn3  eval(parse(text = singleVariable))rn30
                         3.499e-01                          1.575e-01
eval(parse(text = singleVariable))rn31  eval(parse(text = singleVariable))rn32
                        -6.780e-03                          4.879e-02
eval(parse(text = singleVariable))rn33  eval(parse(text = singleVariable))rn34
                         1.913e-01                          1.268e-01
 eval(parse(text = singleVariable))rn4   eval(parse(text = singleVariable))rn5
                         1.268e-01                          1.021e-01
 eval(parse(text = singleVariable))rn6   eval(parse(text = singleVariable))rn7
                         7.411e-02                          1.268e-01
 eval(parse(text = singleVariable))rn8   eval(parse(text = singleVariable))rn9
                         9.935e-02                          1.585e-01

Degrees of Freedom: 2581 Total (i.e. Null);  2548 Residual
Null Deviance:     3579
Residual Deviance: 3564          AIC: 3632
AUC: 0.5432656
```

X Coordinate Position

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                      0.05186                            -0.09256

Degrees of Freedom: 2581 Total (i.e. Null);  2580 Residual
Null Deviance:     3579
Residual Deviance: 3579          AIC: 3583
AUC: 0.5041391
```

Y Coordinate Position

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                       0.4456                             -0.9364

Degrees of Freedom: 2581 Total (i.e. Null);  2580 Residual
Null Deviance:     3579
Residual Deviance: 3534          AIC: 3538
AUC: 0.5740085
```

Distance From Center of Screen

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
```

])

Coefficients:

(Intercept)	eval(parse(text = singleVariable))
0.9696	-3.0426

Degrees of Freedom: 2581 Total (i.e. Null); 2580 Residual
Null Deviance: 3579
Residual Deviance: 3469 AIC: 3473
AUC: 0.6184965

All Node Variables From Above In a Single Model
Call: glm(formula = selected ~ ., family = binomial(link = "logit"),
data = expd.nodes.1[trainingRows, -6])

Coefficients:

(Intercept)	nodeshapeellipse	nodeshapeheptagon
-3.4863605	0.5899565	0.3619640
nodeshapehexagon	nodeshapeoctagon	nodeshapepentagon
0.3062264	1.0323397	0.6199546
nodeshaperectangle	nodeshaperhomboid	nodeshaperoundrectangle
0.3305865	-0.3531556	0.4383240
nodeshapestar	nodeshapetriangle	nodeshapevee
0.4796394	0.3097762	0.2481787
networkrn10	networkrn11	networkrn12
0.2147515	-0.1492117	0.1030205
networkrn13	networkrn14	networkrn15
-0.5446460	-0.5080914	-0.0494916
networkrn16	networkrn17	networkrn18
-0.6305713	-0.4940403	-1.4312409
networkrn19	networkrn2	networkrn20
-0.0515676	0.1206163	0.7059662
networkrn21	networkrn22	networkrn23
-0.3525774	0.1973717	-0.0316591
networkrn24	networkrn25	networkrn26
-0.0915357	-0.4451621	-0.8826462
networkrn27	networkrn28	networkrn29
0.4431387	-0.2762414	-0.5730637
networkrn3	networkrn30	networkrn31
-0.5294955	-0.4396934	-0.5949360
networkrn32	networkrn33	networkrn34
-0.0004802	-0.3930145	-0.1322973
networkrn4	networkrn5	networkrn6
-0.0113852	-0.1575507	0.2078169
networkrn7	networkrn8	networkrn9
-0.1363497	-0.5116286	-0.1576340
nodeheight	numConnected	nodeborderwidth
0.1164512	0.8881960	0.1369196
nodeHue	nodeSaturation	nodeValue

-0.1964144 1.0667418 -2.6051343

Degrees of Freedom: 2581 Total (i.e. Null); 2531 Residual
Null Deviance: 3579
Residual Deviance: 3067 AIC: 3169
AUC: 0.7437774

Edge Width

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                      -1.9367                              0.5083

Degrees of Freedom: 1235 Total (i.e. Null);  1234 Residual
Null Deviance:     1713
Residual Deviance: 1486          AIC: 1490
AUC: 0.76766
```

Edge Color Value

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                        2.378                             -2.933

Degrees of Freedom: 1235 Total (i.e. Null);  1234 Residual
Null Deviance:     1713
Residual Deviance: 1657          AIC: 1661
AUC: 0.6268447
```

Edge Color Saturation

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                      -0.7934                              2.1894

Degrees of Freedom: 1235 Total (i.e. Null);  1234 Residual
Null Deviance:     1713
Residual Deviance: 1586          AIC: 1590
AUC: 0.6689592
```

Edge Color Hue

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                      -0.2411                              0.6840

Degrees of Freedom: 1235 Total (i.e. Null);  1234 Residual
Null Deviance:     1713
Residual Deviance: 1700          AIC: 1704
AUC: 0.5468543
```

Edge Pattern

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                        (Intercept)  eval(parse(text = singleVariable))dotted
                            -0.6274                                -0.0759
 eval(parse(text = singleVariable))solid
                             0.8782

Degrees of Freedom: 1235 Total (i.e. Null);  1233 Residual
Null Deviance:     1713
Residual Deviance: 1667          AIC: 1673
AUC: 0.5834322
```

Network

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                       (Intercept)  eval(parse(text = singleVariable))rn10
                         1.252e-01                            -6.062e-02
eval(parse(text = singleVariable))rn11  eval(parse(text = singleVariable))rn12
                         -4.735e-01                            -1.696e-01
eval(parse(text = singleVariable))rn13  eval(parse(text = singleVariable))rn14
                         -2.877e-01                            -6.360e-01
eval(parse(text = singleVariable))rn15  eval(parse(text = singleVariable))rn16
                          2.899e-02                             4.189e-02
eval(parse(text = singleVariable))rn17  eval(parse(text = singleVariable))rn18
                         -2.922e-01                            -3.934e-01
eval(parse(text = singleVariable))rn19   eval(parse(text = singleVariable))rn2
                         -8.071e-02                             9.798e-02
```

```
eval(parse(text = singleVariable))rn20  eval(parse(text = singleVariable))rn21
                      1.148e-15                       2.670e-03
eval(parse(text = singleVariable))rn22  eval(parse(text = singleVariable))rn23
                      8.615e-02                      -3.387e-01
eval(parse(text = singleVariable))rn24  eval(parse(text = singleVariable))rn25
                     -6.360e-01                       2.113e-01
eval(parse(text = singleVariable))rn26  eval(parse(text = singleVariable))rn27
                      2.626e-01                      -6.800e-02
eval(parse(text = singleVariable))rn28  eval(parse(text = singleVariable))rn29
                      4.919e-02                      -2.122e-01
 eval(parse(text = singleVariable))rn3  eval(parse(text = singleVariable))rn30
                     -1.252e-01                      -1.183e-02
eval(parse(text = singleVariable))rn31  eval(parse(text = singleVariable))rn32
                     -1.252e-01                       1.468e-01
eval(parse(text = singleVariable))rn33  eval(parse(text = singleVariable))rn34
                     -8.071e-02                       1.112e-01
 eval(parse(text = singleVariable))rn4   eval(parse(text = singleVariable))rn5
                     -1.717e-01                       1.625e-01
 eval(parse(text = singleVariable))rn6   eval(parse(text = singleVariable))rn7
                      9.798e-02                       7.551e-02
 eval(parse(text = singleVariable))rn8   eval(parse(text = singleVariable))rn9
                     -7.387e-02                      -4.766e-01

Degrees of Freedom: 1235 Total (i.e. Null);  1202 Residual
Null Deviance:     1713
Residual Deviance: 1699         AIC: 1767
AUC: 0.5591578
```

Edge Length

```
Call:  glm(formula = selected ~ eval(parse(text = singleVariable)),
    family = binomial(link = "logit"), data = dataset[trainingRows,
        ])

Coefficients:
                  (Intercept)  eval(parse(text = singleVariable))
                     -0.4845                          1.0561

Degrees of Freedom: 1235 Total (i.e. Null);  1234 Residual
Null Deviance:     1713
Residual Deviance: 1708         AIC: 1712
AUC: 0.5456128
```

All Edge Variables From Above In a Single Model

```
Call:  glm(formula = selected ~ ., family = binomial(link = "logit"),
    data = expd.nodes.1[trainingRows, -6])

Coefficients:
          (Intercept)       nodeshapeellipse       nodeshapeheptagon
            -3.146654               0.322553               -0.180711
```

nodeshapehexagon	nodeshapeoctagon	nodeshapepentagon
-0.130616	0.568832	0.476437
nodeshaperectangle	nodeshaperhomboid	nodeshaperoundrectangle
0.191983	-0.457952	0.121812
nodeshapestar	nodeshapetriangle	nodeshapevee
-0.141067	0.973111	0.481718
networkrn10	networkrn11	networkrn12
0.341244	-0.067945	0.413010
networkrn13	networkrn14	networkrn15
-0.598890	-0.608173	-0.499902
networkrn16	networkrn17	networkrn18
-0.626762	-0.390869	-0.942892
networkrn19	networkrn2	networkrn20
0.129413	-0.189877	-0.007863
networkrn21	networkrn22	networkrn23
-0.743026	0.107063	0.233762
networkrn24	networkrn25	networkrn26
0.064747	0.042155	-0.737940
networkrn27	networkrn28	networkrn29
0.244629	-0.674022	-0.487793
networkrn3	networkrn30	networkrn31
-0.606290	-0.787151	-0.750390
networkrn32	networkrn33	networkrn34
-0.103603	-0.100025	-0.163747
networkrn4	networkrn5	networkrn6
-0.472530	-0.324130	-0.323797
networkrn7	networkrn8	networkrn9
-0.233620	-0.448732	-0.051007
nodeheight	numConnected	nodeborderwidth
0.103860	0.765371	0.147538
nodeHue	nodeSaturation	nodeValue
-0.292854	1.076880	-1.979983

Degrees of Freedom: 1235 Total (i.e. Null); 1185 Residual
Null Deviance: 1713
Residual Deviance: 1507 AIC: 1609
AUC: 0.7253879

10.9. Appendix I: Links to code and datasets

- Code for systematic review (Chapter 3):
  - https://github.com/ngopal/systematic-review-network-figures
- Code for perception study conducted in (Chapter 7):
  - https://github.com/ngopal/VisualEncodingEngine