## Chapter 7

# Networks in Practice

*This section shows how to take the conceptual framework of network analysis and apply it in practice. A historical network analysis will often require a similar series of steps:*

1. *Deciding on a dataset,*
2. *Encoding, collecting, or cleaning the data,*
3. *Importing the data into a network analysis package,*
4. *Analyzing the data,*
5. *Visualizing the data,*
6. *Interpreting the results,*
7. *Drawing conclusions.*

*The framework is not universal, and the process usually requires a lot of repetition and some steps may be omitted or added depending on circumstance. Steps four and five, while covered in a small section in this book, are extremely open-ended, and historians who wish to learn more are encouraged to delve into the tool of their choice using the Further Reading section of this chapter. The steps we take the most time explaining usually take the least time in practice; less than 5% of the time spent on a project will be time spent analyzing and visualizing data. Most time will be spent on collecting, cleaning, and interpreting.*

### Picking a Dataset

How can you know whether the data you have will be amenable for network analysis? The answer depends on the project, of course, but unfortunately it is often difficult to tell at the outset of a project which data will be most useful, if any, for a network analysis. Many network analyses lead to dead ends, and the more experience you have, the earlier you will begin to

notice when a network analysis might not be going anywhere. It is worth remembering though that a dead end is not a bad result; you will have learned something important about your information!

There are two veins of network analysis that need to be considered as you draw your materials together: will you be using networks for exploration or to prove a hypothesis about the past? Each has its own assumptions about data that carry ramifications for your interpretations. **Exploratory** network analysis is based around the idea that the network is important, but in as-yet unknown ways. **Hypothesis-driven** network analysis requires a preconceived idea about the world that the network analysis will either reinforce or discredit. For historical network analysis these two camps are often blurred,[1] but for the purposes of data creation, the distinction is useful.

Exploratory network analysis is by far the more difficult for creating data. If you do not yet have a strong concept of what may be of interest in a network, you will often be at the mercy of what data are most readily available. A good many network-oriented digital history projects begin this way; a historian is made aware of a pre-existing network dataset, which already comes with a set of pre-defined categories, and they explore that dataset in order to find whatever interesting information may arise from it. This is not a flawed approach, but it can be an extremely limiting one. The creation of these datasets is rarely driven by an interest in historical networks, and the categories available may only be able to explore a limited set of questions.

Hypothesis-driven network analysis, although perhaps more foreign to historians from its formal perspective, will reduce the likelihood of an analysis reaching a dead end. The first step is to **operationalize** a historiographic claim.[2] The claim might be that popularity among early electric blues musicians was deeply influenced by their connections to other musicians rather than to record labels. The claim helps constrain the potential dataset in time and scope, and lends itself to the collection of certain data. First you would need to pick nodes and edges: given our operationalization, this will be musicians and record labels connected by recordings and

[1] Fred Gibbs and Trevor Owens (2013), "The Hermeneutics of Data and Historical Writing," in K. Nawrotzki and J. Dougherty (eds), *Writing History in the Digital Age*, Ann Arbor, MI: University of Michigan Press, Webbook edition available at http://dx.doi.org/10.3998/dh.12230987.0001.001.

[2] Franco Moretti (December 2013), "Operationalizing: or, the function of measurement in modern literary theory," *Stanford Literary Lab Pamphlet* 6, http://litlab.stanford.edu/LiteraryLabPamphlet6.pdf.

ɔt be going anywhere. It is worth
not a bad result; you will have
information!
s that need to be considered as you
using networks for exploration or
:h has its own assumptions about
terpretations. **Exploratory** net-
at the network is important, but
**iven** network analysis requires a
: network analysis will either rein-
nalysis these two camps are often
ation, the distinction is useful.
ır the more difficult for creating
ıcept of what may be of interest
cy of what data are most readily
digital history projects begin this
existing network dataset, which
ategories, and they explore that
ting information may arise from
an be an extremely limiting one.
riven by an interest in historical
only be able to explore a limited

though perhaps more foreign to
vill reduce the likelihood of an
p is to **operationalize** a histori-
popularity among early electric
their connections to other musi-
im helps constrain the potential
to the collection of certain data.
edges: given our operationaliza-
els connected by recordings and

Hermeneutics of Data and Histori-
(eds), *Writing History in the Digi-*
Press, Webbook edition available at

zing: or, the function of measurement
*amphlet* 6, http://litlab.stanford.edu/



contracts and collaborations. You must operationalize popularity; perhaps this can be done through the sale of records or the frequency with which certain songs were played on the radio. You can operationalize the strength of connections as well, perhaps as a function of the number of times two musicians recorded together or recorded with a particular label. Because it might be difficult to tell whether a connection to a musician or a label is more important, it would be useful to include the time of events in the data gathered. With all these data, you could then check whether those connected with certain people or labels tended to become more popular, using path lengths, or whether certain musicians were instrumental in introducing new musicians to record labels, by looking at triadic closure. Going into an analysis with a preformed hypothesis or question will make both the data gathering and analysis step much easier.

## Software

It is important to decide on a software package to use before embarking on a network study because it can dictate which data you plan on collecting and how you plan on collecting them. Thankfully, you are not locked in to any particular software choice; file formats are convertible to one another, although the process can occasionally be occult, and sometimes an analysis requires the use of more than one tool. (Don't neglect the utility of your text editor, too, for generating and creating network files! See the file format section, below.)

### *UCINET*

UCINET can be used to perform quite sophisticated analyses. Nevertheless, we do not recommend UCINET for most historians; it is not free software, is Windows-only, and has a steep learning curve.[3] However, for those who are already familiar with matrix mathematics, UCINET can be more intuitive than the other options. It has a lot of advanced features and has a wide user base among social network analysts. (Mac users can run Windows software using a variety of tools; one we use is http://winebottler.kronenberg.org/.) We do not recommend UCINET for creating visualizations. UCINET is available at https://sites.google.com/site/ucinetsoftware/home.

[3] An excellent guide to network analysis via UCINET is Robert Hanneman and Mark Riddle (2005), *Introduction to Social Network Methods*, Riverside, CA: University of California, Riverside. Online textbook, http://faculty.ucr.edu/~hanneman/nettext/.

### *Pajek*

Pajek is a free program for network analysis with more features and algorithms than any other non-command-line tool. We do not recommend it for those starting out with network analysis, as it can be difficult to learn and is not intuitive. But it can perform sophisticated analyses, so those in need of algorithms they can find in no other tool might find Pajek suits their needs. Pajek is also Windows-only, has a wider user base, and newer versions can scale to fairly large networks. We do not recommend Pajek for creating visualizations. However, the Pajek format (which uses the .NET file extension) is an industry standard and most tools can read or create it. Pajek is available at http://pajek.imfm.si/doku.php.

### *Network Workbench and Sci²*

The NWB and the Sci² Tool (both developed in collaboration with one of the authors of this handbook) are similar free tools for the manipulation of data, the analysis of networks, and the creation of visualizations. The first focuses on network analysis, and the second focuses on scientometrics (citation analysis and other similar goals). These tools have more features than the others with regards to data pre-processing, especially in the creation of networks out of both unstructured and structured data, but fewer specifically analytic tools than UCINET or Pajek. They are particularly useful for converting between file formats, they run on all platforms, and are slightly easier to use than Pajek and UCINET, though not as easy to use as NodeXL or Gephi. NWB is available at http://nwb.cns.iu.edu/. Sci² is available at https://sci2.cns.iu.edu/user/index.php (you will need to complete a free registration process to download it).

### *NodeXL*

NodeXL is a free plugin for Microsoft Excel and is again unfortunately only available on the Windows platform. We recommend NodeXL for historians who are familiar with Excel and are just beginning to explore network analysis. The plugin is not as feature-rich as any of the others in this list, but it does make entering and editing data extremely easy and works very well for small datasets. NodeXL also provides some unique visualizations and the ability to import data from other packages; it does not scale well to networks of more than a few thousand nodes and edges. NodeXL can also be used

with more features and algo-
ol. We do not recommend it
as it can be difficult to learn
isticated analyses, so those in
r tool might find Pajek suits
a wider user base, and newer
do not recommend Pajek for
ormat (which uses the .NET
st tools can read or create it.
ku.php.

in collaboration with one of
e tools for the manipulation
eation of visualizations. The
nd focuses on scientometrics
ese tools have more features
essing, especially in the cre-
d structured data, but fewer
ajek. They are particularly
y run on all platforms, and
INET, though not as easy
e at http://nwb.cns.iu.edu/.
/index.php (you will need to
ad it).

l is again unfortunately only
nend NodeXL for historians
ginning to explore network
of the others in this list, but
ly easy and works very well
nique visualizations and the
es not scale well to networks
. NodeXL can also be used



to mine social media connections (Twitter user lists or hashtags, YouTube, Facebook). NodeXL can be downloaded at http://nodexl.codeplex.com/.

### *Gephi*

Gephi is quickly becoming the tool of choice for network analysts who do not need the full suite of algorithms offered by Pajek or UCINET. Although it does not have the data entry or pre-processing features of NWB, Sci[2], or NodeXL, it is relatively easy to use (eclipsed in this only by NodeXL), it is usable on all platforms, it can analyze fairly large networks, and it creates beautiful visualizations. The development community is also extremely active, with improvements being added constantly. **We recommend Gephi for the majority of historians undertaking serious network analysis research.** Gephi is available at http://gephi.github.io.

### *Networks online: D3.js, gexf-js, and sigma.js*

When it comes to network visualizations, an element of interactivity is at the top of most researchers' wish lists. Until recently, the only truly great options for online, interactive network visualizations came out of the Stanford Visualization Group under Jeffrey Heer, including the work of Mike Bostock. This team was responsible for a number of widely used visualization infrastructures, including the **prefuse** toolkit, a Java-based framework for creating visualizations; **flare**, an ActionScript (Adobe Flash) library for creating visualizations; and **protovis**, a JavaScript library for creating visualizations. The team's most recent venture, **D3.js**, is a highly flexible JavaScript-based framework for developing novel visualizations and is currently the industry standard for interactive, online visualizations. Mike Bostock is now a graphics editor at the *New York Times*, and the Stanford Visualization Group has moved to the University of Washington and is now known as the Interactive Data Lab.

D3.js is a complex language, and it can be difficult for beginners — even those familiar with coding — to use effectively.[4] A number of libraries have been created as a layer around D3.js that attempt to ease the process of creating visualizations, including **vega** and **NVD3**. All of these libraries require some knowledge of coding, however a little effort in learning them

[4] A very good introduction and tutorial to D3.js is by Elijah Meeks (2014), *D3.js in Action*, Shelter Island, NY: Manning Publications. As of this writing, the book is being live-written, with Meeks adapting content to take into account readers' feedback. http://www.manning.com/meeks/.

can be rewarded by highly customized interactive networks online. For those who do not want to code a visualization themselves, there are a few options for creating interactive online visualizations using Gephi. **Seadragon web export**, **Sigmajs Exporter**, and **Gexf-JS Web Viewer** are all plugins available through the Gephi marketplace for creating such visualizations (in Gephi, you can add new plugins by clicking on "tools" then "plugins").

## Data in Abstract

Network data match network theory: their basic components are nodes and edges. There are three ways these are generally represented: as matrices, as adjacency lists, and as node and edge lists. Each have their own strengths and weaknesses, and they will be discussed below.

The same example network will be used in all three descriptions: a network of exchange between four fictional cities. The data types will be used to show how network data can have varying degrees of detail.

### *Matrices*

Although not in most historians' toolboxes, the **matrix** is an extremely useful representation for small networks and it happens to double as a simple network visualization. The below is a network of trade between four fictional cities: Netland, Connectia, Graphville, and Nodopolis. A "0" is placed when there is no trade route between cities, and a "1" if there is.

| | Netland | Connectia | Graphville | Nodopolis |
|---|---|---|---|---|
| Netland | | 1 | 0 | 1 |
| Connectia | | | 1 | 1 |
| Graphville | | | | 0 |
| Nodopolis | | | | |

From this matrix, we can infer that Connectia trades with Netland, Graphville, and Nodopolis; and Nodopolis trades with Netland. Notice only the **upper triangle** of the matrix is filled in; the **diagonal** (shaded) is left unfilled because it is not meaningful for cities to trade with themselves, and the **lower triangle** is left unfilled because, in a symmetrical undirected network, any information in that corner would be redundant and identical. We already know that Netland trades with Connectia so we do not need to repeat ourselves. The network is unweighted because all we know is whether

eractive networks online. For those
hemselves, there are a few options
ons using Gephi. **Seadragon web**
**-JS Web Viewer** are all plugins
e for creating such visualizations
icking on "tools" then "plugins").

r basic components are nodes and
erally represented: as matrices, as
ts. Each have their own strengths
ed below.
used in all three descriptions: a
nal cities. The data types will be
varying degrees of detail.

xes, the **matrix** is an extremely
and it happens to double as a
s a network of trade between four
phville, and Nodopolis. A "0" is
een cities, and a "1" if there is.

| | Graphville | Nodopolis |
|---|---|---|
| | 0 | 1 |
| | 1 | 1 |
| | | 0 |
| | | |

Connectia trades with Netland,
trades with Netland. Notice only
l in; the **diagonal** (shaded) is left
ties to trade with themselves, and
use, in a symmetrical undirected
would be redundant and identical.
h Connectia so we do not need to
ed because all we know is whether



trade exists between two cities (represented by 0 or 1), we do not know the amount of trade.

| | Netland | Connectia | Graphville | Nodopolis |
|---|---|---|---|---|
| Netland | | \$10mil | 0 | \$4mil |
| Connectia | | | \$2mil | \$4mil |
| Graphville | | | | 0 |
| Nodopolis | | | | |

This matrix is identical to the previous, but it now represents a weighted network. It shows \$10 million in trade between Connectia and Netland, \$2 million between Connectia and Graphville, \$4 million between Connectia and Nodopolis, and \$4 million between Netland and Nodopolis. This representation could be extended even further.

| | | Target | | | |
|---|---|---|---|---|---|
| | | Netland | Connectia | Graphville | Nodopolis |
| Source | Netland | | \$6mil | 0 | \$1mil |
| | Connectia | \$4mil | | \$1mil | \$3mil |
| | Graphville | 0 | \$1mil | | 0 |
| | Nodopolis | \$3mil | \$1mil | 0 | |

The matrix now represents a directed, weighted network of trade between cities. The directionality means the trade relationships between cities can be represented asymmetrically, thus broken up into their constituent parts. Directional networks require filling both the upper and lower triangle of the matrix. **Source** and **Target** are the network terms of choice for the nodes that do the sending and those that do the receiving, respectively. In the above matrix, Netland (the source) sends \$6 million to Connectia (the target) and \$1 million to Nodopolis (the target). Netland (the target) receives \$4 million from Connectia (the source) and \$3 million from Nodopolis (the source).

If we wanted to extend this representation even further, we could create multiple parallel matrices. Parallel matrices could represent time slices, so each matrix represents trade between cities in a subsequent year. Alternatively, parallel matrices could represent different varieties of trade, e.g. people, money, and goods. This is one method to encode a multiplex network.

Matrices were, at one point, the standard way to represent networks. They are fairly easy to read, do not take up much space, and a lot of the network analysis algorithms are designed using matrix mathematics. As networks have become larger, however, it is becoming more common to represent them in adjacency lists or node and edge lists. The matrix is still readable by most network software, and some programs are optimized for use with matrices, particularly UCINET and, to a lesser extent, NodeXL.

### *Adjacency Lists*

The **adjacency list** is a simple replacement for the matrix and a bit easier when it comes to data entry. Like a matrix, it can be used to represent many varieties of networks.

| | |
|---|---|
| Netland | Connectia |
| Netland | Nodopolis |
| Connectia | Graphville |
| Connectia | Nodopolis |

This adjacency list represents the same network as the first matrix. It is undirected and unweighted. Adding weights is as easy as adding an additional column of data.

| | | Weight |
|---|---|---|
| Netland | Connectia | $10mil |
| Netland | Nodopolis | $4mil |
| Connectia | Graphville | $2mil |
| Connectia | Nodopolis | $4mil |

Adjacency lists can have any number of additional columns for every additional edge trait. For example, if this were a multiplex network encoding different *varieties* of trade, there might be two additional columns: one for goods, another for people. Each additional column could be filled with numerical values. Alternatively, columns could be used to encode the type of tie. In a family business network, a column for "type of relationship" could be filled in as "trade," "marriage," or "both."

Adjacency lists can also be used to represent directed, asymmetric networks, as below (treating columns as "source" and "target" to indicate directionality).

ard way to represent networks.
ıp much space, and a lot of the
using matrix mathematics. As
is becoming more common to
ıd edge lists. The matrix is still
me programs are optimized for
ıd, to a lesser extent, NodeXL.

: for the matrix and a bit easier
x, it can be used to represent

| ectia |
|---|
| polis |
| hville |
| polis |

e network as the first matrix.
eights is as easy as adding an

| | Weight |
|---|---|
| | $10mil |
| | $4mil |
| | $2mil |
| | $4mil |

f additional columns for every
e a multiplex network encoding
wo additional columns: one for
l column could be filled with
ıld be used to encode the type
mn for "type of relationship"
"both."
sent directed, asymmetric net-
rce" and "target" to indicate



| Source | Target | Weight |
|---|---|---|
| Netland | Connectia | $6mil |
| Netland | Nodopolis | $1mil |
| Connectia | Graphville | $1mil |
| Connectia | Nodopolis | $3mil |
| Connectia | Netland | $4mil |
| Graphville | Connectia | $1mil |
| Nodopolis | Netland | $3mil |
| Nodopolis | Connectia | $1mil |

This is identical to the final matrix. In network visualizations, directedness is represented as an arrow going from the source to the target, which implies the directionality of an edge.

### *Node and Edge Lists*

We recommend this data structure for historians embarking on a network analysis. It is widely used, easy to enter data manually, and allows additional information to be appended to nodes, rather than just to edges. The one down side of **Node and Edge Lists** is that they require more initial work, as they involve the creation of two separate tables: one for nodes and one for edges.

| Nodes | |
|---|---|
| ID | Label |
| 1 | Graphville |
| 2 | Nodopolis |
| 3 | Connectia |
| 4 | Netland |

| Edges | |
|---|---|
| 4 | 3 |
| 4 | 2 |
| 3 | 1 |
| 3 | 2 |

This node and edge list is equivalent to the first adjacency list and the first matrix. Notice particularly that nodes are now given unique IDs that are separate from their labels; this becomes useful if, for example, there are multiple cities with the same name. The two tables below show how to add weights and directionality, as well as additional attributes to individual nodes.

| Nodes | | | |
|---|---|---|---|
| ID | Label | Population | Country |
| 1 | Graphville | 700,000 | USA |
| 2 | Nodopolis | 250,000 | Canada |
| 3 | Connectia | 1,000,000 | Canada |
| 4 | Netland | 300,000 | USA |

| Edges | | |
|---|---|---|
| Source | Target | Weight |
| 4 | 3 | $6mil |
| 4 | 2 | $1mil |
| 3 | 1 | $1mil |
| 3 | 2 | $3mil |
| 3 | 4 | $4mil |
| 1 | 3 | $1mil |
| 2 | 4 | $3mil |
| 2 | 3 | $1mil |

Although node and edge lists require more initial setup, they pay off in the end for their ease of data entry and flexibility. Unfortunately, not all software interprets these data structures in the exact same way, so it will be necessary to convert the gathered data into a format that the program can actually read.

### File Formats

There are many file formats for networks, and most are instantiations of the three main data structures covered in the previous section. Below we describe them in some detail, using the four fictional cities as examples.

to the first adjacency list and the
des are now given unique IDs that
omes useful if, for example, there
The two tables below show how to
additional attributes to individual

| ılation | Country |
|---|---|
| )00 | USA |
| )00 | Canada |
| ),000 | Canada |
| )00 | USA |

| | Weight |
|---|---|
| | $6mil |
| | $1mil |
| | $1mil |
| | $3mil |
| | $4mil |
| | $1mil |
| | $3mil |
| | $1mil |

more initial setup, they pay off in
l flexibility. Unfortunately, not all
; in the exact same way, so it will
ta into a format that the program

ks, and most are instantiations of
in the previous section. Below we
our fictional cities as examples.



### *UCINET*

UCINET's data format is fairly simple plain text but also fairly difficult to edit by hand. The most common format used by UCINET is the "full matrix," which incorporates a list of node labels that are defined before the matrix is written out. The matrix is assumed to have the same nodes in the same order on the horizontal and the vertical, as below.

```
dl N = 4
format = fullmatrix
labels:
netland, connectia, graphland, nodopolis
data:
0 1 0 1
0 0 1 1
0 0 0 0
0 0 0 0
```

Take a moment and notice how this is the same data structure as the first matrix example in the previous section (the Netland/Connectia trading system), although the exact specifications of the file format are unique to UCINET. The first line declares the number of nodes (4), the second declares the format, the third and fourth declare the labels, and subsequent lines describe the edges. While this is just a simple text file, we give it the file extension .dl, thus citynetwork.dl.

### *Pajek, NWB, & Sci²*

The standard file formats for Pajek and NWB/Sci² are similar to one another, as they are both node and edge lists encoded in a single plain text file. In the Pajek file format (NET), we call nodes **vertices**, directed edges **arcs**, and undirected edges **edges**.

```
*Vertices 4
1 "Graphville"
2 "Nodopolis"
3 "Connectia"
4 "Netland"
*Edges
4 3
4 2
3 1
3 2
```

If the edges were directed, the subsection would be **Arcs* instead of **Edges*. Weight can be added to each edge by simply adding a number equivalent to the edge weight at the end of the line featuring that edge.

The NWB/Sci[2] file format (*.NWB) is quite similar, although it requires more declarations of network types and variables. It also requires a declaration of each variable type (int = integer, string = string of text, etc.). An example of an NWB file that has both additional node and edge information would look like this.

```
*Nodes   4
id*int   label*string   population*float   country*string
1    "Graphville"   700000    "USA"
2    "Nodopolis"    250000    "Canada"
3    "Connectia"    1000000   "Canada"
4    "Netland"  300000   "USA"
*DirectedEdges 8
source*int     target*int     weight*float
4    3    6000000
4    2    1000000
3    1    1000000
3    2    3000000
3    4    4000000
1    3    1000000
2    4    3000000
2    3    1000000
```

These file formats are very sensitive to small errors, so it is usually best not to edit them directly. That's why we have network programs!

### *GEXF*

Gephi's XML-based file format, GEXF, is saved in plain text with a .gexf extension. It is fairly verbose but allows quite a bit of detail of a network to be saved.

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.2draft" version="1.2">
    <graph mode="static" defaultedgetype="undirected">
        <nodes>
            <node id="0" label="Graphville" />
```

```
            <node id="1" label="Nodopolis" />
            <node id="2" label="Connectia" />
            <node id="3" label="Netland" />
        </nodes>
        <edges>
            <edge id="0" source="3" target="2" />
            <edge id="0" source="3" target="1" />
            <edge id="0" source="2" target="0" />
            <edge id="0" source="2" target="1" />
        </edges>
    </graph>
</gexf>
```

These files are easy to read, but — again — should not be edited directly unless you really know what you're doing.

### *NodeXL*

NodeXL files are not saved in plain text; instead, they are saved in the default Microsoft Excel format. Entering data directly into this format is the easiest of all the others, as it simply requires opening up the file in Excel and editing or adding as necessary, within the familiar spreadsheet environment.

## Network Visualizations

**Matrix diagrams** tend to be used more by computational social scientists than traditional social network analysts. They are the exact, colorized versions of the matrix data structure discussed earlier in this chapter and are good for showing community patterns in medium-to-large networks. They do not suffer from the same clutter as force-directed visualizations, but they also do not lend themselves to be read at the scale of individual actors.

Figure 7.1 is a matrix visualization of character interactions in Victor Hugo's *Les Misérables*. We made this visualization using Excel to reinforce the fact that matrix visualizations are merely data structures that have been colored and zoomed out. Each column is a character in the book, as is each row, and the list of character names is in the same order horizontally and vertically. A cell is shaded red if the character from that row interacted with the character from that column; it is shaded white if they did not.

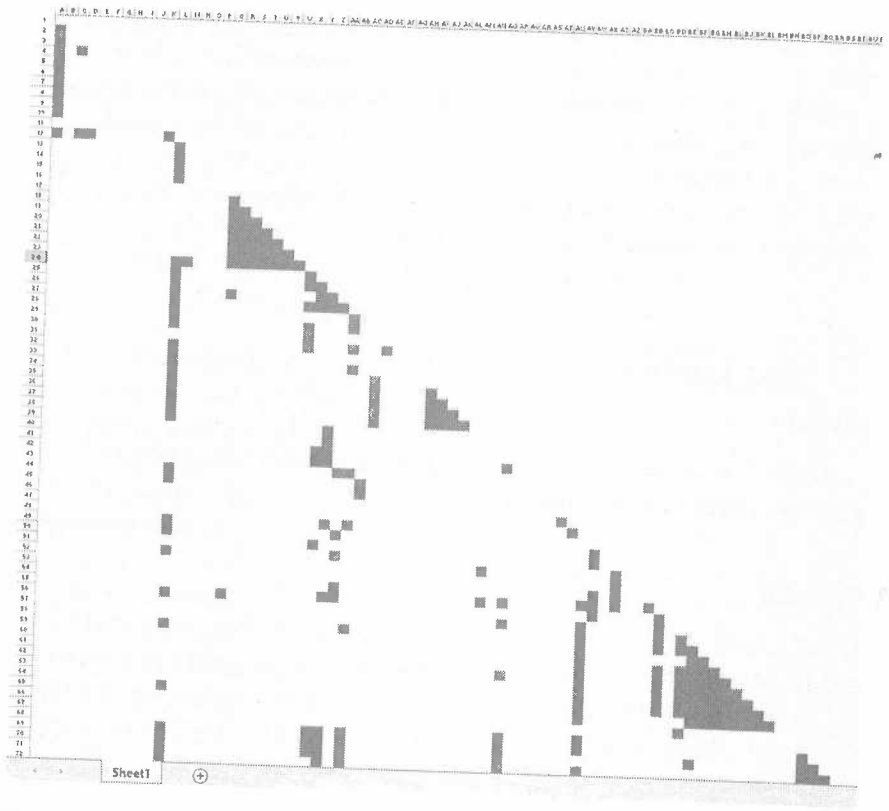
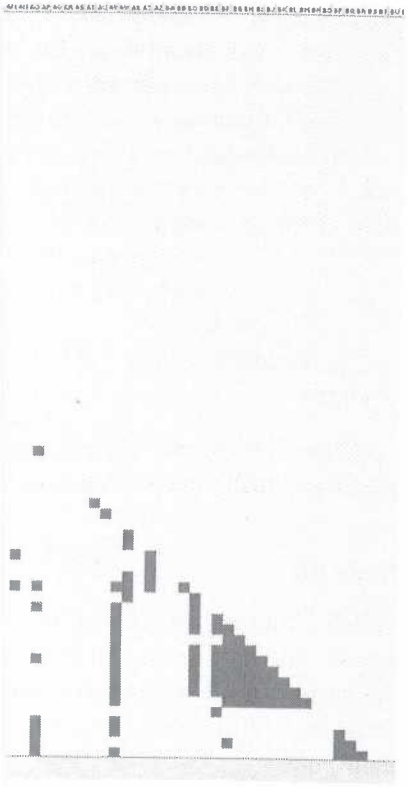



Fig. 7.1 A matrix visualization of *Les Misérables*. Microsoft Excel screenshot used with permission from Microsoft.

Note that only one of the matrix's triangles is filled, because the network is symmetric.

We performed community detection on the network and ordered the characters based on whether they were in a community together. That is why some areas are thick with red cells and others are not; each triangular red cluster represents a community of characters that interact with one another. The vertical columns that feature many red cells are main characters that interact with many other characters in the book. The filled-in column near the left-hand side, for example, is the character interactions of Jean Valjean.

Matrix diagrams can be extended to cover asymmetric networks by using both the matrix's upper and lower triangles. Additional information can be encoded in the intensity of a color (signifying edge weight) or the hue of the shaded cell (indicating different categories of edges).



*les.* Microsoft Excel screenshot used with

igles is filled, because the network

on the network and ordered the
in a community together. That is
and others are not; each triangu-
characters that interact with one
ire many red cells are main char-
aracters in the book. The filled-in
iple, is the character interactions

over asymmetric networks by using
les. Additional information can be
ifying edge weight) or the hue of
gories of edges).



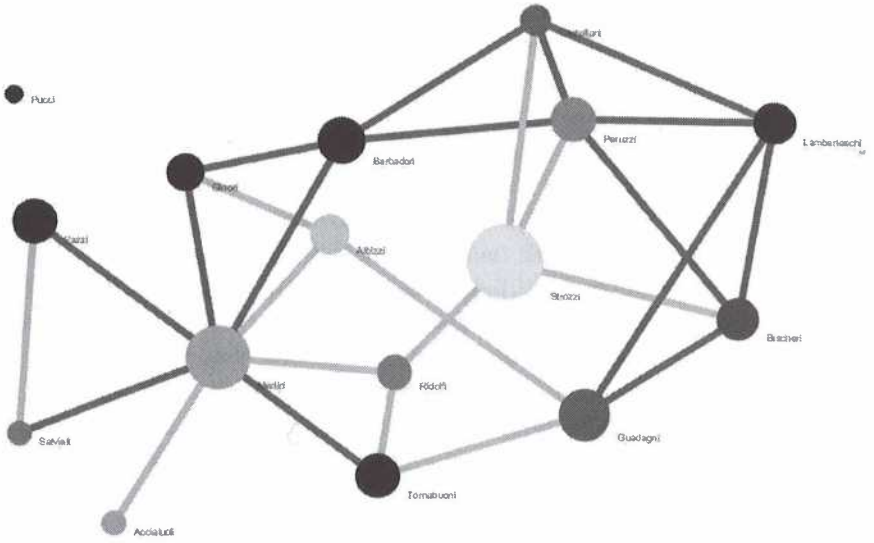



Fig. 7.2 A force-directed layout of inter-familial connections in Renaissance Florence.

**Force-directed layouts** are the visualizations most people think of when they think of network analyses. An example is in Fig. 7.2.

This visualization, created using the Sci[2] Tool, represents a subset of Ansell and Padgett's data concerning the network of connections amongst Renaissance Florentine families.[5] Force-directed layouts like this one attempt to reduce the number of edges that cross one another while simultaneously bringing more directly-connected nodes closer together. They do this by modeling the network as though it were a physics problem, as though each edge were a spring and each node a unit connecting various springs together. The computer simulates this system, letting the springs bounce around until each one is as little stretched as it possibly can be. At the same time, the nodes repel each other, like magnets of the same polarity, so the nodes do not appear too close together. Eventually, the nodes settle into a fairly legible graph like Fig. 7.2. This algorithm is an example of how force-directed layouts work, but they do not all use springs and magnets, and often they are a great deal more complex than how we have described them.

[5] John Padgett and Christopher Ansell (1993), "Robust Action and the Rise of the Medici, 1400–1434," *American Journal of Sociology*, **98**(6), 1259–1319.

There are a few important takeaways from this algorithm. The first is that the layout is generally **stochastic**, or random; there is an element of randomness that will orient the nodes and edges slightly differently every time it is run. The second is that the traditional spatial dimensions (vertical and horizontal) that are so often meaningful in visualizations have no meaning here. There is no $x$- or $y$-axis, and spatial distance from one node to another is not inherently meaningful. For example, had Fig. 7.2 been laid out again, the Acciaiuoli family could just as easily have been closer to the Pazzi than the Salviati family, as opposed to in this case where the reverse is true. To properly read a force-directed network visualization, you need to retrain your visual understanding such that you are aware that it is edges, not spatial distance, which mark nodes as closer or farther away.

This style of visualization becomes more difficult to read as a network grows. Larger instantiations have famously been called "spaghetti-and-meatball visualizations" or "giant hairballs," and it can be impossible to discern any particular details. Still, in some cases, these very large-scale force-directed networks can be useful in discerning patterns at-a-glance.

### *Other visualizations*

Matrix and force-directed visualizations are the two most common network visualizations but they are by no means the only options. A quick search for chord diagrams, radial layouts, arc layouts, hive plots, circle packs, and others will reveal a growing universe of network visualizations. Picking which is appropriate in what situation can be more of an art than a science. Whatever layout you choose: does it help you understand the historical problem? Does it help make the argument more clear?

## When Not To Use Networks

Network analysis can be a powerful method for engaging with a historical dataset but it is often not the most appropriate. Any historical database may be represented as a network with enough contriving but few should be. One could take the Atlantic trade network, including cities, ships, relevant governments and corporations, and connect them all together in a giant multipartite network. It would be extremely difficult to derive any meaning from this network, especially using traditional network analysis methodologies. The LCC, for example, would be meaningless, as it would be impossible for the neighbors of any one node to be neighbors with one another.

Network scientists have developed algorithms for multimodal networks, but they are often created for fairly specific purposes, and one should be very careful before applying them to a different dataset and using that analysis to explore a historiographical question.

Networks, as they are commonly encoded, also suffer from a profound lack of nuance. It is all well to say that, because Henry Oldenburg corresponded with both Gottfried Leibniz and John Milton, he was the short connection between the two men. However, the encoded network holds no information of whether Oldenburg actually transferred information between the two or whether that connection was at all meaningful. In a flight network, Las Vegas and Atlanta might both have very high betweenness centrality because people from many other cities fly to or from those airports, and yet one is much more of a hub than the other. This is because, largely, people tend to fly directly back and forth from Las Vegas, rather than through it, whereas people tend to fly through Atlanta from one city to another. This is the type of information that network analysis, as it is currently used, is not well equipped to handle. Whether this shortcoming affects a historical inquiry depends primarily on the inquiry itself.

When deciding whether to use networks to explore a historiographical question, the first question should be: to what extent is connectivity specifically important to this research project? The next should be: can any of the network analyses my collaborators or I know how to employ be specifically useful in the research project? If either answer is negative, another approach is probably warranted.

## Networks in Action

Once you choose network analysis as a method, you must decide the series of steps to take to turn a historical question into an operational process. There are many ways to reach this goal. No proper path exists; the set of tools and order of steps taken are determined by the task at hand.

For example, let's imagine that you want to find important art dealers in the mid-20th century. You might sit down with the transaction books of private collectors and museums, and enter every art transaction (by hand!) into NodeXL, listing the buyer and seller. If you are on Windows, NodeXL is the obvious choice here because it is the easiest software for entering data by hand. Say that you find a few thousand entities buying or selling art, between whom over 10,000 art pieces changed hands. Knowing this

network is a bit large for NodeXL, you export it in GRAPHML to format and load it into Gephi in order to explore the network. Once in Gephi, you visualize the network in a force-directed layout, noticing immediately some central players in the art world: a few famous dealers, some museums, and so forth. To further refine your search for art dealers worth researching in more depth, you run the appropriate algorithms and look for those with high centrality and low clustering coefficient. These, you reason, must be central figures between whom art flowed to otherwise disconnected communities. You highlight these nodes in red and publish them as an SVG file, to send to your publisher, so the illustration in your book clearly shows how these dealers spanned multiple communities. This would be a good network analysis, and you would detail and discuss these steps so that someone else could re-run your analyses to confirm, refute, or extend.

Other situations would require different steps and different software. Perhaps what you are interested in are the *holes* in the network, the patterning of non-connections. In such a case, you would probably use UCINET. It is beyond the scope of this book to produce tutorials for every potential situation and series of software packages dealing with network analysis. Instead, let us take a simple dataset and work through the process of loading it into the most likely piece of software you might use (Gephi) and see what we might find. We shall do some exploration. The *Further Reading* section at the end of the previous chapter will help the reader move beyond these basics.

Recall that in Chapter 3, you used Notepad++, regular expressions, and OpenRefine to create a comma-separated value file (CSV) of the diplomatic correspondence of the Republic of Texas. The final version of the file you created has a row for each letter listed in the volume, and in each row the name of the sender and the recipient. The file looks like this:

```
source,target
Sam Houston,J. Pinckney Henderson
James Webb,Alc6e La Branche
David G. Burnet,Richard G. Dunlap
...
```

This file is an edge list, as discussed in the previous section on network data types. If you no longer have the file, you can find it online at http://themacroscope.org/2.0/datafiles/texas-correspondence-OpenRefine.csv. Keep this file handy, as you will need to load it after you have installed Gephi. Install Gephi by going to http://gephi.github.io,

export it in GRAPHML to format
e the network. Once in Gephi, you
layout, noticing immediately some
mous dealers, some museums, and
r art dealers worth researching in
lgorithms and look for those with
cient. These, you reason, must be
to otherwise disconnected commu-
nd publish them as an SVG file, to
n in your book clearly shows how
ies. This would be a good network
s these steps so that someone else
fute, or extend.
rent steps and different software.
*holes* in the network, the pattern-
ou would probably use UCINET.
roduce tutorials for every poten-
ges dealing with network analysis.
work through the process of load-
re you might use (Gephi) and see
xploration. The *Further Reading*
will help the reader move beyond

epad++, regular expressions, and
value file (CSV) of the diplomatic
The final version of the file you
the volume, and in each row the
file looks like this:

ed in the previous section on
have the file, you can find
/datafiles/texas-correspondence-
you will need to load it after
going to http://gephi.github.io,



downloading the appropriate install file for your operating system, and running it on your computer.

### *Installing Gephi on OS X Mavericks*[6]

Mac users might have some trouble installing Gephi. We have found that, on Mac OS X Mavericks, Gephi does not load properly after installation. This is a Java-related issue, so you'll need to install an earlier version of Java than the one provided. To fix this, control click (or right-click) on the Gephi package, and select "show package contents." Click on "contents → resources → gephi → etc." Control-click (or right-click) on "gephi.conf" and open with your text editor. Find the line reading:

```
#jdkhome="/path/to/jdk"
```

and paste the following underneath:

```
jdkhome="/System/Library/Java/JavaVirtualMachines/1.6.0.
jdk/Contents/Home"
```

Save that file. Then, go to http://support.apple.com/kb/DL1572 and install the older version of Java (Java 6). Once that is installed, Gephi should run normally.

Run Gephi once it is installed. You will be presented with a welcome window prompting you to open a recent file, create a new project, or load a sample file. Click "New Project" and then click the "Data Laboratory" tab on the horizontal bar at the top of the Gephi window (Fig. 7.3).

You will be presented with a blank screen as you have not yet populated Gephi with the correspondence data. To do so, click "Import Spreadsheet"

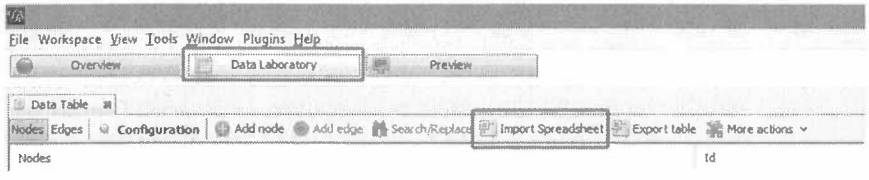



Fig. 7.3 Accessing the Data Laboratory in Gephi.

[6]The Yosemite operating system was released as this text was being finalized. Clement Levallois maintains an excellent web resource of Gephi guides and help files at http://clementlevallois.net/gephi.html including a very detailed guide to installing and running Gephi under Yosemite (as well as Windows 8).

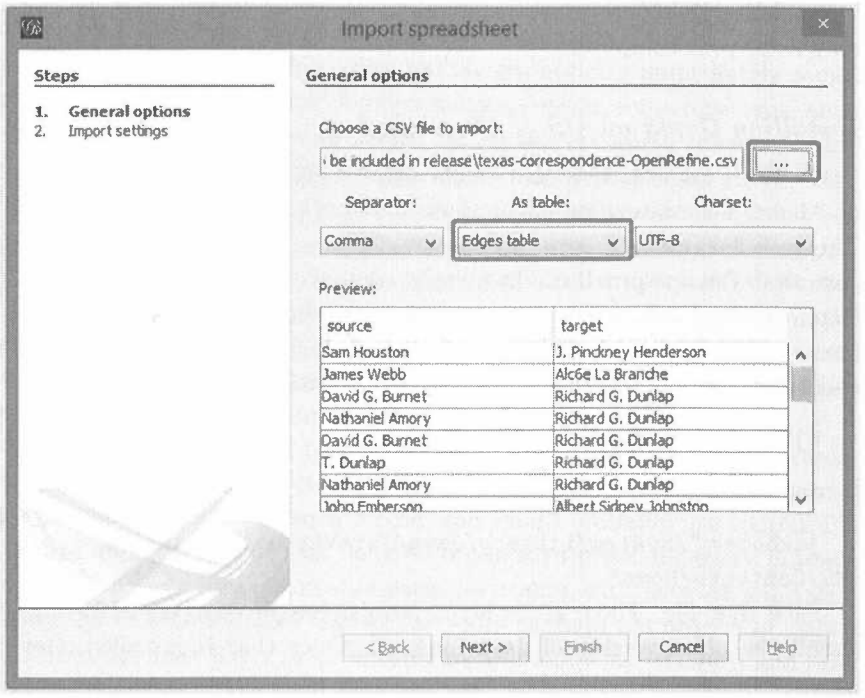
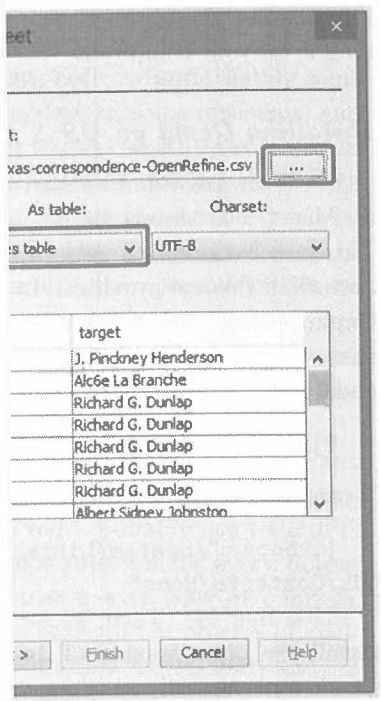



Fig. 7.4 The import spreadsheet dialogue box.

and select the file texas-correspondence-OpenRefine.csv by clicking on the button with the ellipsis (...) and navigating to the file (Fig. 7.4). Make sure that you are importing the file as an "Edges table." Click "Next" and make sure the "Create missing nodes" box has a check mark in it. Click "Finish," and note that you have loaded the Republic of Texas diplomatic correspondence into Gephi.

Gephi is broken up into three panes: Overview, Data Laboratory, and Preview. The Overview pane is used to manipulate the visual properties of the network: change colors of nodes or edges, lay them out in different ways, and so forth. The Overview pane is also where you can apply algorithms to the network, like those you learned about in the previous chapter. The Data Laboratory is for adding, manipulating, and removing data. Once your network is as you want it to be, use the Preview pane to do some final tweaks on the look and feel of the network and to export an image for publication.




eet dialogue box.

penRefine.csv by clicking on the
ing to the file (Fig. 7.4). Make
'Edges table." Click "Next" and
x has a check mark in it. Click
he Republic of Texas diplomatic

Overview, Data Laboratory, and
nipulate the visual properties of
s, lay them out in different ways,
where you can apply algorithms
ut in the previous chapter. The
ting, and removing data. Once
e the Preview pane to do some
work and to export an image for



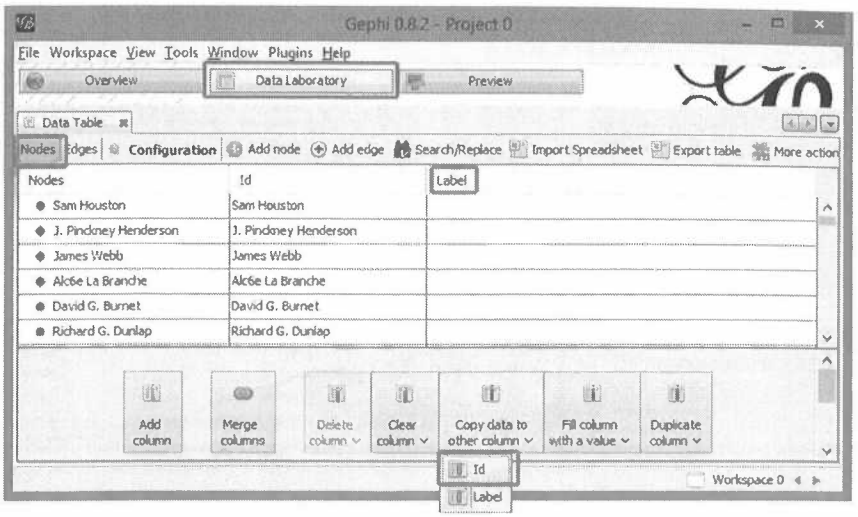



Fig. 7.5 Copying data from one column to another.

There is one tweak that needs to done in the Data Table before the dataset is fully ready to be explored in Gephi. Click on the "Nodes" tab in the Data Table and notice that, of the three columns, "Label" (the furthermost field on the right) is blank in every row. This will be a problem when viewing the network visualization, as those labels are essential for the network to be meaningful.

In the "Nodes" tab, click "Copy data to other column" at the bottom, select "ID", and press "Ok" (Fig. 7.5). Upon doing so, the "Label" column will be filled with the appropriate labels for each correspondent. While you're still in the Data Laboratory, look in the "Edges" tab and notice there is a "Weight" column. Gephi automatically counted every time a letter was sent from correspondent A to correspondent B and summed up all the occurrences, resulting in the "Weight." This means that J. Pinckney Henderson sent three letters to James Webb, because Henderson is in the "Source" column, Webb in the "Target,", and the "Weight" is three.

Clicking on the Overview pane will take you to a visual representation of the network you just imported. In the middle of the screen, you will see your network in the "Graph" tab. The "Context" tab, at the top right, will show that you imported 234 nodes and 394 edges. At first, all the nodes will be randomly strewn across the screen and make little visual sense (Fig. 7.6). Fix this by selecting a layout in the "Layout" tab — the best

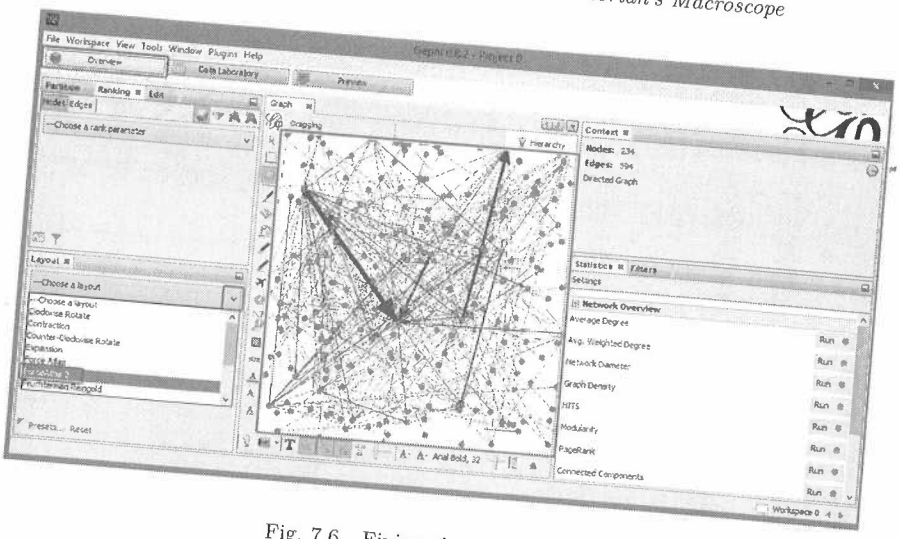


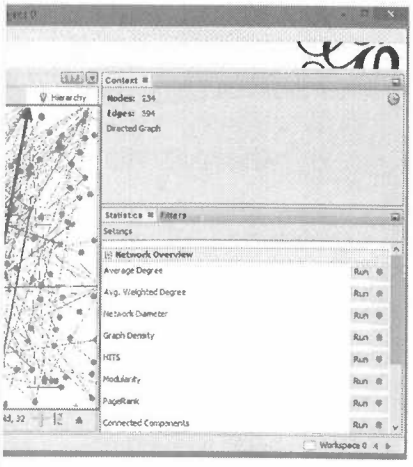

Fig. 7.6 Fixing the layout view.

one for beginners is "Force Atlas 2." Press the "Run" button and watch the nodes and edges reorganize on the screen into something slightly more manageable. After the layout runs for a few minutes, re-press the button (now labeled "Stop") to settle the nodes in their place.

You just ran a force-directed layout, as described earlier in this chapter. Each dot is a correspondent in the network, and lines between dots represent letters sent between individuals. Thicker lines represent more letters, and arrows represent the direction the letters were sent, such that there may be up to two lines connecting any two correspondents (one for each direction).

About two-dozen smaller components of the network will appear to shoot off into the distance, unconnected from the large, connected component in the middle. For the purpose of this exercise, we are not interested in those disconnected components, so the next step will be to filter them out of the network. The first step is to calculate which components of the network are connected to which others; do this by clicking "Run" next to the text that says "Connected Components" in the "Statistics" tab on the right-hand side (Fig. 7.7). Once there, select "UnDirected" and press "OK." Press "Close" when the report pops up indicating that the algorithm has finished running.

Now that this is done, Gephi knows which is the giant connected component (see Chapter 5 for details) and has labeled that component "0". To filter out everything but the giant component, click on the "Filters" tab on the right-hand side and browse to "Component ID *Integer (Node)*" in the



.e layout view.

?ress the "Run" button and watch
screen into something slightly more
a few minutes, re-press the button
es in their place.
as described earlier in this chapter.
ork, and lines between dots represent
er lines represent more letters, and
's were sent, such that there may be
espondents (one for each direction).
nts of the network will appear to
l from the large, connected compo-
this exercise, we are not interested
the next step will be to filter them
calculate which components of the
; do this by clicking "Run" next to
ents" in the "Statistics" tab on the
elect "UnDirected" and press "OK."
o indicating that the algorithm has

s which is the giant connected com-
has labeled that component "0". To
onent, click on the "Filters" tab on
mponent ID *Integer (Node)*" in the



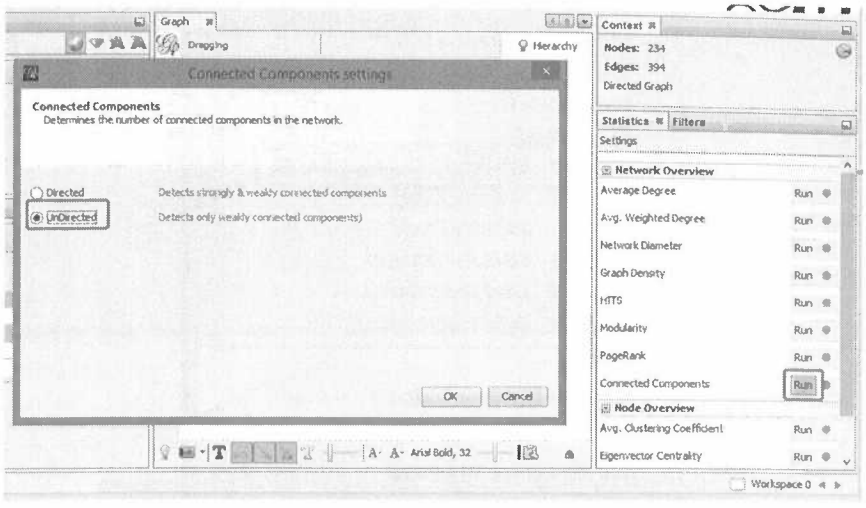



Fig. 7.7 Calculating connected components.

folder directory (you'll find it under "Attributes," then "Equal"). Double-click "Component ID *Integer (Node)*" and click the "Filter" button at the bottom (Fig. 7.8). Doing this removes the disconnected bundles of nodes.

So far, you have gone through the many steps it takes to finally analyze the data. The data-gathering step (1) in this case was pretty easy, as it simply involved downloading a text file from the Internet Archive. The data-cleaning step (2) was a bit more complex, as it involved data mining in Notepad++ using regular expressions and some smoothing in OpenRefine. Data preparation (3) involved loading your CSV into Gephi, adding labels to nodes, and filtering out unwanted data. Two steps are left: analysis (4) and visualization (5). In most history projects, the first three steps take the longest by many orders of magnitude, and each step is often revisited as your thoughts on the project evolve.

There are many possible algorithms you could use for the analysis step, but in this case you will use the PageRank of each node in the network. This measurement calculates the prestige of a correspondent according to how often others write to him or her. The process is circular, such that correspondents with high prestige will confer their prestige on those *they* write to, who in turn pass their prestige along to their own correspondents. The algorithm is described at length in Chapter 5, but for the moment let us take its results to equate with a correspondent's importance in the Republic of Texas letter network.

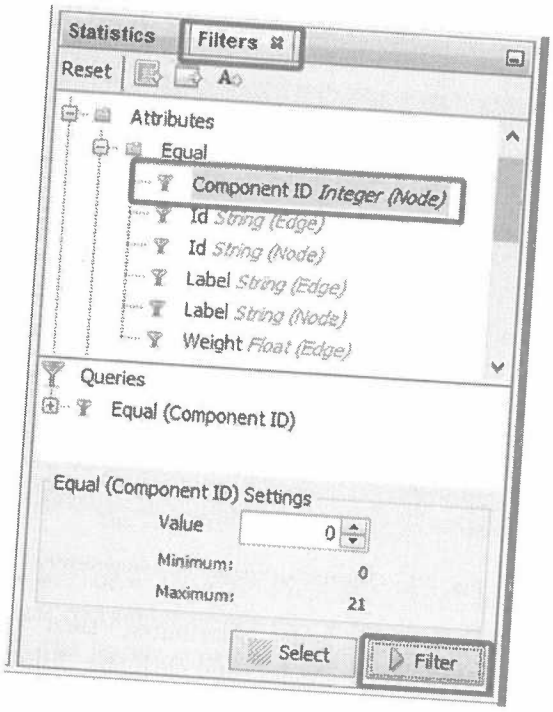
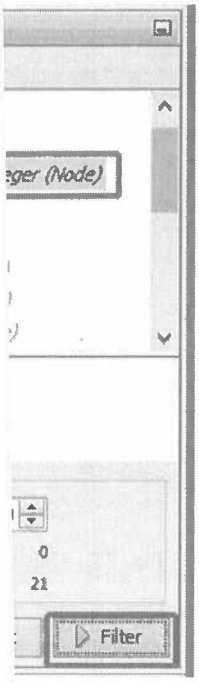



Fig. 7.8 Filtering components.

Calculate the PageRank by clicking on the "Run" button next to "PageRank" in the "Statistics" tab. You will be presented with a prompt asking for a few parameters; make sure "Directed" network is selected and that the algorithm is taking edge weight into account (by selecting "Use edge weight"). Leave all other parameters at their default. Press "OK" (Fig. 7.9).

Once PageRank is calculated, if you click back into the "Data Laboratory" and select the "Nodes" list in the Data Table, you can see that a new "PageRank" column has been added, with values for every node. The higher the PageRank, the more central a correspondent is in the network. Going back to the Overview pane, you can visualize this centrality by changing the size of each correspondent's node based on its PageRank. Do this in the "Ranking" tab on the left side of the Overview pane.

Make sure "Nodes" is selected, press the icon of a little red diamond, and select PageRank from the drop-down menu. In the parameter options just below, enter the "Min size" as 1 and the "Max size" as 10. Press "Apply,"




ponents.

n the "Run" button next to
ill be presented with a prompt
ected" network is selected and
nto account (by selecting "Use
at their default. Press "OK"

k back into the "Data Labora-
Table, you can see that a new
lues for every node. The higher
ident is in the network. Going
ze this centrality by changing
l on its PageRank. Do this in
erview pane.
con of a little red diamond, and
In the parameter options just
ax size" as 10. Press "Apply,"



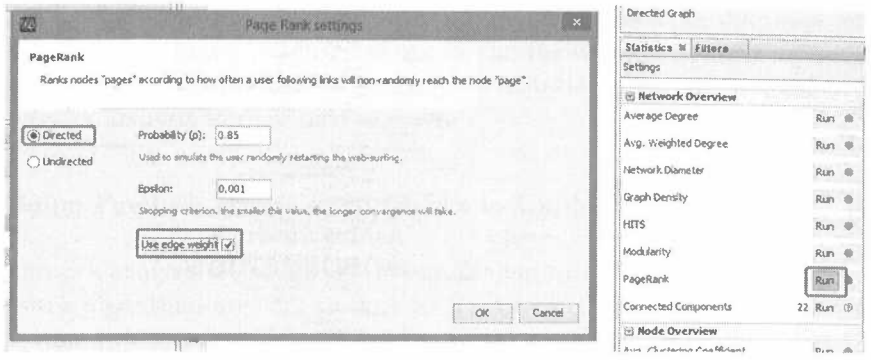



Fig. 7.9 Calculating PageRank.


Fig. 7.10 Resizing nodes by PageRank.

and watch the nodes resize based on their PageRank (Fig. 7.10). To be on the safe side and decrease clutter, re-run the "Force Atlas 2" layout as described above, making sure to keep the "Prevent Overlap" box checked.

At this point, the network is processed enough to visualize in the Preview pane, to finally begin making sense of the data. In Preview, on the left-hand side, select "Show Labels," "Proportional Size," "Rescale Weight,"

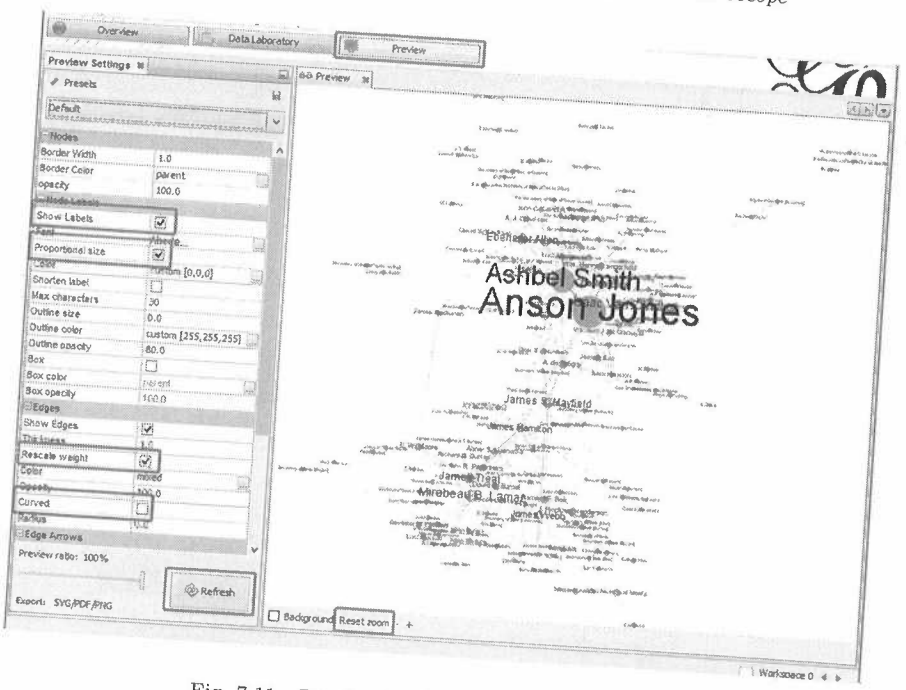
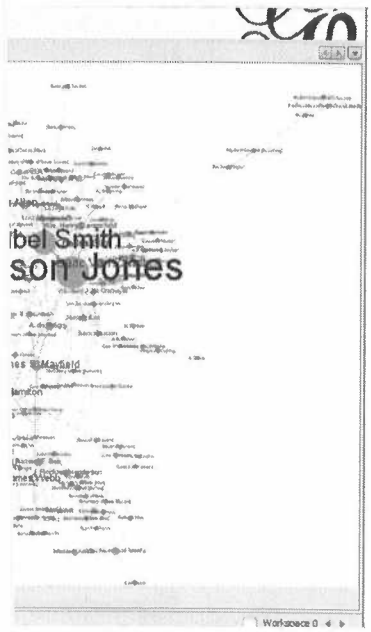



Fig. 7.11 Previewing the network visualization.

and deselect "Curved" edges. Press "Refresh." The resulting visualization should look similar to Fig. 7.11 (although not precisely the same — the process to get to this point, especially the layout, has elements of randomness).

The visualization immediately reveals apparent structure: central figures on the top (Ashbel Smith and Anson Jones) and bottom (Mirabeau B. Lamar, James Webb, and James Treat), and two central figures who connect the two split communities (James Hamilton and James S. Mayfield). A quick search online shows the top network to be associated with the last president of the Republic of Texas, Anson Jones; whereas the bottom network largely revolves around the second president, Mirabeau Lamar. Experts on this period in history could use this analysis to understand the structure of communities building to the annexation of Texas or they could ask meta-questions about the nature of the data themselves. For example, why is Sam Houston, the first and third president of the Republic of Texas, barely visible in this network?

A more nuanced historical view would take into account the temporal, evolving nature of the correspondence network. Gephi can support




rk visualization.

esh." The resulting visualiza-
ough not precisely the same
ly the layout, has elements of

pparent structure: central fig-
nes) and bottom (Mirabeau B.
l two central figures who con-
ilton and James S. Mayfield).
rk to be associated with the
on Jones; whereas the bottom
president, Mirabeau Lamar.
his analysis to understand the
exation of Texas or they could
lata themselves. For example,
dent of the Republic of Texas,

ake into account the tempo-
network. Gephi can support



the analysis and visualization of evolving networks.[7] Such networks are extremely complex, both in terms of creating them and in terms of analysis. However they are the obvious next step for historians interested in applying network analysis to their own research.

### Going Further: Dynamic Networks in Gephi

Network analysis is great when we are dealing with a static network. Most metric algorithms are built to work with networks that capture one moment in time (where moment can be defined as everything from a single day, to a week, to a month, to a span of years). Sometimes, it will make sense to consider a pattern of interactions (a network) that took place over a long period of time *as a single network.* For instance, in archaeological network analysis, given the nature of the material, a single network can encompass 200 years' worth of interactions.[8] But other times it would make more sense to see the shape of a network as an evolving, dynamic set of relationships, as in correspondence networks.

One could create a series of networks in time-slices — the exchange of letters between individuals in 1836 as one network; all those in 1837 as another; and so on. This can be a good approach (depending on your question), but it is subject to *edge effects*: the decision on where to draw the boundary changes the shape of the network. Fortunately, Gephi can deal with dynamic data and with edges that exist for varying durations, and can calculate some metrics on the fly (revisualizing on the fly as well).

But what does "duration" mean for a network? In a correspondence network, perhaps "duration" means "the time the letter is uppermost in one's mind." Perhaps it means "the time until I write a response," thus closing the relationship (and which implies that a correspondence network can have both directed and undirected edges). In Graham's 2006 network of land-owning relationships deduced from the archaeology and epigraphy of Roman stamped bricks, "duration" was considered to be open-ended

[7] For tutorials on how to use some of these features, see Clement Levallois' at http://www.clementlevallois.net/gephi/tuto/gephi_tutorial_dynamics.pdf and https://marketplace.gephi.org/plugin/excel-csv-converter-to-network/.

[8] Graham (2006) *Ex Figlinis: The Network Dynamics of the Tiber Valley Brick Industry in the Hinterland of Rome* BAR International Series 1486. Oxford: John and Erica Hedges Ltd does this with relationships of landholding around Rome in the first three centuries. See also Graham (2014) "On Connecting Stamps: Network Analysis and Epigraphy," *Nouvelles d'archéologie*, **135**, 39–44.

(in broad strokes, the stamps represent a kind of rental agreement and so, unless we find a parcel of land suddenly being leased under a different name, we can assume the relationship continues). The decisions that the historian makes while assembling, cleaning, and representing her data become the objects that the computer manipulates, so these issues are theoretically significant!

At the time of writing, Gephi is being prepared for an update that will alter the way the program imagines "duration" and "time" in networks. For more information about dynamic networks in Gephi, please see our companion website at http://themacroscope.org.

### A Common Sin: Analyzing Two Mode Networks with One Mode Statistics

We often see network analyses where the nodes of the network actually make up more than one kind of thing. A network of small businesses connected to the churches that their workers attend, in small town North America in the 1960s, would be a two-mode network. Visualizing such a network can be a useful heuristic, a good initial exploration of one's data. Clumps and clusters might become apparent, and the role of, say, the Presbyterian church versus the Baptist, as a bridge for a variety of otherwise unconnected businesses might jump out.

However, any network metric calculated in Gephi on such a network would be specious since Gephi's default metrics are all founded on the assumptions of one-mode graphs. That is, the algorithm would work, but since the data *is founded on a different set of assumptions than the algorithm expects*, the resulting numbers are essentially meaningless. The solution then, if one wishes to build an argument based on the properties of these network data, is to transform it from a two-mode graph (where the nodes are two kinds of things, businesses and churches) into two one-mode graphs (where the nodes are one kind of thing, businesses to businesses, connected by virtue of church memberships; churches to churches by virtue of churchgoers' employers).

We can do this transformation automatically within Gephi using a plugin. There are a number of plugins for Gephi, most of which may be found at the "Gephi Marketplace" online or via the Tools → Plugins menu item within Gephi. To convert a two-mode network to two one-mode networks, look for the "multimode networks transformations" plugin. Install

ı kind of rental agreement and so,
ɐing leased under a different name,
). The decisions that the historian
epresenting her data become the
so these issues are theoretically

ɡ prepared for an update that will
ıration" and "time" in networks.
etworks in Gephi, please see our
ope.org.

### ɔde Networks

odes of the network actually make
ɔrk of small businesses connected
d, in small town North America
vork. Visualizing such a network
xploration of one's data. Clumps
the role of, say, the Presbyterian
ı variety of otherwise unconnected

ted in Gephi on such a network
metrics are all founded on the
is, the algorithm would work,
*rent set of assumptions than the*
are essentially meaningless. The
ɹgument based on the properties
om a two-mode graph (where the
and churches) into two one-mode
f thing, businesses to businesses,
ɔs; churches to churches by virtue

matically within Gephi using a
ɔr Gephi, most of which may be
or via the Tools → Plugins menu
le network to two one-mode net-
transformations" plugin. Install



this plugin, and a new panel will open in Gephi, adjacent to the statistics panel. To convert your materials,

1. Make sure your initial CSV file with your data has a column called "type." Fill that column with "undirected." (You can add a new column within Gephi's data laboratory, or do it using a spreadsheet program instead.) The plugin assumes undirected data and therefore doesn't work correctly with directed graphs.
2. Then, once your CSV file is imported, create a new column on the nodes table; call it "node-type" — here you specify what the thing is. Fill it up accordingly. For instance, you might have as node types "worker" and "church".
3. Save your network; this next step will irrevocably change your data. Click "load attributes" on the multimode transformations panel. Under attribute type, select your column you created for step 2. Then, for left matrix, select worker — church; for right matrix, select church — worker.
4. Select "remove edges" and "remove nodes."
5. Hit "run."
6. Save your new one-mode network with a new name. Your new network here will be a network of workers connected to other workers by virtue of shared membership in a particular church.
7. To generate a network where the nodes are churches tied by shared workers, close Gephi, reload your original two-mode network, and in step 3, invert your matrix selections.

While a two-mode network can help us intuit interesting patterns within the data, for actual analysis, we need to transform it so that we have apples to apples, oranges to oranges.

## Conclusion

A network visualization, or a network analysis, can be a powerful part of the historian's macroscope. It has taken us two chapters to discuss what networks are, what they can represent, and how they can be represented. We have shown how to load network data into one of the more popular network visualization and analysis programmes, Gephi. Many kinds of data can be represented as a network, and network methods can be combined with all of the techniques discussed in Chapters 3 and 4.

But should you? If you have detected a note of caution these last two chapters, you are not imagining things. We have been to enough conferences and read enough papers to know that there is, at present, an overabundance of enthusiasm for networks. Sometimes they have been used without due regard for their limitations. Sometimes, they are invoked as a metaphor but their *formal use* has been dismissed as leading to "spaghetti diagrams." That would be to throw the baby out with the bathwater. We believe that network approaches, properly and cautiously used, have much to offer the historian dealing with biggish data. These last two chapters have given you the tools and theoretical apparatus to use them wisely, to good effect, where they can do the most to support your historical understanding.